A Zero Trust Architecture implementation to secure sensitive data inside a corporate network.

Αναγνωστόπουλος Άγγελος Νικόλαος
ΑΜ: up1066593

Επιβλέπων: Φείδας Χρήστος

Πανεπιστήμιο Πατρών

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Οκτώβριος 2022

Το πρόβλημα

Προβλήματα ασφάλειας με το network access management μοντέλο του legacy VPN (Castle and moat).

Roaming users και outsoursing υπηρεσιών σε cloud service providers κάνουν δύσκολο τον ορισμό ενός network perimiter πάνω στο οποίο θα εφαρμοστούν security policies.

Λύση: Zero trust architecture

1. Collect data for every request and action on network
2. **Remove** implicit **trust from actors** from within the network.
3. **Assume breach.**
4. **Use dynamic policy for accessing requests.**
5. **Authorise and authenticate resources before access is allowed.**
6. Secure communications regardless of location
7. Grant access on per-session basis.

Μέχρι τώρα

Proprietary implementations (Cloudfare, Google, CISCO κ.α.)

Αρκετές ερευνητικές εργασίες στην αρχιτεκτονική (Και από πανεπιστήμια και grey literature).

Κάθε βέντορας ασχολείται ιδιωτικά με την ευχρηστότητα του προϊόντος, παρόλα αυτά δεν υπάρχει πολλή βιβλιογραφία στο θέμα αλληλεπίδρασης με το χρήστη.

Όλοι παρέχουν λύσεις όμως αυτές είναι κλειστού κώδικα.

Προτάσεις προς υλοποίηση

Υλοποίηση λογισμικού policy enforcement point για identity based access management.

Υλοποίηση λογισμικού "client – side" για σύνδεση στο δίκτυο.

Γλώσσα συγγραφής πιθανότατα Golang (εναλλακτικά Python).

Ανοιχτά τα πάντα προς το κοινό στο github σαν open source software.

Auditing για κάθε request/event.

Θέματα που δεν θα εστιάσουμε

Εκτεταμένη υλοποίηση zero trust solutions.

Κάλυψη πολλαπλών πλατφορμών/δικτύων.

Κάλυψη αναγκών Bring Your Own Device (BYOD).

Εφαρμογή - Παρουσίαση

Χρήση των papers της BeyondCorp και της αναφοράς του NIST για το zero trust architecture ώστε να φτιάξουμε ένα user friendly λογισμικό με τις αρχές της αρχιτεκτονικής.

Αιτήματα πρόσβασης σε κάποιο resource από διάφορους χρήστες/υπολογιστές (απλό χρήστη, admin, unregistered user, unregistered device).

Συλλογή/Ανάγνωση logs σε κάθε περίπτωση για να δούμε success/failure στα αιτήματα μας.

Γιατί Golang?

Η Golang διαθέτει πακέτα που μας λύνουν τα χέρια όσον αφορά την επικοινωνία servers και το backend programming. Είναι η βασίλισσα στον τομέα αυτό λόγω των ελαφρών της συρρουτίνων που επιτρέπουν τη διαχείρηση πολλών αιτημάτων ταυτόχρονα.

Στην περίπτωσή μας, αν και έχουμε ένα αίτημα την φορά, οι βιβλιοθήκες της θα φανούν πολύ χρήσιμες και ήθελα να ασχοληθώ με την γλώσσα αυτή λόγω της ενασχόλησης μου με DevOps/SecOps. Βρίσκω την πτυχιακή μία εξαιρετική ευκαιρία. Η Python θα παίξει βοηθητικό ρόλο όπου και εάν χριαστεί, λόγω μεγαλύτερης εξοικείωσης μαζί της και περισσότερων resources στο internet.

Τεχνικές Λεπτομέρειες (1)

Θα χρειαστεί να υλοποιηθούν μία σειρά από λειτουργίες, κομμάτια κώδικα και configurations.

Πρέπει πρώτα να ξεκινήσουμε από την αρχιτεκτονική, τα requirements και το systems design diagram.

Τα πάντα θα γίνουν με docker containers ή/και virtual machines.

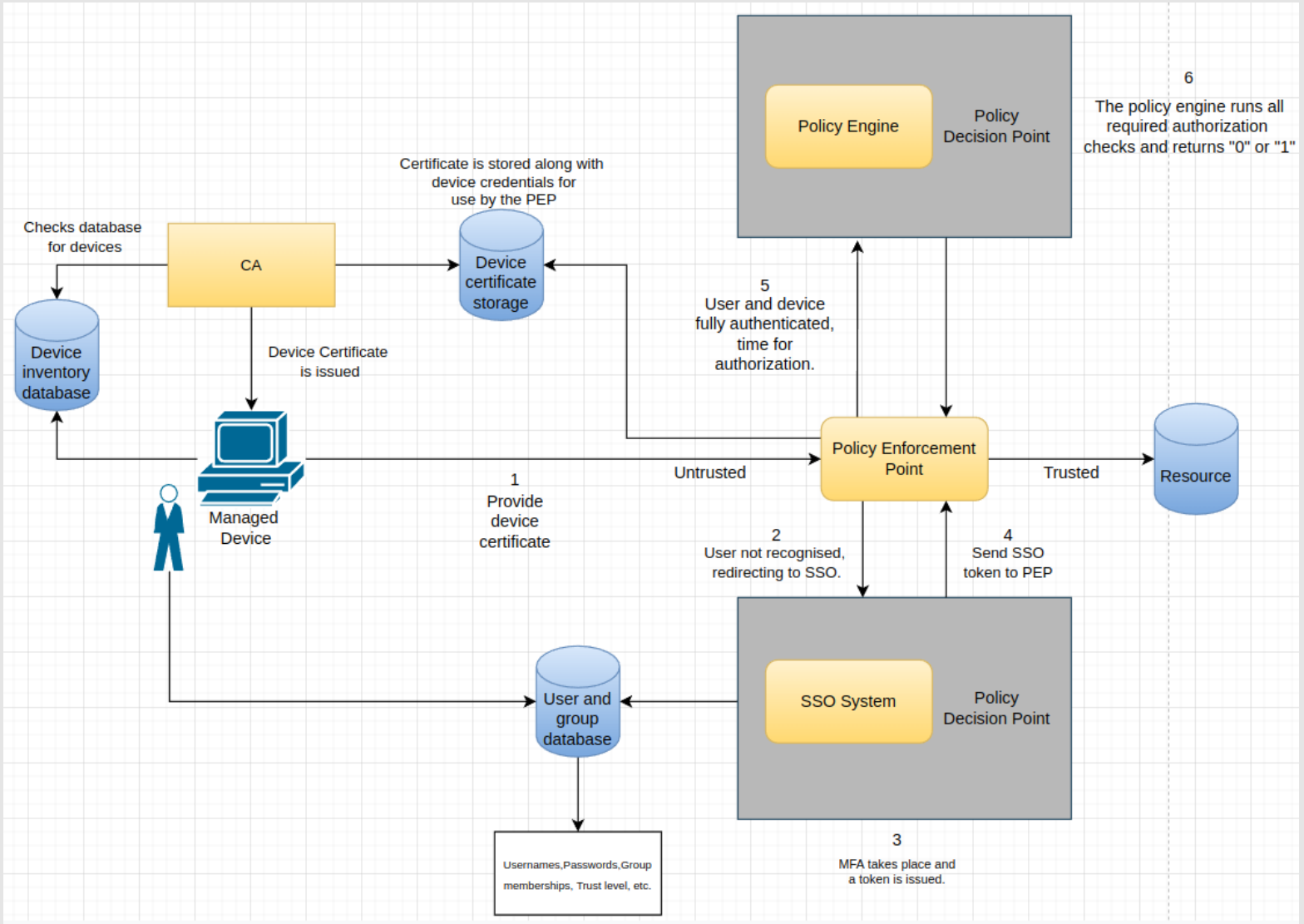
Η αρχιτεκτονική που θα ακολουθήσουμε είναι μία απλοποιημένη εκδοχή αυτών που προτείνονται από το NIST.

Θα πρέπει να υλοποιηθούν ξεχωριστά inventory database, resources databases, resource gateways, PEP, PE, SSO ή κάποιο άλλο authentication service, CA κ.α.

Είναι απαραίτητη η συλλογή logs για ό,τι συμβαίνει στο δίκτυο.

Τεχνικές Λεπτομέρειες (2)

Systems design diagram v1.0



Τεχνικές Λεπτομέρειες (3)

Στην αρχική εκδοχή τους, το CA μπορεί να δίνει απλά strings ή integers για testing και το SSO να είναι ένας απλός συνδυασμός username/password. Ιδανικά αυτά θα εξελιχθούν και ενδεχομένως γίνουν integrate τα industry standards για τα αντίστοιχα σημεία του δικτύου μας.

Θα χρειαστούμε έναν αλγόριθμο εμπιστοσύνης (trust algorithm) με τον οποίο το Policy Engine (PE) θα αποφασίζει για τα εισερχόμενα requests. Αυτός αρχικά θα είναι criteria based, όμως δεν αποκλείεται να επεκταθεί.

Κανονικά, το PEP πρέπει να βρίσκει το που βρίσκεται το προς αίτημα resource (inventory compartmentalization) και να επικοινωνεί με το αντίστοιχο gateway, το οποίο είναι και το μοναδικό που έχει πρόσβαση στο αρχείο. Σε αρχική φάση θα ακολουθηθεί το διάγραμμα v1.0, όπου το PEP έχει άμεση πρόσβαση στις βάσεις δεδομένων μας, υπάρχει όμως δυνατότητα επέκτασης και εδώ.

Οι κρίσιμες περιοχές του δικτύου είναι εφοδιασμένες με firewalls (απλοί κανόνες iptables).

Υπάρχει δυνατότητα να βάλουμε IDS/IPS στο δίκτυο σε τελική φάση.

Τεχνικές Λεπτομέρειες (4)

Μπορούν να χρησιμοποιηθούν τεχνολογίες CI/CD, code testing coverage κ.α metrics για την ποιότητα, καθώς και static code analysis για να είμαστε σίγουροι ότι γράφουμε σωστό, robust κώδικα.

Πιθανότατα στο τέλος να γίνει και κάποιο deployment και χρήση κάποιου container orchestrator (K8s).

Ευχαριστώ για την προσοχή σας!