

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση ελέγχου ταυτότητας πολλών παραγόντων (**multi-factor authentication**) για διαδικτυακές εφαρμογές μέσω της υποδομής **Shibboleth**

Διπλωματικη Εργασια

του

ΓΕΩΡΓΙΟΥ ΘΩΜΑ
Α.Μ. 1059634

Επιβλέπων: Ιωάννης Γαροφαλάκης
Καθηγητής

Μέλη Επιτροπής Αξιολόγησης

Ιωάννης Γαροφαλάκης	Ανδρέας Κομνηνός	Χρήστος Ζαρολιάγκης
Καθηγητής	Επίκουρος Καθηγητής	Καθηγητής

Πάτρα, 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

© Copyright συγγραφής Γεώργιος Θωμάς, 2022.
© Copyright θέματος Ιωάννης Γαροφαλάκης, 2022.
Με την επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Υπεύθυνη Δήλωση

Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στη διπλωματική εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Πατρών.

Γεώργιος Θωμάς

Περίληψη

Σήμερα, οργανώσεις και επιχειρήσεις χρειάζονται τα κατάλληλα εργαλεία για τη διαχείριση και την πιστοποίηση ταυτότητας των χρηστών. Τα εργαλεία αυτά, πλέον, έχουν γίνει ψηφιακά. Περνώντας, όμως, στον ψηφιακό κόσμο, επιτακτική ήταν η ανάγκη να οριστούν κάποιες έννοιες. Μια από αυτές είναι η "ταυτότητα", η οποία με βάση αυτήν γίνεται η διαχείριση και η πιστοποίηση ταυτότητας των χρηστών. Κατά τη διάρκεια της ζωής της ταυτότητας υπάρχουν ορισμένα γεγονότα. Ένα από τα γεγονότα αυτά είναι η ενιαία σύνδεση, η οποία προσφέρει στους χρήστες τη δυνατότητα να συνδέονται μια φορά και να έχουν πρόσβαση σε πολλές εφαρμογές. Για να επιτευχθεί αυτό έχουν οριστεί ορισμένα πρωτόκολλα. Στην εργασία αυτή αναλύεται το πρωτόκολλο SAML το οποίο παρέχει την ενιαία σύνδεση μεταξύ τομέων (cross-domain single sign-on) και την ομοσπονδιακή ταυτότητα (federated identity). Πάνω στο πρωτόκολλο αυτό, έχει βασιστεί και η υποδομή Shibboleth, η οποία προσφέρει ένα ολοκληρωμένο σύστημα ενιαίας σύνδεσης και ταυτοποίησης χρηστών.

Ενώ η υποδομή Shibboleth μπορεί να χαρακτηριστεί ως ένα ολοκληρωμένο σύστημα, η ταυτοποίηση των χρηστών γίνεται με έναν μόνο παράγοντα. Ο παράγοντας αυτός προτρέπει τους χρήστες να εισάγουν ένα όνομα χρήστη και έναν κωδικό πρόσβασης για να πιστοποιηθούν. Ο τρόπος αυτός είναι δοκιμασμένος και τα περισσότερα συστήματα, αν όχι όλα, τον χρησιμοποιούν για να πιστοποιήσουν την ταυτότητα των χρηστών τους. Παρόλα αυτά, για να γίνει η διαδικασία πιστοποίησης ταυτότητας των χρηστών ασφαλέστερη κρίνεται απαραίτητο να χρησιμοποιούνται πολλοί παράγοντες. Η διαδικασία αυτή ονομάζεται έλεγχος ταυτότητας πολλών παραγόντων (Multi-Factor Authentication (MFA)). Η υποδομή Shibboleth, αν και παρέχει εγγενώς κάτι τέτοιο, βασίζεται κυρίως σε ενσωματώσεις με εξωτερικές εφαρμογές. Το Duo είναι μία τέτοια εξωτερική εφαρμογή η οποία ενσωματώνεται με το Shibboleth μέσω ενός προσθέτου και κάνει χρήση MFA με κινητές συσκευές αλλά απαιτεί κόστος. Επίσης, το Shibboleth παρέχει δωρεάν πρόσθετο που πιστοποιεί την ταυτότητα των χρηστών με έναν δεύτερο παράγοντα που κάνει χρήση Time Based One-Time Password (TOTP), ένας τρόπος ο οποίος χρησιμοποιείται σε συνδυασμό με κινητές συσκευές και είναι πολύ διαδεδομένος στα σύγχρονα συστήματα. Παρόλα αυτά, το πρόσθετο αυτό δεν προσφέρει διαχείριση της εγγραφής των χρηστών στον παράγοντα αυτόν. Οπότε, στα πλαίσια της εργασίας αυτής, δημιουργήθηκε ανοικτού κώδικα εφαρμογή η οποία αποτελεί επέκταση του πρόσθετου αυτού. Η εφαρμογή αυτή, δίνει τη δυνατότητα στους χρήστες να εγγραφούν στον δεύτερο παράγοντα ενώ στους διαχειριστές προσφέρεται σύστημα για να διαχειρίζονται τις εγγραφές αυτές.

Abstract

Today, organizations and businesses need the right tools to manage and authenticate users. These tools have now become digital. Moving to the digital world, however, it was imperative to define some concepts. One of these is "identity", which is the basis for managing and authenticating users' identities. During the lifetime of the identity there are certain events. One of these events is single sign-on, which offers users the possibility to log in once and access multiple applications. In order to achieve this, certain protocols have been defined. In this paper we analyze the SAML protocol which provides cross-domain single sign-on and federated identity. On top of this protocol, the Shibboleth infrastructure has been built, which provides an integrated system for single sign-on and user authentication.

While the Shibboleth infrastructure can be described as an integrated system, user identification is done by a single factor. This factor prompts users to enter a username and password to authenticate themselves. This is a proven method and most systems, if not all, use it to authenticate their users. However, in order to make the user authentication process more secure it is necessary to use multiple factors. This process is called Multi-Factor Authentication (MFA). The Shibboleth infrastructure, while inherently providing this, relies primarily on integrations with external applications. Duo is one such external application that integrates with Shibboleth via a plugin and makes use of MFA with mobile devices but requires a cost. Also, Shibboleth provides a free plugin that authenticates users with a second factor that makes use of Time Based One-Time Password (TOTP), a method which is used in combination with mobile devices and is very widespread in modern systems. However, this plugin does not offer management of user enrollment in this factor. So, in the context of this thesis, an open source application was created which is an extension of this plugin. This application, enables users to enroll in the second factor while administrators are offered a system to manage these enrollments.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ιωάννη Γαροφαλάκη και τον επίκουρο καθηγητή κ. Ανδρέα Κομνηνό που μου έδωσαν την ευκαιρία να ασχοληθώ με τον τομέα της πιστοποίησης ταυτότητας. Επίσης, θα ήθελα να ευχαριστήσω τον μεταδιδακτορικό ερευνητή κ. Αθανάσιο Πλέσσα, για την εξαιρετική συνεργασία και καθοδήγηση καθ΄ όλη τη διάρκεια εκπόνησης της διπλωματικής εργασίας. Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου και τους ανθρώπους που με στήριξαν.

Περιεχόμενα

Περίληψη **i**

Abstract **iii**

Ευχαριστίες **v**

Περιεχόμενα **ix**

Κατάλογος Σχημάτων **xi**

1 Εισαγωγή **1**
1.1 Σημασία του Προβλήματος 1
1.2 Στόχος της Διπλωματικής Εργασίας 2
1.3 Μεθοδολογία Προσέγγισης 2
1.4 Συνεισφορά 2
1.5 Διάρθρωση της Διπλωματικής Εργασίας 2

2 Η Ζωή μιας Ταυτότητας **5**
2.1 Ορισμοί 5
2.2 Γεγονότα στη Ζωή μιας Ταυτότητας 6
2.2.1 Provisioning 7
2.2.2 Authorization 7
2.2.3 Authentication 7
2.2.4 Access Policy Enforcement 8
2.2.5 Sessions 8
2.2.6 Single Sign-On (SSO) 9
2.2.7 Stronger Authentication 9
2.2.8 Logout 10
2.2.9 Account Management and Recovery 10
2.2.10 Deprovisioning 11

3 SAML & Shibboleth **13**
3.1 SAML 13

3.1.1 Πρόβλημα Προς Επίλυση . . . 13
3.1.2 Ορισμοί . . . 14
3.1.3 Λειτουργία . . . 14
3.1.4 SP-Initiated Flow . . . 15
3.1.5 Single Sign-On . . . 16
3.1.6 IdP-Initiated Flow . . . 16
3.1.7 Ομοσπονδιακή Ταυτότητα . . . 18
3.1.8 Εναλλακτικά Πρωτόκολλα . . . 19
3.2 Shibboleth . . . 20

4 Multi-Factor Authentication 23
4.1 Το Πρόβλημα με τους Κωδικούς Πρόσβασης . . . 23
4.2 Multi-Factor Authentication . . . 23
4.3 Μέθοδοι MFA . . . 24
4.3.1 SMS . . . 24
4.3.2 Email . . . 24
4.3.3 Ειδοποίηση Push . . . 25
4.3.4 Βιομετρικό Μέσο . . . 25
4.3.5 U2F . . . 25
4.3.6 TOTP . . . 25
4.3.7 Τοποθεσία . . . 26
4.3.8 Backup Κωδικοί . . . 26
4.4 Ενσωμάτωση MFA με Shibboleth . . . 26
4.4.1 Duo MFA . . . 26
4.4.2 Multi Factor Provider (MFaP) . . . 27
4.4.3 Custom MFA Application . . . 27

5 Τεχνική Περιγραφή Υλοποίησης 29
5.1 Βασική Υλοποίηση . . . 29
5.1.1 OpenLDAP . . . 30
5.1.2 Shibboleth IdP . . . 31
5.1.3 Shibboleth SP . . . 37
5.1.4 Σύνδεση SP με IdP . . . 40
5.1.5 Σύνδεση IdP με SP . . . 40
5.2 Ενσωμάτωση Shibboleth IdP με Duo MFA . . . 42
5.2.1 Πλατφόρμα Duo . . . 42
5.2.2 Shibboleth IdP Duo Plugin . . . 43
5.2.3 Διαδικασία Εγγραφής και Σύνδεσης Χρήστη στο Duo MFA . . . 45
5.2.4 Βασικές Λειτουργίες Διαχειριστή . . . 46
5.3 Custom MFA Application . . . 47
5.3.1 Διαδικασία Ελέγχου Ταυτότητας MFA Χρηστών . . . 49

5.3.2 Λειτουργίες Διαχειριστή 50
5.3.3 Αρχιτεκτονική Εφαρμογής 52
5.3.4 Εγκατάσταση Εφαρμογής 55
5.3.5 Ενσωμάτωση Εφαρμογής με Shibboleth IdP 56

6 Συμπεράσματα & Προοπτικές **67**
6.1 Συμπεράσματα 67
6.2 Μελλοντική Εργασία 68

Κατάλογος Σχημάτων

2.1 Γεγονότα στη Ζωή μιας Ταυτότητας [31] 6

3.1 SP-Initiated Flow [31] 15
3.2 IdP-Initiated Flow [31] 17
3.3 Ομοσπονδιακή Ταυτότητα [31] 19
3.4 Shibboleth Flow [26] 21

5.1 Σελίδα Εισόδου Χρήστη 29
5.2 Σελίδα Shibboleth Application 43
5.3 Duo QR Code 45
5.4 Duo Push Notification 46
5.5 Duo OTP 46
5.6 Duo Users Panel 47
5.7 Duo User Panel 47
5.8 Σελίδα Ενεργοποίησης MFA 49
5.9 Σελίδα Εισαγωγής TOTP 50
5.10 Dashboard 50
5.11 Users 51
5.12 User Logs 52
5.13 Αρχιτεκτονική 55
5.14 Σελίδα Εισόδου 57
5.15 Σελίδα Ρυθμίσεων 58

Κεφάλαιο 1

Εισαγωγή

Οι οργανώσεις και οι επιχειρήσεις για να διαχειρίζονται το προσωπικό τους, χρειάζονται τα κατάλληλα συστήματα. Με την ανάπτυξη της τεχνολογίας, τα συστήματα αυτά ξεκίνησαν να γίνονται ψηφιακά. Έτσι, δημιουργήθηκε η ανάγκη να αναπτυχθούν πρωτόκολλα και λογισμικά τα οποία να εξυπηρετούν τις λειτουργίες που χρειάζεται μια οργάνωση ή μια επιχείρηση. Περνώντας στον ψηφιακό κόσμο, όμως, ένα σημαντικό ζήτημα εμφανίστηκε για την ασφαλής λειτουργία των συστημάτων αυτών. Το ζήτημα αυτό είναι η πιστοποίηση ταυτότητας των χρηστών. Η πιστοποίηση ταυτότητας των χρηστών ξεκίνησε με έναν τρόπο που σήμερα είμαστε όλοι γνώριμοι, την εισαγωγή ενός username και password. Με το πέρασμα του χρόνου, ο τρόπος αυτός αν και χρήσιμος και επιτυχής, αποδείχθηκε ότι δεν πρέπει να είναι ο μόνος αλλά να συνδυάζεται με άλλους τρόπους πιστοποίησης ταυτότητας ώστε να αποφευχθούν ζητήματα υποκλοπής και πλαστοπροσωπίας των χρηστών. Έτσι, δημιουργήθηκε η έννοια του ελέγχου ταυτότητας πολλών παραγόντων (Multi-Factor Authentication (MFA)).

1.1 Σημασία του Προβλήματος

Η ασφαλής ταυτοποίηση του χρήστη στη σύγχρονη εποχή αποτελεί ένα καίριο ζήτημα. Αυτό γιατί με τη συνεχή ψηφιοποίηση όλο και περισσότερες πληροφορίες και υπηρεσίες περνούν στον ψηφιακό κόσμο. Με τη μη εξουσιοδοτημένη πρόσβαση από επιτιθέμενους σε συστήματα, οι επιτιθέμενοι έχουν στη διάθεσή τους όχι μόνο ιδιωτικές πληροφορίες αλλά και είναι σε θέση και να εκτελέσουν ενέργειες οι οποίες μπορούν να φέρουν σε κίνδυνο τους χρήστες αλλά και το ίδιο το σύστημα. Ένα παράδειγμα είναι αυτό της μη εξουσιοδοτημένης πρόσβασης σε ένα τραπεζικό σύστημα. Ο επιτιθέμενος έχει πρόσβαση σε προσωπικές πληροφορίες αλλά και έχει και τη δυνατότητα να αφαιρέσει οποιοδήποτε ποσό από τους λογαριασμούς τους θύματος. Οι επιθέσεις αυτές δεν περιορίζονται μόνο σε τραπεζικά συστήματα αλλά σε οποιοδήποτε σύστημα βρίσκεται στον παγκόσμιο ιστό. Με την ανάδειξη των συστημάτων διαχείρισης προσωπικού που χρησιμοποιούν οργανώσεις και επιχειρήσεις, η ανάγκη για μια πιο ασφαλή πιστοποίηση ταυτότητας με πολλούς παράγοντες έγινε ακόμα πιο σημαντική. Αυτό γιατί ένα σημαντικό χαρακτηριστικό των συστημάτων αυτών είναι η ενιαία σύνδεση (single sign-on) η οποία επιτρέπει στον χρήστη να συνδέεται μια φορά και να έχει πρόσβαση σε πολλές εφαρμογές, οι

οποίες με μια μη εξουσιοδοτημένη πρόσβαση μπορούν όλες να παραβιαστούν. Συμπεραίνεται, λοιπόν, ότι ο έλεγχος ταυτότητας πολλών παραγόντων είναι ένα απαραίτητο στοιχείο για να διασφαλιστεί η ακεραιότητα ενός λογαριασμού σε ένα σύστημα.

1.2 Στόχος της Διπλωματικής Εργασίας

Στόχος της Διπλωματικής Εργασίας (ΔΕ) είναι η επέκταση υπάρχοντος συστήματος πιστοποίησης ταυτότητας με τη δημιουργία εφαρμογής η οποία προσφέρει τη δυνατότητα ελέγχου πολλών παραγόντων στο σύστημα αυτό. Αυτό επιτεύχθηκε με την κατάλληλη έρευνα των πρωτοκόλλων και των συστημάτων που χρησιμοποιούνται σήμερα για την πιστοποίηση ταυτότητας χρηστών αλλά και την έρευνα στον τομέα ελέγχου ταυτότητας πολλών παραγόντων που αξιοποιούνται σήμερα στον παγκόσμιο ιστό.

1.3 Μεθοδολογία Προσέγγισης

Αρχικά, πραγματοποιήθηκε έρευνα πάνω στα πρωτόκολλα τα οποία προσφέρουν λειτουργίες πιστοποίησης ταυτότητας. Ύστερα, έγινε μεγαλύτερη ανάλυση στο πρωτόκολλο που αξιοποιεί το σύστημα στο οποίο έγινε επέκταση. Τέλος, εξερευνήθηκαν οι δυνατότητες και τα χαρακτηριστικά του συστήματος αλλά και τυχόν ελλείψεις και αδυναμίες που υπάρχουν στο τομέα ελέγχου ταυτότητας πολλών παραγόντων. Με βάση αυτά, υλοποιήθηκε εφαρμογή που να καλύπτει τις ελλείψεις αυτές, αξιοποιώντας τις δυνατότητες που παρέχει το σύστημα.

1.4 Συνεισφορά

Η διπλωματική εργασία επιχειρεί να ενημερώσει τον αναγνώστη για τις τεχνολογίες που υπάρχουν στον τομέα της πιστοποίησης ταυτότητας των χρηστών και συγκεκριμένα στο καίριο κομμάτι του ελέγχου ταυτότητας πολλών παραγόντων. Επίσης, η διπλωματική εργασία συνεισφέρει με τη δημιουργία εφαρμογής υποστήριξης ελέγχου ταυτότητας πολλών παραγόντων σε υπάρχον σύστημα, ώστε οργανώσεις και επιχειρήσεις να έχουν στη διάθεσή τους ένα πλήρες και ασφαλές σύστημα πιστοποίησης ταυτότητας.

1.5 Διάρθρωση της Διπλωματικής Εργασίας

Η διπλωματική εργασία είναι οργανωμένη σε 6 κεφάλαια. Το Κεφάλαιο 1 είναι εισαγωγικό και αναφέρει τους στόχους της εργασίας. Το Κεφάλαιο 2 ορίζει τι είναι "ταυτότητα" καθώς και περιγράφει τα γεγονότα που υπάρχουν στη ζωή της. Στο Κεφάλαιο 3 παρουσιάζεται το πρωτόκολλο εξουσιοδότησης SAML και η υποδομή Shibboleth που είναι βασισμένη στο πρωτόκολλο αυτό. Επίσης, συνοπτικά παρουσιάζονται εναλλακτικά πρωτόκολλα. Στο Κεφάλαιο 4 ορίζεται ο έλεγχος ταυτότητας πολλών παραγόντων και γίνεται παρουσίαση των σύγχρονων τρόπων εφαρμογής του. Στο Κεφάλαιο 5 καταγράφονται τα βήματα που υλοποιήθηκαν για την

εγκατάσταση και ρύθμιση των τεχνολογιών που χρησιμοποιήθηκαν. Επίσης, γίνεται παρουσίαση των χαρακτηριστικών της εφαρμογής που αναπτύχθηκε καθώς και τα βήματα για την εγκατάστασή και την ενσωμάτωσή της με την υποδομή Shibboleth. Τέλος, στο Κεφάλαιο 6 παρουσιάζονται τα συμπεράσματα και οι προοπτικές από τη διπλωματική εργασία.

Κεφάλαιο 2

Η Ζωή μιας Ταυτότητας

2.1 Ορισμοί

Οι έννοιες της ταυτότητας (identity), του αναγνωριστικού (identifier) και του λογαριασμού (account) είναι στενά συνδεδεμένες αλλά διαφορετικές. Ο όρος "αναγνωριστικό" χρησιμοποιείται για να γίνει αναφορά σε ένα μοναδικό χαρακτηριστικό του οποίου ο σκοπός είναι να ταυτοποιήσει μοναδικά ένα πρόσωπο ή μια οντότητα, μέσα σε ένα συγκεκριμένο πλαίσιο [31]. Οι διευθύνσεις ηλεκτρονικού ταχυδρομείου και οι αριθμοί εργαζομένων είναι παραδείγματα αναγνωριστικών που χρησιμοποιούνται για ανθρώπους. Μια αλφαριθμητική ακολουθία χαρακτήρων που δίνεται σε μη ανθρώπινες οντότητες κατά τη στιγμή του σχηματισμού τους, όπως συσκευές, μπορεί να χρησιμοποιηθεί για την αναγνώρισή τους. Τα αναγνωριστικά είναι ζωτικής σημασίας για τη διαχείριση της ταυτότητας επειδή επιτρέπουν να αναφερόμαστε σε ένα συγκεκριμένο πρόσωπο ή μια μη ανθρώπινη οντότητα.

Ο όρος "ταυτότητα" ορίζεται ως μια συλλογή χαρακτηριστικών που συνδέονται με ένα συγκεκριμένο πρόσωπο ή οντότητα σε ένα συγκεκριμένο πλαίσιο [31]. Μια ταυτότητα αποτελείται από ένα ή περισσότερα αναγνωριστικά και μπορεί επίσης να περιλαμβάνει άλλα χαρακτηριστικά που σχετίζονται με ένα πρόσωπο ή μια οντότητα. Τα χαρακτηριστικά της ταυτότητας ενός ατόμου μπορεί να περιλαμβάνουν στοιχεία όπως όνομα, ηλικία, διεύθυνση, αριθμό τηλεφώνου. Οι μη ανθρώπινες ταυτότητες μπορεί να περιέχουν στοιχεία όπως διεύθυνση IP και αριθμό μοντέλου. Τα χαρακτηριστικά που συνθέτουν μια ταυτότητα μπορούν να χρησιμοποιηθούν για πιστοποίηση ταυτότητας (authentication) και εξουσιοδότηση (authorization), καθώς και για τη μετάδοση πληροφοριών σχετικά με την ταυτότητα σε εφαρμογές.

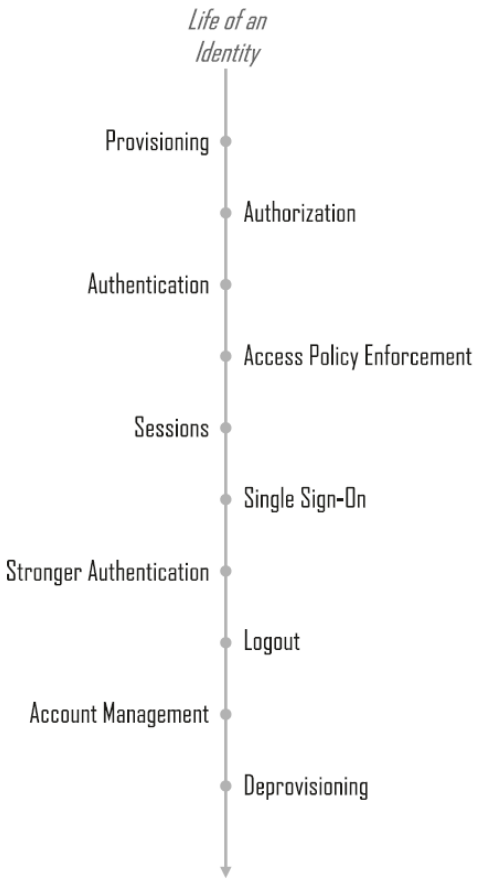
Ο όρος "λογαριασμός" ορίζεται ως μια τοπική κατασκευή μέσα σε μια συγκεκριμένη εφαρμογή ή μια σειρά εφαρμογών που χρησιμοποιείται για την εκτέλεση ενεργειών μέσα σε αυτό το πλαίσιο [31]. Μαζί με την ταυτότητα που συνδέεται με αυτόν, ένας λογαριασμός μπορεί επίσης να έχει το δικό του μοναδικό αναγνωριστικό. Η ύπαρξη ενός αναγνωριστικού λογαριασμού χωριστά από την ταυτότητα που σχετίζεται με το λογαριασμό, παρέχει ένα βαθμό διαχωρισμού. Το αναγνωριστικό λογαριασμού μπορεί να χρησιμοποιηθεί σε άλλες εγγραφές εφαρμογών για να διευκολύνει τους χρήστες να αλλάξουν το όνομα χρήστη ή κάποιο άλλο αναγνωριστικό που σχετίζεται με τον λογαριασμό τους.

Ο όρος "ταυτότητα" χρησιμοποιείται ως το σύνολο των χαρακτηριστικών ενός προσώπου ή μιας οντότητας [31]. Ο όρος "λογαριασμός" χρησιμοποιείται ως μια κατασκευή εντός μιας εφαρμογής που έχει μια ταυτότητα συνδεδεμένη με αυτήν. Θα πρέπει να τονιστεί ότι ένας λογαριασμός μπορεί να συνδέεται με περισσότερες από μία ταυτότητες. Συνοψίζοντας, ένας χρήστης συνδέεται για να χρησιμοποιήσει έναν λογαριασμό που συνδέεται με διάφορα χαρακτηριστικά αναγνώρισης και του επιτρέπει να εκτελεί λειτουργίες μέσα σε ένα σύστημα.

Ένα σύστημα διαχείρισης ταυτότητας (Identity Management (IdM) System)) είναι μια συλλογή υπηρεσιών που διευκολύνει τη δημιουργία, την τροποποίηση και τη διαγραφή ταυτοτήτων και των λογαριασμών που συνδέονται με αυτές, καθώς και την πιστοποίηση ταυτότητας και την εξουσιοδότηση που είναι απαραίτητες για την πρόσβαση σε πόρους (resources). Τα συστήματα διαχείρισης ταυτοτήτων χρησιμοποιούνται για την προστασία των πόρων από ανεπιθύμητη πρόσβαση και αποτελούν κρίσιμο συστατικό μιας ολοκληρωμένης αρχιτεκτονικής ασφάλειας.

2.2 Γεγονότα στη Ζωή μιας Ταυτότητας

Τα γεγονότα στη ζωή μιας ταυτότητας απεικονίζονται στο Σχήμα 2.1.



Σχήμα 2.1: Γεγονότα στη Ζωή μιας Ταυτότητας [31]

2.2.1 Provisioning

Το πρώτο βήμα στη ζωή μιας ταυτότητας είναι η δημιουργία της. Η δημιουργία ενός λογαριασμού και των σχετικών πληροφοριών ταυτότητας αναφέρεται συχνά ως παροχή (provisioning) [31]. Κατά τη διάρκεια του provisioning, οι χρήστες μπορούν να εγγραφούν, τα δεδομένα ταυτότητας μπορούν να εισαχθούν από ένα παλαιό σύστημα ή μπορεί να χρησιμοποιηθεί μια εξωτερική υπηρεσία ταυτότητας. Ο στόχος του provisioning, ανεξάρτητα από τη χρησιμοποιούμενη τεχνολογία, είναι η δημιουργία ενός λογαριασμού με τα σχετικά δεδομένα ταυτότητας. Αυτό συνεπάγεται με τη δημιουργία ενός λογαριασμού, τη σύνδεση στοιχείων προφίλ ταυτότητας με τον λογαριασμό και την απόκτηση ή την ανάθεση ενός μοναδικού αναγνωριστικού που είναι διαφορετικό από εκείνο του λογαριασμού.

Για παράδειγμα, ο Γιάννης γίνεται μέλος ενός πανεπιστημίου ως φοιτητής. Ο Γιάννης θα συμπληρώσει και θα παρέχει τα στοιχεία του στο πανεπιστήμιο όπως όνομα, επίθετο, ημερομηνία γέννησης, τόπος κατοικίας. Το πανεπιστήμιο θα δημιουργήσει έναν λογαριασμό και θα συσχετίσει τις πληροφορίες αυτές με ένα μοναδικό αναγνωριστικό όπως ο αριθμός μητρώου.

Ο Γιάννης θα μπορούσε να γίνει μέλος του πανεπιστημίου ως καθηγητής. Αυτό του επιτρέπει να δημιουργήσει μια δεύτερη ταυτότητα ως καθηγητής εκτός από τον φοιτητικό λογαριασμό. Μια διαδικτυακή ταυτότητα και ένας λογαριασμός δημιουργούνται κατά τη διάρκεια του provisioning και χρησιμοποιούνται για την πρόσβαση σε διαδικτυακές υπηρεσίες.

2.2.2 Authorization

Κατά τη δημιουργία ενός λογαριασμού, συχνά απαιτείται να οριστούν τα δικαιώματα που θα καθορίσουν τι μπορεί να εκτελεί αυτός ο λογαριασμός. Ο όρος εξουσιοδότηση (authorization) χρησιμοποιείται για τη χορήγηση δικαιωμάτων που καθορίζουν τι επιτρέπεται να κάνει ένας λογαριασμός [31].

Όταν δημιουργείται για τον Γιάννη λογαριασμός φοιτητή, το πανεπιστήμιο εξουσιοδοτεί στον λογαριασμό του να έχει πρόσβαση στους βαθμούς των μαθημάτων του. Ο λογαριασμός του δε θα εξουσιοδοτηθεί να προβάλλει βαθμούς των άλλων φοιτητών ούτε να μπορεί να επεξεργαστεί οποιονδήποτε βαθμό. Η εξουσιοδότηση του Γιάννη υποδεικνύει τα δικαιώματα που έχουν παραχωρηθεί στον λογαριασμό του. Η εξουσιοδότηση λογαριασμού πραγματοποιείται κυρίως κατά τη στιγμή της δημιουργίας του λογαριασμού και μπορεί να αλλάξει με την πάροδο του χρόνου.

2.2.3 Authentication

Ο χρήστης, για να αποκτήσει πρόσβαση σε διαδικτυακό περιεχόμενο που δεν είναι διαθέσιμο στο κοινό, πρέπει να πιστοποιήσει την ταυτότητά του. Ο χρήστης παρέχει ένα αναγνωριστικό που προσδιορίζει τον λογαριασμό που επιθυμεί να χρησιμοποιήσει και εισάγει τα στοιχεία σύνδεσης για τον λογαριασμό αυτό. Αυτά επικυρώνονται με βάση τα στοιχεία που έχουν προηγουμένως καταχωρηθεί κατά το provisioning. Τα στοιχεία αυτά μπορεί να αφορούν κάτι που γνωρίζει ο χρήστης (π.χ. κωδικός πρόσβασης), κάτι που έχει ο χρήστης (π.χ.

κινητό τηλέφωνο) ή/και κάτι που είναι ο χρήστης (π.χ. δακτυλικό αποτύπωμα). Η πιστοποίηση ταυτότητας (Authentication) με ένα ή περισσότερα διαπιστευτήρια τα οποία επικυρώνονται με βάση πληροφορίες που έχουν καταχωρηθεί προηγουμένως, αποδεικνύει, σε κάποιο βαθμό εμπιστοσύνης, το δικαίωμα του χρήστη να χρησιμοποιεί ένα λογαριασμό για πρόσβαση σε προστατευόμενους πόρους [31].

Ο Γιάννης μπορεί να έχει πρόσβαση στις ηλεκτρονικές υπηρεσίες του πανεπιστημίου μόλις δημιουργήσει την ηλεκτρονική του ταυτότητα και τον λογαριασμό του στο πανεπιστήμιο. Για να αποκτήσει πρόσβαση σε προστατευμένους πόρους, όπως τους βαθμούς των μαθημάτων του, θα πρέπει να πιστοποιήσει την ταυτότητά του εισάγοντας το αναγνωριστικό και τον κωδικό πρόσβασης που καθορίστηκε κατά το provisioning. Το αναγνωριστικό προσδιορίζει τον λογαριασμό που επιθυμεί να χρησιμοποιήσει και ο κωδικός πρόσβασης αποδεικνύει το δικαίωμά του να χρησιμοποιεί τον λογαριασμό αυτόν.

2.2.4 Access Policy Enforcement

Για να διασφαλιστεί ότι οποιεσδήποτε ενέργειες ενός χρήστη επιτρέπονται από τα δικαιώματα που του έχουν παραχωρηθεί, απαιτείται η επιβολή πολιτικής πρόσβασης μόλις ένας χρήστης πιστοποιηθεί και συνδεθεί με έναν λογαριασμό. Ο όρος επιβολή πολιτικής πρόσβασης (access policy enforcement) χρησιμοποιείται για την επιβολή της πολιτικής πρόσβασης που καθορίζεται από την εξουσιοδότηση [31]. Δηλαδή, η εξουσιοδότηση καθορίζει τι επιτρέπεται να κάνει ένας χρήστης και η επιβολή πολιτικής πρόσβασης επαληθεύει ότι οι ζητούμενες ενέργειες ενός χρήστη επιτρέπονται από τα δικαιώματα που του έχουν χορηγηθεί.

Όταν ο Γιάννης συνδέεται στις ηλεκτρονικές υπηρεσίες του πανεπιστημίου και υποβάλλει ένα αίτημα, η εφαρμογή θα ελέγξει ότι έχει την εξουσιοδότηση για να υποβάλει το αίτημα αυτό. Για παράδειγμα, εάν επιχειρήσει να αποκτήσει πρόσβαση στους βαθμούς των φοιτητών που έχουν εγγραφεί σε ένα μάθημα, θα του απαγορευτεί η πρόσβαση, καθώς δεν έχει εξουσιοδότηση πρόσβασης στην εν λόγω υπηρεσία. Σε αυτή την περίπτωση, η εφαρμογή μπορεί να εμφανίσει μια ειδοποίηση που θα ενημερώνει τον χρήστη ότι δεν επιτρέπεται η πρόσβαση της συγκεκριμένης υπηρεσίας.

2.2.5 Sessions

Μόλις ένας χρήστης πιστοποιηθεί και του δοθεί άδεια πρόσβασης στην εφαρμογή, μπορεί να προβεί σε διάφορες ενέργειες. Ένας χρήστης μπορεί να είναι ενεργός μόνο για ένα χρονικό διάστημα σε ορισμένες εφαρμογές, συνήθως σε εφαρμογές με ευαίσθητες πληροφορίες, πριν του ζητηθεί να συνδεθεί ξανά. Αυτό επιτυγχάνεται ελέγχοντας τη συνεδρία (session) του χρήστη. Μια συνεδρία καταγράφει πληροφορίες όπως το αν ο χρήστης έχει πιστοποιηθεί, και αν ναι, συνήθως καταγράφει επίσης τον μηχανισμό πιστοποίησης και πότε έγινε η πιστοποίηση, εκτός από το αναγνωριστικό χρήστη [31]. Αυτό επιτρέπει σε μια εφαρμογή να προσδιορίσει πότε ο χρήστης πρέπει να πιστοποιηθεί εκ νέου.

Το όριο συνεδρίας (session limit) ή λήξη συνεδρίας (session timeout) είναι το χρονικό διάστημα κατά το οποίο ένας χρήστης μπορεί να είναι ενεργός προτού του ζητηθεί νέα πιστοπο-

ίηση ταυτότητας. Η ευαισθησία των δεδομένων της εφαρμογής επηρεάζει συχνά τις ρυθμίσεις του χρονικού ορίου συνόδου. Οι χρήστες που αφήνουν τις οθόνες τους αφύλακτες ή των οποίων οι πληροφορίες ταυτότητας ενδέχεται να έχουν αλλάξει από τη στιγμή που σχηματίστηκε η σύνοδος προστατεύονται από το όριο συνεδρίας. Ένα όριο συνεδρίας που απαιτεί από έναν χρήστη να προβαίνει σε περιοδική επαναπιστοποίηση χρησιμεύει ως επαλήθευση ότι το άτομο που χρησιμοποιεί το πληκτρολόγιο εξακολουθεί να είναι ο εξουσιοδοτημένος χρήστης. Μπορεί επίσης να έχει ως αποτέλεσμα την ανανέωση των στοιχείων ταυτοποίησης και της κατάστασης λογαριασμού του χρήστη.

Η εφαρμογή του πανεπιστημίου που δίνει τη δυνατότητα για υποβολή εργασιών μπορεί να επιτρέπει μόνο μια σχετικά σύντομη συνεδρία. Μια άλλη, λιγότερο ευαίσθητη εφαρμογή που προσφέρεται από το πανεπιστήμιο, μπορεί να επιτρέπει μεγαλύτερη συνεδρία. Είναι σημαντικό για την εφαρμογή να προσδιορίζει αν ο Γιάννης έχει πιστοποιηθεί αρκετά πρόσφατα για την προβλεπόμενη ενέργεια κάθε φορά που ο Γιάννης υποβάλλει ένα αίτημα σε κάποια από τις εφαρμογές. Εάν ναι, μπορεί να συνεχίσει χωρίς να πιστοποιηθεί ξανά. Εάν έχει περάσει πολύς χρόνος από την τελευταία πιστοποίησή του, θα πρέπει να πιστοποιηθεί ξανά.

2.2.6 Single Sign-On (SSO)

Αφού ένας χρήστης αποκτήσει πρόσβαση σε μια εφαρμογή, μπορεί να επιθυμήσει να κάνει κάτι άλλο που αφορά μια άλλη εφαρμογή. Η ενιαία σύνδεση (Single Sign-On (SSO)) είναι η δυνατότητα που δίνεται στον χρήστη να συνδεθεί μία φορά και στη συνέχεια να έχει πρόσβαση σε πρόσθετες εφαρμογές με τα ίδια στοιχεία πιστοποίησης, χωρίς να χρειάζεται να εισάγονται εκ νέου τα στοιχεία αυτά.

Όταν ο Γιάννης έχει πρόσβαση στον ιστότοπο του πανεπιστημίου, η ενιαία σύνδεση θα παρέχει εύκολη πρόσβαση σε πολλαπλές πανεπιστημιακές υπηρεσίες. Ο Γιάννης μπορεί να αποκτήσει πρόσβαση στο σύστημα στο οποίο μπορεί να δει και να κατεβάσει υλικό που προσφέρουν οι καθηγητές για τα μαθήματά τους (e-class). Στη συνέχεια, μπορεί να αποκτήσει πρόσβαση σε ξεχωριστό σύστημα, για παράδειγμα, αυτό στο οποίο μπορεί να δει τους βαθμούς του κάθε μαθήματος.

Η ενιαία σύνδεση είναι δυνατή όταν ένα σύνολο εφαρμογών έχει αναθέσει τον έλεγχο ταυτότητας στην ίδια οντότητα. Μια πιστοποιημένη συνεδρία στην εν λόγω οντότητα που μπορεί να χρησιμοποιηθεί για πρόσβαση σε πολλαπλές εφαρμογές μέσω ενιαίας σύνδεσης συχνά ονομάζεται SSO συνεδρία (SSO session) [31].

2.2.7 Stronger Authentication

Ο έλεγχος ταυτότητας πολλαπλών παραγόντων (Multi-factor Authentication (MFA)) περιλαμβάνει την πιστοποίηση ταυτότητας ενός χρήστη με ισχυρότερες μορφές πιστοποίησης ταυτότητας (stronger authentication) [31]. Ορισμένες μέθοδοι ελέγχου ταυτότητας, όπως ο έλεγχος ταυτότητας με όνομα χρήστη (username) και κωδικό πρόσβασης (password), θεωρούνται κάπως αδύναμες επειδή βασίζονται αποκλειστικά σε έναν κωδικό πρόσβασης που μπορεί εύκολα να κλαπεί και να χρησιμοποιηθεί από άλλους. Οι ισχυρότεροι μέθοδοι ελέγχου

ταυτότητας απαιτούν πρόσθετα στοιχεία, όπως κάτι που έχει ο χρήστης ή/και κάτι που είναι ο χρήστης. Ο έλεγχος ταυτότητας πολλών παραγόντων είναι ένας τύπος ελέγχου ταυτότητας που απαιτεί την ταυτόχρονη παροχή πολλών παραγόντων. Ένας κωδικός πρόσβασης, μια συσκευή, όπως ένα τηλέφωνο, και ίσως ένας βιομετρικός παράγοντας, όπως ένα δακτυλικό αποτύπωμα, απαιτούνται για τον έλεγχο ταυτότητας πολλαπλών παραγόντων.

Ο Γιάννης συνδέεται στις υπηρεσίες του πανεπιστημίου με το όνομα χρήστη και τον κωδικό πρόσβασης. Για μεγαλύτερη ασφάλεια, έχει την επιλογή να προσθέσει και έναν άλλον παράγοντα για να συνδεθεί στις υπηρεσίες του πανεπιστημίου, όπως έναν ειδικό κωδικό μιας χρήσης που παράγεται από μία εφαρμογή στο τηλέφωνό του. Προσθέτοντας άλλον έναν παράγοντα παρέχεται μεγαλύτερος βαθμός εμπιστοσύνης ότι ο χρήστης που ζητά πρόσβαση είναι ο νόμιμος κάτοχος του λογαριασμού.

2.2.8 Logout

Όταν ένας χρήστης τελειώσει με μια εφαρμογή, θα πρέπει να κάνει αποσύνδεση (logout) για να τερματίσει τη συνεδρία του [31]. Η σύνοδος εφαρμογής του χρήστη θα πρέπει να τερματίζεται όταν αυτός αποσυνδέεται. Για να αποκτήσει πρόσβαση εάν επιστρέψει στην εφαρμογή, θα πρέπει να πιστοποιηθεί ξανά. Όταν χρησιμοποιείται ενιαία σύνδεση, μπορεί να κριθεί απαραίτητο να τερματιστούν πολλές συνεδρίες. Το ποιες συνεδρίες θα πρέπει να τερματίζονται όταν ο χρήστης αποσυνδέεται από μια εφαρμογή αποτελεί επιλογή σχεδιασμού.

Η πράξη της αποσύνδεσης είναι ελαφρώς διαφορετική από τη λήξη συνεδρίας. Στο πρώτο σενάριο, ο χρήστης ζητάει απευθείας να τερματιστεί η συνεδρία του. Όταν λήγει μια συνεδρία, μια εφαρμογή έχει τη δυνατότητα να επιλέξει να τη διατηρήσει σε κατάσταση αναστολής και να τη συνεχίσει μετά την εκ νέου πιστοποίηση ταυτότητας του χρήστη.

Σε ένα σενάριο όπου ο Γιάννης έχει συνδεθεί στον ιστότοπο του πανεπιστήμιου και έχει μία συνεδρία στο e-class και μία συνεδρία στην εφαρμογή που μπορεί να δει τους βαθμούς του. Αν μείνει για λίγα λεπτά σε αδράνεια, η συνεδρία του στο e-class μπορεί να λήξει. Όταν επιστρέψει, θα πρέπει να πραγματοποιήσει εκ νέου πιστοποίηση ταυτότητας για να συνεχίσει. Όταν ολοκληρώσει τις ενέργειες του στο e-class, θα κάνει κλικ στο σύνδεσμο ¨Αποσύνδεση' για να τερματίσει όλες τις ανοικτές συνεδρίες του.

2.2.9 Account Management and Recovery

Διάφορα χαρακτηριστικά προφίλ χρήστη για μια ταυτότητα μπορεί να χρειαστεί να αλλάξουν σε διάφορα σημεία της ύπαρξής της. Για παράδειγμα, ένας χρήστης μπορεί να χρειαστεί να αλλάξει τον αριθμό τηλεφώνου ή τη διεύθυνση ηλεκτρονικού ταχυδρομείου (email) του. Σε ορισμένες περιπτώσεις, ένας χρήστης μπορεί να χρειαστεί να αλλάξει το όνομά του. Μπορεί επίσης να χρειαστεί να αλλάζει περιοδικά τον κωδικό πρόσβασής του. Η διαχείριση λογαριασμού (account management) περιλαμβάνει εργαλεία ή διαδικασίες που επιτρέπουν στους διαχειριστές και τους χρήστες να βλέπουν και να τροποποιούν τις πληροφορίες του προφίλ χρήστη που σχετίζονται με μια ταυτότητα [31].

Επιπλέον, ένας χρήστης μπορεί να ξεχάσει τον κωδικό πρόσβασής του. Εάν συμβεί αυτό,

ο χρήστης πρέπει να δημιουργήσει νέα στοιχεία πιστοποίησης. Αυτό απαιτεί μια διαφορετική μέθοδο επαλήθευσης του λογαριασμού από τον χρήστη πριν του επιτραπεί να αλλάξει τον κωδικό πρόσβασής του. Η ανάκτηση λογαριασμού (account recovery) είναι ένα σύστημα που επαληθεύει ότι ένας χρήστης είναι ο νόμιμος ιδιοκτήτης του λογαριασμού μέσω μιας δευτερεύουσας μεθόδου πριν επιτρέψει στον χρήστη να δημιουργήσει νέα στοιχεία πιστοποίησης [31].

Αν ο Γιάννης ξεχάσει τον κωδικό πρόσβασής του, θα χρειαστεί ένα μέσο επαναφοράς αυτού. Μπορεί να του επιτραπεί να κάνει κλικ σε έναν σύνδεσμο ανάκτησης λογαριασμού μέσω ηλεκτρονικού ταχυδρομείου για να επαναφέρει τον κωδικό πρόσβασής του. Αν ο Γιάννης αποφασίσει να μετακομίσει, θα χρειαστεί τη δυνατότητα να ενημερώσει το προφίλ του στο πανεπιστήμιο ώστε να απεικονίζει τη νέα του διεύθυνση. Κατά τη διάρκεια ζωής μιας ταυτότητας μπορεί να πραγματοποιηθούν διάφορες αλλαγές, οι οποίες απαιτούν την τροποποίηση των χαρακτηριστικών του προφίλ ταυτότητας ή/και τη δημιουργία νέων στοιχείων ταυτότητας.

2.2.10 Deprovisioning

Μπορεί να έρθει η στιγμή που θα χρειαστεί να κλείσει ένας λογαριασμός. Στην περίπτωση αυτή, ο λογαριασμός του χρήστη και τα σχετικά στοιχεία ταυτοποίησης πρέπει να καταργηθούν για να αποτραπεί η περαιτέρω χρήση. Η αποπαροχή (deprovisioning) μπορεί να λάβει τη μορφή της πλήρους διαγραφής του λογαριασμού και των σχετικών πληροφοριών ταυτότητας ή απλώς της απενεργοποίησης του λογαριασμού, ώστε να διατηρηθούν οι πληροφορίες για σκοπούς ελέγχου.

Αν ο Γιάννης αποφοιτήσει και δεν είναι πλέον φοιτητής του πανεπιστήμιου, ο λογαριασμός του θα κλείσει. Δε θα έχει πλέον πρόσβαση στις υπηρεσίες του πανεπιστημίου. Το πανεπιστήμιο, όμως, θα διατηρήσει κάποιες πληροφορίες ώστε να υπάρχει ένα ιστορικό με τη φοίτηση του Γιάννη στο πανεπιστήμιο.

Κεφάλαιο 3

SAML & Shibboleth

3.1 SAML

Το πρωτόκολλο Security Assertion Markup Language (SAML) [11] είναι γνωστό για την παροχή δύο σημαντικών χαρακτηριστικών, την ενιαία σύνδεση μεταξύ τομέων (cross-domain single sign-on) και την ομοσπονδιακή ταυτότητα (federated identity). Πολλές επιχειρήσεις έχουν υιοθετήσει το πρωτόκολλο SAML 2.0 λόγω της ικανότητάς του να επιτρέπει σε πελάτες και εργαζομένους να χρησιμοποιούν εφαρμογές που αναθέτουν την πιστοποίηση ταυτότητας χρήστη σε έναν κεντρικό πάροχο εταιρικής ταυτότητας. Αυτό παρέχει στην επιχείρηση μια κεντρική τοποθεσία για τη διαχείριση και τον έλεγχο των ταυτοτήτων.

Η χρήση του όρου "SAML" αναφέρεται στο SAML 2.0.

3.1.1 Πρόβλημα Προς Επίλυση

Η πιο συνηθισμένη περίπτωση χρήσης για την οποία χρησιμοποιείται το πρωτόκολλο SAML είναι η ενιαία σύνδεση μεταξύ τομέων (cross-domain single sign-on) [31]. Ένας χρήστης σε αυτό το σενάριο πρέπει να έχει πρόσβαση σε πολλές εφαρμογές που βρίσκονται σε διαφορετικούς τομείς (domains), όπως application1.com και application2.com. Ένας χρήστης μπορεί να χρειαστεί να δημιουργήσει έναν λογαριασμό σε κάθε εφαρμογή και να συνδεθεί σε κάθε μία ξεχωριστά, εάν δεν υπάρχει δυνατότητα cross-domain single sign-on. Αυτό σημαίνει ότι ο χρήστης ενδεχομένως να πρέπει να θυμάται διάφορα ονόματα χρήστη και κωδικούς πρόσβασης. Θα ήταν πρόκληση για την εταιρεία να διαχειριστεί όλους τους λογαριασμούς εφαρμογών που μπορεί να δημιουργήσουν οι υπάλληλοί της.

Ένα πλαίσιο βασισμένο στο XML για την ανταλλαγή πληροφοριών ασφαλείας μεταξύ διαδικτυακών επιχειρηματικών εταίρων είναι ο τρόπος με τον οποίο λειτουργεί το πρωτόκολλο SAML [11]. Οι εφαρμογές μπορούν να αναθέσουν την πιστοποίηση των χρηστών σε έναν απομακρυσμένο οργανισμό, γνωστό ως πάροχο ταυτότητας (Identity Provider (IdP)). Ο χρήστης επαληθεύεται από τον πάροχο ταυτότητας, ο οποίος στη συνέχεια επιστρέφει στην εφαρμογή έναν ισχυρισμό που περιέχει λεπτομέρειες σχετικά με τον επαληθευμένο χρήστη και το συμβάν πιστοποίησης ταυτότητας. Ο χρήστης δε θα χρειαστεί να συνδεθεί ξανά εάν αποκτήσει πρόσβαση σε μια δεύτερη εφαρμογή που χρησιμοποιεί τον ίδιο πάροχο ταυτότητας

για την ανάθεση της πιστοποίησης ταυτότητας. Αυτή η δυνατότητα συγκαταλέγεται στην ενιαία σύνδεση.

Επιπλέον, το πρωτόκολλο SAML προσφέρει μια μέθοδο για μια εφαρμογή και έναν πάροχο ταυτότητας να ανταλλάσσουν δεδομένα χρήστη χρησιμοποιώντας ένα κοινό αναγνωριστικό για τον χρήστη. Αυτό το ονομάζεται ομοσπονδιακή ταυτότητα. Η ομοσπονδιακή ταυτότητα μπορεί να χρησιμοποιεί ένα εσωτερικό αναγνωριστικό που αντιστοιχίζεται στο αναγνωριστικό του χρήστη σε κάθε σύστημα ή μπορεί να χρησιμοποιεί το ίδιο αναγνωριστικό σε όλα τα συστήματα.

3.1.2 Ορισμοί

Το πρωτόκολλο SAML ορίζει τους ακόλουθους όρους [31].

- **Subject** - Υποκείμενο (subject) είναι μια οντότητα για την οποία ανταλλάσσονται πληροφορίες ασφαλείας. Το Subject αναφέρεται συνήθως σε ένα πρόσωπο, αλλά μπορεί να είναι οποιαδήποτε οντότητα ικανή για έλεγχο ταυτότητας, συμπεριλαμβανομένου ενός προγράμματος λογισμικού.
- **SAML Assertion** - Ένα μήνυμα βασισμένο σε XML που περιέχει πληροφορίες ασφαλείας για ένα θέμα.
- **SAML Profile** - Μια προδιαγραφή που ορίζει τον τρόπο χρήσης του SAML μηνυμάτων για το cross-domain single sign-on.
- **Identity Provider (IdP)** - Διακομιστής (server) που εκδίδει SAML Assertion για ένα πιστοποιημένο Subject.
- **Service Provider (SP)** - Ένας πάροχος υπηρεσιών (Service Provider) είναι μια οντότητα που παρέχει μια υπηρεσία, συνήθως με τη μορφή εφαρμογής και αναθέτει την πιστοποίηση σε έναν Identity Provider.
- **Trust Relationship** - Μια συμφωνία μεταξύ ενός Service Provider και ενός Identity Provider, σύμφωνα με την οποία ο Service Provider εμπιστεύεται τα Assertions που εκδίδονται από τον Identity Provider.
- **SAML Protocol Binding** - Περιγραφή του τρόπου με τον οποίο τα στοιχεία των μηνυμάτων SAML αντιστοιχίζονται σε τυποποιημένα πρωτόκολλα επικοινωνίας, όπως το HTTP, για τη μετάδοση μεταξύ Service Provider και Identity Provider.

3.1.3 Λειτουργία

Το πιο συνηθισμένο σενάριο πρωτοκόλλου SAML είναι το cross-domain single sign-on [31]. Το Subject σε αυτό το σενάριο είναι ένας χρήστης που θέλει να χρησιμοποιήσει μια εφαρμογή (application). Η εφαρμογή εκτελεί το ρόλο του Service Provider. Η εφαρμογή δίνει σε έναν Identity Provider, ο οποίος μπορεί να βρίσκεται σε διαφορετικό domain, τον έλεγχο

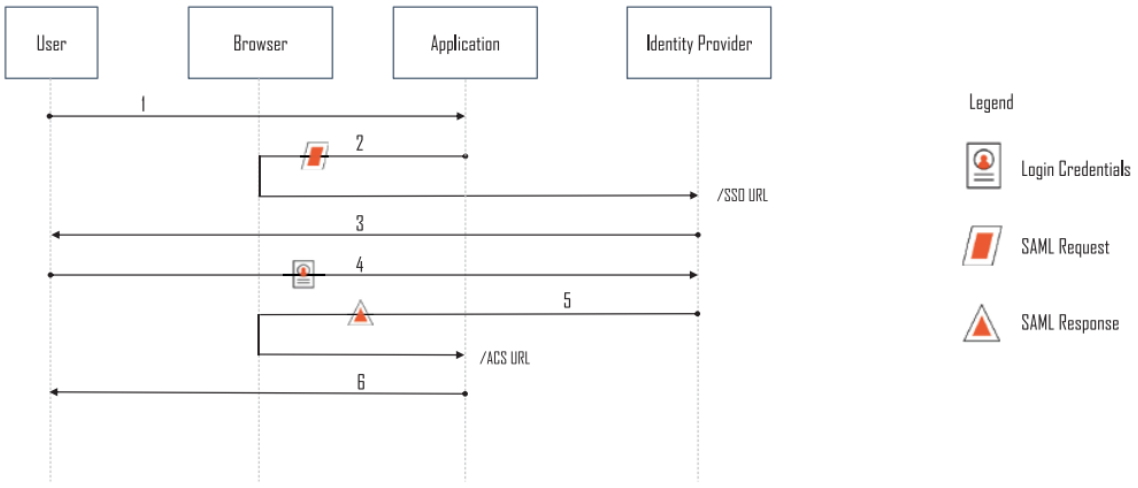
της πιστοποίησης ταυτότητας του χρήστη. Η εφαρμογή λαμβάνει ένα πιστοποιητικό ασφαλείας (security token) γνωστό ως δήλωση SAML Assertion όταν ο Identity Provider πιστοποιεί επιτυχώς τον χρήστη. Οι πληροφορίες σχετικά με το συμβάν πιστοποίησης ταυτότητας και το πιστοποιημένο Subject είναι διαθέσιμες σε ένα SAML Assertion.

Οι οργανισμοί που κατέχουν τον Service Provider και τον Identity Provider ανταλλάσσουν μεταδεδομένα (metadata) προκειμένου να παρέχουν το cross-domain single sign-on. Τα μεταδεδομένα περιλαμβάνουν πληροφορίες που μπορούν να χρησιμοποιηθούν για την επικύρωση ψηφιακά υπογεγραμμένων μηνυμάτων, όπως τα πιστοποιητικά (certificates) και τα URL endpoints. Αυτά τα στοιχεία επιτρέπουν την ανταλλαγή μηνυμάτων μεταξύ των δύο πλευρών. Πριν ο Identity Provider μπορεί να πιστοποιήσει χρήστες για τον Service Provider, πρέπει να δημιουργήσει σχέση εμπιστοσύνης με τον Service Provider χρησιμοποιώντας τα μεταδεδομένα.

Μόλις πραγματοποιηθεί η αμοιβαία διαμόρφωση των παρόχων, όταν ένας χρήστης αποκτά πρόσβαση στον Service Provider (application), αυτός ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη στον Identity Provider με ένα μήνυμα αίτησης πιστοποίησης SAML. Αφού επιβεβαιώσει την ταυτότητα του χρήστη, ο Identity Provider του αποστέλλει ένα μήνυμα απόκρισης πιστοποίησης ταυτότητας SAML και τον παραπέμπει πίσω στην εφαρμογή. Η απόκριση περιλαμβάνει έναν SAML Assertion που περιέχει λεπτομέρειες σχετικά με τον χρήστη και το συμβάν πιστοποίησης ταυτότητας ή ένα σφάλμα εάν υπήρξε κάποιο. Ο Identity Provider μπορεί να προσαρμόσει τα SAML Assertions ξεχωριστά για κάθε Service Provider.

3.1.4 SP-Initiated Flow

Η απλούστερη μορφή cross-domain single sign-on απεικονίζεται στο Σχήμα 3.1 [31]. Σε αυτό το παράδειγμα, ο χρήστης ξεκινά από τον Service Provider (application) και η ροή (flow) αυτή είναι γνωστή ως "SP-initiated".



Σχήμα 3.1: SP-Initiated Flow [31]

1. Ο χρήστης επισκέπτεται έναν Service Provider (application).

2. Ο Service Provider ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη στον Identity Provider με ένα αίτημα πιστοποίησης SAML.

3. Ο Identity Provider αλληλεπιδρά με τον χρήστη για τον έλεγχο ταυτότητας.

4. Ο χρήστης πιστοποιείται. Ο Identity Provider επικυρώνει τα στοιχεία πιστοποίησης.

5. Ο Identity Provider ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη πίσω στον Service Provider με μια απόκριση SAML που περιέχει μια βεβαίωση ελέγχου ταυτότητας SAML. Η απόκριση αποστέλλεται στη διεύθυνση URL του Service Provider.

6. Ο Service Provider επικυρώνει την απάντηση SAML και απαντά στο αρχικό αίτημα του χρήστη (εφόσον ο χρήστης πιστοποιήθηκε επιτυχώς και έχει επαρκή προνόμια για το αίτημα).

3.1.5 Single Sign-On

Στο Σχήμα 3.1 ο χρήστης έχει πρόσβαση σε μία εφαρμογή. Ένας Identity Provider μπορεί να χρησιμοποιείται από πολλούς Service Provider για την πιστοποίηση ταυτότητας των χρηστών. Όταν συμβαίνει αυτό, ένας χρήστης μπορεί να συνδεθεί στον Identity Provider για να δημιουργήσει μια συνεδρία πιστοποίησης ταυτότητας σε μια πρώτη εφαρμογή, όπως φαίνεται στο Σχήμα 3.1. Ο χρήστης μπορεί στη συνέχεια να αποκτήσει πρόσβαση σε μια δεύτερη εφαρμογή που χρησιμοποιεί τον ίδιο Identity Provider, χρησιμοποιώντας το ίδιο πρόγραμμα περιήγησης. Όταν ο χρήστης ανακατευθύνεται στον Identity Provider, ο Identity Provider θα εντοπίσει ότι ο χρήστης έχει ήδη μια σύνοδο και δε θα ζητήσει στον χρήστη να πραγματοποιήσει ξανά πιστοποίηση ταυτότητας. Εάν η πιστοποίηση ταυτότητας είναι επιτυχής, ο Identity Provider θα ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη στην εφαρμογή. Αυτό ονομάζεται ενιαία σύνδεση (single sign-on).

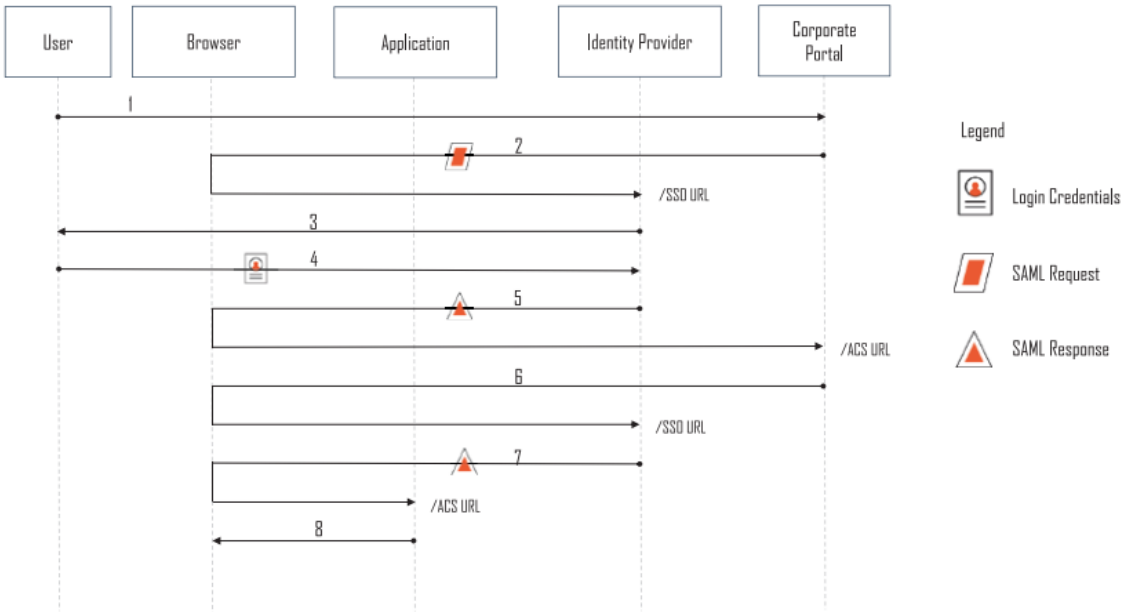
3.1.6 IdP-Initiated Flow

Στο Σχήμα 3.1 απεικονίζεται μια ροή αλληλεπίδρασης με τον χρήστη, ξεκινώντας από τον Service Provider. Επειδή ο χρήστης ξεκινά την αλληλεπίδραση με τον Service Provider, αυτή είναι γνωστή ως "SP-initiated". Η ροή "IdP-initiated", την οποία ορίζει επίσης το πρωτόκολλο SAML, φαίνεται στο Σχήμα 3.2 και ξεκινά με τον Identity Provider, όπου ο χρήστης συνδέεται [31]. Σε αυτή την περίπτωση, ο Identity Provider στέλνει ένα μήνυμα απόκρισης SAML στον Service Provider, αντί ο Service Provider να στείλει ένα αίτημα πιστοποίησης ταυτότητας, ανακατευθύνοντας το πρόγραμμα περιήγησης του χρήστη στον Service Provider. Σε ορισμένες εταιρικές δομές, όπου ένας χρήστης αποκτά πρόσβαση σε προγράμματα μέσω μιας εταιρικής πύλης, υπάρχει αυτή η ροή.

Ο χρήστης κατευθύνεται στον εταιρικό Identity Provider για να συνδεθεί όταν εισέρχεται για πρώτη φορά στην εταιρική πύλη. Ο χρήστης επιστρέφει στην πύλη μετά τη σύνδεση, όπου μπορεί να έχει πρόσβαση σε ένα μενού συνδέσμων εφαρμογών. Ο χρήστης ανακατευθύνεται στον Identity Provider αφού κάνει κλικ σε έναν από αυτούς τους συνδέσμους, με τη διεύθυνση

URL της εφαρμογής ως παράμετρο. Ο Identity Provider αναγνωρίζει ότι ο χρήστης έχει ήδη μία σύνοδο και μεταφέρει το πρόγραμμα περιήγησης του χρήστη στην εφαρμογή.

Οι επιχειρήσεις μπορούν να επωφεληθούν από τη "IdP-initiated" ροή με μια πύλη, καθώς προσφέρει ενιαία σύνδεση και διασφαλίζει ότι οι χρήστες έχουν πρόσβαση στη σωστή διεύθυνση ΥΡΛ για κάθε εφαρμογή, μειώνοντας την πιθανότητα οι χρήστες να πέσουν θύματα απάτης.



Σχήμα 3.2: IdP-Initiated Flow [31]

1. Ο χρήστης επισκέπτεται μια εταιρική πύλη.
2. Η πύλη ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη στον Identity Provider με ένα αίτημα πιστοποίησης ταυτότητας SAML.
3. Ο Identity Provider αλληλεπιδρά με τον χρήστη για την πιστοποίηση ταυτότητας.
4. Ο χρήστης πιστοποιείται. Ο Identity Provider επικυρώνει τα στοιχεία ταυτότητας.
5. Ο Identity Provider ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη πίσω στην πύλη με μια απάντηση SAML. Ο χρήστης συνδέεται στην πύλη, η οποία εμφανίζει ένα κατάλογο εφαρμογών.
6. Ο χρήστης κάνει κλικ σε έναν σύνδεσμο στην πύλη για μια εφαρμογή. Ο σύνδεσμος κατευθύνει το πρόγραμμα περιήγησης του χρήστη στον Identity Provider με μια παράμετρο που υποδεικνύει την επιθυμητή εφαρμογή του Service Provider. Ο Identity Provider ελέγχει τη σύνοδο του χρήστη.
7. Ο Identity Provider ανακατευθύνει το πρόγραμμα περιήγησης του χρήστη στη διεύθυνση URL του Service Provider, με μια νέα απάντηση SAML για τον εν λόγω Service Provider (την εφαρμογή).

8. Ο Service Provider (εφαρμογή) επικυρώνει την απάντηση SAML και εμφανίζει μια κατάλληλη σελίδα για τον χρήστη.

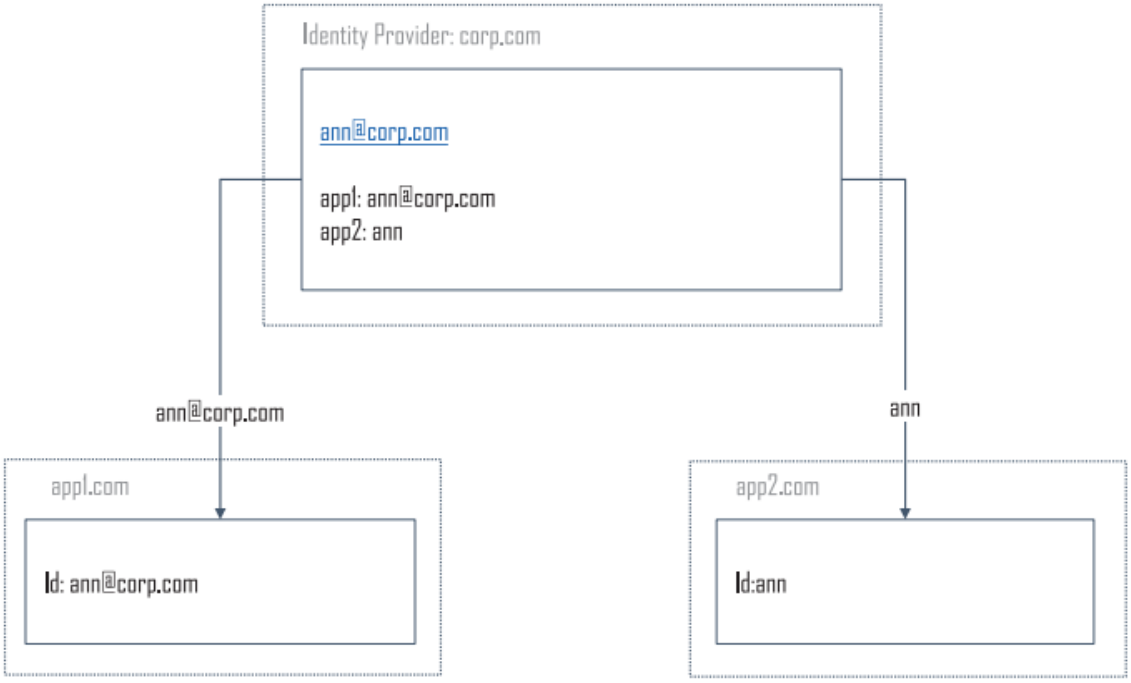
3.1.7 Ομοσπονδιακή Ταυτότητα

Η ομοσπονδιακή ταυτότητα δημιουργεί ένα συμφωνημένο αναγνωριστικό που χρησιμοποιείται μεταξύ ενός Service Provider και ενός Identity Provider για την αναφορά σε ένα χρήστη. Αυτό επιτρέπει σε έναν Service Provider να αναθέσει την πιστοποίηση ταυτότητας του χρήστη σε έναν Identity Provider [31].

Στο Σχήμα 3.3 απεικονίζεται ένα παράδειγμα [31]. Μία χρήστης με το όνομα Ann έχει λογαριασμό σε δύο εφαρμογές, την application1 που φιλοξενείται στο app1.com και την application2 που φιλοξενείται στο app2.com. Στο application1, το αναγνωριστικό του λογαριασμού της είναι ann@corp.com και στο application2, το αναγνωριστικό του λογαριασμού της είναι "ann". Η Ann έχει επίσης έναν λογαριασμό σε έναν εταιρικό Identity Provider, όπου το αναγνωριστικό λογαριασμού της είναι ann@corp.com.

Δύο εφαρμογές, η application1 στον σύνδεσμο app1.com και η application2 στο app2.com, έχουν και οι δύο λογαριασμούς που ανήκουν σε μία χρήστη με το όνομα Ann Smith. Το αναγνωριστικό του λογαριασμού της στο application1 είναι ann@corp.com, ενώ στο application2 είναι "ann". Η Ann έχει επίσης έναν λογαριασμό σε έναν εταιρικό Identity Provider με τη διεύθυνση ηλεκτρονικού ταχυδρομείου ann@corp.com.

Για τη δημιουργία πληροφοριών μεταξύ του application1 και του Identity Provider, οι διαχειριστές του application1 και του Identity Provider επικοινωνούν με μεταδεδομένα σχετικά με τα αντίστοιχα περιβάλλοντά τους. Οι διαχειριστές του Identity Provider και του application2 εκτελούν την ίδια διαδικασία. Συνήθως, οι διαχειριστές ενός Identity Provider τον ρυθμίζουν ώστε να παραδίδει SAML Assertion σε κάθε Service Provider με τις κατάλληλες ταυτότητες για τον συγκεκριμένο Service Provider.



Σχήμα 3.3: Ομοσπονδιακή Ταυτότητα [31]

Όταν η Ann αποκτά πρόσβαση στο application1, αυτό ανακατευθύνει το πρόγραμμα περιήγησής της στον Identity Provider του εργοδότη της στη διεύθυνση "corp.com". Ο Identity Provider πιστοποιεί την Ann και ανακατευθύνει το πρόγραμμα περιήγησής της πίσω στο application1 με έναν ισχυρισμό πιστοποίησης που περιέχει ένα αναγνωριστικό που την προσδιορίζει ως ann@corp.com.

Όταν η Ann αποκτά πρόσβαση στο application2, το πρόγραμμα περιήγησής της ανακατευθύνεται στον Identity Provider, ο οποίος αναγνωρίζει ότι έχει ήδη μια συνεδρία. Το ann@corp.com δε θα γίνει δεκτό ως έγκυρος χρήστης από το application2 εάν ο Identity Provider επιστρέψει αυτό το αναγνωριστικό. Για κάθε Service Provider, ο Identity Provider πρέπει να παραδίδει ένα κατάλληλο αναγνωριστικό. Σε αυτή την περίπτωση, όταν ο Identity Provider πρέπει να ταυτοποιήσει τη χρήστη με το αναγνωριστικό "ann".

3.1.8 Εναλλακτικά Πρωτόκολλα

OAuth 2.0

Το πλαίσιο εξουσιοδότησης (authorization framework) OAuth 2.0 αναπτύχθηκε το 2012 [10] ως βελτίωση του προηγούμενου πρωτοκόλλου OAuth 1.0 [9]. Χρησιμοποιείται ως πλαίσιο εξουσιοδότησης από προμηθευτές, οι οποίοι με τη σειρά τους λειτουργούν ως πάροχοι OAuth. Το πρωτόκολλο αυτό αναπτύχθηκε για τον έλεγχο της προσβασιμότητας third-party εφαρμογών σε υπηρεσίες HTTP.

OpenID Connect

Το OpenID Connect [23] είναι ένα απλό επίπεδο ταυτότητας πάνω από το πρωτόκολλο OAuth 2.0. Επιτρέπει στους πελάτες να επαληθεύουν την ταυτότητα του τελικού χρήστη με βάση τον έλεγχο ταυτότητας που εκτελείται από έναν διακομιστή εξουσιοδότησης (authorization server), καθώς και να λαμβάνουν βασικές πληροφορίες για τον τελικό χρήστη.

CAS

Το Central Authentication Service (CAS) [3] είναι ένα πρωτόκολλο ενιαίας σύνδεσης για το διαδίκτυο. Επιτρέπει σε έναν χρήστη να έχει πρόσβαση σε πολλαπλές εφαρμογές παρέχοντας τα στοιχεία ταυτότητάς του (όπως userid και password) μόνο μία φορά σε μια κεντρική εφαρμογή CAS Server.

3.2 Shibboleth

Το Shibboleth [25] είναι ένα ανοικτού κώδικα διαδικτυακό εργαλείο λογισμικού που υποστηρίζει την ενιαία σύνδεση (Single Sign-On) με τη χρήση του πρωτοκόλλου SAML και αποτελείται από τρία στοιχεία.

- Ο Identity Provider (IdP) είναι υπεύθυνος για την πιστοποίηση ταυτότητας των χρηστών και την παροχή πληροφοριών χρήστη στον Service Provider (SP). Βρίσκεται στον οργανισμό που διατηρεί τον λογαριασμό του χρήστη.
- Ο Service Provider (SP) είναι υπεύθυνος για την προστασία ενός διαδικτυακού πόρου και την κατανάλωση πληροφοριών από τον Identity Provider (IdP). Βρίσκεται στον οργανισμό που παρέχει τον διαδικτυακό πόρο.
- Η υπηρεσία εντοπισμού (Discovery Service (DS)) βοηθά τον Service Provider (SP) να ανακαλύψει τον Identity Provider (IdP) του χρήστη. Μπορεί να βρίσκεται οπουδήποτε στο διαδίκτυο και δεν απαιτείται σε όλες τις περιπτώσεις.

Τα δύο βασικά μέρη του Shibboleth είναι ο Identity Provider (IdP) και ο Service Provider (SP) [27]. Ο Identity Provider παρέχει πληροφορίες σχετικά με τους χρήστες στις υπηρεσίες και ο Service Provider συλλέγει πληροφορίες σχετικά με τους χρήστες για την προστασία των πόρων. Στην τυπική περίπτωση χρήσης, ένα πρόγραμμα περιήγησης ιστού αποκτά πρόσβαση σε έναν προστατευμένο πόρο, πιστοποιείται στον Identity Provider και καταλήγει πίσω στον πόρο συνδεδεμένο.

Ακολουθεί η διαδικασία πιστοποίησης ταυτότητας (Σχήμα 3.4):

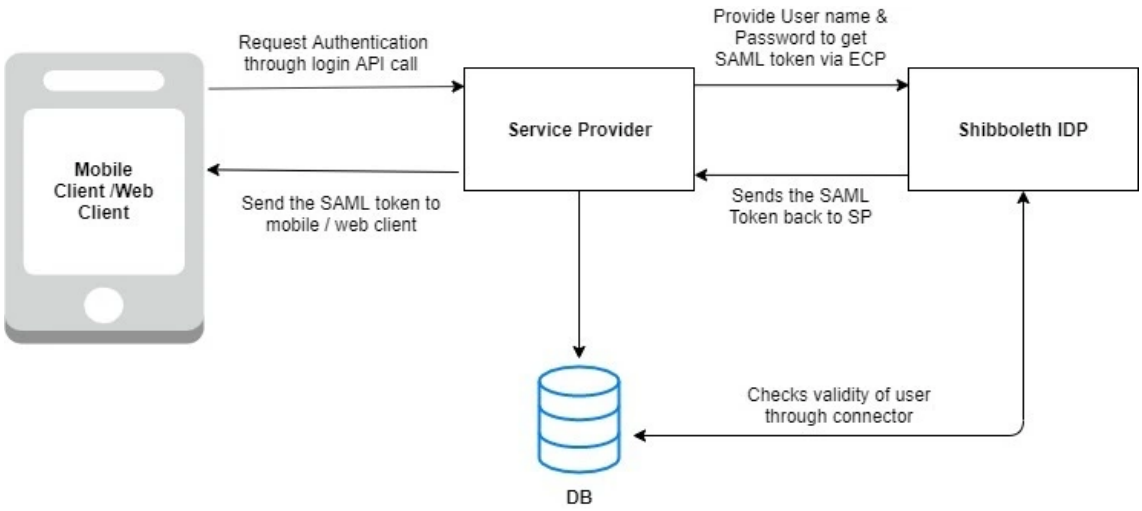
1. Ο χρήστης αποκτά πρόσβαση σε προστατευόμενο πόρο.
2. Ο Service Provider καθορίζει έναν Identity Provider και εκδίδει το αίτημα πιστοποίησης ταυτότητας (authentication request).

3. Ο χρήστης πιστοποιείται στον Identity Provider.

4. Ο Identity Provider εκδίδει απάντηση στον Service Provider.

5. Ο Service Provider αποκωδικοποιεί το μήνυμα και εξάγει τα στοιχεία.

6. Το πρόγραμμα περιήγησης ανακατευθύνεται στον προστατευόμενο πόρο.



Σχήμα 3.4: Shibboleth Flow [26]

Κεφάλαιο 4

Multi-Factor Authentication

4.1 Το Πρόβλημα με τους Κωδικούς Πρόσβασης

Ένας στατικός κωδικός πρόσβασης είναι μια μυστική συμβολοσειρά χαρακτήρων που χρησιμοποιείται επανειλημμένα με την πάροδο του χρόνου από έναν χρήστη για την πιστοποίηση ταυτότητας σε έναν συγκεκριμένο προστατευόμενο πόρο [31]. Αν και οι στατικοί κωδικοί πρόσβασης χρησιμοποιούνται συχνά στις μέρες μας, έχουν σημαντικά μειονεκτήματα. Οι μικροί κωδικοί πρόσβασης μπορούν να μαντευτούν από επιθέσεις που δοκιμάζουν κάθε πιθανό κωδικό πρόσβασης (brute force attacks). Όσον αφορά τους μεγάλους κωδικούς, πολλοί χρήστες θεωρούν δύσκολο να θυμούνται μεγάλους κωδικούς πρόσβασης και για αυτό τους καταγράφουν. Η καταγραφή τους, όμως, αυξάνει τον κίνδυνο κλοπής τους. Ένα όνομα χρήστη και ένας κωδικός πρόσβασης μπορούν να χρησιμοποιηθούν από κάποιον που βρίσκεται μακριά από τον ιδιοκτήτη του λογαριασμού, ο οποίος μπορεί να μη γνωρίζει ότι ο κωδικός πρόσβασης έχει παραβιαστεί μέχρι να χρησιμοποιηθεί από αυτόν που τον έχει κλέψει. Ακόμη χειρότερα, εάν ο παραβιασμένος κωδικός πρόσβασης έχει χρησιμοποιηθεί σε πολλούς ιστότοπους, η κλοπή του μπορεί να θέσει όλους τους ιστότοπους σε κίνδυνο. Η χρήση στατικών κωδικών πρόσβασης έχει γίνει πιο επικίνδυνη από τότε που εφαρμόστηκε η ενιαία σύνδεση (single sign-on), επειδή ένας κλεμμένος κωδικός πρόσβασης μπορεί να δώσει πρόσβαση σε πολλά συστήματα.

4.2 Multi-Factor Authentication

Ο έλεγχος ταυτότητας πολλών παραγόντων (Multi-Factor Authentication (MFA)) είναι μια ασφαλής διαδικασία πιστοποίησης ταυτότητας που απαιτεί περισσότερες από μία τεχνικές πιστοποίησης [5]. Ο έλεγχος ταυτότητας πολλαπλών παραγόντων χρησιμοποιείται όλο και περισσότερο για την πιστοποίηση της ταυτότητας των χρηστών κατά την πρόσβαση σε ένα σύστημα. Το MFA παρέχει μια αποτελεσματικότερη και ασφαλέστερη μέθοδο πιστοποίησης ταυτότητας των χρηστών συνδυάζοντας δύο ή περισσότερες μεθόδους πιστοποίησης ταυτότητας. Οι τρεις βασικοί παράγοντες πιστοποίησης ταυτότητας είναι οι εξής [19]:

- Παράγοντας γνώσης (knowledge factor) - κάτι που γνωρίζει ο χρήστης, όπως ένας κωδικός πρόσβασης

- Παράγοντας ιδιοκτησίας (ownership factor) - κάτι που έχει ο χρήστης, όπως ένα smartphone
- Βιομετρικός παράγοντας (biometric factor) - κάτι που είναι ο χρήστης, δηλαδή βιομετρικά δεδομένα

Ο στόχος του MFA είναι να επικυρώνει τους νόμιμους χρήστες για να διασφαλίζει τις ευαίσθητες πληροφορίες τους παρέχοντας μια πολυεπίπεδη άμυνα και ταυτόχρονα να δυσχεραίνει την πρόσβαση μη εξουσιοδοτημένων ατόμων [5]. Η υιοθέτηση του MFA παρέχει μια αξιόπιστη μέθοδο πιστοποίησης ταυτότητας των χρηστών. Από την πλευρά των επιτιθέμενων, πρέπει να ξεπεράσουν πολλαπλά εμπόδια για να εισέλθουν σε ένα σύστημα.

4.3 Μέθοδοι MFA

Παρακάτω παρουσιάζονται οι πιο συνήθεις μέθοδοι MFA.

4.3.1 SMS

Μία από τις πιο διαδεδομένες μεθόδους MFA είναι τα SMS [21]. Στον χρήστη αποστέλλεται ένας κωδικός επαλήθευσης μιας χρήσης (συνήθως έξι ψηφία) μέσω SMS στο κινητό του τηλέφωνο. Η ευρεία εξάπλωση οφείλεται στο γεγονός ότι οι περισσότεροι χρήστες διαθέτουν ήδη κινητό τηλέφωνο με δυνατότητα SMS. Πιθανά προβλήματα χρηστικότητας μπορεί να περιλαμβάνουν καθυστερημένη παράδοση και έλλειψη υπηρεσιών κινητής τηλεφωνίας (όπως σε μια απομακρυσμένη τοποθεσία).

Η πιστοποίηση με SMS είναι ευάλωτη σε διάφορα στάδια [21]. Επειδή τα δίκτυα κινητής τηλεφωνίας δεν κρυπτογραφούν τα μηνύματα κατά τη μεταφορά, μπορούν να πραγματοποιηθούν επιθέσεις man-in-the-middle [15]. Επιπλέον, μια επικίνδυνη επίθεση είναι και η επίθεση ανταλλαγής SIM [1]. Επίσης, ο server πρέπει να αποθηκεύει με ασφάλεια τον κωδικό μιας χρήσης. Ο κωδικός θα μπορούσε να είναι κατακερματισμένος για την αποτροπή κλοπής, αλλά ένας επιτιθέμενος θα μπορούσε να πραγματοποιήσει brute force attack δεδομένου του σχετικά μικρού αριθμού κωδικών [21]. Οι επιτιθέμενοι μπορούν επίσης να κλέψουν κωδικούς SMS μέσω στοχευμένων επιθέσεων phishing [20]. Ο περιορισμός του αριθμού των ανεπιτυχών προσπαθειών σύνδεσης με έναν κωδικό και η ακύρωση ενός κωδικού μετά από ένα ορισμένο χρονικό διάστημα είναι δύο στρατηγικές για τη μείωση αυτών των κινδύνων.

4.3.2 Email

Η μέθοδος με email είναι όμοια με τη μέθοδο με SMS, αλλά ο κωδικός αποστέλλεται μέσω email [16]. Ο χρήστης πρέπει να συνδεθεί στο email του για να δει τον κωδικό μιας χρήσης. Μπορεί να χρειαστεί περισσότερο χρόνο για να αποκτήσει ο χρήστης πρόσβαση στον λογαριασμό του, αλλά οι χρήστες του λογαριασμού μπορούν να έχουν πρόσβαση στον κωδικό μιας χρήσης χρησιμοποιώντας οποιαδήποτε κινητή συσκευή που μπορεί να λαμβάνει μηνύματα ηλεκτρονικού ταχυδρομείου.

4.3.3 Ειδοποίηση Push

Στη μέθοδο push [21], ο χρήστης λαμβάνει μια ειδοποίηση push στο smartphone του, η οποία του επιτρέπει είτε να εγκρίνει είτε να απορρίψει μια προσπάθεια σύνδεσης. Ο έλεγχος ταυτότητας με ειδοποίηση push απαιτεί πρόσβαση στο διαδίκτυο. Η Google υποστηρίζει αυτή την τεχνική (μέσω του "Google prompt") και είναι επίσης διαθέσιμη μέσω εμπορικών εφαρμογών όπως το DUO Mobile. Επειδή δεν υπάρχουν ψηφία που πρέπει να αντιγραφούν σωστά από την οθόνη του τηλεφώνου, υπάρχει μικρότερη πιθανότητα λάθους του χρήστη με αυτή τη μέθοδο. Η μη ανάγκη πληκτρολόγησης αριθμών καθιστά τη μέθοδο αυτή ως ταχύτερη και πιο εύχρηστη για τους χρήστες σε σχέση με άλλες μεθόδους MFA.

4.3.4 Βιομετρικό Μέσο

Για να γίνει χρήση πιστοποίησης ταυτότητας με βιομετρικό μέσο, ο χρήστης πρέπει να διαθέτει smartphone ή υπολογιστή που επιτρέπει βιομετρική σάρωση, όπως σάρωση δακτυλικών αποτυπωμάτων ή αναγνώριση προσώπου [12]. Ο έλεγχος ταυτότητας με βιομετρικό μέσο είναι ταχύτερος από τις μεθόδους που ο χρήστης πρέπει να περιμένει την αποστολή κωδικού και ύστερα να εισάγει τον κωδικό αυτόν στο σύστημα. Επίσης, είναι από τις πιο ασφαλείς μεθόδους, επειδή είναι από τους πιο δύσκολος τύπους δεδομένων για να τα κλέψει ένας επιτιθέμενος. Ωστόσο, ο χρήστης θα πρέπει να διαθέτει τους κατάλληλους σαρωτές και τις κατάλληλες εφαρμογές στη συσκευή του.

4.3.5 U2F

Το Universal 2nd Factor (U2F) είναι ένα ανοιχτό πρότυπο για την πιστοποίηση ταυτότητας με τη χρήση συσκευής USB [21], το οποίο υποστηρίζεται από τη FIDO (Fast IDentity Online). Ο χρήστης πρέπει να συνδέσει τη συσκευή στον υπολογιστή του και να την ενεργοποιήσει όταν του ζητηθεί.

Το πρότυπο U2F δημιουργήθηκε με γνώμονα την ισχυρή ασφάλεια και την καλή χρηστικότητα. Το πρότυπο U2F προορίζεται να σταματήσει τις επιθέσεις phishing και προσφέρει καλύτερη ασφάλεια και προστασία της ιδιωτικής ζωής από τις μεθόδους που αναφέρθηκαν παραπάνω. Τα δημόσια κλειδιά που δημιουργούνται από τον χρήστη για την πιστοποίηση ταυτότητας U2F πρέπει να αποθηκεύονται στον server, ενώ το μυστικό κλειδί διατηρείται πάντα στη συσκευή U2F. Ο πρωταρχικός κίνδυνος είναι ότι ένας χρήστης μπορεί να χάσει τη συσκευή U2F, ο οποίος κίνδυνος ισχύει και για τις υπόλοιπες μεθόδους που αναφέρθηκαν.

4.3.6 TOTP

Για τη ρύθμιση του TOTP (Time-based One-Time Password), ο χρήστης πρέπει να συγχρονίσει ένα μυστικό κλειδί που παράγεται από τον πάροχο με το smartphone του, συνήθως σαρώνοντας έναν QR κωδικό [21]. Συνδυάζοντας το μυστικό κλειδί με μια χρονοσφραγίδα, κατακερματίζοντας το αποτέλεσμα και στη συνέχεια περικόπτοντας το αποτέλεσμα, το λογισμικό δημιουργεί έναν κωδικό επαλήθευσης (συνήθως 6 ψηφία). Ο server χρησιμοποιεί την

ίδια τεχνική για την επικύρωση του κωδικού που παρέχει ο χρήστης. Το πλεονέκτημα της χρήσης μιας εφαρμογής δημιουργίας κωδικών TOTP (TOTP Authenticator) είναι ότι, μετά τον συγχρονισμό του μυστικού κλειδιού, ο χρήστης δε χρειάζεται σύνδεση στο διαδίκτυο και ούτε χρειάζεται να βασίζεται σε έναν πάροχο κινητής τηλεφωνίας για την παροχή των κωδικών μιας χρήσης, αφαιρώντας έτσι τον κίνδυνο μιας πιθανής επίθεσης. Ωστόσο, ένας επιτιθέμενος μπορεί να υποδυθεί τον χρήστη, αν καταφέρει να αποκτήσει το μυστικό κλειδί του TOTP.

Κάθε κωδικός ισχύει μόνο για ένα προκαθορισμένο χρονικό διάστημα, συνήθως 30 δευτερόλεπτα, μετά το οποίο πρέπει να δημιουργηθεί ένας νέος κωδικός. Αυτό σημαίνει ότι οι χρήστες έχουν στη διάθεσή τους μόλις 30 δευτερόλεπτα για την εισαγωγή του κωδικού. Τόσο ο server όσο και το smartphone πρέπει να έχουν ρολόγια που να είναι επαρκώς συγχρονισμένα. Οι κωδικοί επαλήθευσης πρέπει να εισάγονται χειροκίνητα από τον χρήστη, γεγονός που αυξάνει την πιθανότητα σφάλματος του χρήστη. Επειδή η μέθοδος TOTP απαιτεί τη χρήση μυστικού κλειδιού, για λόγους ασφαλείας το μυστικό κλειδί πρέπει να αποθηκεύεται κρυπτογραφημένο.

4.3.7 Τοποθεσία

Ένας επιπλέον παράγοντας πιστοποίησης ταυτότητας είναι ο παράγοντας τοποθεσίας (location factor) [4]. Η χρήση της γεωγραφικής θέσης της συσκευής (από GPS, διεύθυνση IP κ.λπ.) χρησιμοποιείται για την πιστοποίηση ταυτότητας του χρήστη. Εάν ο χρήστης έχει τα σωστά στοιχεία ταυτότητας, τότε το σύστημα θα εξετάσει την τοποθεσία στην οποία βρίσκεται για να διαπιστώσει εάν ο χρήστης είναι εξουσιοδοτημένος να έχει πρόσβαση στο σύστημα από τη συγκεκριμένη τοποθεσία.

4.3.8 Backup Κωδικοί

Σε περίπτωση που ο χρήστης δεν μπορεί να έχει πρόσβαση στην κύρια μέθοδο MFA, οι backup κωδικοί χρησιμοποιούνται συχνά ως εφεδρική μέθοδος πιστοποίησης ταυτότητας [21]. Ο πάροχος δημιουργεί έναν κατάλογο κωδικών επαλήθευσης, τους οποίους ο χρήστης πρέπει να εκτυπώσει ή να καταγράψει. Τόσο ο server όσο και ο χρήστης πρέπει να τους φυλάξουν με ασφάλεια για να αποφύγουν την κλοπή.

4.4 Ενσωμάτωση MFA με Shibboleth

Παρακάτω παρουσιάζονται τρόποι για την ενσωμάτωση Multi-Factor Authentication στο Shibboleth IdP.

4.4.1 Duo MFA

Το Duo MFA [6] είναι μια υπηρεσία που προσφέρει έλεγχο ταυτότητας πολλαπλών παραγόντων με διάφορες εφαρμογές. Μία από αυτές είναι το Shibboleth IdP, με το οποίο έχει επίσημη ενσωμάτωση μέσω ενός πρόσθετου (plugin) [7]. Το Duo MFA προσφέρει όλους του τρόπους ελέγχου ταυτότητας πολλαπλών παραγόντων που προαναφέρθηκαν. Το Duo MFA

είναι μια closed source εφαρμογή και αποτελεί μια καλή λύση για να ενσωματώσει μια οργάνωση Multi- Factor Authentication στο Shibboleth IdP. Παρόλα αυτά, η χρήση της υπηρεσίας για χρήστες περισσότερων των δέκα είναι επί πληρωμή και ξεκινά με τιμή 3$ ανά χρήστη (τιμή Σεπτεμβρίου 2022). Στην ενότητα 5.2 παρουσιάζονται τα βήματα που πραγματοποιήθηκαν για να γίνει η ενσωμάτωση του Shibboleth IdP με το Duo MFA.

4.4.2 Multi Factor Provider (MFaP)

Το MFaP [22] είναι μια open source εφαρμογή που δημιουργήθηκε για να προσφέρει έλεγχο ταυτότητας πολλαπλών παραγόντων στο Shibboleth IdP. Η εφαρμογή αυτή προσφέρει τους εξής τρόπους MFA: ειδοποίηση push, U2F, TOTP, Backup κωδικοί. Παρόλα αυτά, η εφαρμογή αυτή δημιουργήθηκε για παλιότερη έκδοση του Shibboeth IdP (v3) [24] η οποία δεν υποστηρίζεται πλέον.

4.4.3 Custom MFA Application

Το Duo MFA όντας μια υπηρεσία επί πληρωμή και το MFaP να είναι ξεπερασμένο καθώς λειτουργεί για έκδοση η οποία δεν υποστηρίζεται, δημιουργείται η ανάγκη για εφαρμογή η οποία προσφέρει λειτουργία MFA και ενσωμάτωση με το Shibboleth IdP. Στα πλαίσια της ΔΕ δημιουργήθηκε μια open source εφαρμογή (TOTP Manager [28]) η οποία είναι δωρεάν και λειτουργεί με τη νεότερη έκδοση του Shibboleth IdP (v4). Ως τρόπο επαλήθευσης MFA προσφέρει το TOTP. Περισσότερες λεπτομέρειες για τα χαρακτηριστικά της εφαρμογής, το πως λειτουργεί και πως ενσωματώνεται με το Shibboleth IdP παρουσιάζονται στην ενότητα 5.3.

Κεφάλαιο 5

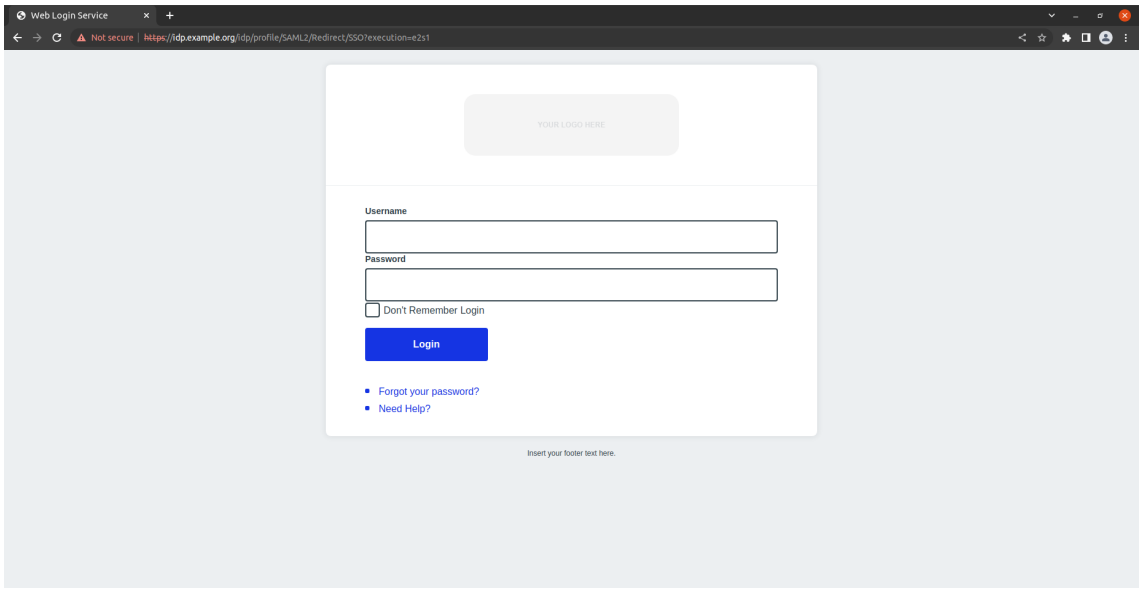
Τεχνική Περιγραφή Υλοποίησης

Στο κεφάλαιο αυτό παρουσιάζονται τα βήματα που πραγματοποιήθηκαν για την εγκατάσταση και τη ρύθμιση όλων των απαραίτητων τεχνολογιών και εργαλείων.

Τα παρακάτω βήματα αφορούν την εγκατάσταση σε τοπικό μηχάνημα και το λειτουργικό σύστημα το οποίο χρησιμοποιήθηκε είναι το Ubuntu 20.04.

5.1 Βασική Υλοποίηση

Η υλοποίηση αυτή αφορά την εγκατάσταση του OpenLDAP, του Shibboleth IdP και του Shibboleth SP καθώς και τη ρύθμιση αυτών ώστε να γίνει η σύνδεση μεταξύ τους. Εφόσον, ολοκληρωθεί η εγκατάσταση όλων των τεχνολογιών, ο χρήστης επισκέπτοντας τη σελίδα `https://sp.example.org/secure` όπως φαίνεται στο Σχήμα 5.1 και εισάγοντας επιτυχώς το username και το password που έχουν τεθεί από τον διαχειριστή μπορεί να συνδεθεί στη σελίδα που έχει θέσει ο SP.



Σχήμα 5.1: Σελίδα Εισόδου Χρήστη

Τα βήματα για την εγκατάσταση και τη ρύθμιση των Shibboleth IdP και Shibboleth SP πραγματοποιήθηκαν ακολουθώντας του αντίστοιχους οδηγούς που βρίσκονται στο idem-tutorials GitHub Repository [13] προσαρμόζοντάς τα στις ανάγκες της υλοποίησης.

5.1.1 OpenLDAP

- Σύνδεση ως root.

```
sudo su -
```

- Ενημέρωση πακέτων.

```
apt update && apt-get upgrade -y --no-install-recommends
```

- Αλλαγή του hostname.

```
hostnamectl set-hostname ldap.example.org
```

- Αντιστοίχηση της διεύθυνσης IP στο hostname.

  Προσθήκη της παρακάτω γραμμής στο αρχείο `/etc/hosts`.

```
127.0.0.1 ldap.example.org
```

- Εγκατάσταση των απαιτούμενων πακέτων.

```
apt install -y slapd ldap-utils
```

  Κατά την εγκατάσταση ο κωδικός που επιλέχθηκε για τον admin ήταν "admin".

- Δημιουργία base DN για τους χρήστες.

  Δημιουργία αρχείου `basedn.ldif` με τα παρακάτω περιεχόμενα.

```
dn: ou=people,dc=example,dc=org
objectClass: organizationalUnit
ou: people
```

  Προσθήκη αρχείου `basedn.ldif` στο OpenLDAP.

```
ldapadd -x -D "cn=admin,dc=example,dc=org" -W -f basedn.ldif
```

- Δημιουργία χρηστών.

  Δημιουργία κωδικού χρήστη.

```
slappasswd
```

  Αποτέλεσμα εντολής.

```
{SSHA}uxx4VIgCv4VTbcqKno7kwm1rg1u+x2fQ
```

  Δημιουργία αρχείου `ldapuser1.ldif` με τα παρακάτω περιεχόμενα.

```
dn: uid=user1,ou=people,dc=example,dc=org
objectClass: inetOrgPerson
cn: user1
sn: user1
mail: user1@example.org
userPassword: {SSHA}uxx4VIgCv4VTbcqKno7kwm1rg1u+x2fQ
```

Προσθήκη αρχείου `ldapuser1.ldif` στο OpenLDAP.

```
ldapadd -x -D "cn=admin,dc=example,dc=org" -W -f ldapuser1.ldif
```

Η ίδια διαδικασία ακολουθείται για τη δημιουργία περισσότερων χρηστών.

5.1.2 Shibboleth IdP

Εγκατάσταση απαιτήσεων λογισμικού

- Σύνδεση ως root.

```
sudo su -
```

- Ενημέρωση πακέτων.

```
apt update && apt-get upgrade -y --no-install-recommends
```

- Εγκατάσταση των απαιτούμενων πακέτων.

```
apt install -y vim wget gnupg ca-certificates openssl apache2 ntp
    libservlet3.1-java liblogback-java --no-install-recommends
```

- Εγκατάσταση Amazon Corretto JDK.

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -
apt-get install software-properties-common
add-apt-repository "deb https://apt.corretto.aws stable main"
apt-get update; sudo apt-get install -y java-11-amazon-corretto-jdk
java -version
```

Ρύθμιση του περιβάλλοντος

- Βεβαίωση ότι το τείχος προστασίας δεν εμποδίζει την κυκλοφορία στις θύρες 443 και 80 για το διακομιστή IdP.

```
ufw allow proto tcp from any to any port 80,443
```

- Αλλαγή του hostname.

```
hostnamectl set-hostname idp.example.org
```

- Αντιστοίχηση της διεύθυνσης IP στο hostname.

  Προσθήκη της παρακάτω γραμμής στο αρχείο `/etc/hosts`

```
127.0.0.1 idp.example.org
```

- Ορισμός της μεταβλητής JAVA_HOME στο αρχείο /etc/environment

```
echo "JAVA_HOME=/usr/lib/jvm/java-11-amazon-corretto" > /etc/
    environment
source /etc/environment
export JAVA_HOME=/usr/lib/jvm/java-11-amazon-corretto
echo $JAVA_HOME
```

Εγκατάσταση Shibboleth Identity Provider v4.2.1

- Λήψη Shibboleth Identity Provider v4.2.1

```
cd /usr/local/src
wget http://shibboleth.net/downloads/identity-provider/4.2.1/
    shibboleth-identity-provider-4.2.1.tar.gz
tar -xzf shibboleth-identity-provider-4.*.tar.gz
```

- Εκτέλεση του install.sh

```
cd /usr/local/src/shibboleth-identity-provider-4.*/bin
bash install.sh -Didp.host.name=$(hostname -f) -Didp.keysize=3072
```

Εγκατάσταση του Jetty 9 Web Server

- Λήψη και εξαγωγή του Jetty.

```
cd /usr/local/src
wget https://repo1.maven.org/maven2/org/eclipse/jetty/jetty-
    distribution/9.4.43.v20210629/jetty-distribution-9.4.43.v20210629.
    tar.gz
tar xzvf jetty-distribution-9.4.43.v20210629.tar.gz
```

- Δημιουργία φακέλου jetty-src ως symbolic link.

```
ln -nsf jetty-distribution-9.4.43.v20210629 jetty-src
```

- Δημιουργία χρήστη συστήματος jetty.

```
useradd -r -M jetty
```

- Δημιουργία προσαρμοσμένων ρυθμίσεων Jetty.

```
mkdir /opt/jetty
wget https://registry.idem.garr.it/idem-conf/shibboleth/IDP4/jetty/
    start.ini -O /opt/jetty/start.ini
```

- Δημιουργία προσωρινού φακέλου.

```
mkdir /opt/jetty/tmp ; chown jetty:jetty /opt/jetty/tmp
chown -R jetty:jetty /opt/jetty /usr/local/src/jetty-src
```

- Δημιουργία φακέλου καταγραφής.

```
mkdir /var/log/jetty
mkdir /opt/jetty/logs
chown jetty:jetty /var/log/jetty /opt/jetty/logs
```

- Ρύθμιση `/etc/default/jetty`.

```
sudo bash -c "cat > /etc/default/jetty <<EOF
JETTY_HOME=/usr/local/src/jetty-src
JETTY_BASE=/opt/jetty
JETTY_USER=jetty
JETTY_START_LOG=/var/log/jetty/start.log
TMPDIR=/opt/jetty/tmp
EOF"
```

- Δημιουργία υπηρεσίας που μπορεί να φορτωθεί από τη γραμμή εντολών.

```
cd /etc/init.d
ln -s /usr/local/src/jetty-src/bin/jetty.sh jetty
update-rc.d jetty defaults
```

- Εκκίνηση του Jetty.

```
service jetty start
```

Ρύθμιση του Jetty

- Ρύθμιση του IdP Context Descriptor.

```
mkdir /opt/jetty/webapps
sudo bash -c 'cat > /opt/jetty/webapps/idp.xml <<EOF
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="war"><SystemProperty name="idp.home"/>/war/idp.war</Set>
  <Set name="contextPath">/idp</Set>
  <Set name="extractWAR">false</Set>
  <Set name="copyWebDir">false</Set>
  <Set name="copyWebInf">true</Set>
  <Set name="persistTempDirectory">false</Set>
</Configure>
EOF'
```

- Προσθήκη δικαιωμάτων στον χρήστη jetty.

```
cd /opt/shibboleth-idp
chown -R jetty logs/ metadata/ credentials/ conf/ war/
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

Ρύθμιση SSL στον Apache2

- Δημιουργία του αρχείου Virtualhost.

```
wget https://registry.idem.garr.it/idem-conf/shibboleth/IDP4/apache2/idp.example.org.conf -O /etc/apache2/sites-available/$(hostname -f).conf
```

- Αφαίρεση της παρακάτω γραμμής από το αρχείο `/etc/apache2/sites-available/idp.example.org.conf`

```
SSLCACertificateFile /etc/ssl/certs/GEANT_OV_RSA_CA_4.pem
```

- Δημιουργία πιστοποιητικών SSL.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/$(hostname -f).key -out /etc/ssl/certs/$(hostname -f).crt
```

- Διαμόρφωση των σωστών δικαιωμάτων για το πιστοποιητικό και το κλειδί SSL που χρησιμοποιούνται από το HTTPS.

```
chmod 400 /etc/ssl/private/$(hostname -f).key
chmod 644 /etc/ssl/certs/$(hostname -f).crt
```

- Ενεργοποίηση των απαιτούμενων Απαχε2 μοδυλες.

```
a2enmod proxy_http ssl headers alias include negotiation
a2ensite $(hostname -f).conf
a2dissite 000-default.conf default-ssl
systemctl restart apache2.service
```

Ρύθμιση σύνδεσης με OpenLDAP

- Προσθήκη κωδικού του admin από το OpenLDAP.

  Ενημέρωση της παρακάτω μεταβλητής του αρχείου `/opt/shibboleth-idp/credentials/secrets.properties` με τον κωδικό του admin από το OpenLDAP.

```
idp.authn.LDAP.bindDNCredential = admin
```

- Ρυθμίσεις `ldap.properties`.

  Τροποποίηση των παρακάτω μεταβλητών του αρχείου `/opt/shibboleth-idp/conf/ldap.properties` ως εξής.

```
idp.authn.LDAP.authenticator = bindSearchAuthenticator
idp.authn.LDAP.ldapURL = ldap://ldap.example.org
idp.authn.LDAP.useStartTLS = false
idp.authn.LDAP.ldapURL = ldap://ldap.example.org
idp.authn.LDAP.bindDN = cn=admin,dc=example,dc=org
idp.attribute.resolver.LDAP.exportAttributes = uid cn sn mail
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

Ρύθμιση του Shibboleth IdP για την αποδέσμευση του persistent NameID

- Δημιουργία computed `persistent-id`.

  Διαμόρφωση του αρχείου `/opt/shibboleth-idp/conf/saml-nameid.properties` ως εξής.

```
idp.persistentId.sourceAttribute = uid
idp.persistentId.encoding = BASE32
idp.persistentId.generator = shibboleth.ComputedPersistentIdGenerator
```

  Ενεργοποίηση της παρακάτω γραμμής στο αρχείο `/opt/shibboleth-idp/conf/saml-nameid.xml`

```
<ref bean="shibboleth.SAML2PersistentGenerator" />
```

  Εκτέλεση εντολής.

```
openssl rand -base64 36
```

  Αποτέλεσμα εντολής.

```
gkKbGpUmFQVS9i53gUF4Rh2k3Sr0DIGtmNY7ERJ+HpuYikTt
```

  Ενεργοποίηση και τροποποίηση της παρακάτω μεταβλητής του αρχείου `/opt/shibboleth-idp/credentials/secrets.properties` ως εξής.

```
idp.persistentId.salt = gkKbGpUmFQVS9i53gUF4Rh2k3Sr0DIGtmNY7ERJ+
    HpuYikTt
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

Ρύθμιση του Attribute Resolver

- Λήψη δείγματος Attribute Resolver.

```
wget https://registry.idem.garr.it/idem-conf/shibboleth/IDP4/attribute
    -resolver-v4-idem-sample.xml -O /opt/shibboleth-idp/conf/attribute
    -resolver.xml
```

- Αφαίρεση των παρακάτω γραμμών από το αρχείο `/opt/shibboleth-idp/conf/attribute-resolver.xml`.

```
useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS}"
trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
```

- Ρύθμιση του σωστού owner.

```
chown jetty /opt/shibboleth-idp/conf/attribute-resolver.xml
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

Ρύθμιση του Shibboleth IdP για την αποδέσμευση του PersonTargetedID

- Δημιουργία αρχείου eduPersonTargetedID.properties

```
wget https://registry.idem.garr.it/idem-conf/shibboleth/IDP4/
    attributes/custom/eduPersonTargetedID.properties -O /opt/
    shibboleth-idp/conf/attributes/custom/eduPersonTargetedID.
    properties
```

- Ρύθμιση του σωστού owner.

```
chown jetty:root /opt/shibboleth-idp/conf/attributes/custom/
    eduPersonTargetedID.properties
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

Ρύθμιση του attribute resolution με το Attribute Registry

- Λήψη του αρχείου `schac.xml`

```
wget https://registry.idem.garr.it/idem-conf/shibboleth/IDP4/
    attributes/schac.xml -O /opt/shibboleth-idp/conf/attributes/schac.
    xml
```

- Ρύθμιση του σωστού owner.

```
chown jetty:root /opt/shibboleth-idp/conf/attributes/schac.xml
```

Ρύθμιση καταγραφών Shibboleth IdP

- Εμπλουτισμός των αρχείων καταγραφής IdP με το σφάλμα ελέγχου ταυτότητας που συνέβη στο OpenLDAP.

```
sed -i '/^    <logger name="org.ldaptive".*/a \\n    <!-- Logs on LDAP
     user authentication - ADDED BY IDEM HOWTO -->' /opt/shibboleth-
    idp/conf/logback.xml
sed -i '/^    <!-- Logs on LDAP user authentication - ADDED BY IDEM
    HOWTO -->/a \ \ \ \ \<logger name="org.ldaptive.auth.Authenticator
    " level="INFO" />' /opt/shibboleth-idp/conf/logback.xml
```

Απενεργοποίηση του καταργημένου πρωτοκόλλου SAML1

Τροποποίηση του αρχείου `/opt/shibboleth-idp/metadata/idp-metadata.xml` ως εξής.

- Στην ενότητα `<EntityDescriptor>` αφαίρεση του validUntil XML attribute.
- Στην ενότητα `<IDPSSODescriptor>`
  - Τροποποίηση του ΑρτιφαστΡεσολυτιονΣεριςε ως εξής.

```
<ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0
    :bindings:SOAP" Location="https://idp.example.org/idp/profile/
    SAML2/SOAP/ArtifactResolution" index="1"/>
```

  - Προσθήκη των παρακάτω γραμμών ακριβώς μετά της γραμμής του προηγούμενου βήματος.

```
<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient<
    /NameIDFormat>
<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
    </NameIDFormat>
```

5.1.3 Shibboleth SP

Εγκατάσταση απαιτήσεων λογισμικού

- Σύνδεση ως root.

```
sudo su -
```

- Ενημέρωση πακέτων.

```
apt update && apt-get upgrade -y --no-install-recommends
```

- Εγκατάσταση των απαιτούμενων πακέτων.

```
apt install ca-certificates vim openssl
```

Ρύθμιση του περιβάλλοντος

- Βεβαίωση ότι το τείχος προστασίας δεν εμποδίζει την κυκλοφορία στις θύρες 443 και 80 για το διακομιστή IdP.

```
ufw allow proto tcp from any to any port 80,443
```

- Αλλαγή του hostname.

```
hostnamectl set-hostname sp.example.org
```

- Αντιστοίχηση της διεύθυνσης IP στο hostname.

  Προσθήκη της παρακάτω γραμμής στο αρχείο `/etc/hosts`.

```
127.0.0.1 sp.example.org
```

Εγκατάσταση Shibboleth Service Provider

- Εγκατάσταση πακέτων.

```
apt install -y apache2 libapache2-mod-shib ntp --no-install-recommends
```

Ρύθμιση SSL στον Apache2

- Δημιουργία του αρχείου Virtualhost.

```
wget https://registry.idem.garr.it/idem-conf/shibboleth/SP3/apache2/sp
    .example.org.conf -O /etc/apache2/sites-available/000-$(hostname -
    f).conf
```

- Αφαίρεση της παρακάτω γραμμής από το αρχείο `/etc/apache2/sites-available/000-sp.example.org.conf`

```
SSLCACertificateFile /etc/ssl/certs/GEANT_OV_RSA_CA_4.pem
```

- Δημιουργία πιστοποιητικών SSL.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/
    private/$(hostname -f).key -out /etc/ssl/certs/$(hostname -f).crt
```

- Διαμόρφωση των σωστών δικαιωμάτων για το πιστοποιητικό και το κλειδί SSL που χρησιμοποιούνται από το HTTPS.

```
chmod 400 /etc/ssl/private/$(hostname -f).key
chmod 644 /etc/ssl/certs/$(hostname -f).crt
```

- Ενεργοποίηση των απαιτούμενων Apache2 modules.

```
a2enmod ssl headers alias include negotiation
a2dissite 000-default.conf default-ssl
a2ensite 000-$(hostname -f).conf
systemctl restart apache2.service
```

Ρύθμιση Shibboleth IdP

- Αλλαγή του SP entityID και διεύθυνσης email.

```
sed -i "s/sp.example.org/$(hostname -f)/" /etc/shibboleth/shibboleth2.
    xml
sed -i "s/root@localhost/admin@example.org/" /etc/shibboleth/
    shibboleth2.xml
sed -i 's/handlerSSL="false"/handlerSSL="true"/' /etc/shibboleth/
    shibboleth2.xml
sed -i 's/cookieProps="http"/cookieProps="https"/' /etc/shibboleth/
    shibboleth2.xml
```

- Δημιουργία διαπιστευτηρίων υπογραφής και κρυπτογράφησης μεταδεδομένων SP.

```
cd /etc/shibboleth
shib-keygen -u _shibd -g _shibd -h $(hostname -f) -y 30 -e https://$(
    hostname -f)/shibboleth -n sp-signing -f
shib-keygen -u _shibd -g _shibd -h $(hostname -f) -y 30 -e https://$(
    hostname -f)/shibboleth -n sp-encrypt -f
/usr/sbin/shibd -t
systemctl restart shibd.service
systemctl restart apache2.service
```

- Ενεργοποίηση ρυθμίσεων Shibboleth Apache2.

```
a2enmod shib
systemctl reload apache2.service
```

Δημιουργία "secure" σελίδας

- Δημιουργία αρχείου `/etc/apache2/conf-available/secure.conf` με τα παρακάτω περιεχόμενα.

```
RedirectMatch    ^/$  /secure

<Location /secure>
  Authtype shibboleth
  ShibRequireSession On
  require valid-user
</Location>
```

Εκτέλεση εντολής.

```
a2enconf secure
```

- Δημιουργία σελίδας.

```
mkdir -p /var/www/html/$(hostname -f)/secure
wget https://registry.idem.garr.it/idem-conf/shibboleth/SP3/secure/
    index.php.txt -O /var/www/html/$(hostname -f)/secure/index.php
```

- Εγκατάσταση των απαραίτητων πακέτων και επανεκκίνηση του Apache2.

```
apt install libapache2-mod-php php
systemctl restart apache2.service
```

Ενεργοποίηση Attribute Support στο Shibboleth SP

- Ενεργοποίηση των σχολιασμένων εντολών στο αρχείο `/etc/shibboleth/attribute-map.xml`

5.1.4 Σύνδεση SP με IdP

- Επεξεργασία του αρχείου `/etc/shibboleth/shibboleth2.xml` ως εξής.

```
<SSO entityID="https://idp.example.org/idp/shibboleth">
   SAML2
</SSO>

<MetadataProvider type="XML" validate="true" path="/opt/shibboleth-idp
    /metadata/idp-metadata.xml"/>
```

- Επανεκκίνηση shibd και Apache2.

```
systemctl restart shibd
systemctl restart apache2
```

5.1.5 Σύνδεση IdP με SP

- Λήψη metadata SP.

```
curl -k -O https://sp.example.org/Shibboleth.sso/Metadata
```

- Μετονομασία metadata αρχείου.

```
mv Metadata sp-metadata.xml
```

- Αντιγραφή metadata στον φάκελο Shibboleth.

```
cp sp-metadata.xml /opt/shibboleth-idp/metadata
```

- Σύνδεση του SP με τον IdP προσθέτοντας τα metadata στο αρχείο `/opt/shibboleth-idp/conf/metadata-providers.xml` ως εξής.

```
<MetadataProvider id="LocalMetadata"  xsi:type="
    FilesystemMetadataProvider" metadataFile="/opt/shibboleth-idp/
    metadata/sp-metadata.xml"/>
```

- Προσθήκη των παρακάτω στο αρχείο `/opt/shibboleth-idp/conf/attribute-filter.xml`

```
<AttributeFilterPolicy id="example-afp">
    <PolicyRequirementRule xsi:type="Requester" value="https://sp.
    example.org/shibboleth" />

  <AttributeRule attributeID="uid" permitAny="true" />
    <AttributeRule attributeID="mail" permitAny="true" />
    <AttributeRule attributeID="eduPersonPrincipalName" permitAny="
    true" />
    <AttributeRule attributeID="displayName" permitAny="true" />
    <AttributeRule attributeID="eduPersonOrcid" permitAny="true" />
    <AttributeRule attributeID="sn" permitAny="true" />
    <AttributeRule attributeID="givenName" permitAny="true" />
```

```
11      <AttributeRule attributeID="eduPersonEntitlement" permitAny="true"
        />
12      <AttributeRule attributeID="cn" permitAny="true" />
13      <AttributeRule attributeID="eduPersonOrgDN" permitAny="true" />
14      <AttributeRule attributeID="title" permitAny="true" />
15      <AttributeRule attributeID="telephoneNumber" permitAny="true" />
16      <AttributeRule attributeID="eduPersonOrgUnitDN" permitAny="true" /
        >
17      <AttributeRule attributeID="schacPersonalTitle" permitAny="true" /
        >
18      <AttributeRule attributeID="schacPersonalUniqueID" permitAny="true
        " />
19      <AttributeRule attributeID="schacHomeOrganization" permitAny="true
        " />
20      <AttributeRule attributeID="schacHomeOrganizationType" permitAny="
        true" />
21      <AttributeRule attributeID="schacUserPresenceID" permitAny="true"
        />
22      <AttributeRule attributeID="mobile" permitAny="true" />
23      <AttributeRule attributeID="schacMotherTongue" permitAny="true" />
24      <AttributeRule attributeID="preferredLanguage" permitAny="true" />
25      <AttributeRule attributeID="schacGender" permitAny="true" />
26      <AttributeRule attributeID="schacDateOfBirth" permitAny="true" />
27      <AttributeRule attributeID="schacPlaceOfBirth" permitAny="true" />
28      <AttributeRule attributeID="schacCountryOfCitizenship" permitAny="
        true" />
29      <AttributeRule attributeID="schacSn1" permitAny="true" />
30      <AttributeRule attributeID="schacSn2" permitAny="true" />
31      <AttributeRule attributeID="schacCountryOfResidence" permitAny="
        true" />
32      <AttributeRule attributeID="schacPersonalUniqueCode" permitAny="
        true" />
33      <AttributeRule attributeID="schacExpiryDate" permitAny="true" />
34      <AttributeRule attributeID="schacUserPrivateAttribute" permitAny="
        true" />
35      <AttributeRule attributeID="schacUserStatus" permitAny="true" />
36      <AttributeRule attributeID="schacProjectMembership" permitAny="
        true" />
37      <AttributeRule attributeID="schacProjectSpecificRole" permitAny="
        true" />
38      <AttributeRule attributeID="schacYearOfBirth" permitAny="true" />
39      <AttributeRule attributeID="eduPersonNickname" permitAny="true" />
40      <AttributeRule attributeID="eduPersonPrimaryAffiliation" permitAny
        ="true" />
41      <AttributeRule attributeID="eduPersonPrimaryOrgUnitDN" permitAny="
        true" />
42      <AttributeRule attributeID="eduPersonAssurance" permitAny="true" /
        >
43      <AttributeRule attributeID="eduPersonPrincipalNamePrior" permitAny
        ="true" />
```

```
    <AttributeRule attributeID="eduPersonUniqueId" permitAny="true" />
    <AttributeRule attributeID="eduPersonUniqueCode" permitAny="true"
    />
    <AttributeRule attributeID="eduPersonTargetedID" permitAny="true"
    />
    <AttributeRule attributeID="eduPersonAffiliation" permitAny="true"
     />
    <AttributeRule attributeID="eduPersonScopedAffiliation" permitAny=
    "true" />
</AttributeFilterPolicy>
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

5.2 Ενσωμάτωση Shibboleth IdP με Duo MFA

Το Duo MFA [6] είναι μια υπηρεσία που καθιστά τον έλεγχο ταυτότητας πολλαπλών παραγόντων εύκολο τόσο για τους διαχειριστές όσο και για τους χρήστες. Το Duo MFA παρέχει μια εύχρηστη, ασφαλή εφαρμογή ελέγχου ταυτότητας για κινητά για γρήγορη έγκριση με χρήση OTP και Push Notification για την επαλήθευση της ταυτότητας του χρήστη. Το Duo MFA προσφέρεται ως δωρεάν υπηρεσία για ως δέκα χρήστες.

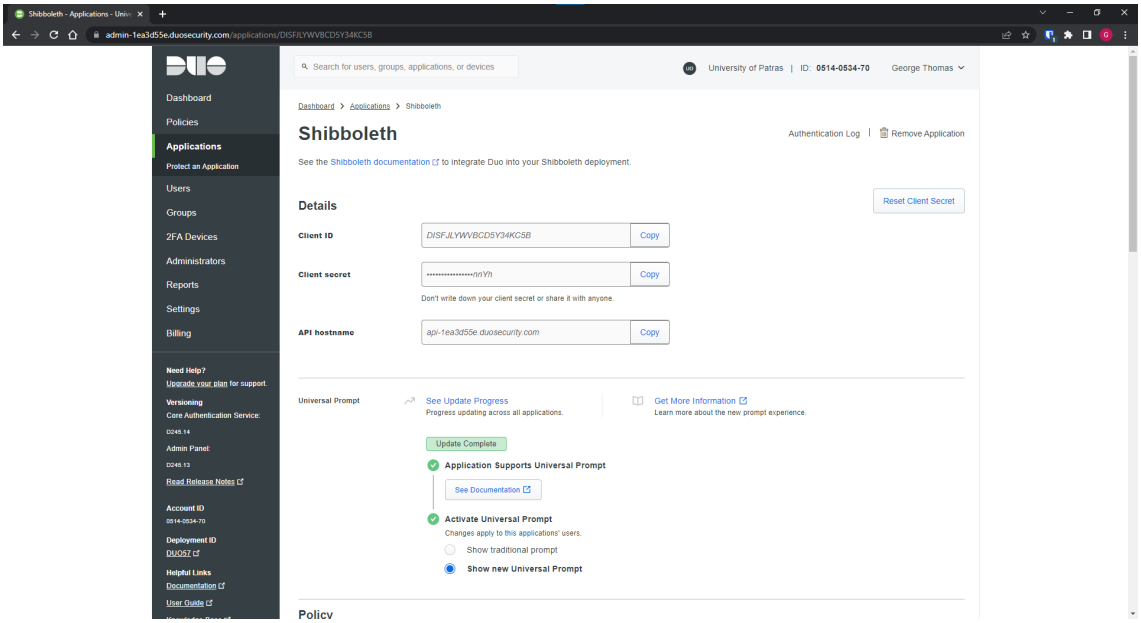
Το Shibboleth IdP προσφέρει επίσημη ενσωμάτωση με την υπηρεσία Duo MFA με τη χρήση plugin [7]. Στην ενότητα αυτή παρουσιάζονται τα βήματα που πραγματοποιήθηκαν για να γίνει η ενσωμάτωση αυτή.

5.2.1 Πλατφόρμα Duo

Για να γίνει η ενσωμάτωση με το Shibboleth IdP πρέπει πρώτα να δημιουργηθεί ένα "Application" στην ιστοσελίδα του Duo ακολουθώντας τα παρακάτω βήματα.

- Εγγραφή στην ιστοσελίδα του Duo [8].
- Μετάβαση στη σελίδα "Applications".
- Μετάβαση στη σελίδα "Protect an Application".
- Επιλογή του "Shibboleth".

Δημιουργώντας το Shibboleth Application, υπάρχει πρόσβαση στα στοιχεία που θα χρησιμοποιηθούν για να γίνει η σύνδεση με το Shibboleth IdP όπως φαίνεται στο Σχήμα 5.2.



Σχήμα 5.2: Σελίδα Shibboleth Application

5.2.2 Shibboleth IdP Duo Plugin

Ενεργοποίηση MFA Flow

- Σύνδεση ως root.

```
sudo su -
```

- Ενεργοποίηση του MFA module.

```
cd /opt/shibboleth-idp
bin/module.sh -t idp.authn.MFA || bin/module.sh -e idp.authn.MFA
```

- Ενεργοποίηση και επεξεργασία της παρακάτω μεταβλητής του αρχείου `/opt/shibboleth-idp/conf/authn/authn.properties` ως εξής.

```
idp.authn.flows = MFA
```

Λήψη και Ενεργοποίηση του Duo Plugin

- Λήψη προαπαιτούμενου OIDCCommon plugin.

```
/opt/shibboleth-idp/bin/plugin.sh -I net.shibboleth.oidc.common
```

- Λήψη Duo plugin.

```
/opt/shibboleth-idp/bin/plugin.sh -I net.shibboleth.idp.plugin.authn.
    duo.nimbus
```

- Ρύθμιση ενσωμάτωσης Duo.

  Επεξεργασία του αρχείου `/opt/shibboleth-idp/conf/authn/duo-oidc.properties` με τις παρακάτω μεταβλητές ορισμένες με τα στοιχεία από την ενότητα 5.2.1.

```
idp.duo.oidc.apiHost = hostname
idp.duo.oidc.clientId = clientid
idp.duo.oidc.redirectURL = https://idp.example.org/idp/profile/Authn/Duo/2FA/duo-callback
idp.duo.oidc.secretKey = key
```

- Επεξεργασία του αρχείου `/opt/shibboleth-idp/conf/authn/mfa-authn-config.xml` ως εξής.

```
<util:map id="shibboleth.authn.MFA.TransitionMap">
    <!-- First rule runs the Password login flow. -->
    <entry key="">
        <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow="authn/Password" />
    </entry>

    <!-- Second rule runs a function if Password succeeds, to determine whether an additional factor is required. -->
    <entry key="authn/Password">
        <bean parent="shibboleth.authn.MFA.Transition" p:nextFlowStrategy-ref="checkSecondFactor" />
    </entry>
    <!-- An implicit final rule will return whatever the final flow returns. -->
</util:map>

<!-- Example script to see if second factor is required. Currently just returns the DuoOIDC flow -->
<bean id="checkSecondFactor" parent="shibboleth.ContextFunctions.Scripted" factory-method="inlineScript">
    <constructor-arg>
        <value>
        <![CDATA[
            nextFlow = "authn/DuoOIDC";

            // Check if second factor is necessary for request to be satisfied.
            //authCtx = input.getSubcontext("net.shibboleth.idp.authn.context.AuthenticationContext");
           // mfaCtx = authCtx.getSubcontext("net.shibboleth.idp.authn.context.MultiFactorAuthenticationContext");
            //if (mfaCtx.isAcceptable()) {
            //    nextFlow = null;
            //}

            nextFlow;   // pass control to second factor or end with the first
```

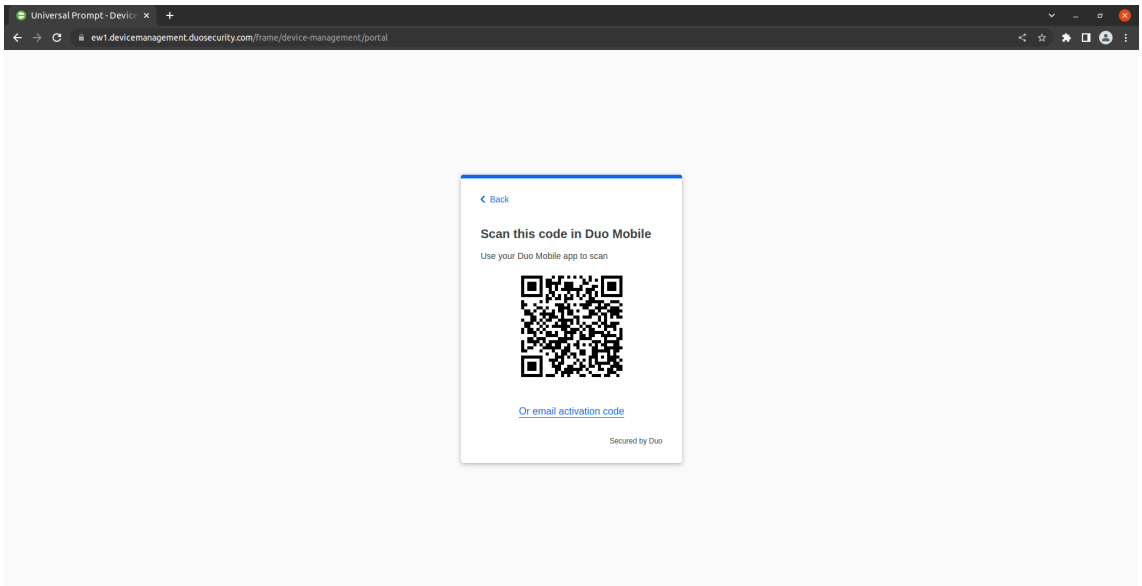
```
        ]]>
        </value>
    </constructor-arg>
</bean>
```

- Επανεκκίνηση Jetty.

```
systemctl restart jetty.service
```

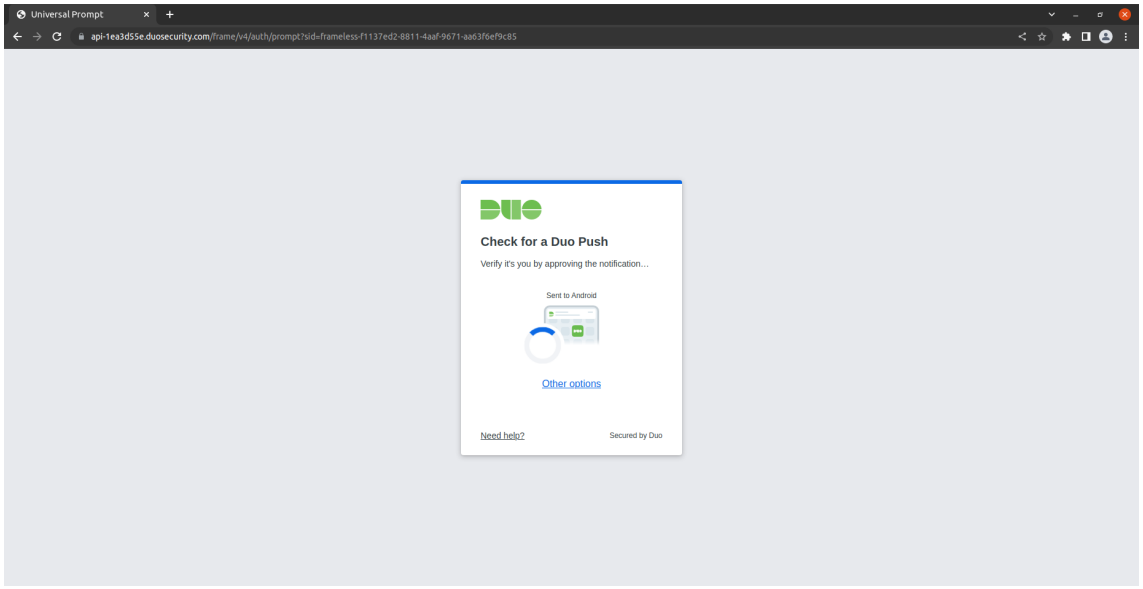
5.2.3 Διαδικασία Εγγραφής και Σύνδεσης Χρήστη στο Duo MFA

Όταν ο χρήστης συνδέεται για πρώτη φορά παραπέμπεται στη σελίδα του Duo ώστε να ενεργοποιήσει το MFA στον λογαριασμό του. Ο χρήστης σαρώνοντας ένα QR κωδικό (Σχήμα 5.3) με την εφαρμογή που διαθέτει το Duo για smartphones ενεργοποιεί το MFA.

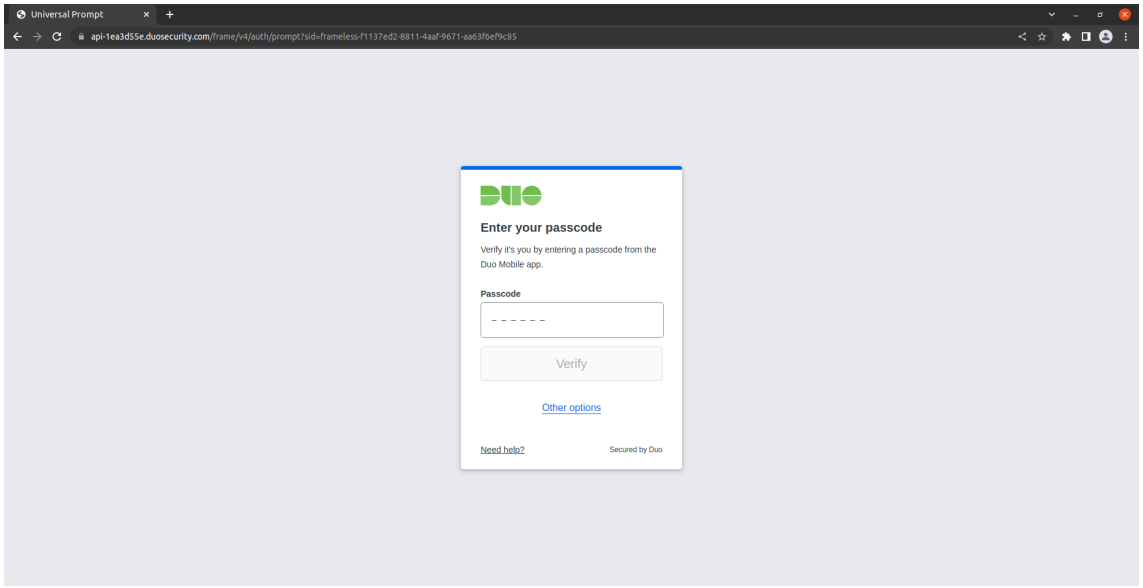


Σχήμα 5.3: Duo QR Code

Τις επόμενες φορές που θα συνδέεται ο χρήστης στον λογαριασμό του θα παραπέμπεται σε σελίδα του Duo η οποία είτε θα στέλνει Push Notification στο smartphone του χρήστη (Σχήμα 5.4) είτε θα έχει πεδίο στο οποίο ο χρήστης θα μπορεί να εισάγει τον κωδικό που εμφανίζεται στην εφαρμογή του smartphone (Σχήμα 5.5) για να επαληθεύσει την ταυτότητα του.



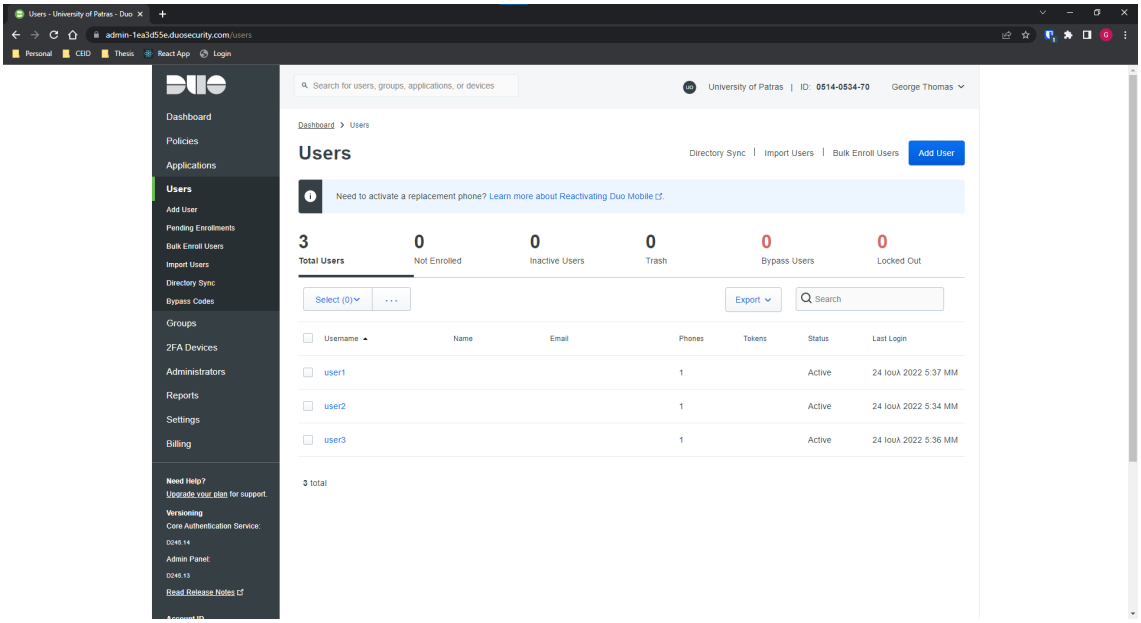
Σχήμα 5.4: Duo Push Notification



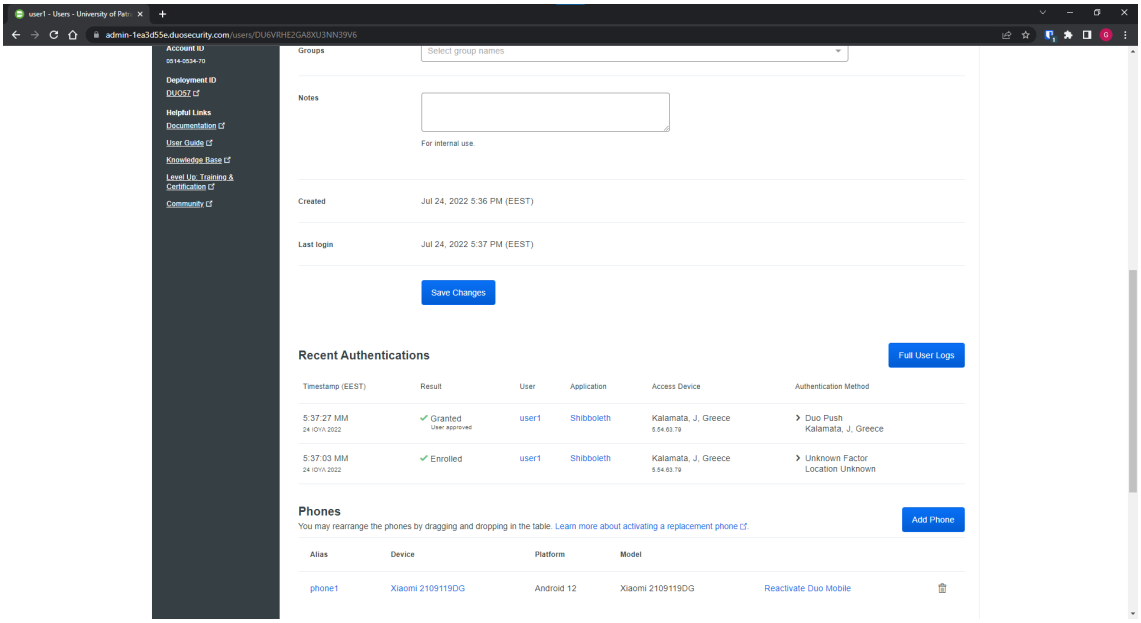
Σχήμα 5.5: Duo OTP

5.2.4 Βασικές Λειτουργίες Διαχειριστή

Ο διαχειριστής στη σελίδα του Duo μπορεί να δει τους χρήστες που έχουν ενεργοποιήσει το MFA (Σχήμα 5.6) καθώς και να επιλέξει κάποιον από αυτούς ώστε να δει επιπλέον πληροφορίες για τον συγκεκριμένο χρήστη (Σχήμα 5.7).



Σχήμα 5.6: Duo Users Panel



Σχήμα 5.7: Duo User Panel

5.3 Custom MFA Application

Το Shibboleth IdP προσφέρει ένα TOTP plugin [30]. Το plugin αυτό παρέχει μια ροή σύνδεσης που υλοποιεί μια απλή μορφή υποστήριξης επαλήθευσης χρήστη με τη χρήση TOTP. Συγκεκριμένα, δημιουργεί μια σελίδα (Σχήμα 5.8), η οποία εμφανίζεται μετά από την επιτυχή σύνδεση του χρήστη με το username και password και παραπέμπει τον χρήστη να εισάγει ένα

εξαψήφιο κωδικό τον οποίο τον επαληθεύει με ένα seed. Αν ο κωδικός είναι σωστός, επιτρέπει την πρόσβαση στη σελίδα που έχει θέσει ο SP. Το seed μπορεί να προέρχεται από μία στατική λίστα με seeds (Κώδικας 5.1). Ο τρόπος αυτός, βέβαια, δεν μπορεί να χρησιμοποιηθεί σε συνθήκες παραγωγικής λειτουργίας, γιατί ο διαχειριστής για κάθε νέο χρήστη θα πρέπει να προσθέτει χειροκίνητα μια εγγραφή στο αρχείο με τα seeds. Εναλλακτικά, το seed μπορεί να προέρχονται από μη στατικές πηγές (π.χ. Βάση Δεδομένων, LDAP). Στην περίπτωση αυτή, το Shibboleth IdP αξιοποιεί το Attribute Resolver για να λάβει το seed. Το Attribute Resolver μπορεί να λάβει την τιμή με πολλούς τρόπους. Για παράδειγμα, μπορεί να χρησιμοποιήσει το LDAP Connector για να λάβει την τιμή από μια εγγραφή του OpenLDAP, το Relational Database Connector για να λάβει την τιμή από μία σχεσιακή βάση δεδομένων ή το HTTP Connector για να λάβει την τιμή από έναν server.

```
<bean id="shibboleth.authn.TOTP.SeedSource" parent="shibboleth.authn.TOTP.
    StaticSeedSource">
  <property name="seeds">
    <map>
      <entry key="jdoe">
        <list>
          <value>K44GQWBAI54IQELEGYPDZG5LAJ6EJX73</value>
        </list>
      </entry>
      <entry key="jsmith">
        <list>
          <value>MLDE4YVTA7YKD5NZTDQX65ZPZ2A2OMW2</value>
        </list>
      </entry>
    </map>
  </property>
</bean>
```

Κώδικας 5.1: Στατική λίστα με seeds

Το TOTP plugin έχει περιορισμούς οι οποίοι κάνουν τη χρήση του αδύνατη για ένα πραγματικό σενάριο. Για παράδειγμα, δεν προσφέρει τρόπο στον χρήστη να ενεργοποιήσει κάποιον TOTP Authenticator ώστε να έχει πρόσβαση στον εξαψήφιο κωδικό. Αυτό συγκαταλέγεται στην αδυναμία του TOTP plugin να προσφέρει οποιαδήποτε μορφής διαχείρισης των χρηστών.

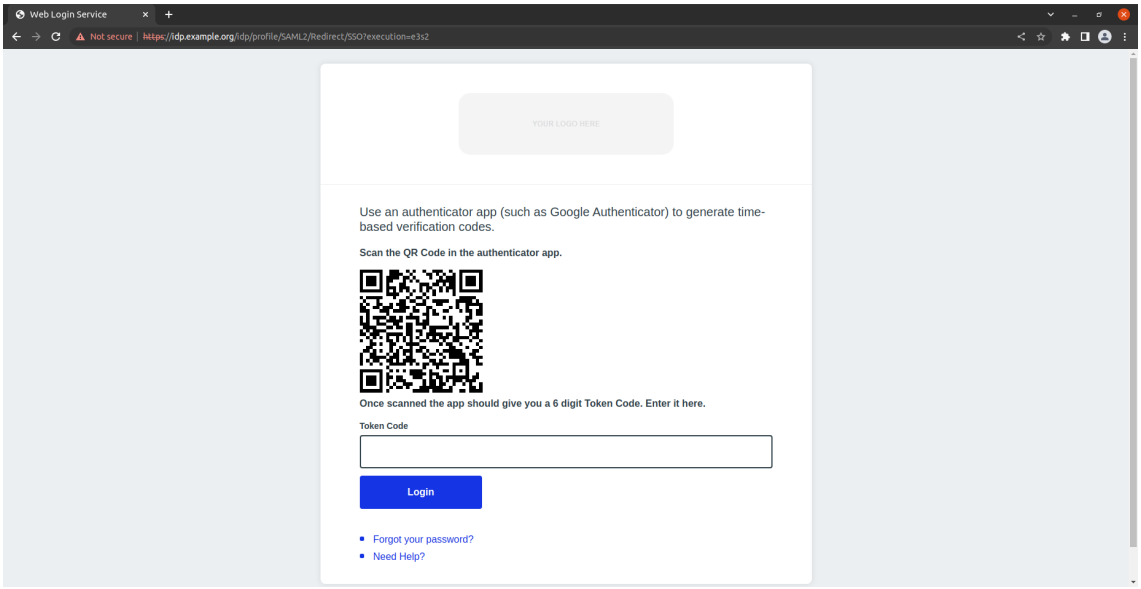
Στα πλαίσια της ΔΕ δημιουργήθηκε μια open source εφαρμογή (TOTP Manager [28]) η οποία καλείται να καλύψει τους περιορισμούς αυτούς αξιοποιώντας και επεκτείνοντας το TOTP plugin δημιουργώντας μια βασική διαχείριση χρηστών αντίστοιχη αυτής του Duo MFA.

Στις ενότητες 5.3.1 και 5.3.2 παρουσιάζονται οι λειτουργίες της εφαρμογής ως προς τον χρήστη και τον διαχειριστή αντίστοιχα, λειτουργίες οι οποίες απουσιάζουν από το TOTP plugin.

5.3.1 Διαδικασία Ελέγχου Ταυτότητας MFA Χρηστών

Ενεργοποίηση MFA Χρήστη

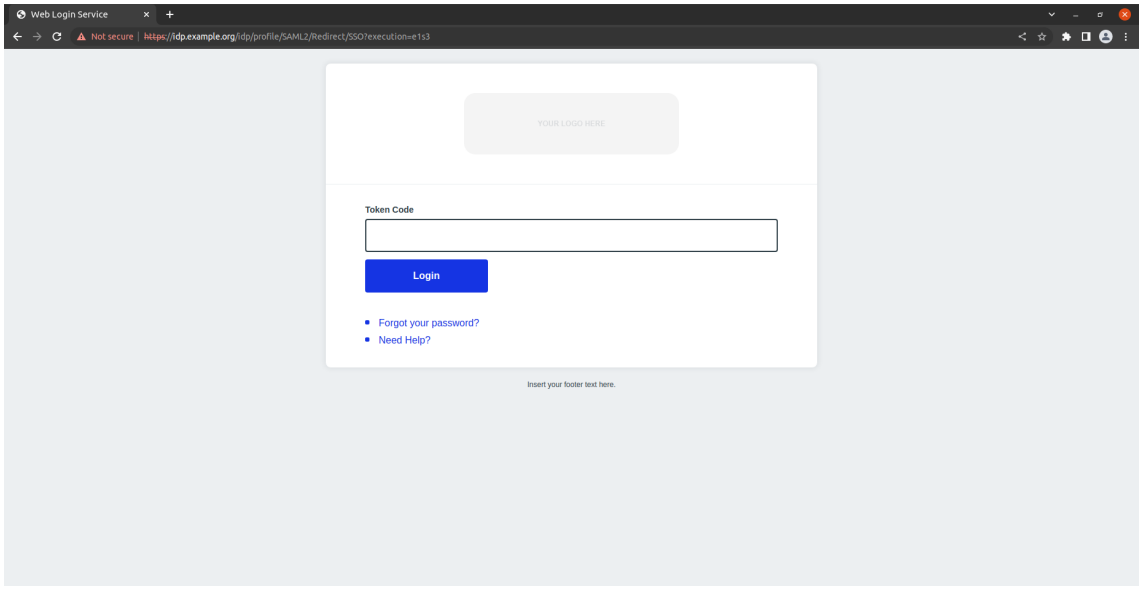
Όταν ο χρήστης συνδέεται για πρώτη φορά επιτυχώς με το username και password, εμφανίζεται μια σελίδα με ένα QR κωδικό (Σχήμα 5.8) τον οποίο ο χρήστης μπορεί να τον σαρώσει με ένα Authenticator (π.χ. Google Authenticator). Όταν τον σαρώσει, θα μπορεί να εισάγει στο πεδίο που έχει η σελίδα τον εξαψήφιο κωδικό που εμφανίζεται στο Authenticator. Με τον τρόπο αυτόν, ο χρήστης ενεργοποιεί το MFA για τον λογαριασμό του και συνδέεται στη σελίδα που έχει θέσει ο SP.



Σχήμα 5.8: Σελίδα Ενεργοποίησης MFA

Χρήση Ελέγχου Ταυτότητας MFA Χρήστη

Όταν ο χρήστης έχει ενεργοποιήσει το MFA στον λογαριασμό του, κάθε φορά που θα συνδέεται με το username και password, θα εμφανίζεται μια σελίδα με ένα πεδίο (Σχήμα 5.9) στο οποίο θα μπορεί να εισάγει τον εξαψήφιο κωδικό που εμφανίζεται στο Authenticator. Εισάγοντας σωστά τον κωδικό θα μπορεί να συνδεθεί στη σελίδα που έχει θέσει ο SP. Σε αντίθετη περίπτωση θα εμφανίζεται μήνυμα λάθους και θα πρέπει να εισάγει τον κωδικό ξανά.

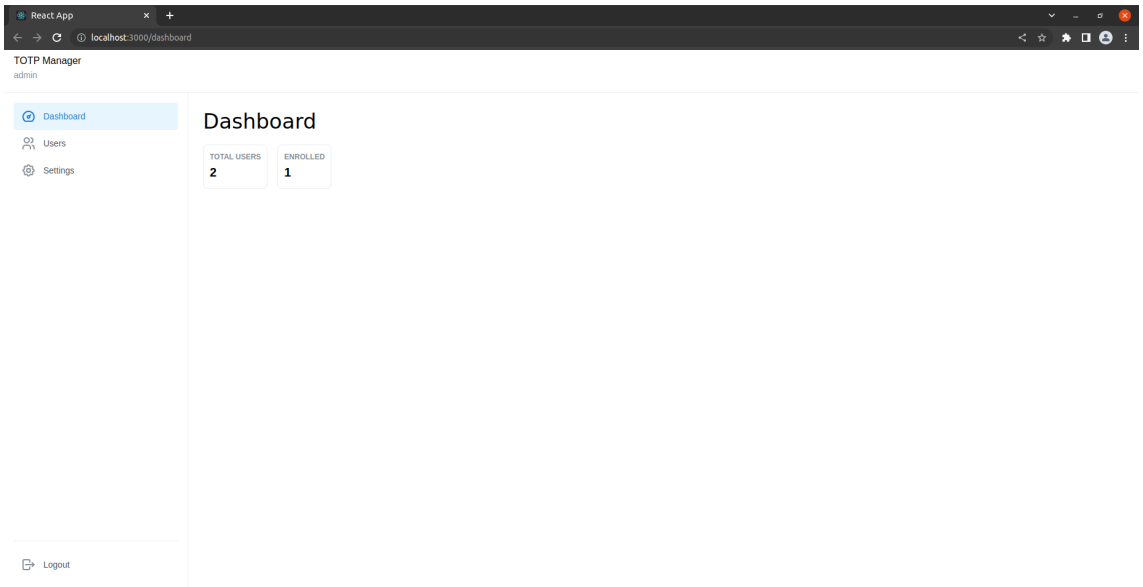


Σχήμα 5.9: Σελίδα Εισαγωγής TOTP

5.3.2 Λειτουργίες Διαχειριστή

Παρουσίαση Βασικών Πληροφοριών

Ο διαχειριστής στο Dashboard της εφαρμογής (Σχήμα 5.10) μπορεί να δει το πλήθος των συνολικών χρηστών και πόσοι από αυτούς έχουν στον λογαριασμό τους ενεργοποιημένο το MFA.

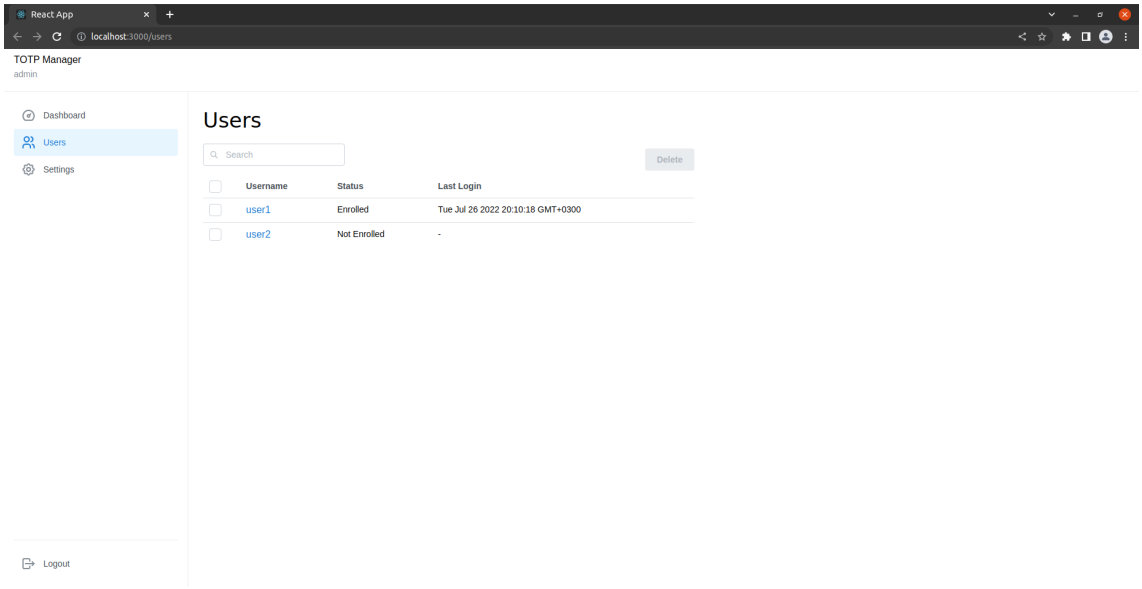


Σχήμα 5.10: Dashboard

Διαχείριση Χρηστών

Ο διαχειριστής στη σελίδα Users της εφαρμογής (Σχήμα 5.11) μπορεί να διαχειρίζεται τους χρήστες. Συγκεκριμένα έχει τις εξής λειτουργίες.

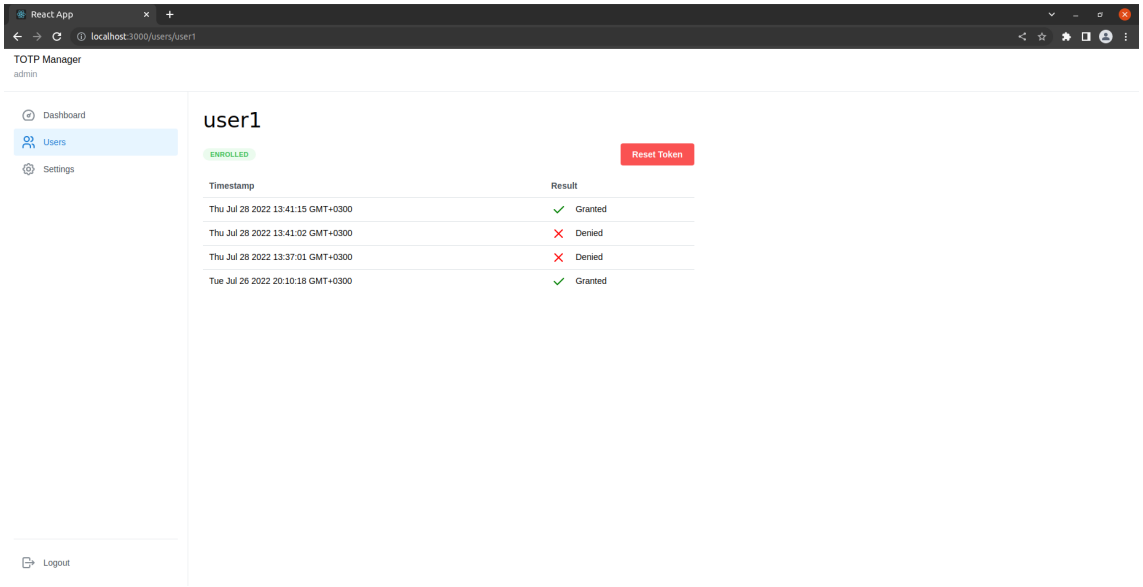
- Παρουσίαση βασικών πληροφοριών των χρηστών.
- Αναζήτηση χρηστών.
- Επιλογή χρηστών για διαγραφή.
- Επιλογή χρήστη για επιπλέον πληροφορίες και διαχείριση αυτού.



Σχήμα 5.11: Users

Διαχείριση Χρήστη

Ο διαχειριστής επιλέγοντας ένα χρήστη παραπέμπεται σε σελίδα (Σχήμα 5.12) όπου μπορεί να δει τα logs του χρήστη καθώς και να κάνει επαναφορά του token. Κάνοντας επαναφορά του token δημιουργείται ένας νέος QR κωδικός για τον χρήστη, ο οποίος θα εμφανιστεί στην επόμενη σύνδεση του χρήστη ώστε να ενεργοποιήσει ξανά το MFA για τον λογαριασμό του.



Σχήμα 5.12: User Logs

Ρυθμίσεις

Ο διαχειριστής στη σελίδα Settings της εφαρμογής (Σχήμα 5.14) μπορεί να δει και να αντιγράψει το Client ID και το Client Secret ώστε να μπορεί να συνδέσει την εφαρμογή με το Shibboleth IdP όπως παρουσιάζεται στην ενότητα 5.3.5.

Επίσης, μπορεί να αλλάξει το όνομα της εφαρμογής. Το όνομα της εφαρμογής φαίνεται στο Authenticator του χρήστη όταν σαρώσει τον QR κωδικό.

Τέλος, μπορεί να κάνει επαναφορά του Client Secret, δηλαδή να δημιουργήσει νέο. Στην περίπτωση αυτή θα πρέπει να ενημερώσει το κατάλληλο αρχείο στο Shibboleth IdP όπως παρουσιάζεται στην ενότητα 5.3.5 με το νέο Client Secret.

5.3.3 Αρχιτεκτονική Εφαρμογής

Βάση Δεδομένων

Η βάση δεδομένων που χρησιμοποιήθηκε είναι η MongoDB [17]. Η MongoDB είναι ένα σύστημα βάσεων δεδομένων ανοικτού κώδικα προσανατολισμένο στα έγγραφα. Αποτελεί μέρος της οικογένειας συστημάτων βάσεων δεδομένων NoSQL [18]. Αντί να αποθηκεύει δεδομένα σε πίνακες, όπως γίνεται σε μια σχεσιακή βάση δεδομένων, η MongoDB αποθηκεύει δομημένα δεδομένα ως έγγραφα τύπου JSON [14] με δυναμικά σχήματα, καθιστώντας την ενσωμάτωση δεδομένων σε ορισμένους τύπους εφαρμογών ευκολότερη και ταχύτερη.

Παρακάτω παρουσιάζονται τα μοντέλα της βάσης δεδομένων. Ένα μοντέλο βάσης δεδομένων είναι ένας τύπος μοντέλου δεδομένων που καθορίζει τη λογική δομή μιας βάσης δεδομένων. Το κάθε μοντέλο έχει ένα μοναδικό id που αναγνωρίζει μοναδικά την κάθε εγγραφή.

- admin

  - username (*String, μοναδικό*)

    Το username του admin για να κάνει login.

  - passwordHash (*String*)

    Το password του admin κατακερματισμένο.

- application

  - name (*String, μοναδικό*)

    Το όνομα της εφαρμογής.

  - clientID (*String, μοναδικό*)

    Το clientID της εφαρμογής κρυπτογραφημένο.

  - clientSecret (*String, μοναδικό*)

    Το clientSecret της εφαρμογής κρυπτογραφημένο.

- user

  - username (*String, μοναδικό*)

    Το username του χρήστη.

  - seed (*String, μοναδικό*)

    Ένα κρυπτογραφημένο αλφαριθμητικό για τη δημιουργία του QR κωδικού.

  - enrolled (*Boolean*)

    Αν ο χρήστης έχει ενεργοποιημένο το MFA.

  - logs (*ObjectId, μοναδικό*)

    Πίνακας που περιέχει τα ids με τα logs του χρήστη.

- log

  - type (*Enum*)

    Ο τύπος του log.

  - date (*Date*)

    Η ημερομηνία που καταγράφτηκε το log.

  - user (*ObjectId, μοναδικό*)

    Το id του χρήστη που ανήκει το log.

Σύνδεση Εφαρμογής με Shibboleth IdP

Όταν ο admin συνδέεται στην εφαρμογή για πρώτη φορά, δημιουργείται ένα ClientID και ένα Client Secret τα οποία μπορεί να θέσει στο Shibboleth IdP όπως παρουσιάζεται στην ενότητα 5.3.5. Οι τιμές αυτές περνάνε ως Authorization Header όταν γίνεται ένα API call από το Shibboleth IdP στο TOTP Manager ώστε να γίνει επαλήθευση του IdP ώστε να επιτραπεί η σύνδεση.

Διαδικασία Σύνδεσης Χρήστη

Όταν συνδέεται ο χρήστης με το username και password, γίνεται ένα API call από το Shibboleth IdP στο TOTP Manager.

Το TOTP Manager εκτελεί τα εξής.

- Αν ο χρήστης δεν έχει ενεργοποιήσει το MFA

  Δημιουργεί στη βάση δεδομένων ένα user, με username το username του χρήστη που συνδέθηκε στο Shibboleth IdP. Για τον χρήστη αυτόν δημιουργεί ένα μοναδικό seed και θέτει την τιμή enrolled ως false.

  Το TOTP Manager επιστρέφει στο Shibboleth IdP ένα JSON με τις εξής τιμές.

  - username

    Το username του χρήστη.
  - seed

    Το seed με βάση το οποίο δημιουργήθηκε ο QR κωδικός για τον χρήστη.
  - enrolled

    Το αν έχει ενεργοποιήσει το MFA.
  - qrCode

    Ένα url με την εικόνα για το QR κωδικό.
  - token

    Ένα αλφαριθμητικό που αναγνωρίζει μοναδικά τον χρήστη.

- Αν ο χρήστης έχει ενεργοποιήσει το MFA

  Αναζητεί τον χρήστη στη βάση δεδομένων και επιστρέφει στο Shibboleth IdP ένα JSON με τις τιμές που αντιστοιχούν στον χρήστη που βρέθηκε αλλά χωρίς το qrCode.

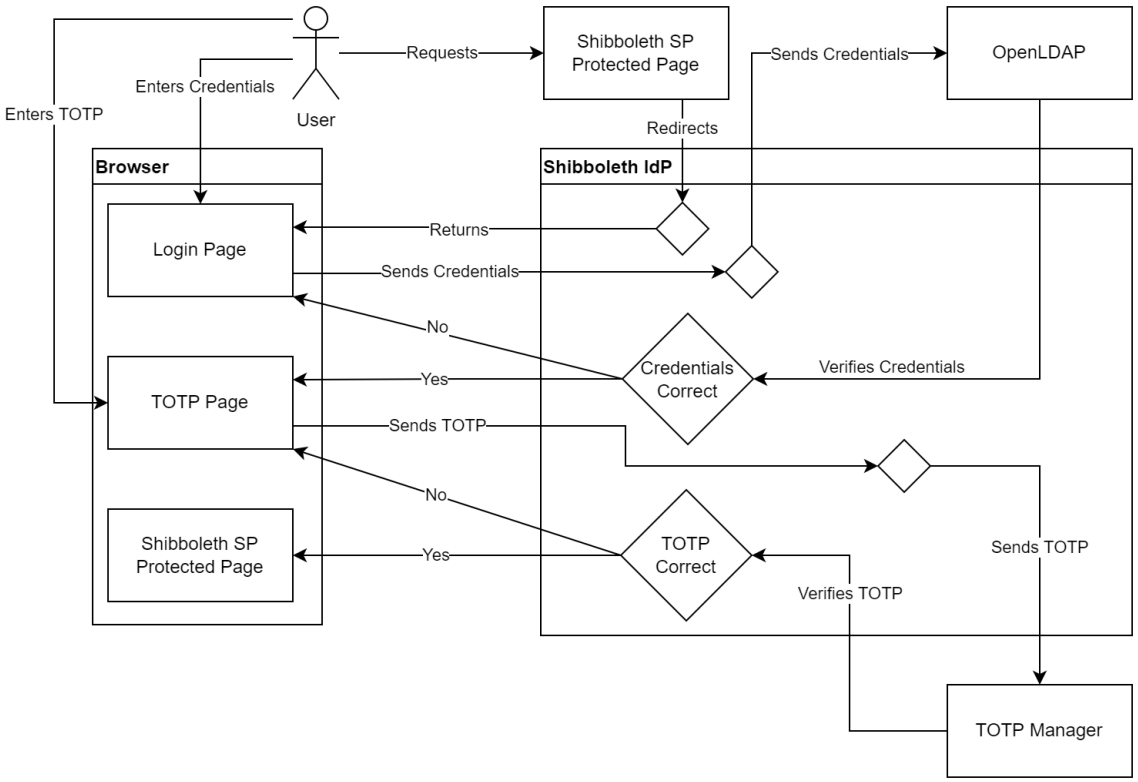
Στη συνέχεια, το Shibboleth IdP με βάση τις τιμές που δέχεται, παρουσιάζει στον χρήστη την κατάλληλη σελίδα. Συγκεκριμένα, αν ο χρήστης δεν έχει ενεργοποιήσει το MFA τότε εμφανίζεται ο QR κωδικός. Αν έχει ενεργοποιήσει το MFA τότε δεν εμφανίζεται ο QR κωδικός.

Μαζί με τη σελίδα έρχεται και ένα script στον φυλλομετρητή του χρήστη το οποίο όταν ο χρήστης εισάγει τον εξαψήφιο κωδικό από το Authenticator στο αντίστοιχο πεδίο της σελίδας στέλνει ένα API call στο TOTP Manager με τον εξαψήφιο κωδικό. Το TOTP Manager ελέγχει αν ο κωδικός αυτός είναι σωστός και δημιουργεί το κατάλληλο είδους log (Granted ή Denied) με την ημερομηνία που συνέβη.

Η ταυτοποίηση και ο έλεγχος ταυτότητας του χρήστη γίνεται με το token που επιστράφηκε κατά το αρχικό API call από το Shibboleth IdP στο TOTP Manager.

Ταυτόχρονα το Shibboleth IdP λαμβάνει από τον χρήστη τον εξαψήφιο κωδικό και τον επαληθεύει με το seed και αν είναι σωστός εμφανίζεται η σελίδα που έχει θέσει ο SP αλλιώς εμφανίζεται μήνυμα σφάλματος να εισάγει τον εξαψήφιο κωδικό ξανά.

Η αρχιτεκτονική παρουσιάζεται στο Σχήμα 5.13.



Σχήμα 5.13: Αρχιτεκτονική

5.3.4 Εγκατάσταση Εφαρμογής

Στο TOTP-Manager GitHub Repository [28] υπάρχουν γενικές οδηγίες για την εγκατάσταση της εφαρμογής. Παρακάτω παρατίθενται ακριβώς τα βήματα που πραγματοποιήθηκαν.

Εγκατάσταση Προαπαιτούμενων

- Εγκατάσταση Node.js

```
cd ~
curl -sL https://deb.nodesource.com/setup_16.x -o /tmp/
    nodesource_setup.sh
sudo bash /tmp/nodesource_setup.sh
sudo apt install -y nodejs
```

- Εγκατάσταση MongoDB.

```
cd ~
curl -fsSL https://www.mongodb.org/static/pgp/server-4.4.asc | sudo
    apt-key add -
sudo apt update
sudo apt install -y mongodb-org
sudo systemctl start mongod.service
sudo systemctl enable mongod
```

Εγκατάσταση & Ρύθμιση Εφαρμογής

- Λήψη εφαρμογής.

```
wget https://github.com/GTS08/TOTP-Manager/releases/download/release/
    totp_manager_release.tar.gz
tar -xzf totp_manager_release.tar.gz
```

- Ορισμός environmental μεταβλητών στο αρχείο .env

```
cd totp_manager_release
echo "MONGODB_URI=mongodb://localhost:27017/totp
PORT=3001
SECRET=ThisIsASecretKey
ENCRYPTION_KEY=43sx4zDiByTwhdpBOBrDIqEy7rYV98wV
ENCRYPTION_VECTOR=OS4g5lqPQ8ffkjkH
" > .env
```

Το ENCRYPTION_KEY πρέπει να είναι 32 αλφαριθμητικών χαρακτήρων.
Το ENCRYPTION_VECTOR πρέπει να 16 αλφαριθμητικών χαρακτήρων.

- Ενεργοποίηση εφαρμογής.

```
npm install
npm start
```

- Δημιουργία διαχειριστή.

```
mongo
use totp
db.admins.insertOne({ username: "admin", passwordHash: "
    $2a$10$joyx4VdSEipbcwVkrwDvYO7j9NssrLiIO.25.loUW.L4OoBNtQwOC"})
exit
```

Ένα password hash μπορεί να δημιουργηθεί χρησιμοποιώντας ένα bcrypt generator [2]. Η τιμή του συγκεκριμένου passwordHash αντιστοιχεί στο password "admin".

5.3.5 Ενσωμάτωση Εφαρμογής με Shibboleth IdP

Στο TOTP-Manager GitHub Repository Wiki [29] υπάρχουν γενικές οδηγίες για την ενσωμάτωση της εφαρμογής με το Shibboleth IdP. Παρακάτω παρατίθενται ακριβώς τα βήματα που πραγματοποιήθηκαν.

Οδηγίες

- Ενεργοποίηση του MFA Flow όπως παρουσιάζεται στην ενότητα 5.2.2.

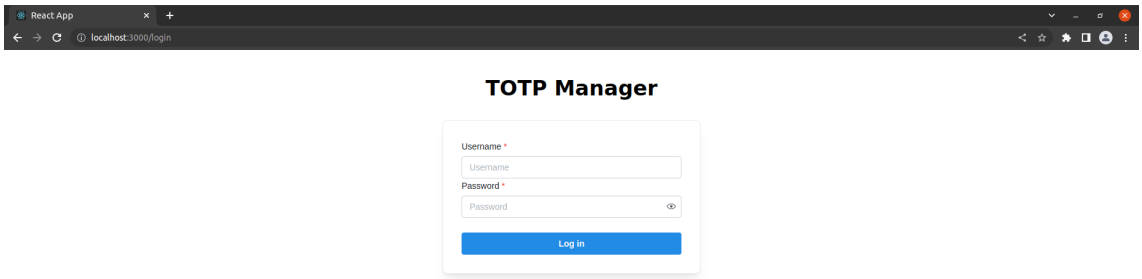
  Το MFA flow είναι μια λειτουργία του Shibboleth IdP που επιτρέπει την χρήση περισσότερων από μία ροή σύνδεσης.

- Λήψη και ενεργοποίηση του TOTP Plugin [30].

  Το TOTP Plugin θα χρησιμοποιηθεί ως δεύτερη ροή σύνδεσης μετά το username και password.

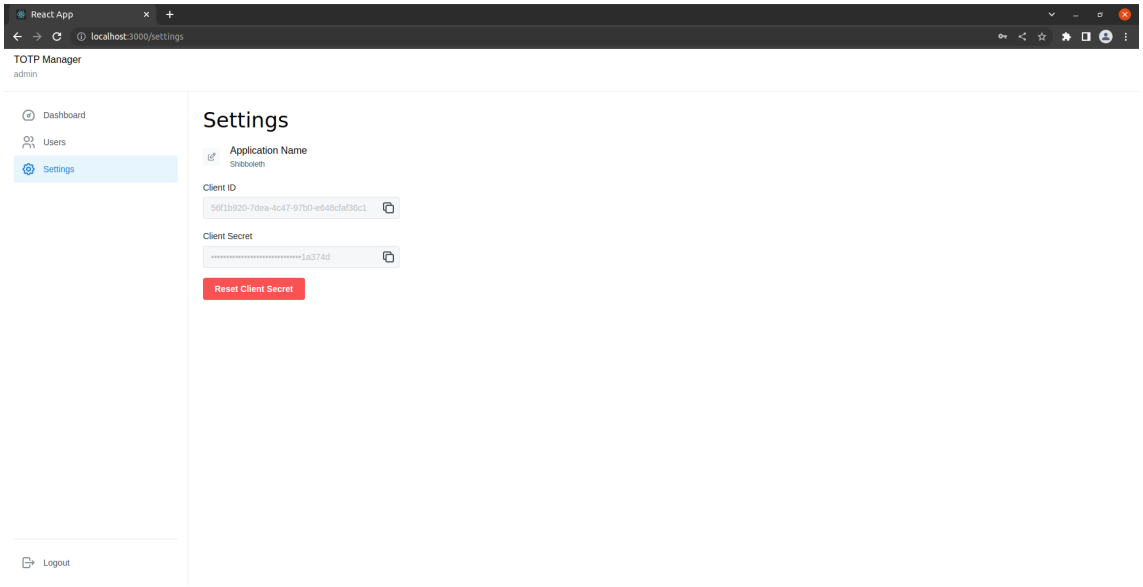
```
sudo su -
/opt/shibboleth-idp/bin/plugin.sh -I net.shibboleth.idp.plugin.authn.
    totp
```

- Σύνδεση του διαχειριστή στην εφαρμογή μέσω ενός φυλλομετρητή επισκέπτοντας τη σελίδα `http://localhost:3001/` (Σχήμα 5.14) με τα στοιχεία που έχουν τεθεί στην ενότητα 5.3.4.



Σχήμα 5.14: Σελίδα Εισόδου

- Μεταφορά στην καρτέλα Settings (Σχήμα 5.15) και αντιγραφή των Client ID και Client Secret.



Σχήμα 5.15: Σελίδα Ρυθμίσεων

- Προσθήκη των παρακάτω στο αρχείο `/opt/shibboleth-idp/conf/idp.properties`

  Οι τιμές αυτές πάρθηκαν από το προηγούμενο βήμα και είναι απαραίτητες για την ασφαλή σύνδεση με το TOTP Manager.

```
# TOTP plugin values
idp.totp.URL = http://localhost:3001
idp.totp.clientID = 56f1b920-7dea-4c47-97b0-e648cfaf36c1
idp.totp.clientSecret = 928484d0-d90e-47b7-9b95-3b13531a374d
```

- Προσθήκη των παρακάτω στο αρχείο `/opt/shibboleth-idp/conf/global.xml`

```
<util:map id="httpHeaders">
    <entry key="Authorization" value="%{idp.totp.clientID} %{idp.totp.clientSecret}" />
</util:map>

<util:map id="shibboleth.CustomViewContext">
    <entry key="view" value-ref="shibboleth.CustomViewContext"/>
</util:map>

<bean id="shibboleth.NonCachingHttpClient" lazy-init="true"
    class="net.shibboleth.idp.profile.spring.relyingparty.metadata.HttpClientFactoryBean"/>
```

  Στο πρώτο map γίνεται ορισμός του HTTP Header που θα χρησιμοποιηθεί για να γίνει η σύνδεση και η εξουσιοδότηση με το TOTP Manager. Το Authorization ορίζεται με τις τιμές από το προηγούμενο βήμα.

  Το δεύτερο map αποτελεί ρύθμιση ώστε να μπορούν να γίνουν διαθέσιμες μεταβλητές στη σελίδα που προσφέρει το TOTP plugin (Σχήμα 5.9).

Το bean αποτελεί ρύθμιση για το caching.

- Προσθήκη των παρακάτω στο αρχείο `/opt/shibboleth-idp/conf/global.xml`

```
<AttributeDefinition xsi:type="Simple" id="tokenSeeds">
  <InputDataConnector ref="myHTTP" attributeNames="seed" />
</AttributeDefinition>

<DataConnector id="myHTTP" xsi:type="HTTP"
    httpClientRef="shibboleth.NonCachingHttpClient"
    acceptTypes="application/json"
    headerMapRef="httpHeaders"
    exportAttributes="body">
    <URLTemplate>
        <![CDATA[
        %{idp.totp.URL}/api/users/enrollment/$pathEscaper.escape($resolutionContext.principal)
        ]]>
    </URLTemplate>

    <ResponseMapping>
        <Script>
        <![CDATA[
        logger = Java.type('org.slf4j.LoggerFactory').getLogger('http-check');

        var HashSet = Java.type("java.util.HashSet");
        var HttpClientSupport = Java.type("net.shibboleth.utilities.java.support.httpclient.HttpClientSupport");
        var IdPAttribute = Java.type("net.shibboleth.idp.attribute.IdPAttribute");
        var StringAttributeValue = Java.type("net.shibboleth.idp.attribute.StringAttributeValue");

        // Limits length to 64k
        var body = HttpClientSupport.toString(response.getEntity(), "UTF-8", 65536);
        var result = JSON.parse(body);

        // Seed attribute
        var attr = new IdPAttribute("seed");
        var value = new HashSet();

        value.add(new StringAttributeValue(result.seed));

        attr.setValues(value);
        connectorResults.add(attr);

        // Response body attribute
        var attr = new IdPAttribute("body");
        var value = new HashSet();
```

```
42
43          value.add(new StringAttributeValue(body));
44
45          attr.setValues(value);
46          connectorResults.add(attr);
47
48          ]]>
49          </Script>
50      </ResponseMapping>
51  </DataConnector>
```

Το Data Connector κάνει ένα API call στο TOTP Manager και επιστρέφει ένα JSON το οποίο κάνει export ώστε να μπορεί να χρησιμοποιηθεί από το MFA flow στο επόμενο βήμα. Το seed γίνεται ξεχωριστά export για να περάσει στο Attribute Definition ώστε η τιμή tokenSeeds να οριστεί με την τιμή που επιστρέφει το TOTP Manager. Η τιμή tokenSeeds είναι η τιμή που το TOTP Plugin θα επαληθεύσει τον εξαψήφιο κωδικό που εισάγει ο χρήστης.

- Ορισμός του αρχείου `/opt/shibboleth-idp/conf/authn/mfa-authn-config.xml` ως εξής.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:context="http://www.springframework.org/schema/context"
4      xmlns:util="http://www.springframework.org/schema/util"
5      xmlns:p="http://www.springframework.org/schema/p"
6      xmlns:c="http://www.springframework.org/schema/c"
7      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8      xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans.xsd
9                          http://www.springframework.org/schema/context
      http://www.springframework.org/schema/context/spring-context.xsd
10                         http://www.springframework.org/schema/util
      http://www.springframework.org/schema/util/spring-util.xsd"
11
12     default-init-method="initialize"
13     default-destroy-method="destroy">
14
15     <util:map id="myMap">
16         <entry key="resolver">
17             <ref bean="shibboleth.AttributeResolverService" />
18         </entry>
19         <entry key="viewRef">
20             <ref bean="shibboleth.CustomViewContext" />
21         </entry>
22     </util:map>
23
24     <util:map id="shibboleth.authn.MFA.TransitionMap">
25         <!-- First rule runs the Password login flow. -->
26         <entry key="">
```

```
27          <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow=
    "authn/Password" />
28      </entry>
29
30      <!-- Second rule runs a function if Password succeeds, to
    determine whether an additional factor is required. -->
31      <entry key="authn/Password">
32          <bean parent="shibboleth.authn.MFA.Transition"
    p:nextFlowStrategy-ref="checkSecondFactor" />
33      </entry>
34      <!-- An implicit final rule will return whatever the final
    flow returns. -->
35  </util:map>
36
37  <!-- Example script to see if second factor is required. Currently
     just returns the TOTP flow -->
38  <bean id="checkSecondFactor" parent="shibboleth.ContextFunctions.
    Scripted" factory-method="inlineScript" p:customObject-ref="myMap"
    >
39      <constructor-arg>
40          <value>
41          <![CDATA[
42              nextFlow = "authn/TOTP";
43
44              logger = Java.type('org.slf4j.LoggerFactory').
    getLogger('mfa-check');
45
46              // Attribute check is required to decide if first
    factor alone is enough.
47              resCtx = input.getSubcontext(
48                  "net.shibboleth.idp.attribute.resolver.context.
    AttributeResolutionContext", true);
49
50              // Look up the username
51              usernameLookupStrategyClass = Java.type("net.
    shibboleth.idp.session.context.navigate.
    CanonicalUsernameLookupStrategy");
52              usernameLookupStrategy = new
    usernameLookupStrategyClass();
53              resCtx.setPrincipal(usernameLookupStrategy.apply(input
    ));
54
55              // resolve the attribute to determine if a first
    factor is sufficient
56              resCtx.getRequestedIdPAttributeNames().add("body");
57              resCtx.resolveAttributes(custom["resolver"]);
58
59              // Check for an attribute value that authorizes use of
     first factor.
60              bodyAttribute = resCtx.getResolvedIdPAttributes().get
```

```
    ("body");
61
62                  bodyAttributeValue = bodyAttribute.getValues();
63
64                  logger.info(bodyAttributeValue[0].value);
65
66                  result = JSON.parse(bodyAttributeValue[0].value);
67
68                  custom["viewRef"].get("view").put("URL", "%{idp.totp.
    URL}");
69
70                  custom["viewRef"].get("view").put("enrolled", result.
    enrolled);
71                  custom["viewRef"].get("view").put("token", result.
    token);
72
73                  if (!result.enrolled) {
74                      custom["viewRef"].get("view").put("qrCode", result
    .qrCode);
75                      custom["viewRef"].get("view").put("seed", result.
    seed);
76                  }
77
78                  input.removeSubcontext(resCtx);   // cleanup
79
80                  nextFlow;   // pass control to second factor or end
    with the first
81              ]]>
82              </value>
83          </constructor-arg>
84      </bean>
85  </beans>
```

Το αρχείο αυτό περιγράφει την σειρά με την οποία πραγματοποιούνται οι ροές σύνδεσης. Πρώτα πραγματοποιείται η ροή σύνδεσης με το username και password και μετά η ροή σύνδεσης με το TOTP. Ακριβώς πριν ενεργοποιηθεί η ροή σύνδεσης TOTP, εκτελείται μια διαδικασία η οποία παίρνει τις τιμές από το JSON που επιστράφηκε από το TOTP Manager και τις αντιστοιχεί σε μεταβλητές οι οποίες μπορούν να χρησιμοποιηθούν στην σελίδα που προσφέρει το TOTP plugin που παρουσιάζεται στο επόμενο βήμα.

- Ορισμός του αρχείου `/opt/shibboleth-idp/views/totp.vm` ως εξής.

```
1
2  #set ($rpContext = $profileRequestContext.getSubcontext('net.
       shibboleth.idp.profile.context.RelyingPartyContext'))
3
4  <!DOCTYPE html>
5  <html>
6      <head>
7          <title>#springMessageText("idp.title", "Web Login Service")</
```

```
    title>
8       <meta charset="UTF-8" />
9       <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
10      <meta name="viewport" content="width=device-width, initial-
    scale=1.0, maximum-scale=5.0">
11      <link rel="stylesheet" type="text/css" href="$request.
    getContextPath()#springMessageText("idp.css", "/css/placeholder.
    css")">
12  </head>
13  <body>
14      <main class="main">
15          <header>
16              <img class="main-logo" src="$request.getContextPath()#
    springMessageText("idp.logo", "/images/placeholder-logo.png")" alt
    ="#springMessageText("idp.logo.alt-text", "logo")" />
17
18              #set ($serviceName = $rpUIContext.serviceName)
19              #if ($serviceName && !$rpContext.getRelyingPartyId().
    contains($serviceName))
20                  <h1>#springMessageText("idp.login.loginTo", "Login
     to") $encoder.encodeForHTML($serviceName)</h1>
21              #end
22          </header>
23
24          <section>
25              <form action="$flowExecutionUrl" method="post">
26                  #parse("csrf/csrf.vm")
27
28                  #*
29                  //
30                  //    SP Description & Logo (optional)
31                  //    These idpui lines will display added
    information (if available
32                  //    in the metadata) about the Service Provider
    (SP) that requested
33                  //    authentication. These idpui lines are "
    active" in this example
34                  //    (not commented out) - this extra SP info
    will be displayed.
35                  //    Remove or comment out these lines to stop
    the display of the
36                  //    added SP information.
37                  //
38                  *#
39                  #set ($logo = $rpUIContext.getLogo())
40                  #if ($logo)
41                      <img class="service-logo" src= "$encoder.
    encodeForHTMLAttribute($logo)" alt="$encoder.
    encodeForHTMLAttribute($serviceName)">
42                  #end
```

```
43                  #set ($desc = $rpUIContext.getServiceDescription()
    )
44                  #if ($desc)
45                      <p>$encoder.encodeForHTML($desc)</p>
46                  #end
47
48                  #parse("totp-error.vm")
49
50                  #if (!$custom.enrolled)
51                      <h2>Use an authenticator app (such as Google
    Authenticator) to generate time-based verification codes.</h2>
52
53                      <h3>Scan the QR Code in the authenticator app
    .</h3>
54
55                      <img src=$custom.qrCode alt="QR Code">
56                      <h3>Once scanned the app should give you a 6
    digit Token Code. Enter it here.</h3>
57                  #end
58
59                  <label for="tokencode">#springMessageText("idp.
    totp.field", "Token Code")</label>
60                  <input id="tokencode" name="tokencode" type="text"
     value="" />
61
62                  <div class="grid">
63                      <div class="grid-item">
64                          <button type="submit" name="
    _eventId_proceed"
65                              onClick="verifyCode()"
66                              >#springMessageText("idp.login.login",
     "Login")</button>
67                      </div>
68                  </div>
69              </form>
70
71              <ul>
72                  <li><a href="#springMessageText("idp.url.password.
    reset", '#')">#springMessageText("idp.login.forgotPassword", "
    Forgot your password?")</a></li>
73                  <li><a href="#springMessageText("idp.url.helpdesk
    ", '#')">#springMessageText("idp.login.needHelp", "Need Help?")</a
    ></li>
74              </ul>
75          </section>
76      </main>
77      <footer class="footer">
78          <div class="cc">
79              <p>#springMessageText("idp.footer", "Insert your
    footer text here.")</p>
```

```
80              </div>
81          </footer>
82
83          <script>
84              function verifyCode() {
85                  const totp = document.getElementById("tokencode").
    value;
86
87                  const token = "${custom.token}";
88
89                  const data = {
90                      totp
91                  };
92
93                  fetch(`${custom.URL}/api/user/log`, {
94                      method: "POST",
95                      headers: {
96                          "Content-Type": "application/json",
97                          "Authorization": `bearer ${token}`,
98                      },
99                      body: JSON.stringify(data),
100                 });
101             }
102         </script>
103     </body>
104 </html>
```

Το αρχείο αυτό αποτελεί το πηγαίο κώδικα της σελίδας του TOTP plugin. Αξιοποιώντας τις μεταβλητές που τέθηκαν στο προηγούμενο βήμα εμφανίζει στον χρήστη το σωστό περιεχόμενο. Συγκεκριμένα, αν ο χρήστης δεν έχει ενεργοποιήσει το MFA (δηλαδή η μεταβλητή enrolled είναι ψευδής) τότε εμφανίζει το QR κωδικό (Σχήμα 5.8) αλλιώς αν ο χρήστης έχει ενεργοποιήσει το MFA (δηλαδή η μεταβλητή enrolled είναι αληθής) τότε δεν εμφανίζει το QR κωδικό (Σχήμα 5.9). Επίσης, στη σελίδα αυτή υπάρχει και ένα script το οποίο κάνει API call στο TOTP Manager ώστε να δημιουργηθούν τα κατάλληλα logs αξιοποιώντας τη μεταβλητή token η οποία περνάει ως Authorization Header και αναγνωρίζει μοναδικά τον χρήστη.

- Επανεκκίνηση Jetty.

```
1 systemctl restart jetty.service
```

Κεφάλαιο 6

Συμπεράσματα & Προοπτικές

6.1 Συμπεράσματα

Σήμερα, οργανώσεις και επιχειρήσεις χρειάζονται συστήματα για τη διαχείριση των χρηστών τους. Η διαχείριση αυτή γίνεται με γνώμονα την "ταυτότητα". Η ταυτότητα, όπως περιγράφεται στο Κεφάλαιο 2, και τα γεγονότα που παρουσιάζονται κατά τη ζωή της είναι απαραίτητα για μια εύκολη και ασφαλής διαχείριση των χρηστών. Ένα από τα γεγονότα αυτά είναι η ενιαία σύνδεση η οποία προσφέρει στους χρήστες τη δυνατότητα να συνδέονται μια φορά και να έχουν πρόσβαση σε πολλές εφαρμογές. Για να υλοποιηθούν αυτά τα συστήματα έχουν δημιουργηθεί πρωτόκολλα. Ένα από αυτά, είναι το πρωτόκολλο SAML, το οποίο προσφέρει ενιαία σύνδεση μεταξύ τομέων και ομοσπονδιακή ταυτότητα. Πάνω σε αυτό το πρωτόκολλο, έχει βασιστεί και η υποδομή Shibboleth, η οποία προσφέρει ένα σύστημα διαχείρισης ταυτότητας.

Η υποδομή Shibboleth για την ταυτοποίηση των χρηστών παρέχει τον κλασσικό τρόπο του συνδυασμού username και password. Η σύγχρονη ταυτοποίηση των χρηστών, βέβαια, απαιτεί ταυτοποίηση με πολλούς παράγοντες για μια πλήρης και ασφαλής ταυτοποίηση. Έτσι, συμπεραίνεται ότι κάθε σύστημα διαχείρισης χρηστών που ταυτοποιεί τους χρήστες της χρειάζεται να προσφέρει πιστοποίηση ταυτότητας με πολλούς παράγοντες. Η υποδομή Shibboleth, αν και παρέχει εγγενώς κάτι τέτοιο, βασίζεται κυρίως σε ενσωματώσεις με εξωτερικές εφαρμογές. Παρόλα αυτά, καμία από αυτές δεν είναι δωρεάν και ενημερωμένη για την τελευταία έκδοση του Shibboleth. Για αυτό, στην εργασία αυτή, δημιουργήθηκε δωρεάν εφαρμογή για να γίνει διαθέσιμη αυτή τη δυνατότητα.

Η εφαρμογή αυτή αξιοποιεί ένα πρόσθετο που προσφέρει το Shibboleth, δηλαδή μια σελίδα που επαληθεύει έναν εξαψήφιο κωδικό με ένα seed. Η σελίδα αυτή εμφανίζεται μετά τη σελίδα της εισαγωγής username και password και αποτελεί τον δεύτερο παράγοντα πιστοποίησης ταυτότητας του χρήστη. Η εφαρμογή επεκτείνει το πρόσθετο αυτό, για να προσφέρει μια διαδικασία στον χρήστη ώστε να μπορεί να ρυθμίσει τον δεύτερο παράγοντα. Ο παράγοντας αυτός κάνει χρήση του TOTP με ένα TOTP Authenticator. Οι χρήστες την πρώτη φορά που συνδέονται μπορούν να σαρώσουν ένα QR κωδικό για να ρυθμίσουν τον TOTP Authenticator. Έτσι, τις επόμενες φορές που συνδέονται θα έχουν στη διάθεσή τους τον εξαψήφιο

κωδικό. Οι διαχειριστές από την πλευρά τους έχουν στη διάθεση τους σύστημα το οποίο τους επιτρέπει να διαχειρίζονται τους χρήστες και την εγγραφή τους στον δεύτερο παράγοντα πιστοποίησης. Στις λειτουργίες του συστήματος συγκαταλέγονται η αναζήτηση, η προβολή, η επαναφορά και η διαγραφή των χρηστών στον δεύτερο παράγοντα πιστοποίησης.

6.2 Μελλοντική Εργασία

Στην εργασία αυτή δημιουργήθηκε εφαρμογή η οποία προσφέρει δυνατότητα ταυτοποίησης χρηστών με χρήση πολλαπλών παραγόντων και ενσωμάτωση με την υποδομή Shibboleth. Ο πρώτος παράγοντας είναι ο συνδυασμός username και password που προσφέρεται από το Shibboleth. Η εφαρμογή που δημιουργήθηκε προσφέρει τον δεύτερο παράγοντα ο οποίος είναι η χρήση του TOTP με κάποιο TOTP Authenticator. Η εφαρμογή, επίσης, προσφέρει και μια βασική διαχείριση των χρηστών όπως περιγράφεται στην ενότητα 5.3.2.

Ως μελλοντική εργασία, η εφαρμογή θα μπορούσε να επεκταθεί ώστε να παρέχει περισσότερους τρόπους για την ταυτοποίηση του χρήστη. Οι μέθοδοι που παρουσιάζονται στην ενότητα 4.3 θα μπορούσαν όλοι να συμπεριληφθούν με τη δυνατότητα στον χρήστη να επιλέγει τον τρόπο που επιθυμεί να ταυτοποιηθεί. Επίσης, θα μπορούσαν να προστεθούν επιπλέον λειτουργίες για τη διαχείριση των χρηστών πέρα των βασικών. Για παράδειγμα, ο διαχειριστής θα μπορούσε να έχει τη δυνατότητα να χωρίζει τους χρήστες σε ομάδες και να διαχειρίζεται κάθε ομάδα ξεχωριστά. Η εφαρμογή είναι open source με αποτέλεσμα να δίνεται η δυνατότητα σε οποιονδήποτε να συνεισφέρει στην επέκτασή της.

Βιβλιογραφία

[1] Nathanael Andrews. Can i get your digits: Illegal acquisition of wireless phone numbers for sim-swap attacks and wireless provider liability. *Nw. J. Tech. & Intell. Prop.*, 16:79, 2018.

[2] Bcrypt generator. URL: https://bcrypt-generator.com/, Πρόσβαση: Σεπτέμβριος 2022.

[3] Shibboleth flow diagram. URL: https://apereo.github.io/cas/6.5.x/protocol/CAS-Protocol-Specification.html, Πρόσβαση: Σεπτέμβριος 2022.

[4] Dipankar Dasgupta, Arunava Roy και Abhijit Nag. *Authentication Basics*, σελίδες 1–36. Springer International Publishing, Cham, 2017.

[5] Dipankar Dasgupta, Arunava Roy και Abhijit Nag. *Multi-Factor Authentication*, σελίδες 185–233. Springer International Publishing, Cham, 2017.

[6] Duo mfa. URL: https://duo.com/editions-and-pricing/duo-mfa, Πρόσβαση: Σεπτέμβριος 2022.

[7] Duo plugin. URL: https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/1374027959/DuoOIDCAuthnConfiguration, Πρόσβαση: Σεπτέμβριος 2022.

[8] Duo website. URL: https://duo.com/, Πρόσβαση: Σεπτέμβριος 2022.

[9] Eran Hammer-Lahav. The oauth 1.0 protocol. Τεχνική Αναφορά υπ. αρίθμ., 2010.

[10] Dick Hardt. The oauth 2.0 authorization framework. Τεχνική Αναφορά υπ. αρίθμ., 2012.

[11] John Hughes και Eve Maler. Security assertion markup language (saml) v2.0 technical overview. *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, 13, 2005.

[12] Sanjar Ibrokhimov, Kueh Lee Hui, Ahmed Abdulhakim Al-Absi, Mangal Sain και others. Multi-factor authentication in cyber physical system: A state of art survey. Στο *2019 21st international conference on advanced communication technology (ICACT)*, σελίδες 279–284. IEEE, 2019.

[13] idem-tutorials. URL: https://github.com/ConsortiumGARR/idem-tutorials, Πρόσβαση: Σεπτέμβριος 2022.

[14] json. URL: https://www.json.org/, Πρόσβαση: Σεπτέμβριος 2022.

[15] Avijit Mallik. Man-in-the-middle-attack: Understanding in simple words. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 2(2):109–134, 2019.

[16] DT Manurung. Designing of user authentication based on multi-factor authentication on wireless networks. *Jour of Adv Research in Dynamical & Control Systems*, 12(1), 2020.

[17] Mongodb. URL: https://www.mongodb.com/, Πρόσβαση: Σεπτέμβριος 2022.

[18] Nosql. URL: https://www.mongodb.com/nosql-explained, Πρόσβαση: Σεπτέμβριος 2022.

[19] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen και Yevgeni Koucheryavy. Multi-factor authentication: A survey. *Cryptography*, 2(1), 2018.

[20] Zulfikar Ramzan. Phishing attacks and countermeasures. *Handbook of information and communication security*, σελίδες 433–448, 2010.

[21] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron και Kent Seamons. A usability study of five Two-Factor authentication methods. Στο *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, σελίδες 357–370, Santa Clara, CA, 2019. USENIX Association.

[22] EmersonRibeiro de Mello, Michelle Silva Wangham, Samuel Bristot Loli, Carlos Eduardoda Silva, GabrielaCavalcanti da Silva, Shirlei Aparecidade Chaves και Bruno Bristot Loli. Multi-factor authentication for shibboleth identity providers. *Journal of Internet Services and Applications*, 11(1):1–21, 2020.

[23] Natsuhiko Sakimura, John Bradley, Mike Jones, Breno De Medeiros και Chuck Mortimore. Openid connect core 1.0. *The OpenID Foundation*, σελίδα Σ3, 2014.

[24] Shibboleth v3. URL: https://shibboleth.atlassian.net/wiki/spaces/IDP30/overview, Πρόσβαση: Σεπτέμβριος 2022.

[25] Shibboleth documentation. URL: https://shibboleth.atlassian.net/wiki/spaces/CONCEPT/overview?homepageId=928645504, Πρόσβαση: Σεπτέμβριος 2022.

[26] Shibboleth flow diagram. URL: https://blog.miniorange.com/what-is-shibboleth, Πρόσβαση: Σεπτέμβριος 2022.

[27] Shibboleth flows and config. URL: https://shibboleth.atlassian.net/wiki/spaces/CONCEPT/pages/928645290/FlowsAndConfig, Πρόσβαση: Σεπτέμβριος 2022.

[28] Totp manager. URL: https://github.com/GTS08/TOTP-Manager, Πρόσβαση: Σεπτέμβριος 2022.

[29] Totp manager wiki. URL: https://github.com/GTS08/TOTP-Manager/wiki/Shibboleth-IdP-Integration, Πρόσβαση: Σεπτέμβριος 2022.

[30] Totp plugin. URL: https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/1376878877/TOTP, Πρόσβαση: Σεπτέμβριος 2022.

[31] Yvonne Wilson και Abhishek Hingnikar. *The Life of an Identity*, σελίδες 9–18. Apress, Berkeley, CA, 2019.

Συντομογραφίες - Αρκτικόλεξα - - Ακρωνύμια

ΔΕ	Διπλωματική Εργασία
κ.λπ.	και λοιπά
IdP	Identity Provider
SP	Service Provider
MFA	Multi-Factor Authentication
SSO	Single Sign-On
SAML	Security Assertion Markup Language
XML	Extensible Markup Language
URL	Uniform Resource Locators
CAS	Central Authentication Service
DS	Discovery Service
DN	Distinguished Name
SSL	Secure Sockets Layer
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
OTP	One-Time Password
TOTP	Time-Based One-Time Password
QR	Quick Response
LDAP	Lightweight Directory Access Protocol
API	Application Programming Interface

Απόδοση ξενόγλωσσων όρων

provisioning	παροχή
authorization	εξουσιοδότηση
authentication	πιστοποίηση ταυτότητας
access policy enforcement	επιβολή πολιτικής πρόσβασης
session	συνεδρία
single sign-on	εννιαία σύνδεση
multi-factor authentication	πιστοποίηση ταυτότητας πολλών παραγόντων
logout	αποσύνδεση
account management	διαχείρηση λογαριασμού
account recovery	ανάκτηση λογαριασμού
deprovisioning	αποπαροχή
cross-domain single sign-on	ενιαία σύνδεση μεταξύ τομέων
federated identity	ομοσπονδιακή ταυτότητα
identity provider	πάροχος ταυτότητας
service provider	πάροχος υπηρεσιών
metadata	μεταδεδομένα
knowledge factor	παράγοντας γνώσης
ownership factor	παράγοντας ιδιοκτησίας
biometric factor	βιομετρικός παράγοντας
location factor	παράγοντας τοποθεσίας
open source	ανοικτός κώδικας
closed source	κλειστός κώδικας