



Πανεπιστήμιο Δυτικής Αττικής

Θέμα

Dynamic Analysis and Test Environment as a Service (DATE-A-A-S)

Συγγραφέας:
Απόστολος
Αναγνωστόπουλος

Επιβλέπων:
Δρ Βασίλειος Μάμαλης

*Μια διπλωματική εργασία που υποβάλλεται για την εκπλήρωση
των απαιτήσεων για το πτυχίο Δίπλωμα Μεταπτυχιακών Σπουδών
(Δ.Μ.Σ.)*

στο

Δίκτυα Επικοινωνιών Νέας Γενιάς και Κατανομημένα Περιβάλλοντα
Εφαρμογών
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

19 Ιουλίου 2024

Dynamic Analysis and Test Environment as a Service (DATE-A-A-S)

Εξεταστική Επιτροπή:

Ιωάννα Καντζάβελου, Αναπληρώτρια Καθηγήτρια

Βασίλειος Μάμαλης, Καθηγητής

Γραμματή Πάντζιου, Καθηγήτρια

Ημερομηνία Εξέτασης: 19/07/2024

Δήλωση συγγραφέα μεταπτυχιακής εργασίας

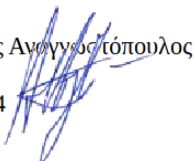
Ο κάτωθι υπογεγραμμένος Απόστολος Αναγνωστόπουλος του Νικολάου, με αριθμό μητρώου mngd21016 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι ο συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από εμένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση των επιβλεπόντων καθηγητών.

Ο Δηλών: Απόστολος Αναγνωστόπουλος

Ημερομηνία: 19/07/2024



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Περίληψη

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Δίπλωμα Μεταπτυχιακών Σπουδών (Δ.Μ.Σ.)

Dynamic Analysis and Test Environment as a Service (DATE-A-A-S)

by Απόστολος Αναγνωστόπουλος

Ανάπτυξη υπηρεσίας που σχετίζεται με δυναμική ανάλυση και δοκιμαστικά περιβάλλοντα στον τομέα της προστασίας των προσωπικών δεδομένων στην ψηφιακή εποχή. Τα στοιχεία θα είναι τα εξής:

- **Δυναμική Ανάλυση:** Η δυναμική ανάλυση θα περιλαμβάνει την αξιολόγηση της συμπεριφοράς ενός προγράμματος κατά τη διάρκεια της εκτέλεσής του. Θα χρησιμοποιηθεί για να κατανοήσουμε πώς αλληλεπιδρά ένα πρόγραμμα με το περιβάλλον του και πώς χειρίζεται τα δεδομένα. Σκοπός είναι η ανίχνευση σφαλμάτων, που δεν είναι προφανή μέσω στατικής ανάλυσης και μόνο.
- **Δοκιμαστικό Περιβάλλον:** Το Δοκιμαστικό Περιβάλλον στοχεύει στην παροχή μιας αυτοματοποιημένης διαδικασίας που επιτρέπει στους προγραμματιστές και στους δοκιμαστές να πραγματοποιήσουν δοκιμές, συμπεριλαμβανομένης της αξιολόγησης της συμπεριφοράς ενός προγράμματος κατά τη διάρκεια της εκτέλεσής του. Οι δοκιμές πραγματοποιούνται χωρίς την ανάγκη φυσικής παρουσίας, προσφέροντας καλύτερη αποτελεσματικότητα και βελτιώνοντας την ευελιξία, όπως κατά την αναθεώρηση κώδικα.

Συνδυάζοντας τα παραπάνω, το "Dynamic Analysis and Test Environment as a Service" (DATE-A-A-S) θα παρέχει δυναμικά παραμετροποιήσιμα δοκιμαστικά περιβάλλοντα ειδικά σχεδιασμένα για δυναμική ανάλυση στα πλαίσια των σχετικών εργαστηριακών μαθημάτων του πανεπιστημίου. Χαρακτηριστικά του DATE-A-A-S :

- **Επεκτασιμότητα:** Δυνατότητα να αυξάνεται ή να μειώνεται το περιβάλλον δοκιμών ανάλογα με τις απαιτήσεις.
- **Απομόνωση:** Περιβάλλον δοκιμών απομονωμένο από άλλα για την αποφυγή παρεμβολών.
- **Αυτοματοποίηση:** Αυτοματοποιημένα εργαλεία και πλαίσια δυναμικής ανάλυσης.
- **Προσαρμογή:** Προσομοίωση διαφορετικών συνθηκών για δοκιμές.
- **Αναφορές:** Αναφορές για τα αποτελέσματα της δυναμικής ανάλυσης

Abstract

Development of a Service Related to Dynamic Analysis and Test Environments in the Field of Personal Data Protection in the Digital Age

The elements will be as follows:

- **Dynamic Analysis:** Dynamic analysis will include evaluating the behavior of a program during its execution. It will be used to understand how a program interacts with its environment and handles data. The purpose is to detect errors that are not obvious through static analysis alone.
- **Test Environment:** The test environment aims to provide an automated process that allows developers and testers to conduct tests, including evaluating a program's behavior during its execution. The tests are carried out without the need for physical presence, offering better efficiency and improving flexibility, such as during code review.

Combining the above, "Dynamic Analysis and Test Environment as a Service" (DATE-A-A-S) will provide dynamically configurable test environments specifically designed for dynamic analysis within the framework of the university's related laboratory courses. Features of DATE-A-A-S include:

- **Scalability:** Ability to increase or decrease the test environment according to requirements.
- **Isolation:** Test environment isolated from others to avoid interference.
- **Automation:** Automated tools and frameworks for dynamic analysis.
- **Customization:** Simulation of different conditions for testing.
- **Reporting:** Reports on the results of the dynamic analysis.

Ευχαριστίες

Η παρούσα διπλωματική εργασία αναλύει ένα εξαιρετικά ενδιαφέρον και δύσκολο γνωστικό αντικείμενο για το οποίο απαιτήθηκε αρκετός χρόνος και προσπάθεια τόσο για την έρευνα όσο και για την υλοποίησή της. Την προσπάθειά μου αυτή υποστήριξαν οι επιβλέποντες καθηγητές μου, τους οποίους θα ήθελα να ευχαριστήσω. Μεγάλη «ευθύνη» στην όλη αυτή διαδρομή και βοήθεια για την επίτευξη του στόχου μου, είχε η οικογένεια μου και σε αυτή αφιερώνω την επιτυχία αυτή.

Περιεχόμενα

Δήλωση συγγραφέα μεταπτυχιακής εργασίας	iii
Περίληψη	v
Ευχαριστίες	vii
Περιεχόμενα	ix
Κατάλογος σχημάτων	xiii
Κατάλογος πινάκων	xv
List Of Source Code Listings	xv
1 Εισαγωγή	1
2 Dynamic Analysis and Testing Environment	3
2.1 Στατική ανάλυση	3
2.2 Δυναμική ανάλυση	4
2.3 Συμπέρασμα	5
3 Σχεδιασμός του "Dynamic Analysis and Testing Environment-As-A-Service"	7
3.1 DATE-A-A-S	7
3.1.1 Αρχιτεκτονική	8
3.1.2 Μεθοδολογία	9
3.1.3 Το "DATE-A-A-S" για χρήση στο Εργαστήριο	9
3.1.4 Το "DATE-A-A-S" on remote client	10
3.2 components	11
3.2.1 mitmproxy	11
3.2.2 WireGuard	14
3.2.3 KernelSU	14
3.2.4 Android Debug Bridge	15
3.2.5 Bliss OS	15
3.2.6 LineageOS	16
3.2.7 Android Open Source Project	17
3.2.8 Appium	18
3.2.9 Allure	19
3.3 Rest API	19
3.3.1 Api Endpoints	19
3.3.2 Api Models	21
DateAAS.ApiResponse	21
DateAAS.Category	21
DateAAS.Lab	21

	DateAAS.Order	22
	DateAAS.Tag	22
	DateAAS.User	22
	3.3.3 Authentication & Authorization	23
3.4	Εργαλεία ενορχήστρωσης	23
	3.4.1 Χαρακτηριστικά και απαιτήσεις	23
	Docker Swarm	23
	Kubernetes	24
	Openshift	25
	3.4.2 Συμπέρασμα	25
4	Υλοποίηση	27
	4.1 Bliss Os	27
	4.1.1 Κύρια Χαρακτηριστικά υλοποίησης	27
	4.1.2 Initial setup	27
	Hardware Virtualization	27
	KVM	27
	4.1.3 run Bliss OS container	27
	Παραμετροποίηση	28
	4.2 WireGuard	28
	4.2.1 base run	28
	Parameters	29
	4.2.2 Server Mode	30
	4.2.3 Client Mode	30
	4.3 kernelSU	31
	4.4 LineageOS	32
	4.4.1 Ενεργοποίηση του "USB debugging"	32
	4.4.2 Ενεργοποίηση "Developer options"	33
	4.4.3 Installing Add-Ons	33
	4.4.4 Heimdall suite	33
	4.4.5 Lineage Recovery	34
	4.4.6 LineageOS	34
	4.5 Appium	34
	4.6 Rest API	35
	4.6.1 addLab	36
	4.6.2 deleteLab	37
	4.6.3 getLabById	37
	4.6.4 findLabsByTags	38
	4.6.5 createUser	38
	4.6.6 createUsersWithArrayInput	39
	4.6.7 deleteUser	39
	4.6.8 loginUser	40
	4.6.9 logoutUser	40
	4.7 Docker Swarm	41
	4.7.1 Βήματα εγκατάστασης	41
	4.7.2 Create a swarm	41
	Εύρεση του Manager key	41
	Εύρεση του Worker key	42
	4.7.3 Προσθήκη Worker στο Swarm	42

5 Proof of concept (PoC)	43
5.1 mitmproxy config	43
5.2 mitmproxy capture	44
5.3 mitmproxy android	45
A' Δεδομένα και ψηφιακή οικονομία	49
B' Εφαρμογές περιόδου και προσωπικά δεδομένα	51
B'.1 Ευρήματα	52
B'.1.1 Maya	52
B'.1.2 MIA	53
B'.1.3 My Period Tracker	54
B'.1.4 Συμπέρασμα	54
Γ' DATE-a-a-S:Source Code	57
Γ'.1 Dockerfiles	57
Γ'.1.1 Bliss OS	57
Γ'.1.2 AndroidSDK	61
Γ'.1.3 Android Emulator	63
Γ'.1.4 WireGuard	65
Γ'.1.5 Appium	66
Appium script - start.sh	69
Appium script - generate_selenium_config	70
Appium script - wireless_autoconnect.sh	71
Appium script - wireless_connect.sh	72
Γ'.2 Compose Files	72
Γ'.2.1 WireGuard	72
Γ'.2.2 Test Environment	73
config file: wdio.conf.js	74
Γ'.2.3 Android and VPN	76
config file: up-wirequard-client.conf	77
Δ' DATE-a-a-S:Api	79
Δ'.1 API Description	79
Δ'.1.1 Yaml	79
Δ'.1.2 HTML	87
Δ'.1.3 NodeJS	88
Γλωσσάρι	89
Βιβλιογραφία	93

Κατάλογος σχημάτων

1.1	Swarmlab-Android-on-Container	2
3.1	DATE-A-A-S-services-arch	8
3.2	DATE-A-A-S-services-components-lab	10
3.3	DATE-A-A-S-services-components-client	11
3.4	DATE-A-A-S-clients-mitm	11
3.5	Arch-mitmproxy-services	13
3.6	AOSP-arch	17
3.7	AOSP-arch-components	18
3.8	AOSP-based-os	18
3.9	Arch-services-components-allure	19
3.10	Arch-services-components-swarm	26
4.1	DATE-A-A-S-blissOs-arch	28
4.2	DATE-A-A-S-WG-arch	31
4.3	Download mode	33
4.4	Recovery mode	34
5.1	DATE-A-A-S-mitmproxy-config	43
5.2	DATE-A-A-S-mitmproxy-run	44
5.3	DATE-A-A-S-mitmproxy-capture-webserver	44
5.4	DATE-A-A-S-mitmproxy-key-google	45
5.5	DATE-A-A-S-mitmproxy-capture-google-login	46
5.6	DATE-A-A-S-mitmproxy-capture-serverlog	46
5.7	DATE-A-A-S-android-run-virtual	47
5.8	DATE-A-A-S-android-run-serverlog	47
5.9	DATE-A-A-S-android-run-installappl	48
B'1	menstruations	53
B'2	menstruations	54

Κατάλογος πινάκων

3.1	API Endpoints	20
3.2	Πληροφορίες ApiResponse	21
3.3	Πληροφορίες Category	21
3.4	Πληροφορίες Lab	21
3.5	Πληροφορίες Order	22
3.6	Πληροφορίες Tag	22
3.7	Πληροφορίες User	22
4.1	Παράμετρος: lab	36
4.2	Αίτημα HTTP - addLab	36
4.3	Αίτημα HTTP - deleteLab	37
4.4	Παράμετρος: get labId	38
4.5	Παράμετρος: tags	38
4.6	Παράμετρος: User object	39
4.7	Παράμετρος: array of User objects	39
4.8	Παράμετρος: username	40
4.9	Παράμετρος: user login	40

Listings

4.1	Bliss OS - Host Initial Setup	27
4.2	Bliss OS - Host enable kvm	27
4.3	Bliss OS - Run Docker Container (Δείτε Γ.1.1)	27
4.4	WireGuard - Run Docker Container (Δείτε Γ.1.4)	28
4.5	WireGuard - Run Docker Container "Server Mode" (Δείτε Γ.2.1)	30
4.6	WireGuard - Run Docker Container "Client Mode" (Δείτε Γ.2.1)	30
4.7	KernelSU - Install	32
4.8	Appium - Enable Virtualization	35
4.9	Appium - Build docker image	35
4.10	Appium - Run Server	35
4.11	Appium - Share volume	35
4.12	addLab	36
4.13	deleteLab	37
4.14	getLabById	37
4.15	findLabsByTags	38
4.16	createUser	38
4.17	createUsersWithArrayInput	39
4.18	deleteUser	39
4.19	loginUser	40
4.20	logoutUser	40
4.21	Create a swarm	41
4.22	swarm: manager key	41
4.23	swarm: worker key	42
4.24	swarm: dwarm join	42
Γ.1	Dockerfile Bliss OS	57
Γ.2	Dockerfile Appium	66
Γ.3	Dockerfile DockerfileAppiumScript-start	69
Γ.4	Dockerfile DockerfileAppiumScript-generate_selenium_config	70
Γ.5	Dockerfile DockerfileAppiumScript-wireless_autoconnect	71
Γ.6	Dockerfile DockerfileAppiumScript-wireless_connect	72

Αφιερώνω αυτή τη διπλωματική εργασία στην οικογένειά μου, για την αδιάκοπη υποστήριξη και αγάπη τους, και στον καθηγητή μου, Δρ. Β. Μάμαλη, για την πολύτιμη καθοδήγησή του και την αμέριστη βοήθειά του καθ' όλη τη διάρκεια των μεταπτυχιακών σπουδών μου. Χωρίς τη συμβολή τους, αυτό το έργο δεν θα είχε ολοκληρωθεί.

Κεφάλαιο 1

Εισαγωγή

Στην εποχή της ψηφιακής τεχνολογίας, η εκθετική αύξηση της δημιουργίας δεδομένων αποτελεί μια αδιαμφισβήτητη πραγματικότητα. Τα δεδομένα διεισδύουν σε κάθε τομέα της ανθρώπινης δραστηριότητας, μεταβάλλοντας τον τρόπο που ζούμε, εργαζόμαστε και αλληλεπιδρούμε με το περιβάλλον μας. Η εξάπλωση των smartphones και των συσκευών του Διαδικτύου των Πραγμάτων (IoT) έχει ενισχύσει περαιτέρω αυτή την ανάπτυξη, καθώς οι αλληλεπιδράσεις των χρηστών, οι μετρήσεις αισθητήρων και τα δεδομένα από τις συσκευές συμβάλλουν στο διαρκώς αυξανόμενο οικοσύστημα δεδομένων (Παράρτημα: A').

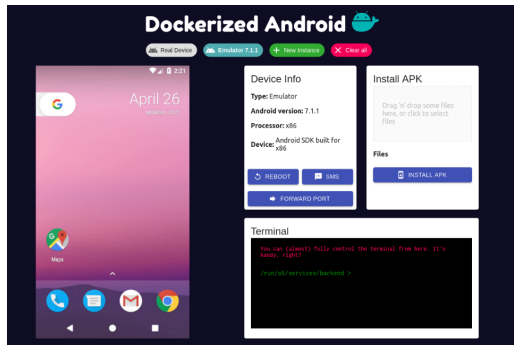
Το τεράστιο μέγεθος των παραγόμενων δεδομένων εγείρει θεμελιώδη ερωτήματα σχετικά με την ιδιωτικότητα, τη δεοντολογία, την ιδιοκτησία και τη δίκαιη πρόσβαση σε αυτά (Banerjee *and others*, 2011) (Guardian, 2018). Οι φορητές συσκευές αποτελούν επίσης συχνά το πρώτο ευάλωτο σημείο, τόσο για τους "χάκερς" όσο και για τις αμφιλεγόμενες πρακτικές των κατασκευαστών των εφαρμογών (Παράρτημα: B') (Luo *and others*, 2022) (Xue *and others*, 2019). Είναι, επομένως, ζωτικής σημασίας να διαθέτουμε εργαλεία που επιτρέπουν τη δοκιμή και την ανάλυση εφαρμογών σε ένα ελεγχόμενο και ασφαλές περιβάλλον, χωρίς τον κίνδυνο πρόκλησης ζημιών στον πραγματικό κόσμο, μειώνοντας τον κίνδυνο παραβίασης της ιδιωτικότητας και προστατεύοντας τους χρήστες από κακόβουλες επιθέσεις και πρακτικές, ενώ μέσω της συστηματικής δοκιμής και ανάλυσης διασφαλίζεται ότι οι εφαρμογές πληρούν τα πρότυπα ασφάλειας και προστασίας δεδομένων, διαφυλάσσοντας έτσι τα δικαιώματα και την ασφάλεια των χρηστών.

Η παρούσα εργασία αποσκοπεί στην υλοποίηση του πλαισίου DAATE-A-A-S, το οποίο αντιμετωπίζει το σημαντικό ζήτημα της ιδιωτικότητας σε περιβάλλον Dockerized_Service. Η έρευνα καλύπτει ευρέως φορητές και μη φορητές συσκευές, καθώς και web εφαρμογές, ανεξαρτήτως του περιβάλλοντος λειτουργίας, και περιγράφει εκτενώς τον σχεδιασμό και την υλοποίηση της πλατφόρμας, επιδεικνύοντας πώς οι διάφορες λειτουργίες μπορούν να εφαρμοστούν επιτυχώς για την υλοποίηση περίπλοκων σεναρίων σε διάφορες συσκευές και υπηρεσίες. Στόχος είναι η παρουσίαση ενός αποτελεσματικού πλαισίου που διευκολύνει την ομαλή ένταξη του συστήματος σε περιβάλλον βασισμένο σε containers. Το "Δυναμικό Περιβάλλον Ανάλυσης και Δοκιμών" που αναπτύσσεται, αντιμετωπίζει θέματα προστασίας δεδομένων, παρέχοντας ένα ρεαλιστικό και ασφαλές περιβάλλον για μελέτη και εκπαίδευση. Επιπλέον, προσφέρει περιβάλλον προσομοίωσης για συσκευές Android, αυτόνομες εφαρμογές H/Y και web εφαρμογές σε περιβάλλον βασισμένο σε containers.

Ιδιαίτερη έμφαση έχει δοθεί στις συσκευές Android για διάφορους λόγους.

- Είναι ανοικτού κώδικα, επομένως είναι δυνατή η εύκολη αναπαραγωγή του περιβάλλοντος.
- Τρέχει σχεδόν σε κάθε συσκευή.

- Υπάρχουν πολλά εξωτερικά εργαλεία που μπορούν εύκολα να ενσωματωθούν για να παρέχουν επεκταμένες δυνατότητες.



Σχήμα 1.1: Android-on-Container¹

¹Πηγή: <https://arxiv.org/pdf/2205.09493.pdf>

Κεφάλαιο 2

Dynamic Analysis and Testing Environment

2.1 Στατική ανάλυση

Η στατική ανάλυση κώδικα εφαρμόζεται στα πρώτα στάδια της ανάπτυξης και πριν ξεκινήσει ο δοκιμαστικός έλεγχος του λογισμικού, υποστηρίζοντας το DevOps στο να δημιουργήσει έναν αυτοματοποιημένο κύκλο ανατροφοδότησης και βοηθώντας τις ομάδες ανάπτυξης να παραδίδουν ποιοτικές εκδόσεις εγκαίρως, τηρώντας ταυτόχρονα τις προδιαγραφές προγραμματισμού και συμμόρφωσης.

Οι περιορισμοί των εργαλείων στατικής ανάλυσης κώδικα περιλαμβάνουν την έλλειψη κατανόησης της πρόθεσης του προγραμματιστή και τον περιορισμένο τους ρόλο σε συγκεκριμένα στάδια ανάπτυξης. Στο παρακάτω παράδειγμα, ένα εργαλείο στατικής ανάλυσης μπορεί να ανιχνεύσει πιθανό overflow, αλλά δεν μπορεί να προσδιορίσει ότι η συνάρτηση ουσιαστικά δεν κάνει αυτό που αναμένεται!

```
int calculateArea(int length, int width)
{
    return (length + width);
}
```

Ωστόσο, η στατική ανάλυση αποτελεί ουσιαστικό εργαλείο για τη διασφάλιση της αξιοπιστίας, της ασφάλειας και της συντηρησιμότητας των εφαρμογών λογισμικού.

Τα οφέλη της στατικής ανάλυσης περιλαμβάνουν:

- **Πρόωρη ανίχνευση προβλημάτων:** Με την ενσωμάτωση της στατικής ανάλυσης στη διαδικασία ανάπτυξης, οι προγραμματιστές μπορούν να εντοπίζουν προβλήματα εγκαίρως, επιτρέποντάς τους να τα αντιμετωπίζουν πριν εξελιχθούν σε μεγαλύτερα προβλήματα. Αυτό μειώνει τον χρόνο και την προσπάθεια που απαιτούνται για αποσφαλμάτωση και συντήρηση και εξασφαλίζει ότι ο κώδικας είναι αξιόπιστος και ασφαλής.
- **Μείωση του κόστους:** Η αντιμετώπιση προβλημάτων στα πρώιμα στάδια του SDLC¹ μπορεί να μειώσει το κόστος επισκευής σφαλμάτων και άλλων προβλημάτων αργότερα στη διαδικασία. Αυτό μπορεί να εξοικονομήσει χρόνο και πόρους και να μειώσει τον κίνδυνο καθυστερήσεων ή άλλων προβλημάτων που μπορεί να επηρεάσουν το χρονοδιάγραμμα του έργου.
- **Βελτίωση της ποιότητας του κώδικα:** Η στατική ανάλυση βοηθά στην εντοπισμό παραβιάσεων προτύπων προγραμματισμού και άλλων προβλημάτων που μπορεί να επηρεάσουν την ποιότητα του κώδικα. Με την αντιμετώπιση αυτών των προβλημάτων εγκαίρως, οι προγραμματιστές μπορούν να

¹Software Development Life Cycle (SDLC)

εξασφαλίσουν ότι ο κώδικας είναι καλογραμμένος, συντηρήσιμος και πιο εύκολος στην αποσφαλμάτωση.

- **Ενίσχυση της ασφάλειας:** Τα εργαλεία στατικής ανάλυσης μπορούν να εντοπίσουν ευπάθειες ασφαλείας στον κώδικα, επιτρέποντας στους προγραμματιστές να αντιμετωπίσουν αυτά τα προβλήματα πριν ο κώδικας κυκλοφορήσει στην παραγωγή. Αυτό μπορεί να μειώσει τον κίνδυνο παραβιάσεων ασφαλείας και άλλων προβλημάτων που μπορεί να επηρεάσουν την ασφάλεια της εφαρμογής.

2.2 Δυναμική ανάλυση

Η δυναμική ανάλυση αναφέρεται στην εξέταση και αξιολόγηση της συμπεριφοράς ενός προγράμματος κατά τη διάρκεια της εκτέλεσής του. Περιλαμβάνει την παρατήρηση του τρόπου με τον οποίο το λογισμικό αλληλεπιδρά με το περιβάλλον του, επεξεργάζεται την είσοδο, εκτελεί υπολογισμούς και παράγει έξοδο κατά τη διάρκεια της εκτέλεσης.

Οι τεχνικές δυναμικής ανάλυσης χρησιμοποιούνται συχνά για διάφορους σκοπούς, συμπεριλαμβανομένων:

- **Δοκιμή(Testing):** Η δυναμική ανάλυση χρησιμοποιείται ευρέως στον έλεγχο λογισμικού για την εντοπισμό ελαττωμάτων, σφαλμάτων και ευπαθειών που ενδέχεται να μην είναι προφανή μέσω στατικής ανάλυσης (εξέταση του κώδικα χωρίς να τον εκτελέσει). Παρατηρώντας τη συμπεριφορά του λογισμικού κατά τη διάρκεια της εκτέλεσης, οι δοκιμαστές μπορούν να ανιχνεύσουν σφάλματα κατά την εκτέλεση, διαρροές μνήμης, σημεία μπλοκαρίσματος απόδοσης και άλλα θέματα.
- **Προφίλ(Profiling):** Το προφίλ είναι μια μορφή δυναμικής ανάλυσης που επικεντρώνεται στη μέτρηση διαφόρων πτυχών της εκτέλεσης ενός προγράμματος, όπως η χρήση CPU, η χρήση μνήμης και ο χρόνος εκτέλεσης. Τα εργαλεία προφίλ συλλέγουν δεδομένα σχετικά με τον τρόπο με τον οποίο το πρόγραμμα συμπεριφέρεται κατά τη διάρκεια της εκτέλεσης, βοηθώντας τους προγραμματιστές να ανιχνεύσουν τα σημεία απόδοσης και τις περιοχές για βελτιστοποίηση.
- **Ανάλυση ασφαλείας(Security Analysis):** Η δυναμική ανάλυση είναι απαραίτητη στις δοκιμές ασφαλείας για τον εντοπισμό ευπαθειών και δυνητικών κινδύνων ασφαλείας σε εφαρμογές λογισμικού. Με την ανάλυση της συμπεριφοράς του προγράμματος κατά τη διάρκεια της εκτέλεσης, οι αναλυτές ασφαλείας μπορούν να ανιχνεύσουν ελαττώματα ασφαλείας όπως υπερχείλιση του buffer, injection attacks και παράκαμψη πιστοποίησης.
- **Ανάλυση συμπεριφοράς(Behavioral Analysis):** Η δυναμική ανάλυση μπορεί να χρησιμοποιηθεί για την κατανόηση της συμπεριφοράς ενός προγράμματος σε πραγματικές σενάρια (Tran and others, 2012). Παρατηρώντας τον τρόπο με τον οποίο το λογισμικό αντιδρά σε διαφορετικές εισόδους, οι προγραμματιστές μπορούν να "καταλάβουν" τη λειτουργικότητα, τις εξαρτήσεις και τις αλληλεπιδράσεις του με εξωτερικά συστήματα.

Οι τεχνικές δυναμικής ανάλυσης περιλαμβάνουν τη χρήση ειδικών εργαλείων για την παρακολούθηση της εκτέλεσης του προγράμματος και τη συλλογή σχετικών δεδομένων.

2.3 Συμπέρασμα

Το έργο αυτό έχει ως στόχο την υλοποίηση ενός πλαισίου που αντιμετωπίζει και αναλύει το πρόβλημα της ιδιωτικότητας για φορητές και μη συσκευές, καθώς και για web εφαρμογές. Σε αυτό το πλαίσιο, τα εργαλεία για δυναμική ανάλυση θα παίξουν καθοριστικό ρόλο στην επίτευξη του στόχου μας. Η δυναμική ανάλυση θα μας επιτρέψει να εξερευνήσουμε τη συμπεριφορά του συστήματος κατά την εκτέλεση, ενώ τα εργαλεία θα μας παρέχουν τις απαραίτητες λειτουργίες για την ανάλυση και την επίλυση προβλημάτων που προκύπτουν από την πολυπλοκότητα των εικονικών σεναρίων. Είναι σημαντικό να σημειωθεί ότι η στατική ανάλυση δεν αποτελεί τον κύριο παράγοντα εδώ, καθώς η δυναμική προσέγγιση είναι αυτή που επιτρέπει την πλήρη κατανόηση και αντιμετώπιση των προκλήσεων που συναντώνται.

Κεφάλαιο 3

Σχεδιασμός του "Dynamic Analysis and Testing Environment-As-A-Service"

3.1 DATE-A-A-S

Σκοπός αυτού του κεφαλαίου είναι ο σχεδιασμός της υπηρεσίας "Dynamic Analysis and Testing Environment As A Service". Σε αυτό το κεφάλαιο, θα περιγραφούν τα διάφορα συστατικά μέρη της υπηρεσίας, που θα παρέχουν ενημερώσεις και ειδοποιήσεις για κάθε συμβάν ή ενέργεια που σχετίζεται με τις λειτουργίες των εφαρμογών.

Περιβάλλον Λειτουργίας

Οι τεχνολογίες που θα εφαρμοστούν θα λειτουργούν μέσα σε ένα περιβάλλον βασισμένο σε container. Αυτό θα προσφέρει τα εξής πλεονεκτήματα:

- **Αξιοπιστία:** Τα containers επιτρέπουν τη δημιουργία ενός σταθερού και προβλέψιμου περιβάλλοντος λειτουργίας, το οποίο ελαχιστοποιεί τα προβλήματα που μπορεί να προκύψουν από διαφορές στις ρυθμίσεις του συστήματος.
- **Ευκολία Διαχείρισης:** Τα containers διευκολύνουν τη διαχείριση και τη συντήρηση του περιβάλλοντος, καθώς επιτρέπουν την εύκολη ανάπτυξη, αναβάθμιση και κλιμάκωση των εφαρμογών.
- **Φορητότητα:** Τα containers προσφέρουν τη δυνατότητα εύκολης μεταφοράς των εφαρμογών μεταξύ διαφορετικών περιβαλλόντων, διασφαλίζοντας ότι οι εφαρμογές θα λειτουργούν με τον ίδιο τρόπο ανεξάρτητα από την πλατφόρμα ή την υποδομή.

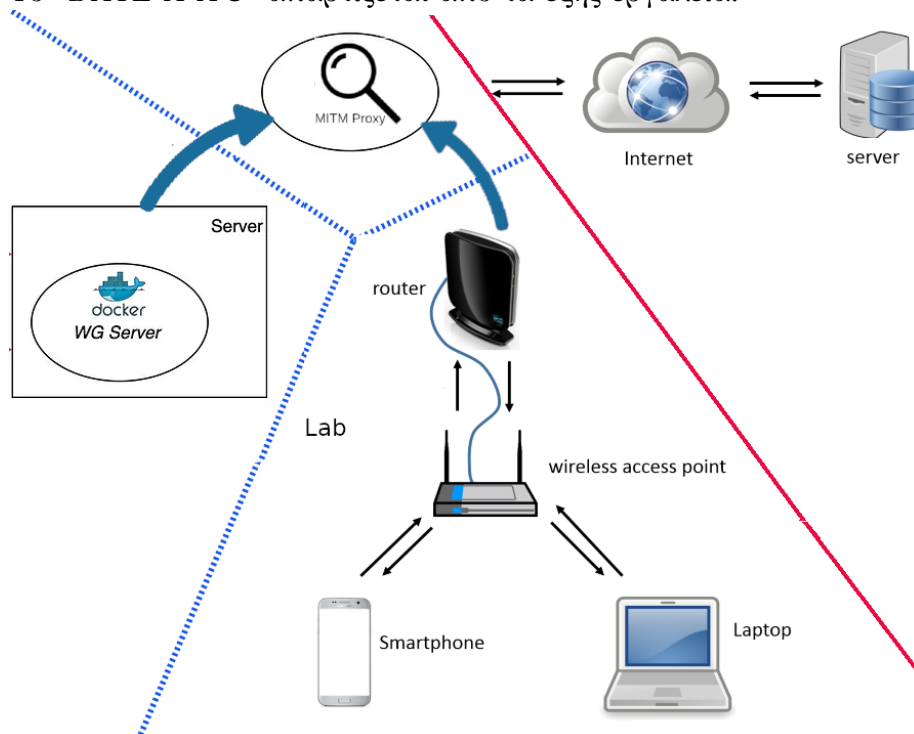
Στόχος

Ο στόχος είναι να δημιουργηθεί ένα αξιόπιστο και εύκολα διαχειρίσιμο περιβάλλον για τη μελέτη και εκπαίδευση. Σε αυτό το περιβάλλον, οι χρήστες θα μπορούν να αναλύουν και να δοκιμάζουν τις εφαρμογές τους με ασφάλεια και αποτελεσματικότητα, λαμβάνοντας ταυτόχρονα ενημερώσεις και ειδοποιήσεις για κάθε σημαντικό συμβάν που σχετίζεται με τις λειτουργίες των εφαρμογών τους.

Η υπηρεσία θα λειτουργεί τόσο απομακρυσμένα όσο και σε εργαστήριο, παρέχοντας στους φοιτητές την ευελιξία να χρησιμοποιούν την πλατφόρμα από οπουδήποτε, αλλά και να συμμετέχουν σε πρακτικά μαθήματα και εργαστήρια στο χώρο του πανεπιστημίου.

3.1.1 Αρχιτεκτονική

Το "DATE-A-A-S" απαρτίζεται από τα εξής εργαλεία:



Σχήμα 3.1: DATE-A-A-S-services-arch

- Container με WireGuard - VPN
- Container με Mitmproxy: Το mitmproxy λειτουργεί σε "διαφανή" λειτουργία, προσφέροντας λειτουργία παρέμβασης χωρίς την γνώση του πελάτη. Περιλαμβάνει τα απαραίτητα εργαλεία για τη δημιουργία λειτουργικού σημείου πρόσβασης στο δίκτυο.
- Container με Bliss OS for PC - "Android for PC"
- Ενσωμάτωση του KernelSU - "A kernel-based root solution for Android"
- Κινητό Android: Εκτελείται με Lineage OS για υποστήριξη εκδόσεων Android.
- Φορητός Υπολογιστής με ADB: Χρησιμοποιείται για την εγκατάσταση του mitmproxy πιστοποιητικού στο Σύστημα Trust λόγω περιορισμών ασφαλείας.
- Container με Appium: Ένα εργαλείο στον τομέα του αυτοματοποιημένου ελέγχου λογισμικού, για να διαπιστώσει εάν η λειτουργικότητα μιας εφαρμογής λειτουργεί όπως αναμένεται, προσβάσιμο από πολλές γλώσσες προγραμματισμού όπως Java, Python, Ruby, JS, κ.ά.
- Wi-Fi hotspot: Το Wi-Fi hotspot επιτρέπει στις συσκευές, όπως τα κινητά τηλέφωνα, να συνδεθούν στο διαδίκτυο μέσω του κινητού τηλεφώνου ή ενός άλλου συμβατού συστήματος που λειτουργεί ως hotspot¹

¹Σε αυτό το σενάριο:

- PC με λειτουργικό σύστημα Linux ως τον βασικό υπολογιστή στον οποίο θα εκτελούνται native εφαρμογές.
 - VM με Windows μέσω Oracle VirtualBox: Το Packer χρησιμοποιείται για τη δυναμική δημιουργία του VM.
 - Container με Linux: Αυτόματη εγκατάσταση του λογισμικού που πρόκειται να ελεγχθεί.

3.1.2 Μεθοδολογία

Η δυναμική ανάλυση των εφαρμογών περιλαμβάνει τα παρακάτω βήματα:

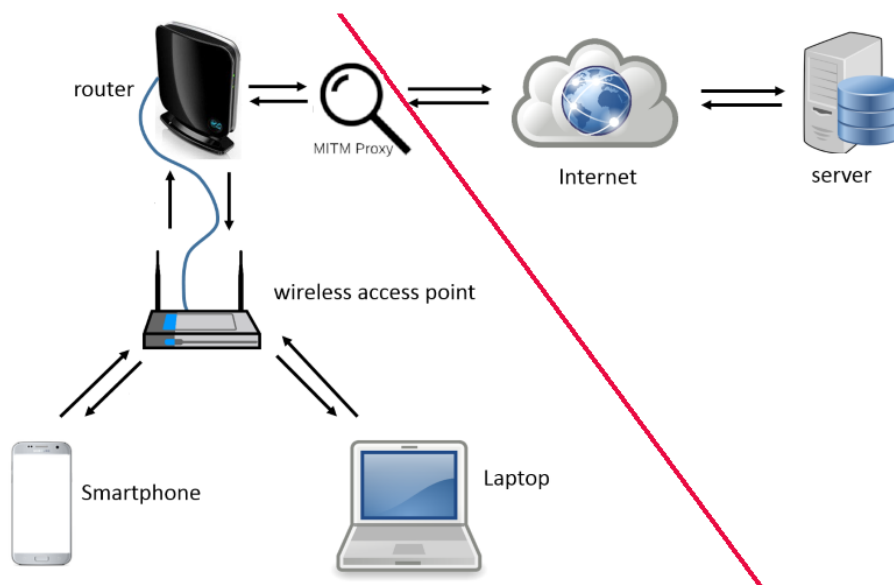
- Σύνδεση σε Wi-Fi χωρίς παρέμβαση.
- Χρήση εφαρμογών.
 - Εφαρμογές Windows/Linux.
 - Λήψη εφαρμογών από το Google Play Store.
- Ανακατεύθυνση στο mitmproxy.
 - Αποθήκευση δικτυακής συνεδρίας στη βάση Mongo
- Άνοιγμα εφαρμογής, εγγραφή, και κανονικές δραστηριότητες.
- Αποθήκευση στιγμιότυπων οθόνης και διακοπή ροής στο mitmproxy.
- Απεγκατάσταση της εφαρμογής.

Ο συγκεκριμένος τρόπος λειτουργίας επιτρέπει την ομαλή ενσωμάτωση και τη ρύθμιση του περιβάλλοντος εργαστηρίου, επιτρέποντας την αξιοποίηση των προηγούμενα περιγραφέντων τεχνολογιών στη μελέτη και την εκπαίδευση.

3.1.3 Το "DATE-A-A-S" για χρήση στο Εργαστήριο

Το "DATE-A-A-S" απαρτίζεται από τα εξής εργαλεία σε Container για Χρήση στο Εργαστήριο:

- Κινητά Τηλέφωνα Android: Τα κινητά τηλέφωνα Android συνδέονται στο Wi-Fi hotspot που παρέχεται από το Bliss OS ή το ίδιο το κινητό τηλέφωνο.
- Bliss OS: Το Bliss OS λειτουργεί ως ένα είδος εικονικού μηχανήματος Android για το PC. Σε αυτήν την περίπτωση, μπορεί επίσης να δημιουργήσει ένα Wi-Fi hotspot για να επιτρέψει στα κινητά τηλέφωνα να συνδεθούν.
- Διαφανές Wi-Fi: Η σύνδεση σε ένα Wi-Fi χωρίς παρέμβαση είναι σημαντική για την εκτέλεση των δοκιμών, καθώς διευκολύνει την καταγραφή της επικοινωνίας μεταξύ των εφαρμογών και των διακομιστών χωρίς παρεμβολή.

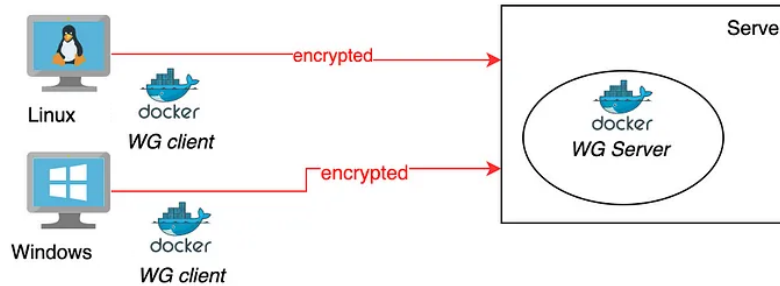


Σχήμα 3.2: DATE-A-A-S-services-components-lab

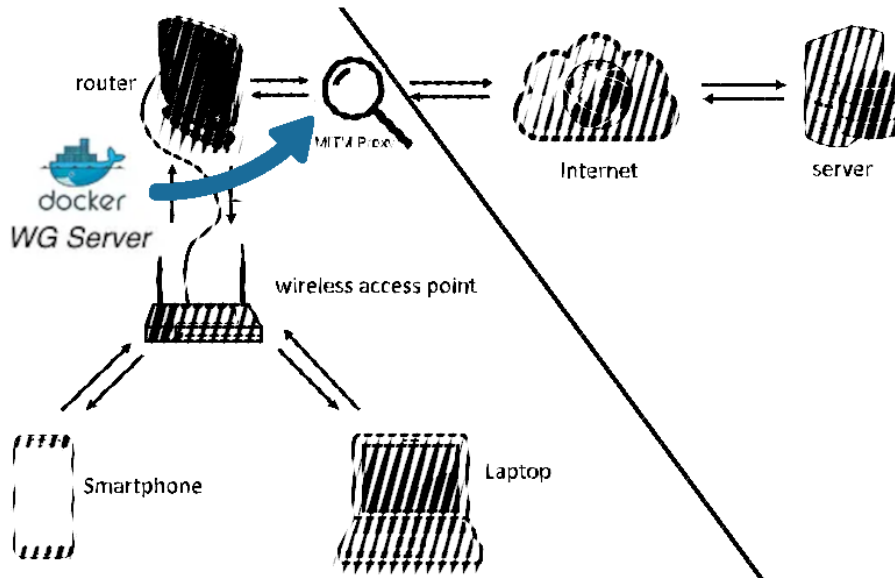
3.1.4 Το "DATE-A-A-S" on remote client

Το "DATE-A-A-S" απαρτίζεται από τα εξής εργαλεία Περιβάλλοντος Container για απομακρυσμένη χρήση.

- Σύνδεση σε docker WG server.
- Χρήση εφαρμογών.
 - Εφαρμογές Windows/Linux.
 - Λήψη εφαρμογών από το Google Play Store.
- Ανακατεύθυνση στο mitmproxy.
- Άνοιγμα εφαρμογής, εγγραφή, και κανονικές δραστηριότητες.
- Αποθήκευση στιγμιότυπων οθόνης και διακοπή ροής στο mitmproxy.
- Απεγκατάσταση της εφαρμογής.



Σχήμα 3.3: DATE-A-A-S-services-components-client



Σχήμα 3.4: DATE-A-A-S-clients-mitm

3.2 components

3.2.1 mitmproxy

Το mitmproxy αποτελεί ένα σύνολο εργαλείων που παρέχουν ένα διαδραστικό, SSL/TLS-ικό intercepting proxy για HTTP/1, HTTP/2 και WebSockets (Pingle, Mairaj and Javaid, 2018). Παρέχει διάφορα χαρακτηριστικά που επιτρέπουν τον έλεγχο, την εγγραφή και την επεξεργασία της κυκλοφορίας HTTP και HTTPS (Yaacoub and others, 2023).

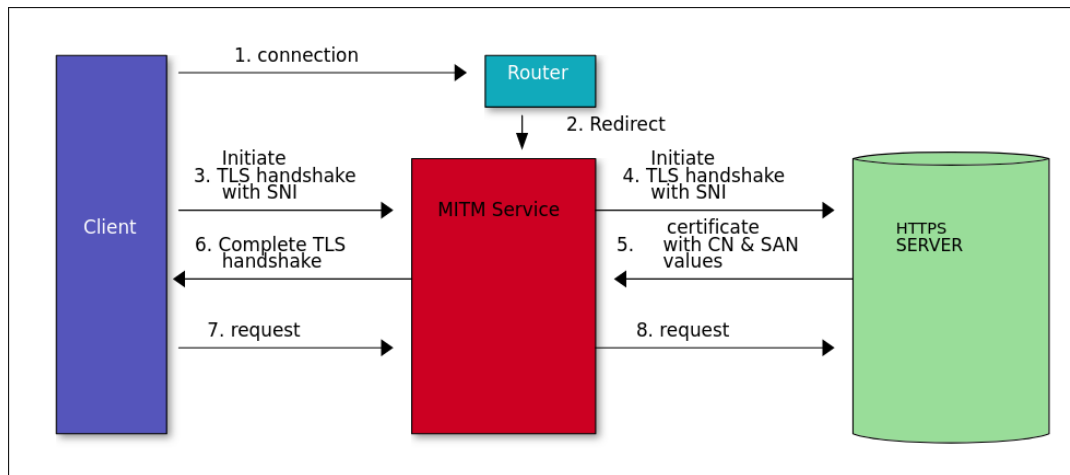
Χαρακτηριστικά:

- **Επέμβαση σε HTTP & HTTPS:** Επιτρέπει την παρέμβαση σε αιτήσεις και απαντήσεις HTTP και HTTPS, καθώς και την τροποποίησή τους κατά τη διάρκεια της μετάδοσης.

- **Αποθήκευση συνομιλιών HTTP:** Μπορεί να αποθηκεύει πλήρεις συνομιλίες HTTP για μετέπειτα επαναφορά και ανάλυση.
- **Επανάληψη συνομιλίας:** Επιτρέπει την επανάληψη του πελάτη ή των απαντήσεων ενός προηγούμενως καταγεγραμμένου server.
- **Λειτουργία αντίστροφου proxy:** Μπορεί να λειτουργήσει ως αντίστροφο proxy για την προώθηση κίνησης σε έναν καθορισμένο διακομιστή.
- **Λειτουργία διαφανούς proxy:** Υποστηρίζει διαφανή λειτουργία proxy σε macOS και Linux, επιτρέποντας τη χρήση του χωρίς να χρειάζεται αλλαγή ρυθμίσεων.
- **Επικεφαλίδες HTTP με Python:** Παρέχει τη δυνατότητα δημιουργίας σεναρίων με τη χρήση Python για επεξεργασία της κυκλοφορίας HTTP.
- **Δυναμική δημιουργία πιστοποιητικών SSL/TLS:** Δημιουργεί πιστοποιητικά SSL/TLS κατά τη διάρκεια της επέμβασης.

Το mitmproxy αποτελεί ένα ισχυρό εργαλείο για τον έλεγχο της κυκλοφορίας σε εφαρμογές που χρησιμοποιούν πρωτόκολλα HTTP και HTTPS.

Η διαδικασία που θα εφαρμοστεί στο DATE-A-A-S είναι:



Σχήμα 3.5: mitmproxy

1. Ο πελάτης πραγματοποιεί σύνδεση με τον εξυπηρετητή (server).
2. Ο δρομολογητής ανακατευθύνει τη σύνδεση στο mitmproxy, το οποίο συνήθως είναι σε λειτουργία ακρόασης σε ένα τοπικό θύρα του ίδιου υπολογιστή. Το mitmproxy στη συνέχεια εξετάζει το μηχανισμό δρομολόγησης για να καθορίσει ποιος ήταν ο αρχικός προορισμός.
3. Ο πελάτης πιστεύει ότι επικοινωνεί με τον απομακρυσμένο εξυπηρετητή και εκκινεί τη σύνδεση TLS. Χρησιμοποιεί το SNI για να υποδείξει το όνομα του κεντρικού υπολογιστή (hostname) στο οποίο συνδέεται.
4. Το mitmproxy συνδέεται στον εξυπηρετητή και καθιερώνει μια σύνδεση TLS χρησιμοποιώντας το όνομα του κεντρικού υπολογιστή (hostname) που υποδείχθηκε από τον πελάτη.
5. Ο εξυπηρετητής ανταποκρίνεται με το αντίστοιχο πιστοποιητικό, το οποίο περιέχει τις τιμές CN (Common Name) και SAN (Subject Alternative Name) που απαιτούνται για τη δημιουργία του πιστοποιητικού παρέμβασης.
6. Το mitmproxy δημιουργεί το πιστοποιητικό παρέμβασης και συνεχίζει τη σύνδεση TLS του πελάτη που είχε παραμείνει στο στάδιο 3.
7. Ο πελάτης αποστέλλει το αίτημά του μέσω της καθιερωμένης σύνδεσης TLS.
8. Το mitmproxy μεταβιβάζει το αίτημα στον εξυπηρετητή μέσω της σύνδεσης TLS που ξεκίνησε στο στάδιο 4.

3.2.2 WireGuard

Το WireGuard είναι ένα πρωτόκολλο επικοινωνίας και λογισμικό ανοιχτού κώδικα που υλοποιεί κρυπτογραφημένα εικονικά ιδιωτικά δίκτυα (VPNs) και σχεδιάστηκε με στόχους ευκολίας χρήσης, υψηλής απόδοσης και χαμηλού επιπέδου επίθεσης. Στοιχείει σε καλύτερη απόδοση από τα IPsec και OpenVPN, πρωτόκολλα τούνελ. Το πρωτόκολλο WireGuard μεταδίδει την κυκλοφορία δεδομένων μέσω του πρωτοκόλλου UDP

Τον Μάρτιο του 2020, η έκδοση του λογισμικού για το Linux έφτασε σε μια σταθερή παραγωγική έκδοση και ενσωματώθηκε στον πυρήνα Linux 5.6, ενώ προστέθηκε και σε προηγούμενες εκδόσεις του πυρήνα Linux². ο WireGuard χαρακτηρίζεται από τη χρήση προηγμένης κρυπτογραφίας, ενσωματώνοντας τεχνολογίες όπως το πλαίσιο πρωτοκόλλου Noise, το Curve25519, το ChaCha20, το Poly1305, το BLAKE2, το SipHash24, το HKDF, και ασφαλείς κατασκευές. Αυτό δίνει στο WireGuard τη δυνατότητα να προσφέρει υψηλή ασφάλεια επικοινωνίας. (Dowling and Paterson, 2018) Οι επιλογές του WireGuard είναι συντηρητικές και λογικές, έχοντας υποβληθεί σε αναθεώρηση από κρυπτογράφους. Η χρήση πρωτοκόλλων όπως το Curve25519 για την ανταλλαγή κλειδιών, το ChaCha20 για τη συμμετρική κρυπτογράφηση, και το Poly1305 για τους κωδικούς ελέγχου ακεραιότητας επιδιώκει υψηλή απόδοση και ασφάλεια. Το γεγονός ότι βασίζεται μόνο στο πρωτόκολλο UDP το καθιστά ελαφρύ και κατάλληλο για γρήγορη επικοινωνία, ενώ η χρήση του BLAKE2s, του SipHash24 και του HKDF προσθέτει επιπλέον στρώματα ασφαλείας. (Donenfeld, 2017) (Jumakhan and Mirzaeinia, 2024)

Γνωστοί Περιορισμοί:

- Deep Packet Inspection: Το WireGuard δεν επικεντρώνεται στην συγκεκριμένη λειτουργία (obfuscation). Αντιθέτως, η συγκάλυψη θα πρέπει να συμβαίνει σε ένα επίπεδο πάνω από το WireGuard(μπορεί να επιτευχθεί από έργα όπως το Shadowsocks³), το οποίο επικεντρώνεται στην παροχή στερεάς κρυπτογραφίας με μια απλή υλοποίηση.
- Λειτουργία TCP: Το WireGuard δεν υποστηρίζει ρυθμίσεις μεταφοράς μέσω TCP λόγω της κακής απόδοσης του δικτύου κατά την μεταφορά TCP-επάνω-από-TCP. Η μετατροπή των πακέτων UDP του WireGuard σε TCP είναι δουλειά ενός ανώτερου επιπέδου και μπορεί να επιτευχθεί από έργα όπως τα udrtunnel και udp2raw.

3.2.3 KernelSU

- kernel-based root λύση πρόσβασης για Android: Όπως υποδηλώνει το όνομά, το KernelSU λειτουργεί υπό τον πυρήνα Linux, προσφέροντας της μεγαλύτερο έλεγχο επί των εφαρμογών χρήστη στον χώρο των χρηστών (userspace).
- Root Access Control: Μόνο εφαρμογές που έχουν εξουσιοδοτηθεί μπορούν να έχουν πρόσβαση ή να δουν το su, όλες οι άλλες εφαρμογές δεν είναι ενημερωμένες γι' αυτό.
- Προσαρμόσιμα προνόμια Root: Το KernelSU επιτρέπει την προσαρμογή του su ως προς το uid, gid, groups, δυνατότητες (capabilities), και κανόνες SELinux, κλειδώνοντας τα προνόμια ρίζας.

²Με άδεια χρήσης GNU General Public License (GPL) έκδοση 2

³Το Shadowsocks είναι ένα εργαλείο που χρησιμοποιείται για την ανάπτυξη ενός ιδιωτικού proxy server, που επιτρέπει στους χρήστες να παρακάμπτουν την λογοκρισία στο Διαδίκτυο και να προστατεύουν την ιδιωτικότητά τους.

- Modules: Μπορούν να τροποποιήσουν το /system χωρίς επίπτωση στο σύστημα χρησιμοποιώντας το overlaysfs.

Το KernelSU είναι χρήσιμο για τη μελέτη και τον έλεγχο εφαρμογών, ιδίως όταν πρέπει να ξεπεραστούν περιορισμοί που έχουν ενσωματωθεί στο Android.

Τρόπος χρήσης του στο περιβάλλον DATE-A-A-S:

- Θα επιτρέψει την εκτέλεση διαφόρων ερευνητικών δραστηριοτήτων, όπως η ανάλυση του συστήματος ή η εποπτεία της κίνησης δεδομένων.
- Θα επιτρέψει να τεστάρετε εφαρμογές υπό διάφορες συνθήκες, συμπεριλαμβανομένης της πρόσβασης σε δεδομένα που συνήθως είναι απροσπέλαστα.
- Προσαρμογή του λειτουργικού συστήματος, προσθέτοντας προηγμένες ρυθμίσεις και δυνατότητες, χωρίς περιορισμούς.

3.2.4 Android Debug Bridge

Το Android Debug Bridge (ADB) είναι ένα εργαλείο που παρέχεται από το Android SDK (Software Development Kit) και χρησιμοποιείται για την επικοινωνία με συσκευές Android, όπως smartphones και tablets, από έναν υπολογιστή.

Ορισμένα χαρακτηριστικά του ADB περιλαμβάνουν (Regupathy, 2014):

- Σύνδεση συσκευών: Επιτρέπει τη σύνδεση του υπολογιστή με μια συσκευή Android μέσω USB, Wi-Fi ή Bluetooth.
- Εκτέλεση Εντολών: Εκτέλεση εντολών από υπολογιστή στη συσκευή Android. Για παράδειγμα, εγκατάσταση και απεγκατάσταση εφαρμογών, αντιγραφή αρχείων κ.λπ.
- Αποσφαλμάτωση (Debugging): Το ADB επιτρέπει την εκτέλεση εντολών αποσφαλμάτωσης στη συσκευή Android, όπως εκτύπωση καταγραφικών μηνυμάτων και εκκίνηση του λειτουργικού συστήματος σε λειτουργία αποσφαλμάτωσης.

θα χρησιμοποιηθεί για την εγκατάσταση του mitmproxy πιστοποιητικού στο σύστημα Trust λόγω περιορισμών ασφαλείας⁴

3.2.5 Bliss OS

Το Bliss OS είναι ένα λειτουργικό σύστημα ανοικτού κώδικα βασισμένο στο Android Open Source Project (AOSP), σχεδιασμένο για να λειτουργεί σε προσωπικούς υπολογιστές, φορητούς υπολογιστές και άλλες συμβατές συσκευές. Το ουσιαστικό είναι ότι φέρνει την εμπειρία του Android σε παραδοσιακές πλατφόρμες υπολογιστών, προσφέροντας στους χρήστες τη γνωστή διεπαφή του Android σε μεγαλύτερες οθόνες.

Κύρια χαρακτηριστικά του Bliss OS είναι:

⁴Το "Σύστημα Trust" (Trust System) ή "Σύστημα Πιστοποίησης" ή "Διαχείρισης Πιστοποιητικών" (Certificate Management System) είναι υπεύθυνο για τη διαχείριση και τον έλεγχο των πιστοποιητικών ασφαλείας σε μια Android συσκευή. Όταν μια εφαρμογή ή μια υπηρεσία χρειάζεται πρόσβαση σε ασφαλείς πόρους, όπως συνδέσεις HTTPS, χρησιμοποιείται το πιστοποιητικό για να επιβεβαιωθεί η ταυτότητα του διακομιστή. Αυτά τα πιστοποιητικά αποθηκεύονται και διαχειρίζονται στο Σύστημα Trust. Με την εγκατάσταση πιστοποιητικών του mitmproxy, σημαίνει ότι θα επιτρέψει στο mitmproxy να παρεμβαίνει σε ασφαλείς συνδέσεις και να αναλύει την κίνηση HTTPS.

- **Συμβατότητα:** Το Bliss OS σχεδιάστηκε για να λειτουργεί σε μια ευρεία γκάμα συσκευών, συμπεριλαμβανομένων Chromebooks, Η/Υ και tablet που κυκλοφόρησαν την τελευταία δεκαετία.
- **Βελτιστοποιήσεις και Χαρακτηριστικά:** Το λειτουργικό σύστημα περιλαμβάνει βελτιστοποιήσεις και επιπλέον χαρακτηριστικά πέρα από την τυπική λειτουργία Android. Αυτές οι βελτιστοποιήσεις είναι σχεδιασμένες για να ενισχύσουν την απόδοση, την εμπειρία χρήστη και τη συμβατότητα με διάφορες ρυθμίσεις υλικού.
- **Υποστήριξη Εφαρμογών ARM/ARM64:** Το Bliss OS επεκτείνει τη συμβατότητά του υποστηρίζοντας εφαρμογές ARM και ARM64 σε υπολογιστές. Αυτό σημαίνει ότι ο χρήστης μπορεί να εκτελεί εφαρμογές που αρχικά σχεδιάστηκαν για διάφορες αρχιτεκτονικές.
- **Hardware Acceleration:** Το Bliss OS περιλαμβάνει χαρακτηριστικά όπως η επιτάχυνση υλικού VA-API για την αποκωδικοποίηση βίντεο.
- **XtMapper Keymapper:** Διευκολύνει τη χρήση πληκτρολογίου και ποντικιού για τον έλεγχο εφαρμογών με ευκολία.
- **Immutable System:** Εξασφαλίζει ότι ο πυρήνας του λειτουργικού συστήματος παραμένει αναλλοίωτος, ενώ το ριζικό αρχείο συστήματος για το σύστημα παραμένει προσβάσιμο μόνο για ανάγνωση. Αυτό παρέχει έναν ασφαλή τρόπο ενημέρωσης και συντήρησης της κρίσιμης δομής του συστήματος.

Το Bliss OS είναι επίσης συμβατό με εικονικό περιβάλλον (virtual environment) και Docker, μια δυνατότητα που θα αξιοποιήσουμε εκτενώς στην υλοποίηση του DATE-A-A-S.

3.2.6 LineageOS

Το LineageOS είναι ένα ανοικτού κώδικα λειτουργικό σύστημα βασισμένο στο Android Open Source Project (AOSP), σχεδιασμένο να λειτουργεί σε διάφορες συσκευές, όπως smartphones, tablets και smart TVs. Πρόκειται για έναν διάδοχο του CyanogenMod, ένα άλλο δημοφιλές προσαρμοσμένο Android λειτουργικό σύστημα που διακόπηκε το 2016.

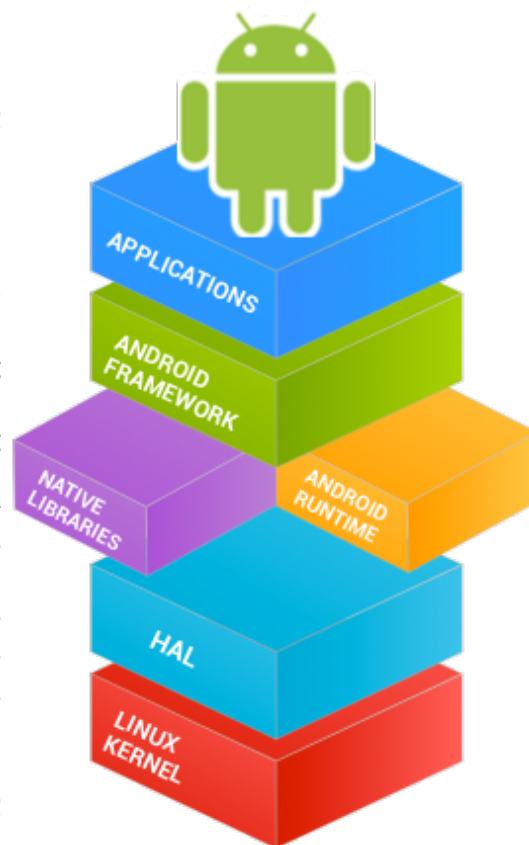
Ορισμένα βασικά χαρακτηριστικά και πλεονεκτήματα του LineageOS περιλαμβάνουν (Ponakala and Dailey, 2020)

- **Ανοικτός Κώδικας:** Το LineageOS είναι ελεύθερο λογισμικό και ανοικτού κώδικα, πράγμα που σημαίνει ότι ο πηγαίος κώδικας του είναι ελεύθερα προσβάσιμος για παραμετροποίηση και προσαρμογή.
- **Υποστήριξη Πολλών Συσκευών:** Υποστηρίζει μια εκτεταμένη λίστα συσκευών, καλύπτοντας πολλά μοντέλα και εταιρείες.
- **Ενημερώσεις Ασφαλείας και Ενημερώσεις Android:** Το LineageOS παρέχει τακτικές ενημερώσεις ασφαλείας και ενημερώσεις βάσης Android, ακόμα και για συσκευές που δεν λαμβάνουν πλέον επίσημες ενημερώσεις από τους κατασκευαστές τους.
- **Κοινότητα Χρηστών:** Υπάρχει μια ενεργή κοινότητα πίσω από το LineageOS που παρέχει υποστήριξη, συμβουλές και επίλυση προβλημάτων μέσω φόρουμ και άλλων πλατφορμών.

Είναι μια εξαιρετική επιλογή για το DATE-A-A-S, κυρίως λόγω των εκτενών δυνατοτήτων προσαρμογής και ρύθμισης του λειτουργικού συστήματος. Η δυνατότητα αυτή μας παρέχει την ευελιξία να προσαρμόσουμε το περιβάλλον σύμφωνα με τις ακριβείς ανάγκες του έργου μας. Σε συνδυασμό με την ευρεία υποστήριξη συσκευών του LineageOS, ανοίγει τον δρόμο για την υποστήριξη περισσότερων συσκευών που θα εξασφαλίζει ότι το DATE-A-A-S θα λειτουργεί αποτελεσματικά και απροβλημάτιστα σε μια ποικιλία συσκευών.

3.2.7 Android Open Source Project

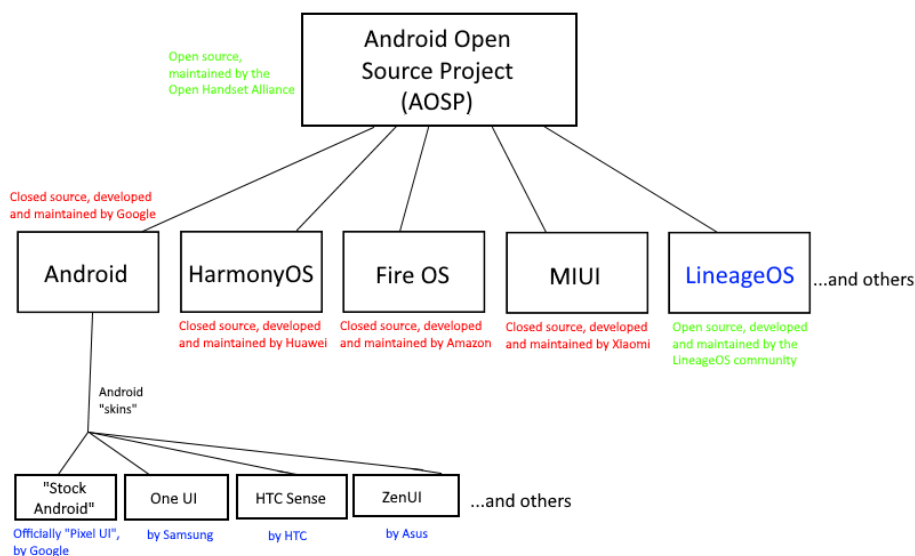
Το AOSP είναι ένα λειτουργικό σύστημα για κινητές συσκευές και είναι ένα έργο ανοιχτού κώδικα. Πρόκειται για ένα πλήρες λειτουργικό σύστημα υψηλής ποιότητας, διαθέτοντας πηγαίο κώδικα που είναι προσαρμόσιμος και μπορεί να μεταφερθεί σε σχεδόν οποιαδήποτε συσκευή. Επίσης, παρέχει δημόσια τεκμηρίωση που είναι προσβάσιμη σε όλους (Song and Yang, 2017) (K.V, Sharmila and Manjunath, 2014). Ωστόσο, η πλειονότητα των συσκευών λειτουργεί με την ιδιόκτητη έκδοση του Android που αναπτύχθηκε από τη Google που περιλαμβάνει κλειστού κώδικα λογισμικό που προεγκαθίσταται, κυρίως τις Google Mobile Services (GMS), που περιλαμβάνουν πυρήνα εφαρμογών όπως το Google Chrome, την πλατφόρμα διανομής Google Play, και τις Google Play Services. Περισσότερο από το 70% των smartphones που βασίζονται στο Android Open Source Project λειτουργούν με το οικοσύστημα της Google, γνωστό απλώς ως Android. Ορισμένα από αυτά παρουσιάζουν προσαρμοσμένα περιβάλλοντα χρήστη και σουίτες λογισμικού, όπως τα TouchWiz και One UI από τη Samsung, καθώς και το HTC Sense. Επιπλέον, υπάρχουν και άλλα οικοσυστήματα και παρακλάδια του AOSP, όπως το Fire OS από την Amazon, το ColorOS από την Oppo, το OriginOS από την Vivo, το MagicUI από την Honor, καθώς και προσαρμοσμένες ROM όπως το LineageOS (Krutz and others, 2015)



Σχήμα 3.6: AOSP-arch



Σχήμα 3.7: AOSP-arch-components



Σχήμα 3.8: AOSP-based-os

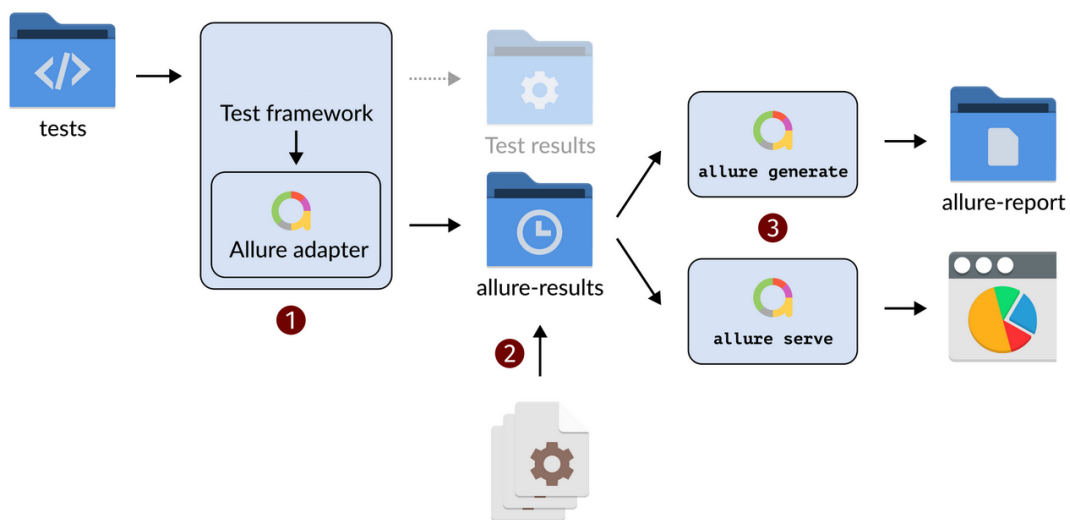
3.2.8 Appium

Το Appium είναι ένα έργο ανοιχτού κώδικα, σχεδιασμένο για την αυτοματοποίηση χρήσης εφαρμογών σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των κινητών (iOS, Android), περιηγητών (Chrome, Firefox, Safari), τηλεόρασης (Android TV, Samsung) και πολλών άλλων. Χρησιμοποιείται κυρίως στην αυτοματοποίηση δοκιμών λογισμικού και τον έλεγχο της λειτουργικότητας των εφαρμογών

(π.χ. αν λειτουργεί σύμφωνα με τις προσδοκίες). Το Appium παρέχει ένα ομοιόμορφο σύνολο εργαλείων που υποστηρίζουν αυτόν τον τύπο αυτοματοποίησης σε οποιαδήποτε πλατφόρμα. Ενσωματώνει αυτές τις τεχνολογίες μέσω των δημοφιλών γλωσσών προγραμματισμού, όπως Python, Ruby, JS, κ.ά.

3.2.9 Allure

Το Allure Framework αποτελείται από ένα σύνολο προσαρμογέων και παραδειγμάτων για το Allure Report. Είναι ένα ευέλικτο, ελαφρύ εργαλείο αναφοράς δοκιμών με υποστήριξη πολλαπλών γλωσσών. Με τη χρήση του Allure Framework, οι ομάδες ανάπτυξης μπορούν να δημιουργήσουν όχι μόνο εξαιρετικά δοκιμαστικά περιβάλλοντα αλλά και αναφορές δοκιμών που παρέχουν σαφή εικόνα του ποιος τεστ έχει εκτελεστεί, τα αποτελέσματά τους και τις πιθανές αποκλίσεις.



Σχήμα 3.9: Allure How it works

3.3 Rest API

3.3.1 Api Entpoints

Όλες οι διευθύνσεις URI είναι σχετικές με το <https://195.130.109.103/v2>

Class	Μέθοδος	Method info url	HTTP request	Περιγραφή
DateAAS.LabApi	<i>addLab</i>	docs/LabApi.md#addLab)	POST /lab	Προσθήκη νέου εργαστηρίου στον server
DateAAS.LabApi	<i>deleteLab</i>	docs/LabApi.md#deleteLab)	DELETE /lab/labId	Διαγραφή ενός εργαστηρίου
DateAAS.LabApi	<i>findLabsByStatus</i>	docs/LabApi.md#findLabsByStatus)	GET /lab/findByStatus	Εύρεση εργαστηρίων κατά κατάσταση
DateAAS.LabApi	<i>findLabsByTags</i>	docs/LabApi.md#findLabsByTags)	GET /lab/findByTags	Εύρεση εργαστηρίων με ετικέτες
DateAAS.LabApi	<i>getLabById</i>	docs/LabApi.md#getLabById)	GET /lab/labId	Εύρεση εργαστηρίου με βάση το ID
DateAAS.LabApi	<i>updateLabWithForm</i>	docs/LabApi.md#updateLabWithForm)	POST /lab/labId	Ενημέρωση εργαστηρίου
DateAAS.StoreApi	<i>getInventory</i>	docs/StoreApi.md#getInventory)	GET /store/inventory	Λίστα εργαστηρίων κατά κατάσταση
DateAAS.UserApi	<i>createUser</i>	docs/UserApi.md#createUser)	POST /user	Δημιουργία χρήστη
DateAAS.UserApi	<i>createUsersWithArrayInput</i>	docs/UserApi.md#createUsersWithArrayInput)	POST /user/createWithArray	Δημιουργία χρηστών με πίνακα
DateAAS.UserApi	<i>createUsersWithListInput</i>	docs/UserApi.md#createUsersWithListInput)	POST /user/createWithList	Δημιουργία χρηστών με λίστας
DateAAS.UserApi	<i>deleteUser</i>	docs/UserApi.md#deleteUser)	DELETE /user/username	Διαγραφή χρήστη
DateAAS.UserApi	<i>getUserByName</i>	docs/UserApi.md#getUserByName)	GET /user/username	Εύρεση χρήστη με βάση το όνομα χρήστη
DateAAS.UserApi	<i>loginUser</i>	docs/UserApi.md#loginUser)	GET /user/login	Σύνδεση χρήστη στο σύστημα
DateAAS.UserApi	<i>logoutUser</i>	docs/UserApi.md#logoutUser)	GET /user/logout	Αποσύνδεση της τρέχουσας συνεδρίας χρήστη

Πίνακας 3.1: API Endpoints

3.3.2 Api Models

- DateAAS.ApiResponse
- DateAAS.Category
- DateAAS.Lab
- DateAAS.Order
- DateAAS.Tag
- DateAAS.User
- DateAAS.ApiResponse
- DateAAS.Category
- DateAAS.Lab
- DateAAS.Order
- DateAAS.Tag
- DateAAS.User

DateAAS.ApiResponse

Name	Type	Notes
code	Number	[optional]
type	String	[optional]
message	String	[optional]

Πίνακας 3.2: Πληροφορίες ApiResponse

DateAAS.Category

Name	Type	Notes
id	Number	[optional]
name	String	[optional]

Πίνακας 3.3: Πληροφορίες Category

DateAAS.Lab

Name	Type	Notes
id	Number	[optional]
name	String	
category	Category	[optional]
tags	[Tag]	[optional]
status	String	[optional]

Πίνακας 3.4: Πληροφορίες Lab

Enum: StatusEnum

- **available** (value: "available")
- **pending** (value: "pending")
- **error** (value: "error")

DateAAS.Order

Name	Type	Notes
id	Number	[optional]
labId	Number	[optional]
quantity	Number	[optional]
runDate	Date	[optional]
complete	Boolean	[optional]
status	String	[optional]

Πίνακας 3.5: Πληροφορίες Order

Enum: StatusEnum

- **running** (value: "running")
- **approved** (value: "approved")
- **scheduled** (value: "scheduled")

DateAAS.Tag

Name	Type	Notes
id	Number	[optional]
name	String	[optional]

Πίνακας 3.6: Πληροφορίες Tag

DateAAS.User

Name	Type	Notes
id	Number	[optional]
username	String	[optional]
firstName	String	[optional]
lastName	String	[optional]
email	String	[optional]
password	String	[optional]
phone	String	[optional]
userStatus	Number	[optional]

Πίνακας 3.7: Πληροφορίες User

3.3.3 Authentication & Authorization

api_key

- **Type:** API key
- **API key parameter name:** api_key
- **Location:** HTTP header

lab_auth

- **Type:** OAuth
- **Flow:** implicit
- **Authorization URL:** <https://195.130.109.103/oauth/authorize>
- **Scopes:**
 - **read:labs:** read labs
 - **write:labs:** modify labs

3.4 Εργαλεία ενορχήστρωσης

Τα υφιστάμενα πλαίσια εργασίας που μπορούν να βοηθήσουν στην υλοποίηση, είναι:

- **Kubernetes:** μια πλατφόρμα ενορχήστρωσης "υπηρεσιών-σε-κοντέινερ" ανοιχτού κώδικα που επιτρέπει στους χρήστες να αυτοματοποιούν την ανάπτυξη, την κλιμάκωση και τη διαχείριση εφαρμογών με κοντέινερ.
- **Docker Swarm:** ένα εγγενές εργαλείο ομαδοποίησης για το Docker που επιτρέπει στους χρήστες να δημιουργούν και να διαχειρίζονται ένα σμήνος κόμβων Docker.
- **OpenShift:** μια πλατφόρμα ενορχήστρωσης "υπηρεσιών-σε-κοντέινερ", είναι χτισμένο πάνω στο Kubernetes και παρέχει πρόσθετα χαρακτηριστικά και λειτουργίες για τη διαχείριση υβριδικών και multi-cloud περιβαλλόντων.

3.4.1 Χαρακτηριστικά και απαιτήσεις

Docker Swarm

Το Docker Swarm είναι ένα εργαλείο ενορχήστρωσης "υπηρεσίες-σε-κοντέινερ" που χρησιμοποιείται για τη διαχείριση και την κλιμάκωση ενός συνόλου κοντέινερ που εκτελούνται σε πολλαπλούς υπολογιστικούς κόμβους. Η αρχιτεκτονική του είναι σχεδιασμένη για να επιτρέπει στους χρήστες να δημιουργούν και να διαχειρίζονται ένα σύνολο Docker hosts, γνωστό ως σμήνος ή cluster.

Τα βασικά στοιχεία της αρχιτεκτονικής του Docker Swarm είναι:

- **Κόμβοι (Nodes):** Ένας κόμβος είναι ένας διακομιστής σε ένα Docker Swarm και μπορεί να είναι είτε ένας Manager (Διαχειριστής) είτε ένας Worker (Εργάτης). Οι διαχειριστές διαχειρίζονται το σμήνος και την κατάσταση του, ενώ οι εργάτες εκτελούν τα κοντέινερ.

- **Manager Nodes/Swarm manager** (Κόμβοι Διαχείρισης): Οι κόμβοι διαχείρισης διαχειρίζονται το σμήνος και τις λειτουργίες του, όπως την κατανομή εργασιών, την εξισορρόπηση φορτίου και την επικοινωνία μεταξύ των κόμβων. Για λόγους ανθεκτικότητας, συνήθως υπάρχουν πολλαπλοί κόμβοι διαχείρισης σε ένα σμήνος.
- **Worker Nodes/Swarm worker** (Κόμβοι Εργασίας): Οι κόμβοι εργασίας είναι υπεύθυνοι για την εκτέλεση των κοντέινερ που έχουν ανατεθεί από τους κόμβους διαχείρισης. Δεν έχουν εξουσία λήψης αποφάσεων στο σμήνος και απλά εκτελούν τις εντολές που λαμβάνουν.
- **Services και Tasks** (Υπηρεσίες και Εργασίες): Στο Docker Swarm, μια υπηρεσία είναι ο ορισμός των εργασιών που πρέπει να εκτελεστούν. Οι εργασίες είναι οι πραγματικές εντολές που εκτελούνται στους κόμβους εργασίας και αντιπροσωπεύουν μια εκτέλεση ενός κοντέινερ.
- **Load Balancing** (Εξισορρόπηση Φορτίου): Το Docker Swarm παρέχει ενσωματωμένη εξισορρόπηση φορτίου για να διασφαλίσει ότι κανένας κόμβος δεν υπερφορτώνεται.
- **Scaling** (Κλιμάκωση): Το Docker Swarm επιτρέπει την εύκολη κλιμάκωση των εφαρμογών, επιτρέποντας την προσθήκη ή την αφαίρεση κοντέινερ ανάλογα με τις απαιτήσεις του φορτίου.

Ένα από τα κύρια πλεονεκτήματα του Docker Swarm είναι η ελαφριά του φύση σε συνδυασμό με την αυτοματοποιημένη εξισορρόπηση φορτίου, διασφαλίζοντας την αποδοτική κατανομή των πόρων μεταξύ των κοντέινερ. Επιπλέον, η άρρηκτη σύνδεσή του με το Docker CLI και άλλα εργαλεία του Docker, όπως το Docker Compose, ενισχύει την ευελιξία και την ευκολία χρήσης, προσφέροντας έξυπνες στρατηγικές επιλογής κόμβων και ένα ειδικό API για το Swarm⁵.

Kubernetes

Μια συστάδα Kubernetes περιλαμβάνει επίσης δύο τύπους κόμβων: **κύριους κόμβους** και **κόμβους εργασίας**. Οι κύριοι κόμβοι φιλοξενούν τις πτυχές του επιπέδου ελέγχου της συστάδας, συμπεριλαμβανομένου του χρονοπρογραμματισμού για τα pods(**Pod**) σε όλους τους πόρους και το τελικό σημείο API με το οποίο αλληλεπιδρούν οι χρήστες. Οι κόμβοι υπολογισμού εκτελούν φόρτους εργασίας για τους χρήστες της συστάδας. Η συστάδα διαθέτει εγγενείς υπηρεσίες Kubernetes, όπως etcd(**etcd**) και DNS, που αναπτύσσονται ως pods(**Pod**) εντός του Kubernetes, οι οποίες είναι υποχρεωτικές ή προαιρετικές ανάλογα με τον τύπο της υπηρεσίας.

Υπάρχουν τρεις κύριες αρχιτεκτονικές συστάδων Kubernetes: **single-master**, **multi-master** και **multi-master με load balancer**. Οι κόμβοι master είναι κρίσιμοι για τη λειτουργία της συστάδας και το μέγεθος και η ποσότητά τους απαιτούν ιδιαίτερη προσοχή. Οι υπολογιστικοί κόμβοι μπορούν να προστίθενται και να αφαιρούνται από τη συστάδα ανάλογα με τις ανάγκες, ώστε να προσαρμόζονται στην κλίμακα των εφαρμογών που φιλοξενούν.

⁵Είναι σημαντικό να κατανοήσουμε τις διαφορές μεταξύ του Docker Swarm και του Docker Compose. Παρόλο που και τα δύο εργαλεία διευκολύνουν τη διαχείριση πολλαπλών κοντέινερ, το Docker Compose εστιάζει στην αυτοματοποίηση εφαρμογών σε έναν μόνο υπολογιστή, ενώ το Docker Swarm επεκτείνεται σε μια ολόκληρη συστάδα υπολογιστών.

OpenShift

Το OpenShift είναι μια πλατφόρμα ως υπηρεσία (PaaS) που βασίζεται στο Kubernetes, αλλά προσθέτει κάποια επιπλέον χαρακτηριστικά και λειτουργίες που δεν παρέχονται από το Kubernetes από μόνο του. Οι βασικές διαφορές και προσθήκες που κάνει το OpenShift στον πυρήνα του Kubernetes είναι:

- **Ενσωματωμένη Ασφάλεια:** Το OpenShift περιλαμβάνει περισσότερες ενσωματωμένες πολιτικές ασφαλείας και χρήστη, όπως αυθεντικοποίηση και έλεγχο ταυτότητας, ενώ το Kubernetes από μόνο του δεν προσφέρει αυτές τις λειτουργίες σε τόσο εκτεταμένο βαθμό.
- **Αυτόματη Εγκατάσταση και Ενημερώσεις:** Το OpenShift παρέχει ευκολότερη διαχείριση και αυτοματοποίηση για την εγκατάσταση, την αναβάθμιση και την κλιμάκωση του συστήματος χωρίς downtime.
- **Περιβάλλοντα Ανάπτυξης και Εκτέλεσης:** Το OpenShift περιλαμβάνει ενσωματωμένα περιβάλλοντα ανάπτυξης (IDEs) και εργαλεία για την κατασκευή, τον σχεδιασμό και την ανάπτυξη εφαρμογών, κάτι που το Kubernetes δεν προσφέρει από μόνο του.
- **Ενσωματωμένος Εξισορροπητής Φορτίου και Διαχείριση Κυκλοφορίας:** Το OpenShift περιλαμβάνει ενσωματωμένους εξισορροπητές φορτίου και προηγμένες δυνατότητες διαχείρισης της κυκλοφορίας δικτύου.
- **Ενσωματωμένο CI/CD:** Το OpenShift περιλαμβάνει ενσωματωμένα εργαλεία για συνεχή ολοκλήρωση και συνεχή παράδοση (CI/CD), ενώ το Kubernetes συνήθως απαιτεί εξωτερικά εργαλεία για αυτήν τη λειτουργία.
- **Υποστήριξη και Κοινότητα:** Το OpenShift προσφέρεται από τη Red Hat με επαγγελματική υποστήριξη και εκτεταμένη κοινότητα, ενώ το Kubernetes είναι ένα ανοιχτού κώδικα project που υποστηρίζεται από την Cloud Native Computing Foundation (CNCF).

Συνοψίζοντας, το OpenShift επεκτείνει τις λειτουργίες του Kubernetes προσθέτοντας περισσότερες ενσωματωμένες λειτουργίες, ευκολία χρήσης και επαγγελματική υποστήριξη, καθιστώντας το πιο κατάλληλο για επιχειρηματικές εφαρμογές και περιβάλλοντα παραγωγής.

3.4.2 Συμπέρασμα

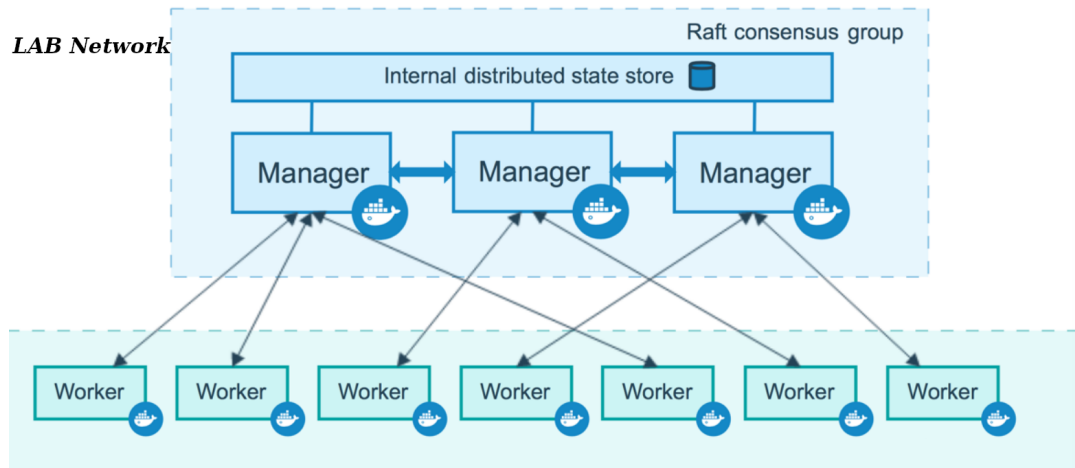
Όλες οι πλατφόρμες έχουν τη δική τους καμπύλη εκμάθησης και χρειάζεται να επενδύσει κανείς χρόνο για να τα χρησιμοποιήσει αποτελεσματικά⁶

- Το Docker Swarm είναι σχετικά απλό στην εγκατάσταση και τη διαχείριση και ενσωματώνεται καλά με άλλα εργαλεία Docker.
- Το Kubernetes είναι πιο πολύπλοκο και απαιτεί περισσότερη εγκατάσταση, αλλά χρησιμοποιείται ευρέως και διαθέτει μια μεγάλη κοινότητα χρηστών και συνεργατών.

⁶Είναι δυνατή η ενσωμάτωση ετερογενών πλατφορμών ανοιχτού κώδικα "υπηρεσιών-σε-κοντέινερ", κάτι τέτοιο όμως απαιτεί σημαντική προσπάθεια ανάπτυξης και προσαρμοσμένες λύσεις και δεν υπάρχει καμία ενσωματωμένη λειτουργία που να διευκολύνει αυτή τη διαδικασία.

- Το OpenShift είναι χτισμένο πάνω στο Kubernetes και παρέχει πρόσθετα χαρακτηριστικά για επιχειρηματικά περιβάλλοντα, αλλά μπορεί να είναι πιο δύσκολο στην εγκατάσταση και τη διαχείριση.

Η αρχιτεκτονική του Docker Swarm είναι σχεδιασμένη για ad-hoc χρήση, αυτοματοποίηση, και ευελιξία, καθιστώντας το ιδανικό για περιβάλλοντα που απαιτούν γρήγορη κλιμάκωση και συνεχή διαθεσιμότητα.



Σχήμα 3.10: Docker swarm

Κεφάλαιο 4

Υλοποίηση

4.1 Bliss Os

4.1.1 Κύρια Χαρακτηριστικά υλοποίησης

- να λειτουργεί ως αυτόνομο λειτουργικό σύστημα σε συμβατό υλικό.
- να παρέχει πλήρη εμπειρία Android σε μη-κινητές συσκευές.
- να επιτρέπει την προσαρμογή και ρύθμιση για διάφορες χρήσεις.
- και την δυνατότητα scale

4.1.2 Initial setup

Hardware Virtualization

Ενεργοποίηση του "hardware virtualization" στο BIOS και εγκατάσταση του QEMU και των εξαρτήσεων στον υπολογιστή:

```
sudo apt install qemu qemu-kvm libvirt-clients libvirt-daemon-system
bridge-utils virt-manager
```

Listing 4.1: Bliss OS - Host Initial Setup

KVM

Ενεργοποίηση του libvirt και ενσωμάτωση του KVM στον πυρήνα:

```
sudo systemctl enable --now libvirtd
sudo systemctl enable --now virtlogd

echo 1 | sudo tee /sys/module/kvm/parameters/ignore_msrs

sudo modprobe kvm
```

Listing 4.2: Bliss OS - Host enable kvm

4.1.3 run Bliss OS container

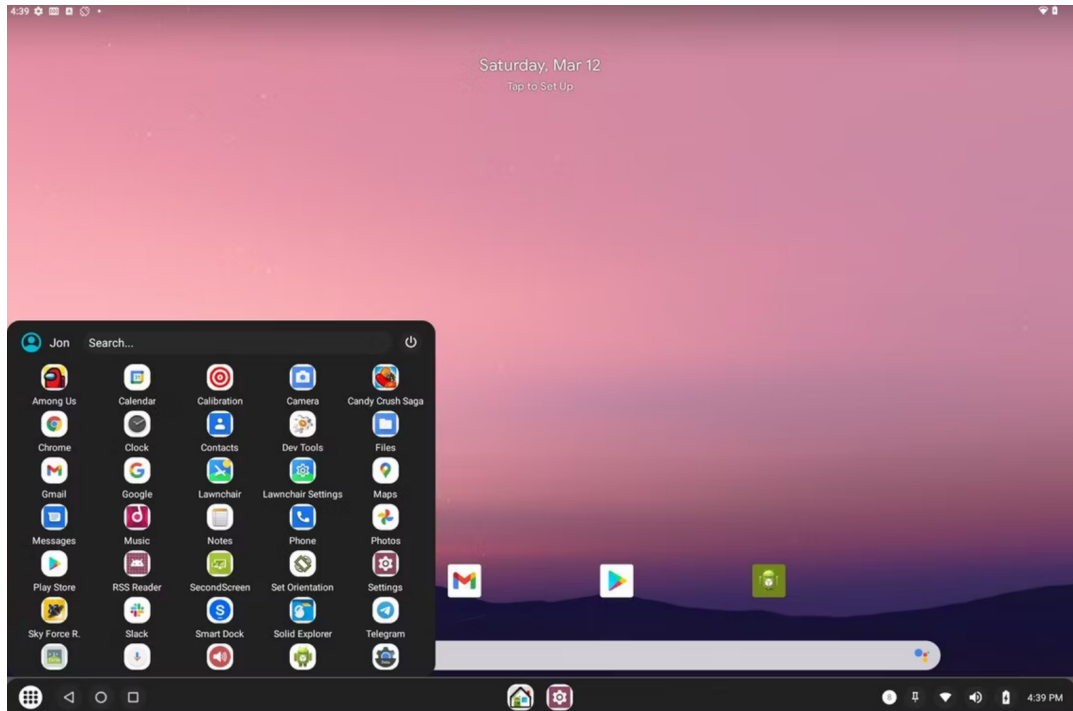
```
docker run -it \
  --privileged \
  --device /dev/kvm \
  -v /tmp/.X11-unix:/tmp/.X11-unix \
  -e "DISPLAY=\${DISPLAY:-:0.0}" \
  -p 5555:5555 \
  -p 50922:10022 \
```

```

--device /dev/video0 \
-e EXTRA='-device usb-host,hostbus=3,hostaddr=3' \
-e RAM=4 \
--device /dev/snd \
sickcodes/dock-droid:latest

```

Listing 4.3: Bliss OS - Run Docker Container (Δείτε Γ.4.1)



Σχήμα 4.1: DATE-A-A-S-blissOs-arch

Παραμετροποίηση

- Αύξηση μνήμης RAM στη γραμμή: `-e RAM=4`
- Ρύθμιση κάμερας και ήχου
 1. `v4l2-ctl --list-devices`
 2. `lsusb` για να βρεθεί το `hostbus` και το `hostaddr`
 3. Προσθήκη γραμμής `-device usb-host,hostbus=3,hostaddr=3`

4.2 WireGuard

4.2.1 base run

```

version: "2.1"
services:
  heimdall:
    image: linuxserver/heimdall
    container_name: wireguard
    volumes:
      #- /home/[user]/appdata/client:/config
      - /home/myuser/appdata/client:/config

```



```
environment:
  - PUID=1000
  - PGID=1000
  - TZ=Europe/Athens
  #- PEERS=5
ports:
  - 80:80
  - 443:443
restart: unless-stopped
```

Listing 4.4: WireGuard - Run Docker Container (Δείτε Γ.1.4)

Parameters

- **-p 51820/udp** wireguard port
- **-e PUID=1000** for UserID - see below for explanation
- **-e PGID=1000** for GroupID - see below for explanation
- **-e TZ=Etc/UTC** specify a timezone to use, see this list.
- **-e SERVERURL=myhost.testdoamin.com** External IP or domain name for docker host. Used in server mode. If set to auto, the container will try to determine and set the external IP automatically
- **-e SERVERPORT=51820** External port for docker host. Used in server mode.
- **-e PEERS=1** Number of peers to create confs for. Required for server mode. Can also be a list of names: myPC,myPhone,myTablet (alphanumeric only)
- **-e PEERDNS=auto** DNS server set in peer/client configs (can be set as 8.8.8.8). Used in server mode. Defaults to auto, which uses wireguard docker host's DNS via included CoreDNS forward.
- **-e INTERNAL_SUBNET=10.13.13.0** Internal subnet for the wireguard and server and peers (only change if it clashes). Used in server mode.
- **-e ALLOWEDIPS=0.0.0.0/0** The IPs/Ranges that the peers will be able to reach using the VPN connection. If not specified the default value is: '0.0.0.0/0, ::0/0' This will cause ALL traffic to route through the VPN, if you want split tunneling, set this to only the IPs you would like to use the tunnel AND the ip of the server's WG ip, such as 10.13.13.1.
- **-e PERSISTENTKEEPALIVE_PEERS=** Set to all or a list of comma separated peers (ie. 1,4,laptop) for the wireguard server to send keepalive packets to listed peers every 25 seconds. Useful if server is accessed via domain name and has dynamic IP. Used only in server mode.
- **-e LOG_CONFS=true** Generated QR codes will be displayed in the docker log. Set to false to skip log output.
- **-v /config** Contains all relevant configuration files.
- **-v /lib/modules** Host kernel modules for situations where they're not already loaded.
- **-sysctl=** Required for client mode.

4.2.2 Server Mode

Εάν η μεταβλητή περιβάλλοντος "PEERS" έχει οριστεί σε έναν αριθμό ή μια λίστα από συμβολοσειρές που χωρίζονται με κόμμα, το container θα λειτουργεί σε Server Mode και θα δημιουργηθούν οι απαραίτητες ρυθμίσεις για τον διακομιστή και τους χρήστες/πελάτες. Οι κωδικοί QR για τις ρυθμίσεις των χρηστών/πελατών θα αποθηκευτούν στα logs του Docker (και σε μορφή κειμένου και png κάτω από τον φάκελο /config/peerX) εάν η μεταβλητή "LOG_CONFS" έχει τιμή "true".

Οι μεταβλητές **SERVERURL**, **SERVERPORT**, **INTERNAL_SUBNET**, **PEERDNS**, **INTERFACE**, **ALLOWEDIPS** και **PERSISTENTKEEPALIVE_PEERS** είναι προαιρετικές μεταβλητές που χρησιμοποιούνται για τη λειτουργία διακομιστή. Οποιαδήποτε αλλαγή σε αυτές τις μεταβλητές περιβάλλοντος θα ενεργοποιήσει την δημιουργία των ρυθμίσεων του διακομιστή και των πελατών. Οι ρυθμίσεις των πελατών θα αναδημιουργηθούν με τα υπάρχοντα κλειδιά.

Για να εμφανίσουμε τους κωδικούς QR των ενεργών πελατών, χρησιμοποιούμε:

```
docker exec -it wireguard /app/show-peer ή
docker exec -it wireguard /app/show-peer 0 4 5 ή
docker exec -it wireguard /app/show-peer client1 client2
```

Οι ρυθμίσεις που χρησιμοποιούνται για τον διακομιστή και τους πελάτες αποθηκεύονται στον φάκελο /config/templates.

Για να εκκινήσουμε την υπηρεσία χρησιμοποιούμε:

```
docker compose up
docker compose down
```

Listing 4.5: WireGuard - Run Docker Container "Server Mode" (Δείτε Γ.2.1)

4.2.3 Client Mode

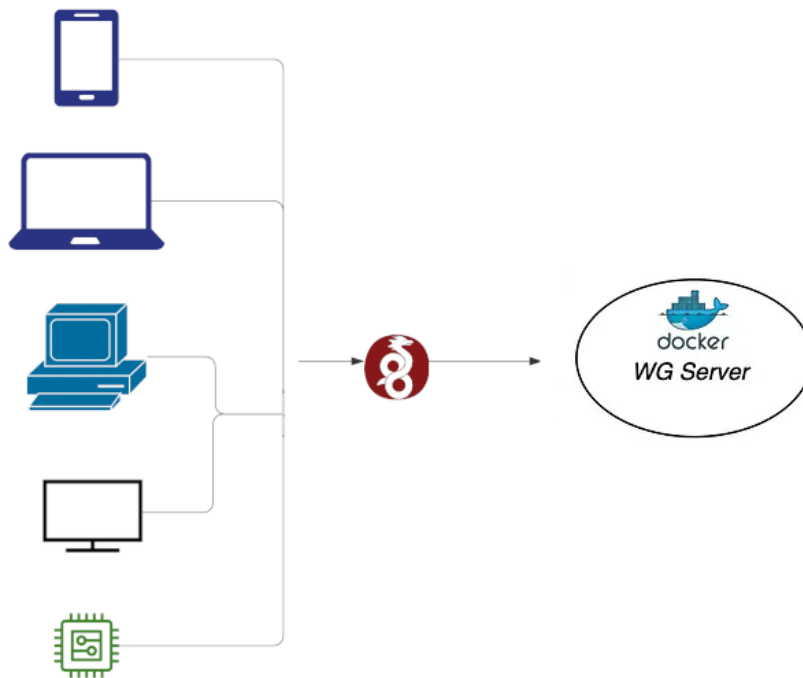
- Δεν ορίζουμε τη μεταβλητή περιβάλλοντος **PEERS**.
- Αποθηκεύουμε τις ρυθμίσεις του πελάτη στον φάκελο config ως /config/wg_confs/<όνομα τούνελ>.conf.

Για να εκκινήσουμε την υπηρεσία χρησιμοποιούμε:

```
docker compose up
docker compose down
```

Listing 4.6: WireGuard - Run Docker Container "Client Mode" (Δείτε Γ.2.1)

¹Android: WireGuard client από Google Play Store(<https://play.google.com/store/apps/details?id=com.wireguard.android>)

Σχήμα 4.2: DATE-A-A-S-WG-arch¹

4.3 kernelSU

Σημαντική Προειδοποίηση για τη Διαδικασία Flashing

Οι παρακάτω οδηγίες μπορεί να οδηγήσουν σε απώλεια δεδομένων και πρέπει να χρησιμοποιηθούν μόνο σε συσκευές που ΔΕΝ περιέχουν σημαντικά δεδομένα ή ΔΕΝ χρειάζονται για κανονική, καθημερινή χρήση. Σε περίπτωση που η συσκευή περιέχει σημαντικά δεδομένα, ΜΗΝ τη χρησιμοποιήσετε, προκειμένου να αποφύγετε απώλεια πληροφοριών.

Η διαδικασία Flashing προορίζεται για δοκιμαστικούς σκοπούς και όχι για κανονική παραγωγή ή καθημερινή χρήση.

Για την εγκατάσταση θα χρειαστούμε:

- το αρχείο boot.img για τη συσκευή ²
- το αρχείο zip AnyKernel3 που παρέχεται από τον KernelSU και ταιριάζει με την έκδοση KMI της συσκευής
- Από το πακέτο AnyKernel3 χρειαζόμαστε το αρχείο Image³
- το τελευταίο Magisk από τη σελίδα⁴
- Μετονομάζουμε το Magisk-*.apk σε Magisk-vesion.zip και το αποσυμπιέζουμε.

Στη συνέχεια γράφουμε το αρχείο libmagiskboot.so από το Magisk-v25.2/lib/arm64-v8a/ στη συσκευή μέσω adb:

²<https://github.com/ssut/payload-dumper-go>

³<https://github.com/tiann/KernelSU/releases>

⁴<https://github.com/topjohnwu/Magisk/releases>

```
adb push Magisk-v25.2/lib/arm64-v8a/libmagiskboot.so /data/local/tmp/
magiskboot

# Γράφουμε το stock boot.img και το αρχείο Image από το AnyKernel3 στη
  συσκευή.
# Εναρξη shell μέσω adb και εκτελούμαι
cd /data/local/tmp/
chmod +x magiskboot
cd /data/local/tmp/

# αποσυμπίεση του boot.img
./magiskboot unpack boot.img

# Αντιγράφουμε το Image στον πυρήνα
mv -f Image kernel

# Δημιουργία του boot.img (new-boot.img) και εγγραφή στη συσκευή με το
  fastboot.
./magiskboot repack boot.img

# reboot bootloader για να έχουμε το fastboot mode,
adb reboot bootloader
fastboot flash boot boot.img
```

Listing 4.7: KernelSU - Install

4.4 LineageOS

4.4.1 Ενεργοποίηση του "USB debugging"

- Άνοιγμα του μενού "Settings" και επιλογή του "About".
- Επτά φορές κλικ στο "Build number".
- Επιστροφή στο αρχικό μενού και επιλογή του "Developer options".
- Ενεργοποίηση της επιλογής "Android debugging"
- Σύνδεση της συσκευής στον υπολογιστή.
- Εκτέλεση της εντολής **adb devices**.
- Θα εμφανιστεί ένα παράθυρο διαλόγου στη συσκευή, ζητώντας άδεια για "usb debugging". Επιλογή του "always allow".

4.4.2 Ενεργοποίηση "Developer options"



Σχήμα 4.3: Download mode

1. Σύνδεση της συσκευής σε ασύρματο δίκτυο.
2. Άνοιγμα του μενού "Settings" και επιλογή του "About".
3. Επτά φορές κλικ στο "Build number".
4. Επιστροφή στο αρχικό μενού και επιλογή του "Developer options".
5. Ενεργοποίηση της επιλογής "OEM unlock".
6. Επανεκκίνηση σε λειτουργία λήψης:
 - Με τη συσκευή απενεργοποιημένη, πάτημα ταυτόχρονα των πλήκτρων "Μείωση της έντασης Ήχου" και του Bixby και ταυτόχρονη σύνδεση στο USB.
 - Εκτέλεση των ενεργειών που εμφανίζονται στην οθόνη.
7. Η συσκευή θα επανεκκινήσει. Αποσύνδεση από το USB καλώδιο.
8. Η συσκευή θα ζητήσει την διαμόρφωση των δεδομένων χρήστη.
9. Ολοκλήρωση της εγκατάστασης του Android, αγνοώντας ό,τι είναι δυνατόν.
10. Ενεργοποίηση του "Developer options", πατώντας αυτλη την φορά 10 φορές την επιλογή "Build Number".

4.4.3 Installing Add-Ons

- Click Apply Update,
- Apply from ADB
- adb -d sideload filename.zip for all desired packages in sequence.

4.4.4 Heimdall suite

- Εκτέλεση των 1,2,3, 4 και 5 από το βήμα 2
- Εγκατάσταση του "Heimdall suite"⁵ με την εντολή: **heimdall print-pit**

⁵<https://androidfilehost.com/?fid=4349826312261708650>

4.4.5 Lineage Recovery

- Κατέβασμα του Lineage Recovery⁶ recovery.img.
- Εκτέλεση των 1,2,3, 4 και 5 από το βήμα 2
- Εκτέλεση της εντολής: **heimdall flash -RECOVERY recovery.img -no-reboot**
- Αφαίρεση του USB και επανακίνηση

4.4.6 LineageOS

- Σύνδεση της συσκευής σε ασύρματο δίκτυο.
- Κατέβασμα του LineageOS installation package⁷.
- Επανεκκίνηση σε λειτουργία ανάκτησης:
 - Επανεκκίνηση με πατημένα τα πλήκτρα "Volume Up + Bixby + Power" ενώ η συσκευή είναι συνδεδεμένη σε υπολογιστή.
- Επιλογή "Factory Reset" και "Format data / factory reset"
- Sideload το "LineageOS .zip" με επιλογή του "Apply Update" και στη συνέχεια το "Apply from ADB" και στη συνέχεια εκτέλεση της εντολής: **adb -d sideload filename.zip**.



Σχήμα 4.4: Recovery mode

4.5 Appium

- Ενεργοποίηση του Virtualization

⁶<https://download.lineageos.org/devices/beyond0lte>

⁷<https://download.lineageos.org/devices/beyond0lte>

```

- sudo apt install cpu-checker
  kvm-ok

```

Listing 4.8: Appium - Enable Virtualization

- Build Appium server (Δείτε Γ.1.5)

```

- docker build -t appiumserver .

```

Listing 4.9: Appium - Build docker image

- Run Appium Server

```

- docker run --privileged -d -p 6080:6080 -p 5554:5554 -p
  5555:5555 -p 4723:4723 -e DEVICE="Samsung Galaxy S6" -e
  APPIUM=true --name android-container appiumserver

```

Listing 4.10: Appium - Run Server

- Share Volume

```

- docker run --privileged -d -p 6080:6080 -p 4723:4723 -p
  5554:5554 -p 5555:5555 -v \${PWD}/example/sample_apk:/root/tmp
  -e DEVICE="Nexus 5" -e APPIUM=true -e CONNECT_TO_GRID=true
  -e APPIUM_HOST="127.0.0.1" -e APPIUM_PORT=4723 -e
  SELENIUM_HOST="172.17.0.1" -e SELENIUM_PORT=4444 --name
  android-container appiumserver

```

Listing 4.11: Appium - Share volume

4.6 Rest API

Λίστα API Classes

- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.LabApi
- DateAAS.StoreApi
- DateAAS.UserApi
- DateAAS.UserApi
- DateAAS.UserApi
- DateAAS.UserApi

- DateAAS.UserApi
- DateAAS.UserApi
- DateAAS.UserApi

Στη συνέχεια ακολουθούν Παραδείγματα Υλοποίησης

4.6.1 addLab

Προσθήκη Εργαστηρίου

```
var DateAAS = require('date_a_a_s');
var defaultClient = DateAAS.ApiClient.instance;

// Configure OAuth2 access token for authorization: lab_auth
var lab_auth = defaultClient.authentications['lab_auth'];
lab_auth.accessToken = 'YOUR ACCESS TOKEN';

var apiInstance = new DateAAS.LabApi();

var body = new DateAAS.Lab(); // Lab | Lab object that needs to be added
to the server

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully.');
  }
};
apiInstance.addLab(body, callback);
```

Listing 4.12: addLab

Παράμετροι

Όνομα	Τύπος	Περιγραφή
body	Lab	Αντικείμενο Εργαστηρίου (lab) (Περιγραφή: 3.3.2)

Πίνακας 4.1: Παράμετρος: lab

Τύπος Επιστροφής

- null (κενό σώμα απάντησης)

Εξουσιοδότηση (Περιγραφή: [3.3.3](#))

- **Απαιτείται:** lab_auth

Κεφαλίδες Αιτήματος HTTP

Κεφαλίδα	Τιμή
Content-Type	application/json
Accept	application/json

Πίνακας 4.2: Αίτημα HTTP - addLab

4.6.2 deleteLab

Διαγραφή Εργαστηρίου

```
var DateAAS = require('date_a_a_s');
var defaultClient = DateAAS.ApiClient.instance;

// Configure OAuth2 access token for authorization: lab_auth
var lab_auth = defaultClient.authentications['lab_auth'];
lab_auth.accessToken = 'YOUR ACCESS TOKEN';

var apiInstance = new DateAAS.LabApi();

var labId = 789; // Number | Lab id to delete

var opts = {
  'apiKey': "apiKey_example" // String |
};

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully.');
```

Listing 4.13: deleteLab

Παράμετροι

Όνομα	Τύπος	Περιγραφή	Σημειώσεις
labId	Number	ID Εργαστηρίου	
apiKey	String		προαιρετικό

Πίνακας 4.3: Αίτημα HTTP - deleteLab

Εξουσιοδότηση: [4.6.1](#)

Κεφαλίδες Αιτήματος HTTP: [4.2](#)

4.6.3 getLabById

Εύρεση Εργαστηρίου με labId

```
var DateAAS = require('date_a_a_s');
var defaultClient = DateAAS.ApiClient.instance;

// Configure API key authorization: api_key
var api_key = defaultClient.authentications['api_key'];
api_key.apiKey = 'YOUR API KEY';
// Uncomment the following line to set a prefix for the API key, e.g. "
// Token" (defaults to null)
//api_key.apiKeyPrefix = 'Token';

var apiInstance = new DateAAS.LabApi();

var labId = 789; // Number | ID of Lab to return

var callback = function(error, data, response) {
  if (error) {
```

```

    console.error(error);
  } else {
    console.log('API called successfully. Returned data: ' + data);
  }
};
apiInstance.getLabById(labId, callback);

```

Listing 4.14: getLabById

Όνομα	Τύπος	Περιγραφή
labId	Number	Id Εργαστηρίου (lab)

Πίνακας 4.4: Παράμετρος: get labId

Εξουσιοδότηση: [4.6.1](#)

Κεφαλίδες Αιτήματος HTTP: [4.2](#)

4.6.4 findLabsByTags

Εύρεση Εργαστηρίου με tags

```

var DateAAS = require('date_a_a_s');
var defaultClient = DateAAS.ApiClient.instance;

// Configure OAuth2 access token for authorization: lab_auth
var lab_auth = defaultClient.authentications['lab_auth'];
lab_auth.accessToken = 'YOUR ACCESS TOKEN';

var apiInstance = new DateAAS.LabApi();

var tags = ["tags_example"]; // [String] | Tags to filter by

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully. Returned data: ' + data);
  }
};
apiInstance.findLabsByTags(tags, callback);

```

Listing 4.15: findLabsByTags

Όνομα	Τύπος	Περιγραφή
tags	String	Tags

Πίνακας 4.5: Παράμετρος: tags

Εξουσιοδότηση: [4.6.1](#)

Κεφαλίδες Αιτήματος HTTP: [4.2](#)

4.6.5 createUser

Δημιουργία Χρήστη

```

var DateAAS = require('date_a_a_s');
var apiInstance = new DateAAS.UserApi();

```

```

var body = new DateAAS.User(); // User | Created user object

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully.');
```

Listing 4.16: createUser

Όνομα	Τύπος	Περιγραφή
body	User	Αντικείμενο Χρήστη (User object 3.7)

Πίνακας 4.6: Παράμετρος: User object

Εξουσιοδότηση: Αυτή η ενέργεια μπορεί να γίνει μόνο από τον Διαχειριστή.
Κεφαλίδες Αιτήματος HTTP: 4.2

4.6.6 createUsersWithArrayInput

Δημιουργία Χρηστών με Πίνακα

```

var DateAAS = require('date_a_a_s');
var apiInstance = new DateAAS.UserApi();
var body = [new DateAAS.User()]; // [User] | List of user object

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully.');
```

Listing 4.17: createUsersWithArrayInput

Όνομα	Τύπος	Περιγραφή
body	[User]	Array με Αντικείμενο Χρήστη (array of User object 3.7)

Πίνακας 4.7: Παράμετρος: array of User objects

Εξουσιοδότηση: Αυτή η ενέργεια μπορεί να γίνει μόνο από τον Διαχειριστή.
Κεφαλίδες Αιτήματος HTTP: 4.2

4.6.7 deleteUser

Διαγραφή Χρήστη

```

var DateAAS = require('date_a_a_s');
var apiInstance = new DateAAS.UserApi();
var username = "username_example"; // String | The name that needs to be
  deleted

var callback = function(error, data, response) {
```

```

if (error) {
  console.error(error);
} else {
  console.log('API called successfully.');
```

Listing 4.18: deleteUser

Όνομα	Τύπος	Περιγραφή
username	String	Username Χρήστη (User object 3.7)

Πίνακας 4.8: Παράμετρος: username

Εξουσιοδότηση: Αυτή η ενέργεια μπορεί να γίνει μόνο από τον Διαχειριστή.
Κεφαλίδες Αιτήματος HTTP: 4.2

4.6.8 loginUser

Σύνδεση Χρήστη

```

var DateAAS = require('date_a_a_s');
var apiInstance = new DateAAS.UserApi();
var username = "username_example"; // String | The user name for login
var password = "password_example"; // String | The password for login in
clear text

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
    console.log('API called successfully. Returned data: ' + data);
  }
};
apiInstance.loginUser(username, password, callback);
```

Listing 4.19: loginUser

Όνομα	Τύπος	Περιγραφή
username	String	Το όνομα χρήστη για τη σύνδεση
password	String	Ο κωδικός χρήστη για τη σύνδεση σε απλό κείμενο

Πίνακας 4.9: Παράμετρος: user login

Κεφαλίδες Αιτήματος HTTP: 4.2

4.6.9 logoutUser

Αποσύνδεση Χρήστη

```

var DateAAS = require('date_a_a_s');
var apiInstance = new DateAAS.UserApi();

var callback = function(error, data, response) {
  if (error) {
    console.error(error);
  } else {
```

```

    console.log('API called successfully.');
```

Listing 4.20: logoutUser

Κεφαλίδες Αιτήματος HTTP: 4.2

4.7 Docker Swarm

Αυτό το έργο απαιτεί τρεις Linux υπολογιστές που έχουν εγκατεστημένο το Docker και μπορούν να επικοινωνούν μέσω δικτύου.

4.7.1 Βήματα εγκατάστασης

1. **Εγκατάσταση Docker Engine σε Linux μηχανήματα**
Φυσικοί υπολογιστές ή εικονικοί υπολογιστές βασισμένοι σε Linux ως hosts.
2. **Η διεύθυνση IP του διαχειριστή**
Δεδομένου ότι άλλοι κόμβοι επικοινωνούν με τον κόμβο διαχειριστή μέσω της διεύθυνσης IP του, χρησιμοποιούμε μια σταθερή διεύθυνση IP.
3. **Άνοιγμα πρωτοκόλλων και θυρών μεταξύ των hosts**
Οι παρακάτω θύρες πρέπει να είναι διαθέσιμες. Σε ορισμένα συστήματα, αυτές οι θύρες είναι ανοιχτές από προεπιλογή.
 - Θύρα 2377 TCP για επικοινωνία με και μεταξύ των κόμβων διαχειριστών
 - Θύρα 7946 TCP/UDP για ανακάλυψη κόμβων overlay δικτύου
 - Θύρα 4789 UDP (ρυθμιζόμενη) για κίνηση overlay δικτύου

Η θύρα 4789 είναι η προεπιλεγμένη τιμή για τη θύρα διαδρομής δεδομένων του Swarm, γνωστή και ως θύρα VXLAN.

```
# Παράδειγμα κανόνα iptables
iptables -I INPUT -m udp --dport 4789 -m policy --dir in --pol none -j DROP
```

4.7.2 Create a swarm

```
docker swarm init --advertise-addr 195.130.109.103
```

Listing 4.21: Create a swarm

Η παραπάνω διαδικασία δημιουργεί και τα αντίστοιχα keys

Εύρεση του Manager key

```
docker swarm join-token manager
```

Listing 4.22: swarm: manager key

Εύρεση του Worker key

```
docker swarm join-token worker
```

Listing 4.23: swarm: worker key

4.7.3 Προσθήκη Worker στο Swarm

Για την συμμετοχή ως worker στο swarm:

```
docker swarm join \  
--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8  
vxv8rssmk743ojnwacrr2e7c \  
195.130.109.103:2377
```

Listing 4.24: swarm: dwarm join

Το key: *SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c* θα αντικατασταθεί από το αντίστοιχο που θα δημιουργηθεί κατά τη δημιουργία του εργαστηρίου. ο

Η συμμετοχή στο δίκτυο και στο swarm που έχει δημιουργηθεί είναι ελευθερη από οποιοδήποτε σημείο εντός και εκτός πανεπιστημίου

Κεφάλαιο 5

Proof of concept (PoC)

Ένα σημαντικό μέρος της παρούσας εργασίας αποτελεί το οπτικοακουστικό υλικό "poc-diplomatiki-android" και "poc-diplomatiki" σε μορφή MP4, το οποίο έχει ως σκοπό να περιγράψει αναλυτικά το proof of concept¹. Το συγκεκριμένο υλικό παρέχει μια λεπτομερή παρουσίαση του σεναρίου υλοποίησης, καταγράφοντας κάθε βήμα της διαδικασίας με σαφήνεια και ακρίβεια.

Η χρήση του οπτικοακουστικού υλικού κρίνεται απαραίτητη για την καλύτερη κατανόηση των τεχνικών και θεωρητικών πτυχών της εργασίας, επιτρέποντας στους αναγνώστες και αξιολογητές να αποκτήσουν μια πιο ολοκληρωμένη εικόνα της υλοποίησης. Μέσα από αυτό, δίνεται η δυνατότητα να παρακολουθήσει κανείς τη διαδικασία εφαρμογής και τις λειτουργικές λεπτομέρειες, όπως αυτές αναπτύσσονται βήμα προς βήμα.

Για την υποστήριξη και τη διευκόλυνση της κατανόησης του περιεχομένου, στο παρόν σημείο της εργασίας θα επισυνάψω επιλεκτικά ορισμένα στιγμιότυπα (screenshots) από τα βίντεο. Τα στιγμιότυπα αυτά έχουν επιλεγεί για να αποδώσουν τα κύρια σημεία και τα κρίσιμα βήματα της υλοποίησης, προσφέροντας μια σαφή και συνοπτική απεικόνιση των επιμέρους φάσεων του έργου.

5.1 mitmproxy config

```
1 #!/bin/bash
2
3 apt update && apt install -y procps
4 update-ca-certificates
5 /usr/local/bin/python /usr/local/bin/mitmweb --web-host 0.0.0.0 --set ssl insecure=true
```

Σχήμα 5.1: DATE-A-A-S-mitmproxy-config²

¹Δείτε επίσης: <https://codeberg.org/DATE-A-A-S/dateaas>

²00:54 poc-diplomatiki.mp4

³00:54 poc-diplomatiki.mp4

⁴02:26 poc-diplomatiki.mp4

```

1 #!/bin/bash
2
3 docker run --rm -it \
4   -p 8080:8080 \
5   -p 127.0.0.1:8081:8081 \
6   -v ./pem:/pem \
7   -v ./pem/mitmproxy.crt:/home/mitmproxy/.mitmproxy/mitmproxy.crt \
8   -v ./pem/mitmproxy/mitmproxy.crt:/usr/local/share/ca-certificates/mitmproxy.crt \
9   -v ./run1.sh:/run1.sh \
10  --name mitmproxy \
11  --entrypoint /run1.sh \
12  mitmproxy/mitmproxy

```

Σχήμα 5.2: DATE-A-A-S-mitmproxy-run³

Path	Method	Status	Size	Time	Request	Response	Connection	Timing
https://www.google.com/gen_204?atyp=i&bb=1&ei=O...	POST	204	0	190ms	POST https://play.google.com/log?hasfast=true&authuser=0&format=json HTTP/2.0			
https://www.google.com/gen_204?atyp=i&bb=1&ei=O...	POST	204	0	129ms		user-agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0		
https://dooh.waterfox.net/	POST	200	915b	110ms		accept: /*/*		
https://dooh.waterfox.net/	POST	200	915b	115ms		accept-language: en-US,en;q=0.5		
https://services.addons.mozilla.org/api/v4/addons/search...	GET	200	82b	270ms		accept-encoding: gzip, deflate, br		
https://firefox.settings.services.mozilla.com/v1/buckets/...	GET	200	54b	25ms		referrer: https://accounts.google.com/		
https://dooh.waterfox.net/	POST	200	915b	165ms		content-type: text/plain;charset=UTF-8		
https://aus1.waterfox.net/update/SystemAddons/G6.0.7/...	GET	404	40b	193ms		content-length: 538		
https://push.services.mozilla.com/	WSS	101	277b	398ms		origin: https://accounts.google.com		
https://dooh.waterfox.net/	POST	200	915b	85ms		sec-gpc: 1		
https://dooh.waterfox.net/	POST	200	915b	85ms		cookie: SOCS=CAESHAgBEhJnd3NfmjAyMjA5MjktMF9SdzEAAAJVIAEaBgIAkvoZBg		
https://firefox.settings.services.mozilla.com/v1/buckets/...	GET	200	588b	107ms		cookie: AEC=Ae3NU9099VXss4wPqwA6oTQqwmnQjK6i-XTGdZra187dBLkSx2T-BnBYQ		
https://www.google.com/gen_204?atyp=i&bb=1&ei=O...	POST	204	0	91ms		cookie: __Secure-		
https://www.google.com/gen_204?atyp=i&bb=1&ei=O...	POST	204	0	170ms		ENID=17.SE=sHpRoSj1DgIJeSfAmU57z_y9VAq1pwGgobgHUMYPnH9UctTiZuXHIHafbaP0j4GvezgQYwRa-		

Σχήμα 5.3: DATE-A-A-S-mitmproxy-capture-webserver⁴

5.2 mitmproxy capture

⁵03:06 poc-diplomatiki.mp4

⁶05:15 poc-diplomatiki.mp4

⁷06:40 poc-diplomatiki.mp4

Waterfox | about:certificate?cert=MIIDjCCAgqgAwIBAgIU BZRkVfsvh2wJvAt2PO9dxKOGfAlwDQYJKoZIhvcNAQELBQAwKDESMBAGAlUEAwWJbWl0bXBByb

Certificate

www.google.com		mitmproxy
Subject Name		
Common Name	www.google.com	
Issuer Name		
Common Name	mitmproxy	
Organization	mitmproxy	
Validity		
Not Before	Mon, 15 Jan 2024 17:49:10 GMT	
Not After	Thu, 16 Jan 2025 17:49:10 GMT	
Subject Alt Names		
DNS Name	www.google.com	
Public Key Info		
Algorithm	RSA	
Key Size	2048	

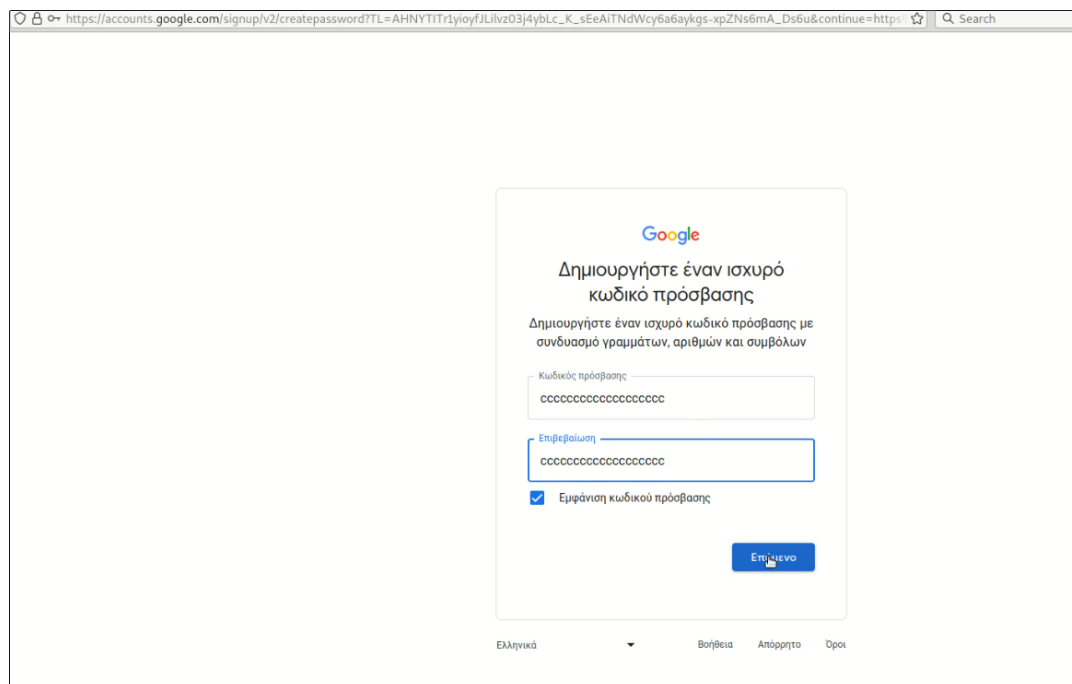
Σχήμα 5.4: DATE-A-A-S-mitmproxy-key-google⁵

5.3 mitmproxy android

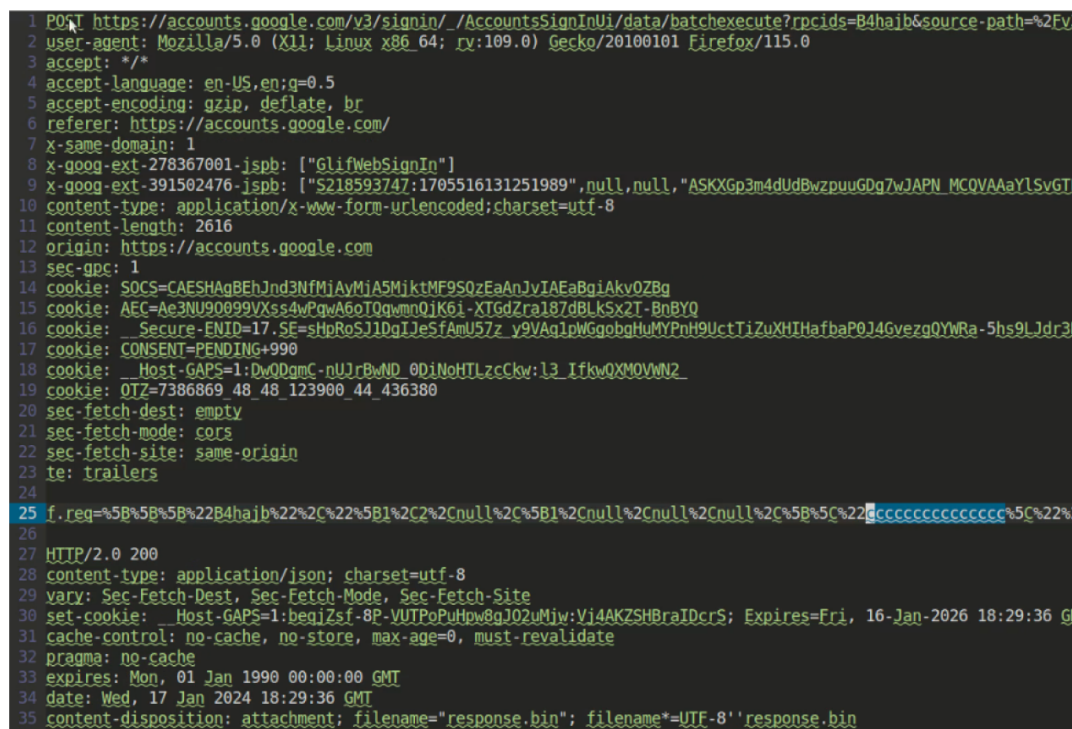
⁸01:09 poc-diplomatiki-android.mp4

⁹03:10 poc-diplomatiki-android.mp4

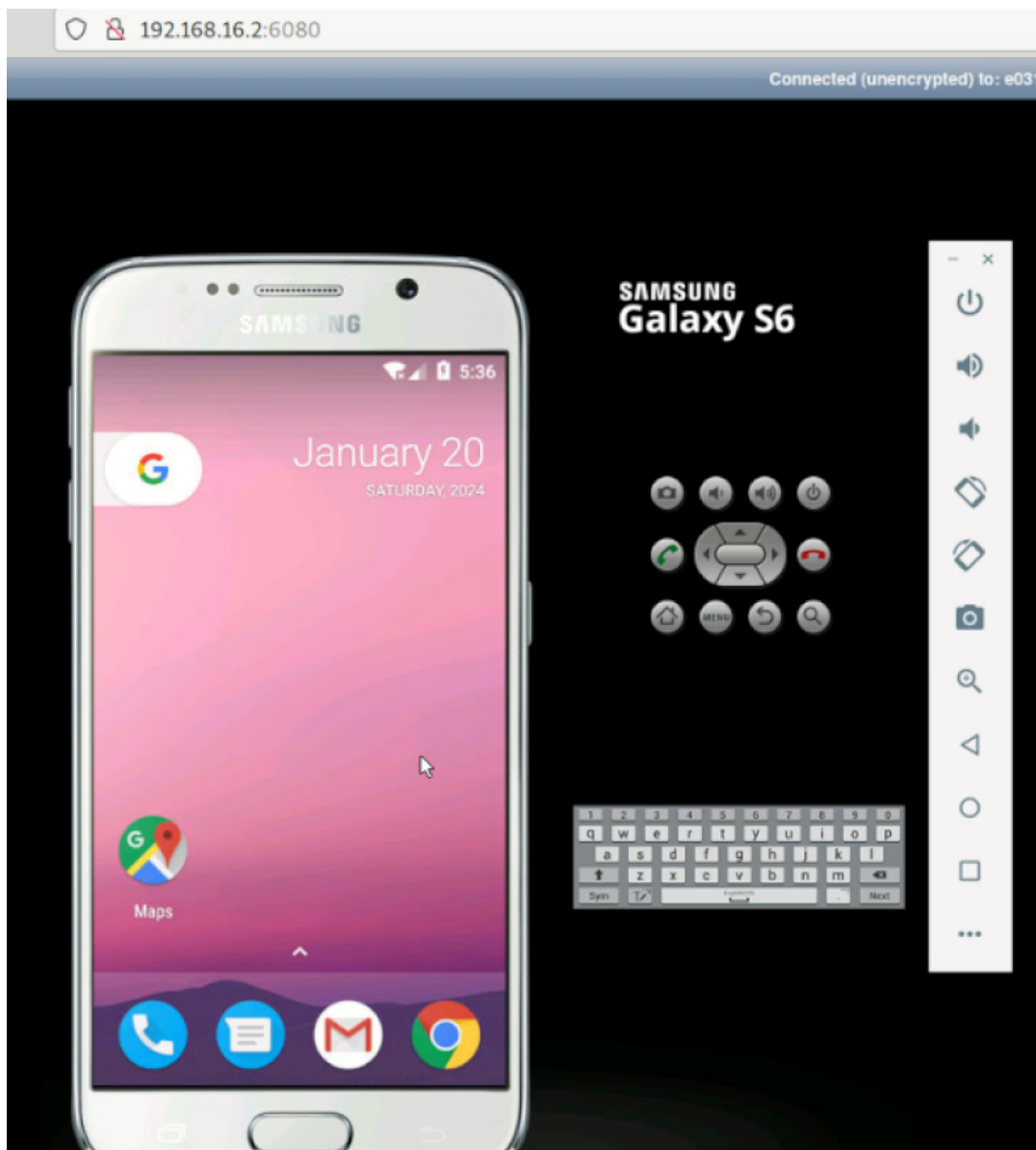
¹⁰03:59 poc-diplomatiki-android.mp4



Σχήμα 5.5: DATE-A-A-S-mitmproxy-capture-google-login⁶



Σχήμα 5.6: DATE-A-A-S-mitmproxy-capture-serverlog⁷



Σχήμα 5.9: DATE-A-A-S-android-run-installapp1¹⁰

Παράρτημα Α΄

Δεδομένα και ψηφιακή οικονομία

”Στην ψηφιακή οικονομία παρατηρείται μια τάση προς την εταιρική συγκέντρωση των προσωπικών δεδομένων. Αυτό ισχύει ιδιαίτερα για τις ψηφιακές πλατφόρμες, όπως τα μέσα κοινωνικής δικτύωσης πλατφόρμες, οι μηχανές αναζήτησης, η ψηφιακή ψυχαγωγία ή οι διαδικτυακοί έμποροι λιανικής πώλησης. Ο τρόπος με τον οποίο με τον οποίο μετράται παραδοσιακά η κυριαρχία στην αγορά δεν καταγράφει πάντα την την έκταση της ισχύος των εν λόγω εταιρειών στην αγορά, καθώς τα προϊόντα και οι υπηρεσίες τους είναι συχνά ”δωρεάν” για τους καταναλωτές. Η τάση αυτή τροφοδοτείται από την αυξανόμενη εξάρτηση πολλών τομέων της οικονομίας από τα δεδομένα, ιδίως τα προσωπικά δεδομένα. Η πρόσβαση στα προσωπικά δεδομένα γίνεται αντιληπτή ως ολοένα και πιο πολύτιμη ικανότητα στην ψηφιακή οικονομία και η απόκτηση των τεράστιες κλίμακες είναι αυτό που επιτρέπει στις μεγάλες εταιρείες τεχνολογίας να κερδίζουν δισεκατομμύρια δολάρια κάθε ετησίως μέσω της στοχευμένης διαφήμισης. Το 2020, για παράδειγμα, η μητρική εταιρεία της Google, Alphabet, απέκτησε πάνω από το 80% των εσόδων της ύψους 182,5 δισεκατομμυρίων δολαρίων από την παροχή διαφημίσεις στους χρήστες των πολλών υπηρεσιών της που απευθύνονται στους χρήστες, οι οποίες, μεταξύ άλλων μεταξύ άλλων, περιλαμβάνουν το λειτουργικό σύστημα Android, το Google Search, το YouTube και Gmail. Ομοίως, τα έσοδα του Facebook το 2020 ήταν 85,97 δισεκατομμύρια δολάρια με διαφημίσεις τα έσοδα από τη διαφήμιση να αντιπροσωπεύουν σχεδόν το 98% αυτών (84,17 δις. δολάρια). Η αξία της πρόσβασης στα προσωπικά δεδομένα αυξάνεται καθώς όλο και περισσότερα δεδομένα γίνονται εκμεταλλεύσιμα μέσω της απόκτησης, του συνδυασμού και της πρόσθετης επεξεργασίας και αυτό δίνει κίνητρο στις εταιρείες να ακολουθήσουν επιχειρηματικές στρατηγικές που αποσκοπούν στη συλλογή όσο το δυνατόν περισσότερα δεδομένα.

Με την ανάπτυξη και την ενσωμάτωση των τεχνητών τεχνολογιών τεχνητής νοημοσύνης (AI), τα δεδομένα των χρηστών έχουν γίνει ακόμη πιο σημαντικά για τις εταιρείες αυτές, δεδομένου ότι αποτελούν βασική εισροή για την εκπαίδευση των μοντέλων TN, ιδίως εκείνων που κατηγοριοποιούν τους ανθρώπους με βάση τα χαρακτηριστικά τους και προφίλ. Και δεδομένης της αυξανόμενης σημασίας των προσωπικών δεδομένων σε όλους τους τομείς της οικονομίας, οι συγκεντρώσεις δεδομένων είναι πιθανό να συνεχιστούν και να επεκταθούν σε άλλες αγορές.

Καθώς οι τεράστιες συγκεντρώσεις εξουσίας που προκύπτουν από την αξία των προσωπικών δεδομένων στην ψηφιακής οικονομίας, είναι εξαιρετικά σημαντικό τα δεδομένα των εταιρειών να να βρίσκονται στο επίκεντρο των ανταγωνιστικών ελέγχων των ρυθμιστικών αρχών αντιμονοπωλιακής νομοθεσίας. Πράγματι, η σημασία της κατοχής δεδομένων αναγνωρίζεται πολύ καλά από τις ψηφιακές πλατφόρμες που θεωρούν σταθερά τα δεδομένα των καταναλωτών ως επιχειρηματικό περιουσιακό στοιχείο. Τα προσωπικά δεδομένα είναι επίσης αναπόσπαστο μέρος των επιχειρηματικών μοντέλων αυτών των εταιρειών και, ως εκ τούτου, της

αγοραίας αξίας τους. Κατά την αξιολόγηση της ισχύος στην αγορά, οι αρχές ανταγωνισμού έτειναν, στο παρελθόν, να επικεντρώνονται στις τιμές και τις εκροές, δίνοντας ελάχιστη έως καθόλου σημασία στο ρόλο των προσωπικών δεδομένων στις διάφορες σχετικές αγορές (π.χ. διαφήμιση, μέσα κοινωνικής δικτύωσης, μηχανές αναζήτησης, διαδικτυακή ψυχαγωγία κ.λπ.) Αυτή η στενή προσέγγιση έχασε την όλο και πιο σημαντικές επιπτώσεις της επεξεργασίας των προσωπικών δεδομένων στον ανταγωνισμό. δεδομένων, ιδίως όταν γίνεται σε κλίμακα. Με τη σειρά της δεν έλαβε υπόψη της τις πολλαπλές επιπτώσεις που έχει η απόκτηση προσωπικών δεδομένων σε ορισμένους τύπους ψηφιακών υπηρεσιών, όπως περιγράφεται ανωτέρω. Οι δικτυακές επιδράσεις στις επιγραμμικές αγορές μπορούν να αυξήσουν το σημασία της απόκτησης ή της απώλειας ενός χρήστη λόγω της σημασίας των προσωπικών δεδομένων (σε κλίμακα) για τη λειτουργία ορισμένων αλγορίθμων, όπως εκείνων που στηρίζουν την αποτελεσματικότητα της στοχευμένης διαφήμισης.

Τα τελευταία χρόνια οι ρυθμιστικές αρχές ανταγωνισμού αναγνώρισαν την ανάγκη να εξετάσουν το ρόλο των προσωπικών δεδομένων στις αξιολογήσεις του ανταγωνισμού, ιδίως στις ψηφιακές αγορές. Ορισμένες αρχές ανταγωνισμού έχουν προχωρήσει ένα βήμα παραπέρα, ώστε να αντιμετωπίσουν τις επιπτώσεις στην ιδιωτική ζωή και την προστασία των δεδομένων. Για παράδειγμα, τον Φεβρουάριο του 2019 η γερμανική αρχή ανταγωνισμού σημείωσε ότι "η παρακολούθηση της επεξεργασίας δεδομένων δραστηριοτήτων των δεσπόζουσων επιχειρήσεων αποτελεί, επομένως, βασικό καθήκον μιας αρχής ανταγωνισμού αρχής ανταγωνισμού, το οποίο δεν μπορούν να εκπληρώσουν οι υπεύθυνοι προστασίας δεδομένων". Σε παρόμοιο πνεύμα, οι ρυθμιστικές αρχές δεδομένων έχουν προειδοποιήσει για τις απειλές κατά της ιδιωτικής ζωής που θέτει η αυξημένη αγορά συγκέντρωσης της αγοράς και έχουν υποστηρίξει την ανάγκη ενσωμάτωσης της προστασίας των δεδομένων στην αξιολόγηση των πιθανών καταχρήσεων δεσπόζουσας θέσης και των συγχωνεύσεων των εταιρειών που δραστηριοποιούνται στην ψηφιακή αγορά." ¹.

¹Πηγή: <https://privacyinternational.org/sites/default/files/2022-05/Personal%20Data%20and%20Competition%20May%202022%20EN.pdf>

Παράρτημα Β΄

Εφαρμογές περιόδου και προσωπικά δεδομένα

Τον Δεκέμβριο του 2018, το Privacy International¹ διεξήγαγε έρευνα σε 36 από τις πιο δημοφιλείς εφαρμογές στον κόσμο(privacyinternational.org, 2019a) Από τις 36 εφαρμογές που δοκίμασε κατά τη διάρκεια της έρευνας (Μεθοδολογία έρευνας για Δυναμική Ανάλυση: (privacyinternational.org, 2019b)), διαπιστώθηκε ότι το 61% μεταφέρει αυτόματα δεδομένα στο Facebook από τη στιγμή που ένας χρήστης ανοίγει την εφαρμογή. Αυτό συμβαίνει ανεξάρτητα από το εάν ο χρήστης έχει λογαριασμό στο Facebook ή όχι και αν είναι συνδεδεμένος στο Facebook ή όχι. Επίσης, διαπιστώθηκε ότι ορισμένες από αυτές τις εφαρμογές συστηματικά αποστέλλουν στο Facebook ακριβή και, σε ορισμένες περιπτώσεις, ευαίσθητα προσωπικά δεδομένα².

Η έρευνα υπογραμμίζει ότι οι εφαρμογές περιόδου, που αποτελούσε μια ομάδα εφαρμογών της έρευνας(privacyinternational.org, 2019g),(privacyinternational.org, 2019c), (privacyinternational.org, 2019d), (privacyinternational.org, 2019f), (privacyinternational.org, 2019h), (privacyinternational.org, 2019e), προκαλούν σοβαρές ανησυχίες όσον αφορά τη συμμόρφωσή τους με τις υποχρεώσεις τους σύμφωνα με τον Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR), ειδικά όσον αφορά τη συγκατάθεση και τη διαφάνεια³.

¹<https://privacyinternational.org/>

²Δεν είχε σημασία εάν οι χρήστες είχαν αποσυνδεθεί από το Facebook ή δεν είχαν λογαριασμό.

³Διαπιστώθηκε ότι μοιράζονται τα δεδομένα των χρηστών με τρίτους, συμπεριλαμβανομένων διαφημιστών και εταιρειών ανάλυσης. Οι πληροφορίες περιλαμβάνουν προσωπικές λεπτομέρειες σχετικά με την κυκλοφορία των χρηστών, τις διακυμάνσεις της διάθεσης και άλλα προσωπικά δεδομένα υγείας.

Β.1 Ευρήματα

Β.1.1 Maya

Η εφαρμογή επεξεργάζεται σημαντικές ποσότητες προσωπικών δεδομένων⁴, συμπεριλαμβανομένων δεδομένων υγείας που μπορούν να θεωρηθούν ως ευαίσθητα δεδομένα σύμφωνα με τους νόμους προστασίας δεδομένων της ΕΕ.

Παρά το γεγονός ότι η εταιρεία εδρεύει στην Ινδία, η εφαρμογή εξυπηρετεί χρήστες στην ΕΕ μέσω του Google Play Store UK (αλλά ακόμη και μετά το Brexit η εφαρμογή είναι διαθέσιμη ακόμη και σήμερα στην Ε.Ε., όπως διαπιστώνουμε εδώ: https://play.google.com/store/apps/developer?id=Plackal+Tech&hl=en_US&gl=US). Η έλλειψη διαφάνειας σχετικά με τους αποδέκτες των δεδομένων των χρηστών και οι πρακτικές διαμοιρασμού δεδομένων με διαφημιστές προκαλούν ανησυχία⁵. Ταυτόχρονα, οι πρακτικές αυτές παραβιάζουν τους κανονισμούς προστασίας δεδομένων της ΕΕ, ιδίως όσον αφορά την απαγόρευση επεξεργασίας ευαίσθητων κατηγοριών δεδομένων χωρίς τη ρητή συναίνεση του χρήστη.

Η έλλειψη πληροφοριών σχετικά με τα δικαιώματα των χρηστών της ΕΕ προσθέτει ακόμη ένα στοιχείο σχετικά με την προστασία των προσωπικών δεδομένων. Συγκεκριμένα, η πολιτική απορρήτου της εφαρμογής δεν παρέχει επαρκείς πληροφορίες σχετικά με το δικαίωμα των χρηστών να διορθώσουν τα προσωπικά τους δεδομένα ή να υποβάλλουν παράπονο στην αρμόδια αρχή. Αυτό οδηγεί σε ανησυχίες σχετικά με την ενημέρωση και την επικοινωνία με τους χρήστες της ΕΕ σχετικά με τα δικαιώματα που τους παρέχονται στο πλαίσιο των νομοθεσιών προστασίας δεδομένων.

Ο ευρωπαϊκός νόμος περί προστασίας δεδομένων απαγορεύει την επεξεργασία ειδικών κατηγοριών δεδομένων, εκτός από συγκεκριμένες περιπτώσεις, όπως με την ρητή συναίνεση του χρήστη (Δείτε παράρτημα: ??). Σε αυτήν την περίπτωση, είναι αμφίβολο εάν η Maya θα μπορούσε να ισχυριστεί ότι έχει λάβει την ενημερωμένη και ρητή συναίνεση των χρηστών για την κοινοποίηση δεδομένων, δεδομένου ότι τα προσωπικά δεδομένα κοινοποιούνται πριν οι χρήστες δουν, ακόμη και συμφωνήσουν, στην πολιτική απορρήτου. Με άλλα λόγια, είναι δύσκολο να δούμε πώς

⁴ Στην πολιτική απορρήτου της εφαρμογής (έως τις 19 Αυγούστου 2019), η Plackal Tech αναφέρει ρητά ότι η Maya συλλέγει πληροφορίες σχετικά με "σημειώσεις, συμπτώματα ή διαθέσεις", καθώς και "πληροφορίες που εισάγετε στην εφαρμογή, συμπεριλαμβανομένου του μήκους του ημικυκλικού σας κύκλου, και γενικές πληροφορίες σχετικά με την υγεία σας, όπως το βάρος, η διάθεση, η θερμοκρασία και / ή οποιαδήποτε σωματική οικειότητα". Η Plackal Tech αναφέρει επίσης ότι "μπορεί επίσης να συλλέγει την ακριβή τοποθεσία της συσκευής σας όταν η εφαρμογή λειτουργεί στο προσκήνιο ή στο παρασκήνιο". Μπορεί επίσης να παράγει "την περίπου τοποθεσία σας από τη διεύθυνση IP σας". Είναι αμφίβολο εάν αυτή η εκτεταμένη συλλογή δεδομένων είναι απαραίτητη αυστηρά για την παροχή της υπηρεσίας που ζητούν οι χρήστες και, αντίστοιχα, δημιουργεί μια σειρά ερωτημάτων σχετικά με τη συμβατότητα αυτών των εφαρμογών με τον ευρωπαϊκό νόμο περί προστασίας δεδομένων. Για παράδειγμα, η αρχή της ελαχιστοποίησης των δεδομένων απαιτεί από τους παροχούς των υπηρεσιών να επεξεργάζονται την ελάχιστη ποσότητα προσωπικών δεδομένων που είναι απαραίτητη για την παροχή της υπηρεσίας.

⁵ Ενώ η πολιτική απορρήτου της Maya αναφέρει ότι οι πληροφορίες μπορεί να αποκαλυφθούν σε τρίτους, δεν παρέχει ακριβείς πληροφορίες σχετικά με τις κατηγορίες προσωπικών δεδομένων των χρηστών που αποκαλύπτονται ή οποιεσδήποτε ακριβείς πληροφορίες σχετικά με τους τρίτους αυτούς.

Παρόλο που αναφέρει ότι δεν διαρρέει προσωπικά δεδομένα στους διαφημιστές, η πολιτική απορρήτου αναφέρει ότι τα προσωπικά δεδομένα των χρηστών μπορεί να χρησιμοποιηθούν "για να συμμορφωθούμε με τις επιθυμίες των διαφημιζόμενων μας προς εμφάνιση των διαφημίσεων τους σε αυτό το στόχο κοινό", χωρίς να διευκρινίζει εάν αυτό περιλαμβάνει επίσης δεδομένα που αφορούν την υγεία.

ένας μέσος χρήστης θα συμφωνούσε ακόμα και συμπεριφορικά να μοιράζεται τέτοιες προσωπικές λεπτομέρειες της υγείας και της σεξουαλικής του ζωής με το Facebook, καθώς αυτό υπερβαίνει αυτό που κανείς δεν θα περίμενε λογικά σε αυτό το πλαίσιο.

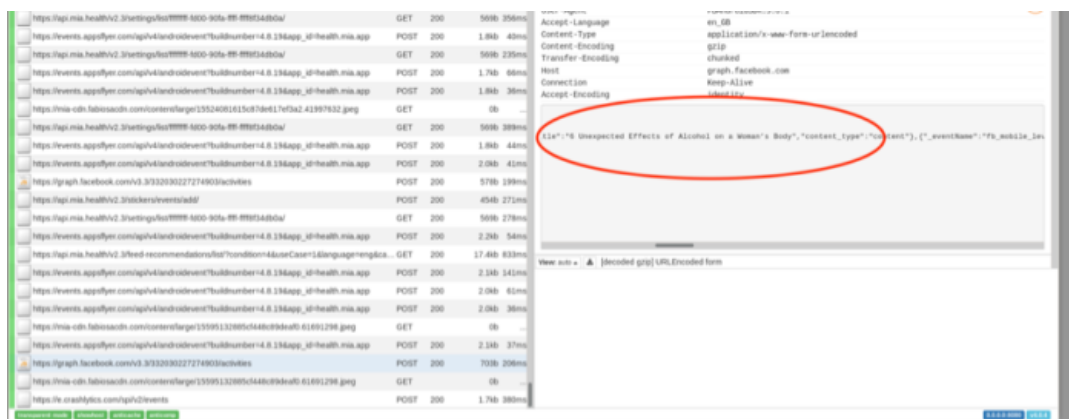
B.1.2 MIA

MIA από την Mobapp Development Limited

Στο σημείο της ερώτησης της συλλογής "ευαίσθητων δεδομένων" η πολιτική απορρήτου της MIA κατά την διάρκεια της έρευνας (Αυγούστου 2019), αναφέρει σαφώς ότι η εφαρμογή μπορεί να συλλέξει "ημερομηνίες κύκλου, συμπτώματα σχετικά με τον κύκλο, πληροφορίες σχετικά με την υγεία και τις δραστηριότητες (ύπνος, διάθεση, ασθένειες, σεξ, βήματα κ.λπ.), μετρήσεις σώματος, οι οποίες μπορεί να περιλαμβάνουν πληροφορίες σχετικά με προσωπικά θέματα υγείας που παρέχετε, συμπεριλαμβανομένων πληροφοριών σχετικά με την φυσική κατάστασή σας".

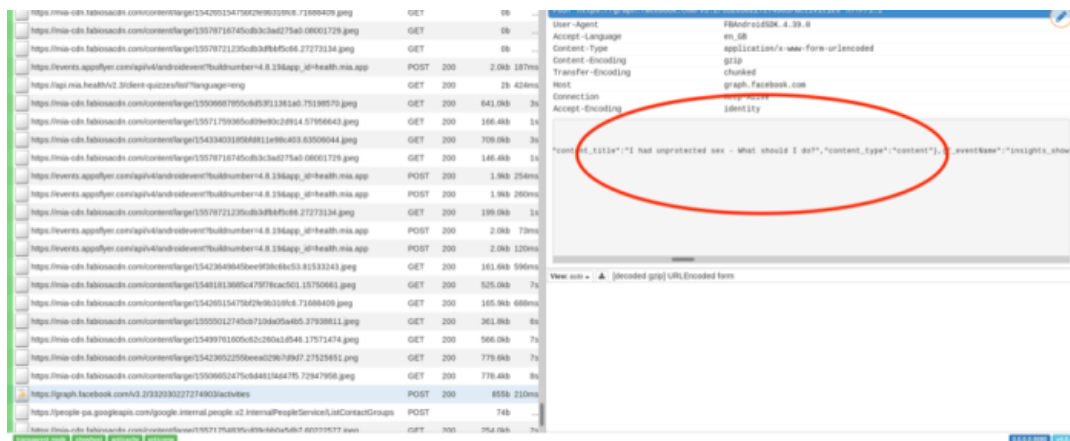
Επιπλέον, η MIA ανέφερε στην πολιτική απορρήτου της ότι μπορεί να χρησιμοποιήσει τα προσωπικά δεδομένα που συλλέγονται για αρκετούς σκοπούς, συμπεριλαμβανομένου του "εκπαίδευση αλγορίθμων μηχανικής μάθησης" και του "εκτέλεση ελέγχων φόντου σε χρήστες". Ωστόσο, η πολιτική απορρήτου δεν προσδιορίζει ποιες ακριβώς κατηγορίες προσωπικών δεδομένων μπορεί να χρησιμοποιηθούν για αυτούς τους σκοπούς και εάν αυτό περιλαμβάνει ευαίσθητα δεδομένα που αφορούν τη σεξουαλική υγεία. Αυτό εγείρει σοβαρές ανησυχίες σχετικά με τη διαφάνεια, καθώς οι χρήστες χρειάζεται να λαμβάνουν σημαντικές πληροφορίες σχετικά με τη χρήση αλγορίθμων από αυτές τις υπηρεσίες, και ιδιαίτερα πώς αυτή η χρήση μπορεί να τους επηρεάσει.

Το GDPR ισχύει για την MIA καθώς ο υπεύθυνος επεξεργασίας δεδομένων είναι εγκατεστημένος στην ΕΕ (Κύπρος) και η εφαρμογή είναι διαθέσιμη για λήψη στο Google Play Store. Αυτό σημαίνει ότι, εφόσον οι χρήστες της ΕΕ μπορούν να κατεβάσουν και να χρησιμοποιήσουν την εφαρμογή, η MIA χρειάζεται να τηρεί τις υποχρεώσεις του GDPR.



Σχήμα Β.1: Κοινή χρήση δεδομένων με το Facebook: κατανάλωση αλκοόλ⁶

⁶Πηγή: https://www.privacyinternational.org/sites/default/files/flysystem/styles/middle_column_small/local-default/2019-09/Picture%2020.png?itok=oE1pLcLN0



Σχήμα Β'.2: Κοινή χρήση δεδομένων με το Facebook: πληροφορίες για τη σεξουαλική ζωή⁷

Β.1.3 My Period Tracker

Το My Period Tracker από τη Linchpin Health είναι επίσης διαθέσιμο για λήψη από χρήστες της ΕΕ, καθώς εμφανίζεται στο Google Play Store, πράγμα που μπορεί να σημαίνει ότι εξυπηρετεί μια ευρωπαϊκή κοινότητα και ως εκ τούτου χρειάζεται να συμμορφωθεί με τους νόμους προστασίας δεδομένων της ΕΕ (GDPR). Ωστόσο, δεν υπάρχει λειτουργικός σύνδεσμος προς την πολιτική απορρήτου της εφαρμογής ή ακόμη και στην ιστοσελίδα στο Google Play Store, κάτι που μπορεί να αποτελεί παράβαση του GDPR και αποτυχία της εταιρείας/υπεύθυνου επεξεργασίας/εφαρμογής να ενημερώσει επαρκώς τους υποκείμενους δεδομένων σχετικά με τις χρήσεις των δεδομένων τους.

Β.1.4 Συμπέρασμα

Η ευρεία επίδραση των εφαρμογών που εξετάστηκαν στην έρευνά οδηγεί στο συμπέρασμα ότι προσωπικές λεπτομέρειες από την ιδιωτική ζωή εκατομμυρίων χρηστών σε όλο τον κόσμο κοινοποιούνται στο Facebook και άλλους τρίτους χωρίς την ελεύθερη, σαφή και ενημερωμένη συγκατάθεση τους, ειδικά όσον αφορά ευαίσθητα προσωπικά δεδομένα, όπως δεδομένα που αφορούν την υγεία ή τη σεξουαλική ζωή ενός χρήστη.

Η έρευνά υπογραμμίζει ότι υπάρχουν σοβαρές ανησυχίες όσον αφορά την συμμόρφωσή τους με τις υποχρεώσεις τους σύμφωνα με τον Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR), ειδικά όσον αφορά τη συγκατάθεση και τη διαφάνεια.

Αυτό θέτει ενδιαφέροντα ζητήματα.

- Πρώτον, ακόμη και όταν ο GDPR ισχύει, για παράδειγμα, στις χώρες της ΕΕ, αυτό δεν σημαίνει ότι οι ελεγκτές τηρούν τον κανονισμό. Όπως αποδεικνύει η έρευνά, οι εφαρμογές που στοχεύουν τους χρήστες της ΕΕ πρέπει να συμμορφώνονται, μεταξύ άλλων, με αυστηρές υποχρεώσεις συγκατάθεσης και διαφάνειας όσον αφορά την επεξεργασία προσωπικών δεδομένων, αλλά συχνά αποτυγχάνουν να το κάνουν.

⁷Πηγή: https://www.privacyinternational.org/sites/default/files/flysystem/styles/middle_column_small/local-default/2019-09/Picture%2021.png?itok=TNxFWip4

- Δεύτερον, ενώ οι εφαρμογές που βρίσκονται στην Ευρώπη ενδέχεται να μην πληρούν τις υποχρεώσεις τους σύμφωνα με το GDPR, οι χρήστες της ΕΕ εξακολουθούν να έχουν το κατάλληλο δικαίωμα προσφυγής, όπως η δυνατότητα να θέσουν το θέμα στον ελεγκτή απευθείας ή να υποβάλουν καταγγελία στην εθνική αρχή εποπτείας ή ακόμη να καταγγείλουν τον ελεγκτή στα εθνικά δικαστήρια. Ωστόσο, το ίδιο δεν ισχύει για τους χρήστες που βρίσκονται σε χώρες χωρίς κατάλληλους νόμους προστασίας δεδομένων ή με νόμους προστασίας δεδομένων που δεν διαθέτουν αποτελεσματική επιβολή.

Οι πρακτικές που αναδεικνύονται από αυτήν την έρευνα θα πρέπει να λειτουργήσουν ως παράδειγμα κατάχρησης που θα πρέπει να ωθήσει τους νομοθέτες και τους ρυθμιστές να υπερασπιστούν τα δικαιώματα των χρηστών.

Οι εταιρείες επίσης δεν πρέπει να αποφεύγουν τις ευθύνες τους. Η Facebook ανακοίνωσε ότι θα ξεκινήσει ένα εργαλείο που θα επιτρέπει στους χρήστες της να σταματήσουν τις εφαρμογές και τις επιχειρήσεις να κοινοποιούν τα δεδομένα τους στο κοινωνικό δίκτυο, το οποίο θα αντιμετωπίσει το πρόβλημα για ορισμένους χρήστες. Ωστόσο, είναι ανεπαρκές, καθώς δεν θα προστατεύσει τους χρήστες των εφαρμογών που δεν διαθέτουν προφίλ στο Facebook.

Η ευθύνη δεν πρέπει να βαρύνει τους χρήστες για να ανησυχούν για το τι μοιράζονται με τις εφαρμογές που έχουν επιλέξει. Η ευθύνη πρέπει να βαρύνει τις εταιρείες

Σημείωση

Το Privacy International έχει πραγματοποιήσει και άλλες σχετικές έρευνες, αποκαλύπτοντας παρόμοιες πρακτικές σχετικά με τη χρήση δεδομένων. Παρότι δεν είναι δυνατόν να παρουσιαστούν όλες εδώ, παραθέτουμε τα σχετικά URL για μελέτη:

- <https://privacyinternational.org/report/2647/how-apps-android-share-data>
- <https://privacyinternational.org/blog/2758/appdata-update>
- https://media.ccc.de/v/35c3-9941-how_facebook_tracks_you_on_android
- <https://privacyinternational.org/appdata>

Παράρτημα Γ'

DATE-a-a-S:Source Code

Γ'.1 Dockerfiles

Δείτε: <https://codeberg.org/DATE-A-A-S/dateaas>

Γ'.1.1 Bliss OS

```
#!/usr/bin/docker
# Title:          Dock-Droid (Docker Android)
# Author:         Sick.Codes https://twitter.com/sickcodes
# Version:        1.0
# License:        GPLv3+
# Repository:     https://github.com/sickcodes/dock-droid
# Website:        https://sick.codes
#
# This Dockerfile is a wrapper for Android x86 raw or qcow2 images.
#
# Build:  docker build -t dock-droid .
#

FROM archlinux:base-devel

MAINTAINER 'https://twitter.com/sickcodes' <https://sick.codes>

SHELL ["/bin/bash", "-c"]

# OPTIONAL: Arch Linux server mirrors for super fast builds
# set RANKMIRRORS to any value other than nothing, e.g. -e RANKMIRRORS=
# true
ARG RANKMIRRORS
ARG MIRROR_COUNTRY=US
ARG MIRROR_COUNT=10

RUN if [[ "${RANKMIRRORS}" ]]; then \
    { pacman -Sy wget --noconfirm || pacman -Syu wget --noconfirm ; } \
    \
    ; wget -O ./rankmirrors "https://raw.githubusercontent.com/sickcodes/dock-droid/master/rankmirrors" \
    ; wget -O- "https://www.archlinux.org/mirrorlist/?country=\${MIRROR\_COUNTRY:-US}&protocol=https&use\_mirror\_status=on" \
    | sed -e 's/^#Server/Server/' -e '/^#/d' \
    | head -n "${MIRROR_COUNT:-10}" \
    | bash ./rankmirrors --verbose --max-time 5 - > /etc/pacman.d/mirrorlist \
    && tee -a /etc/pacman.d/mirrorlist <<< 'Server = http://mirrors.evowise.com/archlinux/\$repo/os/\$arch' \
    && tee -a /etc/pacman.d/mirrorlist <<< 'Server = http://mirror.rackspace.com/archlinux/\$repo/os/\$arch' \
```

```

    && tee -a /etc/pacman.d/mirrorlist <<< 'Server = https://mirror.
rackspace.com/archlinux/$repo/os/$arch' \
    && cat /etc/pacman.d/mirrorlist \
; fi

RUN pacman -Syu git zip vim nano alsa-utils openssh unzip usbutils --
noconfirm \
&& ln -s /bin/vim /bin/vi \
&& useradd arch -p arch \
&& tee -a /etc/sudoers <<< 'arch ALL=(ALL) NOPASSWD: ALL' \
&& mkdir /home/arch \
&& chown arch:arch /home/arch

# allow ssh to container
RUN mkdir -m 700 /root/.ssh

WORKDIR /root/.ssh
RUN touch authorized_keys \
&& chmod 644 authorized_keys

WORKDIR /etc/ssh
RUN tee -a sshd_config <<< 'AllowTcpForwarding yes' \
&& tee -a sshd_config <<< 'PermitTunnel yes' \
&& tee -a sshd_config <<< 'X11Forwarding yes' \
&& tee -a sshd_config <<< 'PasswordAuthentication yes' \
&& tee -a sshd_config <<< 'PermitRootLogin yes' \
&& tee -a sshd_config <<< 'PubkeyAuthentication yes' \
&& tee -a sshd_config <<< 'HostKey /etc/ssh/ssh_host_rsa_key' \
&& tee -a sshd_config <<< 'HostKey /etc/ssh/ssh_host_ecdsa_key' \
&& tee -a sshd_config <<< 'HostKey /etc/ssh/ssh_host_ed25519_key'

USER arch

ENV USER arch

WORKDIR /home/arch

RUN git clone https://aur.archlinux.org/android-sdk-platform-tools.git \
&& cd android-sdk-platform-tools \
&& makepkg -si --nocheck --force --noconfirm \
; source /etc/profile.d/android-sdk-platform-tools.sh || exit 1

RUN git clone https://aur.archlinux.org/binfmt-qemu-static.git \
&& cd binfmt-qemu-static \
&& makepkg -si --nocheck --force --noconfirm || exit 1

RUN git clone https://aur.archlinux.org/qemu-user-static-bin.git \
&& cd qemu-user-static-bin \
&& makepkg -si --nocheck --force --noconfirm || exit 1

WORKDIR /home/arch

# optional --build-arg to change branches for testing
ARG BRANCH=master
ARG REPO='https://github.com/sickcodes/dock-droid.git'
RUN git clone --recurse-submodules --depth 1 --branch "${BRANCH}" "${REPO}
}"

WORKDIR /home/arch/dock-droid

RUN touch ./enable-ssh.sh \

```

```

&& chmod +x ./enable-ssh.sh \
&& tee -a enable-ssh.sh <<< '[[ -f /etc/ssh/ssh_host_rsa_key ]] || \' \
\
&& tee -a enable-ssh.sh <<< '[[ -f /etc/ssh/ssh_host_ed25519_key ]] \
|| \' \
&& tee -a enable-ssh.sh <<< '[[ -f /etc/ssh/ssh_host_ed25519_key ]] \
|| \' \
&& tee -a enable-ssh.sh <<< 'sudo /usr/bin/ssh-keygen -A' \
&& tee -a enable-ssh.sh <<< 'nohup sudo /usr/bin/sshd -D &'

RUN yes | sudo pacman -Syu qemu virglrenderer libvirt dnsmasq virt-
manager bridge-utils openresolv jack ebttables edk2-ovmf netctl
libvirt-dbus wget --overwrite --noconfirm \
&& yes | sudo pacman -Scc

ARG LINUX=true

# # required to use libguestfs inside a docker container, to create
# bootdisks for docker-osx on-the-fly
# RUN if [[ "${LINUX}" == true ]]; then \
#     sudo pacman -Syu linux libguestfs --overwrite --noconfirm \
#     ; fi

ARG COMPLETE=true

ARG CDROM_IMAGE_URL=https://sourceforge.net/projects/blissos-x86/files/
Official/bleeding_edge/Generic%20builds%20-%20Pie/11.13/Bliss-v11
.13--OFFICIAL-20201113-1525_x86_64_k-k4.19.122-ax86-ga-rmi_m-20.1.0-
llvm90_dgc-t3_gms_intelhd.iso
# ARG CDROM_IMAGE_URL=https://sourceforge.net/projects/blissos-dev/files/
Android-Generic/PC/bliss/R/gapps/BlissOS-14.3-x86_64-202106261907_k-
android12-5.10.46-ax86_m-21.1.3_r-x86_emugapps_cros-hd.iso
# ARG CDROM_IMAGE_URL=https://sourceforge.net/projects/blissos-dev/files/
Android-Generic/PC/bliss/R/gapps/BlissOS-14.3-x86_64-202106181339_k-
google-5.4.112-lts-ax86_m-r_emugapps_cros-hd_gearlock.iso

ENV CDROM_IMAGE_URL="${CDROM_IMAGE_URL}"

# use the COMPLETE arg, for a complete image, ready to boot.
# otherwise use your own image: -v "$PWD/disk.img":/image
ARG WGET_OPTIONS=
# ARG WGET_OPTIONS='--no-verbose'

RUN if [[ "${COMPLETE}" ]]; then \
    echo "Downloading 1GB image... This step might take a while...
    Press Ctrl+C if you want to abort." \
    && wget ${WGET_OPTIONS} "${CDROM_IMAGE_URL}" || exit 1 \
    ; fi

ARG QCOW_SIZE=50G

RUN qemu-img create -f qcow2 /home/arch/dock-droid/android.qcow2 "${
QCOW_SIZE}"

# RUN [[ -z "${VDI}" ]] && qemu-img convert -f vdi -O qcow2 "${VDI}"
# android.qcow2
# RUN [[ -z "${ISO}" ]] && -cdrom \

#### Mount disk inside container

# sudo modprobe nbd \
# sudo qemu-nbd --connect=/dev/nbd0 android2.qcow2 -f qcow2 \

```

```

# sudo fdisk /dev/nbd0 -l\
# mkdir /tmp/image /tmp/system
# sudo mount /dev/nbd0p1 /tmp/image

# sudo mount /tmp/image/bliss-x86-11.13/system.img /tmp/system
# sudo tee -a /tmp/system/build.prop <<< 'ro.adb.secure=0'
# sudo umount /tmp/system
# sudo umount /tmp/image
# sudo qemu-nbd -d /dev/nbd0

RUN wget -O supergrub2.iso https://telkomuniversity.dl.sourceforge.net/
  project/supergrub2/2.04s2-beta2/super_grub2_disk_2.04s2-beta2/
  supergrub2-2.04s2-beta2-multiarch-CD.iso

# RUN sudo guestfish -a /home/user/bliss/android2.qcow2 \

# sudo guestmount -a android.qcow2 -m /dev/vg0 /mnt

#### SPECIAL RUNTIME ARGUMENTS BELOW

# env -e ADDITIONAL_PORTS with a comma
# for example, -e ADDITIONAL_PORTS=hostfwd=tcp::23-:23,
ENV ADDITIONAL_PORTS=

# add additional QEMU boot arguments
ENV BOOT_ARGS=

# edit the CPU that is beign emulated
ENV CPU=host
ENV CPUID_FLAGS='+invts,vmware-cpuid-freq=on,+pcid,+ssse3,+sse4.2,+
  popcnt,+avx,+aes,+xsave,+xsaveopt,check,'

ENV DISPLAY=:0.0
ENV DISPLAY_ARGUMENTS='-vga vmware'

ENV ENABLE_KVM='-enable-kvm'

ENV IMAGE_PATH=/home/arch/dock-droid/android.qcow2
ENV IMAGE_FORMAT=qcow2

ENV KVM='accel=kvm:tcg'

# ENV NETWORKING=e1000-82545em
ENV NETWORKING=vmxnet3

# add libguestfs debug output
ENV LIBGUESTFS_DEBUG=1
ENV LIBGUESTFS_TRACE=1

ENV PATH="\${PATH}:/opt/android-sdk/platform-tools"

# dynamic RAM options for runtime
ENV RAM=4
# ENV RAM=max
# ENV RAM=half

# ENV WEBCAM=/dev/video0
ENV WEBCAM=

RUN touch Launch.sh \
  && chmod +x ./Launch.sh \
  && tee -a Launch.sh <<< '#!/bin/bash' \

```



```

&& tee -a Launch.sh <<< 'set -eux' \
&& tee -a Launch.sh <<< 'source /etc/profile.d/android-sdk-platform-
tools.sh' \
&& tee -a Launch.sh <<< 'sudo chown $(id -u):$(id -g) /dev/kvm 2>/
dev/null || true' \
&& tee -a Launch.sh <<< 'sudo chown -R $(id -u):$(id -g) /dev/snd 2>/
dev/null || true' \
&& tee -a Launch.sh <<< 'sudo chown -R $(id -u):$(id -g) /dev/video
{0..10} 2>/dev/null || true' \
&& tee -a Launch.sh <<< 'sudo qemu-system-x86_64 -m ${RAM:-4}000 \' \
&& tee -a Launch.sh <<< '${ENABLE_KVM}-enable-kvm' \' \
&& tee -a Launch.sh <<< '-cpu ${CPU-host},${CPUID_FLAGS}-+invts,
vmware-cpuid-freq=on,+pcid,+ssse3,+sse4.2,+popcnt,+avx,+aes,+xsave,+
xsaveopt,check,">${BOOT_ARGS} \' \
&& tee -a Launch.sh <<< '-smp ${CPU_STRING:-$(nproc)} \' \
&& tee -a Launch.sh <<< '-machine q35,${KVM-"accel=kvm:tcg"} \' \
&& tee -a Launch.sh <<< '-smp ${CPU_STRING:-${SMP:-4},cores=${CORES
:-4}} \' \
&& tee -a Launch.sh <<< '-hda "${IMAGE_PATH:=/home/arch/dock-droid/
android.qcow2}" \' \
&& tee -a Launch.sh <<< '-usb -device usb-kbd -device usb-tablet \' \
&& tee -a Launch.sh <<< '-smbios type=2 \' \
&& tee -a Launch.sh <<< '-audiodev ${AUDIO_DRIVER:-alsa},id=hda -
device ich9-intel-hda -device hda-duplex,audiodev=hda \' \
&& tee -a Launch.sh <<< '-device usb-ehci,id=ehci \' \
&& tee -a Launch.sh <<< '-netdev user,id=net0,hostfwd=tcp::${
INTERNAL_SSH_PORT:-10022}-:22,hostfwd=tcp::${SCREEN_SHARE_PORT
:-5900}-:5900,hostfwd=tcp::${ADB_PORT:-5555}-:5555,${ADDITIONAL_PORTS
} \' \
&& tee -a Launch.sh <<< '-device ${NETWORKING:-vmxnet3},netdev=net0,
id=net0,mac=${MAC_ADDRESS:-00:11:22:33:44:55} \' \
&& tee -a Launch.sh <<< '-monitor stdio \' \
&& tee -a Launch.sh <<< '-boot menu=on \' \
&& tee -a Launch.sh <<< '-cdrom "${CDROM:-${CDROM}}" \' \
&& tee -a Launch.sh <<< '${DISPLAY_ARGUMENTS:=--vga vmware} \' \
&& tee -a Launch.sh <<< '${WEBCAM:-} \' \
&& tee -a Launch.sh <<< '${EXTRA:-}'

VOLUME ["/tmp/.X11-unix"]

CMD export CDROM="${CDROM:=$(basename "${CDROM_IMAGE_URL}")}" \
&& touch ./android.qcow2 "${CDROM}" \
&& ./enable-ssh.sh \
&& /bin/bash -c ./Launch.sh

```

Listing Γ.1: Dockerfile Bliss OS

Γ.1.2 AndroidSDK

Docker Image with a pre-installed Android SDK

```

# =====
#
# Android SDK Docker Image
# =====
#
# Base image
# -----
#
FROM ubuntu:22.04

```

```

# Author
# -----
#
LABEL maintainer "thyrlian@gmail.com"

# support multiarch: i386 architecture
# install Java
# install essential tools
ARG JDK_VERSION=17
RUN dpkg --add-architecture i386 && \
  apt-get update && \
  apt-get dist-upgrade -y && \
  apt-get install -y --no-install-recommends libncurses5:i386 libc6:
i386 libstdc++6:i386 lib32gcc-s1 lib32ncurses6 lib32z1 zlib1g:i386 && \
  \
  apt-get install -y --no-install-recommends openjdk-${JDK_VERSION}-jdk
&& \
  apt-get install -y --no-install-recommends git wget unzip && \
  apt-get clean && rm -rf /var/lib/apt/lists/*

# download and install Gradle
# https://services.gradle.org/distributions/
ARG GRADLE_VERSION=8.3
ARG GRADLE_DIST=bin
RUN cd /opt && \
  wget -q https://services.gradle.org/distributions/gradle-${
GRADLE_VERSION}-${GRADLE_DIST}.zip && \
  unzip gradle*.zip && \
  ls -d */ | sed 's/\/*$//g' | xargs -I{} mv {} gradle && \
  rm gradle*.zip

# download and install Kotlin compiler
# https://github.com/JetBrains/kotlin/releases/latest
ARG KOTLIN_VERSION=1.9.10
RUN cd /opt && \
  wget -q https://github.com/JetBrains/kotlin/releases/download/v${
KOTLIN_VERSION}/kotlin-compiler-${KOTLIN_VERSION}.zip && \
  unzip *kotlin*.zip && \
  rm *kotlin*.zip

# download and install Android SDK
# https://developer.android.com/studio#command-line-tools-only
ARG ANDROID_SDK_VERSION=10406996
ENV ANDROID_HOME /opt/android-sdk
RUN mkdir -p ${ANDROID_HOME}/cmdline-tools && \
  wget -q https://dl.google.com/android/repository/commandlinetools-
linux-${ANDROID_SDK_VERSION}_latest.zip && \
  unzip *tools*linux*.zip -d ${ANDROID_HOME}/cmdline-tools && \
  mv ${ANDROID_HOME}/cmdline-tools/cmdline-tools ${ANDROID_HOME}/
cmdline-tools/tools && \
  rm *tools*linux*.zip

# set the environment variables
ENV JAVA_HOME /usr/lib/jvm/java-${JDK_VERSION}-openjdk-amd64
ENV GRADLE_HOME /opt/gradle
ENV KOTLIN_HOME /opt/kotlinc
ENV PATH ${PATH}:${GRADLE_HOME}/bin:${KOTLIN_HOME}/bin:${ANDROID_HOME}/
cmdline-tools/latest/bin:${ANDROID_HOME}/cmdline-tools/tools/bin:${
ANDROID_HOME}/platform-tools:${ANDROID_HOME}/emulator
# WORKAROUND: for issue https://issuetracker.google.com/issues/37137213
ENV LD_LIBRARY_PATH ${ANDROID_HOME}/emulator/lib64:${ANDROID_HOME}/
emulator/lib64/qt/lib

```

```

# patch emulator issue: Running as root without --no-sandbox is not
  supported. See https://crbug.com/638180.
# https://doc.qt.io/qt-5/qtwebengine-platform-notes.html#sandboxing-
  support
ENV QTWEBENGINE_DISABLE_SANDBOX 1

# accept the license agreements of the SDK components
ADD license_accepter.sh /opt/
RUN chmod +x /opt/license_accepter.sh && /opt/license_accepter.sh
  $ANDROID_HOME

# setup adb server
EXPOSE 5037

# install and configure SSH server
EXPOSE 22
ADD sshd-banner /etc/ssh/
ADD accredited_keys /tmp/
RUN apt-get update && \
  apt-get install -y --no-install-recommends openssh-server supervisor
  locales && \
  mkdir -p /var/run/sshd /var/log/supervisord && \
  locale-gen en en_US en_US.UTF-8 && \
  apt-get remove -y locales && apt-get autoremove -y && \
  apt-get clean && rm -rf /var/lib/apt/lists/* && \
  FILE_SSHD_CONFIG="/etc/ssh/sshd_config" && \
  echo "\nBanner /etc/ssh/sshd-banner" >> $FILE_SSHD_CONFIG && \
  echo "\nPermitUserEnvironment=yes" >> $FILE_SSHD_CONFIG && \
  ssh-keygen -q -N "" -f /root/.ssh/id_rsa && \
  FILE_SSH_ENV="/root/.ssh/environment" && \
  touch $FILE_SSH_ENV && chmod 600 $FILE_SSH_ENV && \
  printenv | grep "JAVA_HOME\|GRADLE_HOME\|KOTLIN_HOME\|ANDROID_HOME\|
  LD_LIBRARY_PATH\|PATH" >> $FILE_SSH_ENV && \
  echo "\nauth required pam_env.so envfile=$FILE_SSH_ENV" >> /etc/pam.d
  /sshd && \
  FILE_AUTH_KEYS="/root/.ssh/authorized_keys" && \
  touch $FILE_AUTH_KEYS && chmod 600 $FILE_AUTH_KEYS && \
  for file in /tmp/*.pub; \
  do if [ -f "$file" ]; then echo "\n" >> $FILE_AUTH_KEYS && cat $file
  >> $FILE_AUTH_KEYS && echo "\n" >> $FILE_AUTH_KEYS; fi; \
  done && \
  (rm /tmp/*.pub 2> /dev/null || true)

ADD supervisord.conf /etc/supervisor/conf.d/
CMD ["/usr/bin/supervisord"]

```

F.1.3 Android Emulator

Emulator with different device profile and skins, such as Samsung Galaxy S6, LG Nexus 4, HTC Nexus One and more.

```

ARG DOCKER_ANDROID_VERSION
FROM budtmo/docker-android:base_${DOCKER_ANDROID_VERSION}

#=====
# Android Packages
#=====
ARG EMULATOR_ANDROID_VERSION
ARG EMULATOR_API_LEVEL
ENV EMULATOR_ANDROID_VERSION=${EMULATOR_ANDROID_VERSION} \
  EMULATOR_API_LEVEL=${EMULATOR_API_LEVEL} \
  EMULATOR_SYS_IMG=x86_64 \

```

```

    EMULATOR_IMG_TYPE=google_apis \
    EMULATOR_BROWSER=chrome
ENV PATH ${PATH}:${ANDROID_HOME}/build-tools
RUN yes | sdkmanager --licenses \
    && sdkmanager "platforms;android-${EMULATOR_API_LEVEL}" \
    "system-images;android-${EMULATOR_API_LEVEL};${EMULATOR_IMG_TYPE};${
    EMULATOR_SYS_IMG}" "emulator" \
    && ln -s ${ANDROID_HOME}/emulator/emulator /usr/bin/

#####
# UI Packages
#-----
# ffmpeg
#   Video recorder
# feh
#   Screen background
# libxcomposite-dev
#   Window System for Emulator
# menu
#   Debian menu
# openbox
#   Windows manager
# x11vnc
#   VNC server
# xterm
#   Terminal emulator
#####
RUN apt-get -qqy update && apt-get -qqy install --no-install-recommends \
    ffmpeg \
    feh \
    libxcomposite-dev \
    menu \
    openbox \
    x11vnc \
    xterm \
    && apt autoremove -y \
    && apt clean all \
    && rm -rf /var/lib/apt/lists/*

#####
# noVNC
#####
ENV NOVNC_VERSION="1.4.0" \
    WEBSOCKIFY_VERSION="0.11.0" \
    OPT_PATH="/opt"
RUN wget -nv -O noVNC.zip "https://github.com/novnc/novnc/archive/refs/
tags/v${NOVNC_VERSION}.zip" \
    && unzip -x noVNC.zip \
    && rm noVNC.zip \
    && mv noVNC-${NOVNC_VERSION} ${OPT_PATH}/noVNC \
    && wget -nv -O websockify.zip "https://github.com/novnc/websockify/
archive/refs/tags/v${WEBSOCKIFY_VERSION}.zip" \
    && unzip -x websockify.zip \
    && mv websockify-${WEBSOCKIFY_VERSION} ${OPT_PATH}/noVNC/utils/
websockify \
    && rm websockify.zip \
    && ln ${OPT_PATH}/noVNC/vnc.html ${OPT_PATH}/noVNC/index.html

ENV DISPLAY=:0 \
    SCREEN_NUMBER=0 \
    SCREEN_WIDTH=1600 \
    SCREEN_HEIGHT=900 \

```

```

    SCREEN_DEPTH=24+32 \
    VNC_PORT=5900 \
    WEB_VNC_PORT=6080

EXPOSE 5900 6080

#=====
# Copy app
#=====
RUN rm -rf ${SCRIPT_PATH}
ENV SCRIPT_PATH="docker-android"
ENV WORK_PATH="/home/androidusr"
ENV APP_PATH=${WORK_PATH}/${SCRIPT_PATH}
RUN mkdir -p ${APP_PATH}
COPY mixins ${APP_PATH}/mixins
COPY cli ${APP_PATH}/cli
RUN chown -R 1300:1301 ${APP_PATH} \
  && pip install --quiet -e ${APP_PATH}/cli

#=====
# Configure OpenBox
#=====
RUN echo ${APP_PATH}/mixins/configs/display/.fehbg >> /etc/xdg/openbox/
  autostart

#=====
# Use created user
#=====
USER 1300:1301
ENV LOG_PATH=${WORK_PATH}/logs \
  WEB_LOG_PORT=9000
EXPOSE 9000
RUN mkdir -p ${LOG_PATH}

#=====
# Run App
#=====
STOPSIGNAL SIGTERM
ENV DEVICE_TYPE=emulator
ENTRYPOINT ["/home/androidusr/docker-android/mixins/scripts/run.sh"]

```

F.1.4 WireGuard

```

# syntax=docker/dockerfile:1

FROM ghcr.io/linuxserver/baseimage-alpine:3.18

# set version label
ARG BUILD_DATE
ARG VERSION
ARG WIREGUARD_RELEASE
LABEL build_version="Linuxserver.io version:- ${VERSION} Build-date:- ${
  BUILD_DATE}"
LABEL maintainer="thespad"

RUN \
  echo "**** install dependencies ****" && \
  apk add --no-cache --virtual=build-dependencies \
    build-base \
    elfutils-dev \
    git \

```

```

    linux-headers && \
apk add --no-cache \
    bc \
    coredns \
    grep \
    iproute2 \
    iptables \
    ip6tables \
    iputils \
    libcap-utils \
    libqrencode \
    net-tools \
    openresolv && \
echo "wireguard" >> /etc/modules && \
echo "**** install wireguard-tools ****" && \
if [ -z ${WIREGUARD_RELEASE+x} ]; then \
    WIREGUARD_RELEASE=$(curl -sX GET "https://api.github.com/repos/
WireGuard/wireguard-tools/tags" \
    | jq -r .[0].name); \
fi && \
cd /app && \
git clone https://git.zx2c4.com/wireguard-tools && \
cd wireguard-tools && \
git checkout "${WIREGUARD_RELEASE}" && \
sed -i 's|\\[[ $proto == -4 \\]| && cmd sysctl -q net.ipv4.conf.all
\\.src_valid_mark=1|[[ $proto == -4 ]] \\&& [[ $(sysctl -n net.ipv4.
conf.all.src_valid_mark) != 1 ]] \\&& cmd sysctl -q net.ipv4.conf.all
\\.src_valid_mark=1|' src/wg-quick/linux.bash && \
make -C src -j$(nproc) && \
make -C src install && \
rm -rf /etc/wireguard && \
ln -s /config/wg_confs /etc/wireguard && \
echo "**** clean up ****" && \
apk del --no-network build-dependencies && \
rm -rf \
    /tmp/*

# add local files
COPY /root /

# ports and volumes
EXPOSE 51820/udp

```

Γ.1.5 Appium

```

FROM ubuntu:focal-20231211

ENV DEBIAN_FRONTEND=noninteractive

#=====
# General Packages
#-----
# ca-certificates
#   SSL client
# curl
#   Transfer data from or to a server
# gnupg
#   Encryption software. It is needed for nodejs
# libgconf-2-4
#   Required package for chrome and chromedriver to run on Linux

```

```

# libqt5webkit5
#   Web content engine (Fix issue in Android)
# openjdk-11-jdk
#   Java
# sudo
#   Sudo user
# tzdata
#   Timezone
# unzip
#   Unzip zip file
# wget
#   Network downloader
# xvfb
#   X virtual framebuffer
# zip
#   Make a zip file
#=====
RUN apt-get -qqy update && \
    apt dist-upgrade -y && \
    apt-get -qqy --no-install-recommends install \
    ca-certificates \
    curl \
    gnupg \
    libgconf-2-4 \
    libqt5webkit5 \
    openjdk-11-jdk \
    sudo \
    tzdata \
    unzip \
    wget \
    xvfb \
    zip \
    && rm -rf /var/lib/apt/lists/*

#=====
# Set JAVA_HOME
#=====
ENV JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64" \
    PATH=$PATH:$JAVA_HOME/bin

#=====
# Set Timezone (UTC as default)
#=====
ENV TZ "UTC"
RUN echo "${TZ}" > /etc/timezone \
    && dpkg-reconfigure --frontend noninteractive tzdata

#=====
# Create a user
#=====
ARG USER_PASS=secret
RUN groupadd androidusr \
    --gid 1301 \
    && useradd androidusr \
    --uid 1300 \
    --gid 1301 \
    --create-home \
    --shell /bin/bash \
    && usermod -aG sudo androidusr \
    && echo androidusr:${USER_PASS} | chpasswd \
    && echo 'androidusr ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers

```

```

WORKDIR /home/androidusr

#=====
# Install Android SDK
#=====
ENV SDK_VERSION=commandlinetools-linux-8512546_latest
ENV ANDROID_BUILD_TOOLS_VERSION=34.0.0
ENV ANDROID_FOLDER_NAME=cmdline-tools
ENV ANDROID_DOWNLOAD_PATH=/home/androidusr/${ANDROID_FOLDER_NAME} \
  ANDROID_HOME=/opt/android \
  ANDROID_TOOL_HOME=/opt/android/${ANDROID_FOLDER_NAME}

RUN wget -O tools.zip https://dl.google.com/android/repository/${
  SDK_VERSION}.zip && \
  unzip tools.zip && rm tools.zip && \
  chmod a+x -R ${ANDROID_DOWNLOAD_PATH} && \
  chown -R 1300:1301 ${ANDROID_DOWNLOAD_PATH} && \
  mkdir -p ${ANDROID_TOOL_HOME} && \
  mv ${ANDROID_DOWNLOAD_PATH} ${ANDROID_TOOL_HOME}/tools
ENV PATH=$PATH:${ANDROID_TOOL_HOME}/tools:${ANDROID_TOOL_HOME}/tools/bin

# https://askubuntu.com/questions/885658/android-sdk-repositories-cfg-
# could-not-be-loaded
RUN mkdir -p ~/.android && \
  touch ~/.android/repositories.cfg && \
  echo y | sdkmanager "platform-tools" && \
  echo y | sdkmanager "build-tools;${ANDROID_BUILD_TOOLS_VERSION}" && \
  mv ~/.android .android && \
  chown -R 1300:1301 .android
ENV PATH=$PATH:${ANDROID_HOME}/platform-tools:${ANDROID_HOME}/build-tools

#=====
# Install latest nodejs, npm, appium
# Using this workaround to install Appium -> https://github.com/appium/
# appium/issues/10020 -> Please remove this workaround asap
#=====
ENV NODE_VERSION=18
ENV APPIUM_VERSION=2.3.0
RUN curl -sL https://deb.nodesource.com/setup_${NODE_VERSION}.x | bash && \
  \
  apt-get -qq install nodejs && \
  npm install -g appium@${APPIUM_VERSION} && \
  exit 0 && \
  npm cache clean && \
  apt-get remove --purge -y npm && \
  apt-get autoremove --purge -y && \
  rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* && \
  apt-get clean

#=====
# Fix permission issue to download e.g. chromedriver
#=====
RUN chown -R 1300:1301 /usr/lib/node_modules/appium

#=====
# Copy scripts
#=====
ENV SCRIPT_PATH="appium-docker-android"
RUN mkdir -p ${SCRIPT_PATH}
COPY start.sh \
  generate_selenium_config.sh \
  wireless_autoconnect.sh \

```



```

    wireless_connect.sh \
    ${SCRIPT_PATH}/
RUN chown -R 1300:1301 ${SCRIPT_PATH}
ENV APP_PATH=/home/androidusr/${SCRIPT_PATH}

#=====
# Use created user
#=====
USER 1300:1301

#=====
# Install basic Android drivers
#=====
ENV APPIUM_DRIVER_ESPRESSO_VERSION="2.30.0"
ENV APPIUM_DRIVER_FLUTTER_VERSION="2.3.0"
ENV APPIUM_DRIVER_GECKO_VERSION="1.2.10"
ENV APPIUM_DRIVER_UIAUTOMATOR2_VERSION="2.37.0"
RUN appium driver install --source=npm appium-espresso-driver@${
  APPIUM_DRIVER_ESPRESSO_VERSION} && \
  appium driver install --source=npm appium-flutter-driver@${
  APPIUM_DRIVER_FLUTTER_VERSION} && \
  appium driver install --source=npm appium-geckodriver@${
  APPIUM_DRIVER_GECKO_VERSION} && \
  appium driver install --source=npm appium-uiautomator2-driver@${
  APPIUM_DRIVER_UIAUTOMATOR2_VERSION}

#=====
# Expose Port
#-----
# 4723
# Appium port
#=====
EXPOSE 4723

#=====
# Start script
#=====
CMD ./${SCRIPT_PATH}/start.sh

```

Listing Γ.2: Dockerfile Appium

Appium script - start.sh

```

#!/bin/bash

# It is workaround to access adb from androidusr
echo "Prepare adb to have access to device"
sudo /opt/android/platform-tools/adb devices >/dev/null
sudo chown -R 1300:1301 .android
echo "adb can be used now"

# Connect device via wireless
if [ "${REMOTE_ADB}" = true ]; then
  echo "Connect device via wireless"
  # Avoid lost connection
  ${APP_PATH}/wireless_autoconnect.sh && \
  ${APP_PATH}/wireless_connect.sh
fi

# Command to start Appium
APPIUM_LOG="${APPIUM_LOG:-/var/log/appium.log}"

```

```

command="xvfb-run appium --log $APPIUM_LOG"

# Adding Selenium configurations if needed
if [ "${CONNECT_TO_GRID}" = true ]; then
  NODE_CONFIG_JSON="${APP_PATH}/nodeconfig.json"
  if [ "${CUSTOM_NODE_CONFIG}" != true ]; then
    ${APP_PATH}/generate_selenium_config.sh ${NODE_CONFIG_JSON}
  fi
  command+=" --nodeconfig ${NODE_CONFIG_JSON}"
fi

if [ "${DEFAULT_CAPABILITIES}" = true ]; then
  DEFAULT_CAPABILITIES_JSON="${APP_PATH}/defaultcapabilities.json"
  command+=" --default-capabilities ${DEFAULT_CAPABILITIES_JSON}"
fi

# Adding additional Appium configuration if needed
command+=" ${APPIUM_ADDITIONAL_PARAMS}"

# Run the whole command
pkill -x xvfb-run
rm -rf /tmp/.X99-lock
${command}

```

Listing Γ.3: Dockerfile DockerfileAppiumScript-start

Appium script - generate_selenium_config

```

#!/bin/bash

node_config_json=${1}

if [ -z "${PLATFORM_NAME}" ]; then
  PLATFORM_NAME="Android"
fi

if [ -z "${APPIUM_HOST}" ]; then
  APPIUM_HOST=$(hostname -i)
fi

if [ -z "${APPIUM_PORT}" ]; then
  APPIUM_PORT=4723
fi

if [ -z "${SELENIUM_HOST}" ]; then
  SELENIUM_HOST="172.17.0.1"
fi

if [ -z "${SELENIUM_PORT}" ]; then
  SELENIUM_PORT=4444
fi

if [ -z "${BROWSER_NAME}" ]; then
  BROWSER_NAME="android"
fi

if [ -z "${NODE_TIMEOUT}" ]; then
  NODE_TIMEOUT=300
fi

#Get device names

```

```

devices=$(adb devices | grep -oP "\K([^\s]+)(?=\sdevice(\W|$))")
echo "Devices found: ${#devices[@]}"

#Create capabilities json configs
function create_capabilities() {
  capabilities=""
  for name in ${devices[@]}; do
    os_version=$(adb -s $name shell getprop ro.build.version.release |
tr -d '\r')
    serial_number=$(adb -s $name shell getprop ro.serialno | tr -d '\r')
    capabilities+=$(cat <<_EOF
{
  "platform": "$PLATFORM_NAME",
  "platformName": "$PLATFORM_NAME",
  "version": "$os_version",
  "browserName": "$BROWSER_NAME",
  "deviceName": "$name",
  "maxInstances": 1,
  "applicationName": "$serial_number"
}
_EOF
)
    if [ ${devices[-1]} != $name ]; then
      capabilities+=', '
    fi
  done
  echo "$capabilities"
}

#Final node configuration json string
nodeconfig=$(cat <<_EOF
{
  "capabilities": [$(create_capabilities)],
  "configuration": {
    "cleanUpCycle": 2000,
    "timeout": $NODE_TIMEOUT,
    "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
    "url": "http://$APPIUM_HOST:$APPIUM_PORT/wd/hub",
    "host": "$APPIUM_HOST",
    "port": $APPIUM_PORT,
    "maxSession": 6,
    "register": true,
    "registerCycle": 5000,
    "hubHost": "$SELENIUM_HOST",
    "hubPort": $SELENIUM_PORT
  }
}
_EOF
)
echo "${nodeconfig}" > ${node_config_json}

```

Listing

Γ.4:

Dockerfile

DockerfileAppiumScript-generate_selenium_config

Appium script - wireless_autoconnect.sh

```

#!/bin/bash

if [ ! -z "${REMOTE_ADB}" ]; then

```

```

if [ -z "${REMOTE_ADB_POLLING_SEC}" ]; then
    REMOTE_ADB_POLLING_SEC=5
fi

function connect() {
    while true; do
        #to avoid immediate run
        sleep ${REMOTE_ADB_POLLING_SEC}
        ${APP_PATH}/wireless_connect.sh
    done
}

( trap "true" HUP ; connect ) >/dev/null 2>/dev/null </dev/null &
disown

fi

```

Listing Γ.5: Dockerfile DockerfileAppiumScript-wireless_autoconnect

Appium script - wireless_connect.sh

```

#!/bin/bash

if [ ! -z "${ANDROID_DEVICES}" ]; then
    IFS=',' read -r -a array <<<"${ANDROID_DEVICES}"
    for i in "${!array[@]}"; do
        array_device=$(echo ${array[$i]} | tr -d " ")
        #string contains check
        if [[ ${connected_devices} != *${array_device}* ]]; then
            echo "Connecting to: ${array_device}"
            adb connect ${array_device} >/dev/null 2>/dev/null
            #Give time to finish connection
            sleep 2
            adb devices
            echo "Success!"
        fi
    done
fi

```

Listing Γ.6: Dockerfile DockerfileAppiumScript-wireless_connect

Γ.2 Compose Files

Γ.2.1 WireGuard

```

version: "2.1"
services:
  heimdall:
    image: linuxserver/heimdall
    container_name: heimdall
    volumes:
      - /home/user/appdata/heimdall:/config
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/London
    ports:
      - 80:80

```

```
- 443:443
restart: unless-stopped

nginx:
  image: linuxserver/nginx
  container_name: nginx
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=Europe/London
  volumes:
    - /home/user/appdata/nginx:/config
  ports:
    - 81:80
    - 444:443
  restart: unless-stopped

mariadb:
  image: linuxserver/mariadb
  container_name: mariadb
  environment:
    - PUID=1000
    - PGID=1000
    - MYSQL_ROOT_PASSWORD=ROOT_ACCESS_PASSWORD
    - TZ=Europe/London
  volumes:
    - /home/user/appdata/mariadb:/config
  ports:
    - 3306:3306
  restart: unless-stopped
```

Γ.2.2 Test Environment

```
version: "3"
services:
  chrome:
    image: selenium/node-chrome:4.16.1-20231219
    shm_size: 2gb
    depends_on:
      - selenium-hub
    environment:
      - SE_EVENT_BUS_HOST=selenium-hub
      - SE_EVENT_BUS_PUBLISH_PORT=4442
      - SE_EVENT_BUS_SUBSCRIBE_PORT=4443
    networks:
      wdio-selenium-net:

  firefox:
    image: selenium/node-firefox:4.16.1-20231219
    shm_size: 2gb
    depends_on:
      - selenium-hub
    environment:
      - SE_EVENT_BUS_HOST=selenium-hub
      - SE_EVENT_BUS_PUBLISH_PORT=4442
      - SE_EVENT_BUS_SUBSCRIBE_PORT=4443
    networks:
      wdio-selenium-net:

selenium-hub:
```

```

image: selenium/hub:4.16.1-20231219
container_name: selenium-hub
#   ports:
#     - "4442:4442"
#     - "4443:4443"
#     - "4444:4444"
expose:
  - "4442"
  - "4443"
  - "4444"
networks:
  wdio-selenium-net:

wdio:
  image: wdio
  container_name: wdio
  depends_on:
    - selenium-hub
  volumes:
    - wdio-data:/home/node/workspace
  networks:
    wdio-selenium-net:

allure:
  image: frankescobar/allure-docker-service
  container_name: allure
  depends_on:
    - selenium-hub
  environment:
    - CHECK_RESULTS_EVERY_SECONDS=3
    - KEEP_HISTORY=1
  ports:
    - "5050:5050"
  volumes:
    - wdio-data:/data
  networks:
    wdio-selenium-net:

networks:
  wdio-selenium-net:
    driver: bridge

volumes:
  wdio-data:

```

config file: wdio.conf.js

```

exports.config = {
  //
  // =====
  // Runner Configuration
  // =====
  runner: 'local', //
  // =====
  // Server Configurations
  // =====
  hostname: 'selenium-hub',
  port: 4444,
  path: '/',
  //
  // =====

```

```
// Specify Test Files
// =====
specs: [
  './test/specs/**/*.js'
],
// Patterns to exclude.
exclude: [
  // 'path/to/excluded/files'
],
//
// =====
// Capabilities
// =====
maxInstances: 10,
//
capabilities: [{
  browserName: 'chrome'
}, {
  browserName: 'firefox'
}],
//
// =====
// Test Configurations
// =====
// Level of logging verbosity: trace | debug | info | warn | error |
silent
logLevel: 'info',
//
// Set specific log levels per logger
// loggers:
// - webdriver, webdriverio
// - @wdio/browserstack-service, @wdio/devtools-service, @wdio/sauce-
service
// - @wdio/mocha-framework, @wdio/jasmine-framework
// - @wdio/local-runner
// - @wdio/sumologic-reporter
// - @wdio/cli, @wdio/config, @wdio/utils
// Level of logging verbosity: trace | debug | info | warn | error |
silent
// logLevels: {
//   webdriver: 'info',
//   '@wdio/appium-service': 'info'
// },
//
// If you only want to run your tests until a specific amount of
tests have failed use
// bail (default is 0 - don't bail, run all tests).
bail: 0,
//
baseUrl: 'http://selenium-hub',
//
// Default timeout for all waitFor* commands.
waitForTimeout: 10000,
//
// Default timeout in milliseconds for request
// if browser driver or grid doesn't send response
connectionRetryTimeout: 120000,
//
// Default request retries count
connectionRetryCount: 3,
//
```

```

// Test runner services
services: ['docker'],

// Framework you want to run your specs with.
framework: 'mocha',
//
// The number of times to retry the entire specfile when it fails as
// a whole
// specFileRetries: 1,
//
// Delay in seconds between the spec file retry attempts
// specFileRetriesDelay: 0,
//
// Whether or not retried spec files should be retried immediately or
// deferred to the end of the queue
// specFileRetriesDeferred: false,
//
//reporters: ['spec'],
reporters: [['allure', {
// test results will generate in the directory mentioned below
  outputDir: 'allure-results',
}]],

// Options to be passed to Mocha.
// See the full list at http://mochajs.org/
mochaOpts: {
  ui: 'bdd',
  timeout: 60000
}
}

```

Γ.2.3 Android and VPN

```

services:
  wireguard:
    image: lscr.io/linuxserver/wireguard
    container_name: wireguard
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Athens
    volumes:
      #- ./wg-server:/config
      - ./up-wireguard-client.conf:/config/wg_confs/wg0.conf
      - /lib/modules:/lib/modules
    ports:
      - "8080:8080"
      - "5800:5800" # firefox
#      - "6080:6080" # android
#      - "5554:5554" # android
#      - "5555:5555" # android
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=1
    restart: unless-stopped

# firefox:
#   image: jlesage/firefox

```



```
#   container_name: firefox
#   network_mode: "service:wireguard"
#   volumes:
#     - "./firefox:/config:rw"

android:
  image: butomol1989/docker-android-x86-7.1.1
  container_name: android
  network_mode: "service:wireguard"
  privileged: true
  environment:
    - DEVICE=Samsung Galaxy S6
    #   ports:
    #     - "6080:6080" # android
    #     - "5554:5554" # android
    #     - "5555:5555" # android
```

config file: up-wireguard-client.conf

```
[Interface]
PrivateKey = 9kscUmtUkYiXXan6fXZ0aaH5PwMDvIXaaYPP3+poqvQ=
Address = 10.0.0.1/32
DNS = 8.8.8.8

PostUp = iptables -t nat -A POSTROUTING -o wg+ -j MASQUERADE
PreDown = iptables -t nat -D POSTROUTING -o wg+ -j MASQUERADE
[Peer]
PublicKey = MfLzz0bTns3BuU7jv9wqsH7xtPQBd1ZndbxC3NrKxnw=
AllowedIPs = 0.0.0.0/0
Endpoint = 83.212.77.133:51820
```


Παράρτημα Δ΄

DATE-a-a-S:Api

Δ΄.1 API Description

Δείτε: <https://codeberg.org/DATE-A-A-S/dateaas>

Δ΄.1.1 Yaml

```

---
swagger: "2.0"
info:
  description: "This is a DATE-A-A-S REST API."
  version: "1.0.7"
  title: "DATE-A-A-S"
  license:
    name: "AGPL-3.0-or-later"
    url: "https://spdx.org/licenses/AGPL-3.0-or-later.html"
host: "195.130.109.103"
basePath: "/v2"
tags:
- name: "DATE-A-A-S"
  description: "Everything about DATE-A-A-S Rest API"
  externalDocs:
    description: "Find out more"
    url: "https://codeberg.org/DATE-A-A-S/dateaas"
- name: "API"
  description: "Access to DATE-A-A-S"
- name: "user"
  description: "Operations about user"
schemes:
- "https"
- "http"
paths:
  /lab:
    post:
      tags:
      - "lab"
      summary: "Add a new lab to the server"
      description: ""
      operationId: "addLab"
      consumes:
      - "application/json"
      - "application/xml"
      produces:
      - "application/json"
      - "application/xml"
      parameters:
      - in: "body"
        name: "body"
        description: "Lab object that needs to be added to the server"

```

```

    required: true
    schema:
      $ref: "#/definitions/Lab"
  responses:
    405:
      description: "Invalid input"
  security:
  - lab_auth:
    - "write:labs"
    - "read:labs"
/lab/findByStatus:
  get:
    tags:
    - "lab"
    summary: "Finds Labs by status"
    description: "Multiple status values can be provided with comma
separated strings"
    operationId: "findLabsByStatus"
    produces:
    - "application/json"
    - "application/xml"
    parameters:
    - name: "status"
      in: "query"
      description: "Status values that need to be considered for filter
"
      required: true
      type: "array"
      items:
        type: "string"
        enum:
        - "available"
        - "pending"
        - "error"
        default: "available"
      collectionFormat: "multi"
    responses:
      200:
        description: "successful operation"
        schema:
          type: "array"
          items:
            $ref: "#/definitions/Lab"
      400:
        description: "Invalid status value"
  security:
  - lab_auth:
    - "write:labs"
    - "read:labs"
/lab/findByTags:
  get:
    tags:
    - "lab"
    summary: "Finds LABs by tags"
    description: "Multiple tags can be provided with comma separated
strings. Use\
  \ tag1, tag2, tag3 for testing."
    operationId: "findLabsByTags"
    produces:
    - "application/json"
    - "application/xml"
    parameters:

```

```
- name: "tags"
  in: "query"
  description: "Tags to filter by"
  required: true
  type: "array"
  items:
    type: "string"
  collectionFormat: "multi"
responses:
  200:
    description: "successful operation"
    schema:
      type: "array"
      items:
        $ref: "#/definitions/Lab"
  400:
    description: "Invalid tag value"
security:
- lab_auth:
  - "write:labs"
  - "read:labs"
deprecated: true
/lab/{labId}:
get:
  tags:
  - "lab"
  summary: "Find Lab by ID"
  description: "Returns a single lab"
  operationId: "getLabById"
  produces:
  - "application/json"
  - "application/xml"
  parameters:
  - name: "labId"
    in: "path"
    description: "ID of Lab to return"
    required: true
    type: "integer"
    format: "int64"
  responses:
    200:
      description: "successful operation"
      schema:
        $ref: "#/definitions/Lab"
    400:
      description: "Invalid ID supplied"
    404:
      description: "Lab not found"
  security:
  - api_key: []
post:
  tags:
  - "lab"
  summary: "Updates a Lab in the server with form data"
  description: ""
  operationId: "updateLabWithForm"
  consumes:
  - "application/x-www-form-urlencoded"
  produces:
  - "application/json"
  - "application/xml"
  parameters:
```

```
- name: "labId"
  in: "path"
  description: "ID of Lab that needs to be updated"
  required: true
  type: "integer"
  format: "int64"
- name: "name"
  in: "formData"
  description: "Updated name of the Lab"
  required: false
  type: "string"
- name: "status"
  in: "formData"
  description: "Updated status of the Lab"
  required: false
  type: "string"
responses:
  405:
    description: "Invalid input"
security:
- lab_auth:
  - "write:labs"
  - "read:labs"
delete:
  tags:
  - "lab"
  summary: "Deletes a Lab"
  description: ""
  operationId: "deleteLab"
  produces:
  - "application/json"
  - "application/xml"
  parameters:
  - name: "api_key"
    in: "header"
    required: false
    type: "string"
  - name: "labId"
    in: "path"
    description: "Lab id to delete"
    required: true
    type: "integer"
    format: "int64"
  responses:
    400:
      description: "Invalid ID supplied"
    404:
      description: "Lab not found"
  security:
  - lab_auth:
    - "write:labs"
    - "read:labs"
/store/inventory:
  get:
    tags:
    - "store"
    summary: "Returns Lab inventories by status"
    description: "Returns a map of status codes to quantities"
    operationId: "getInventory"
    produces:
    - "application/json"
    parameters: []
```

```
responses:
  200:
    description: "successful operation"
    schema:
      type: "object"
      additionalProperties:
        type: "integer"
        format: "int32"
    security:
      - api_key: []
/user/createWithList:
  post:
    tags:
      - "user"
    summary: "Creates list of users with given input array"
    description: ""
    operationId: "createUsersWithListInput"
    consumes:
      - "application/json"
    produces:
      - "application/json"
      - "application/xml"
    parameters:
      - in: "body"
        name: "body"
        description: "List of user object"
        required: true
        schema:
          type: "array"
          items:
            $ref: "#/definitions/User"
    responses:
      default:
        description: "successful operation"
/user/{username}:
  get:
    tags:
      - "user"
    summary: "Get user by user name"
    description: ""
    operationId: "getUserByName"
    produces:
      - "application/json"
      - "application/xml"
    parameters:
      - name: "username"
        in: "path"
        description: "The name that needs to be fetched. Use user1 for
testing. "
        required: true
        type: "string"
    responses:
      200:
        description: "successful operation"
        schema:
          $ref: "#/definitions/User"
      400:
        description: "Invalid username supplied"
      404:
        description: "User not found"
  delete:
    tags:
```

```

- "user"
summary: "Delete user"
description: "This can only be done by the logged in user."
operationId: "deleteUser"
produces:
- "application/json"
- "application/xml"
parameters:
- name: "username"
  in: "path"
  description: "The name that needs to be deleted"
  required: true
  type: "string"
responses:
  400:
    description: "Invalid username supplied"
  404:
    description: "User not found"
/user/login:
get:
  tags:
  - "user"
  summary: "Logs user into the system"
  description: ""
  operationId: "loginUser"
  produces:
  - "application/json"
  - "application/xml"
  parameters:
  - name: "username"
    in: "query"
    description: "The user name for login"
    required: true
    type: "string"
  - name: "password"
    in: "query"
    description: "The password for login in clear text"
    required: true
    type: "string"
  responses:
  200:
    description: "successful operation"
    headers:
      X-Expires-After:
        type: "string"
        format: "date-time"
        description: "date in UTC when token expires"
      X-Rate-Limit:
        type: "integer"
        format: "int32"
        description: "calls per hour allowed by the user"
    schema:
      type: "string"
  400:
    description: "Invalid username/password supplied"
/user/logout:
get:
  tags:
  - "user"
  summary: "Logs out current logged in user session"
  description: ""
  operationId: "logoutUser"

```



```
produces:
  - "application/json"
  - "application/xml"
parameters: []
responses:
  default:
    description: "successful operation"
/user/createWithArray:
  post:
    tags:
      - "user"
    summary: "Creates list of users with given input array"
    description: ""
    operationId: "createUsersWithArrayInput"
    consumes:
      - "application/json"
    produces:
      - "application/json"
      - "application/xml"
    parameters:
      - in: "body"
        name: "body"
        description: "List of user object"
        required: true
        schema:
          type: "array"
          items:
            $ref: "#/definitions/User"
    responses:
      default:
        description: "successful operation"
/user:
  post:
    tags:
      - "user"
    summary: "Create user"
    description: "This can only be done by the logged in user."
    operationId: "createUser"
    consumes:
      - "application/json"
    produces:
      - "application/json"
      - "application/xml"
    parameters:
      - in: "body"
        name: "body"
        description: "Created user object"
        required: true
        schema:
          $ref: "#/definitions/User"
    responses:
      default:
        description: "successful operation"
securityDefinitions:
  api_key:
    type: "apiKey"
    name: "api_key"
    in: "header"
  lab_auth:
    type: "oauth2"
    authorizationUrl: "https://195.130.109.103/oauth/authorize"
    flow: "implicit"
```

```
scopes:
  read:labs: "read your labs"
  write:labs: "modify labs in your account"
definitions:
  ApiResponse:
    type: "object"
  properties:
    code:
      type: "integer"
      format: "int32"
    type:
      type: "string"
    message:
      type: "string"
  Category:
    type: "object"
  properties:
    id:
      type: "integer"
      format: "int64"
    name:
      type: "string"
    xml:
      name: "Category"
  Lab:
    type: "object"
  required:
  - "name"
  properties:
    id:
      type: "integer"
      format: "int64"
    category:
      $ref: "#/definitions/Category"
    name:
      type: "string"
      example: "mtmproxy"
    tags:
      type: "array"
      xml:
        wrapped: true
      items:
        xml:
          name: "tag"
          $ref: "#/definitions/Tag"
    status:
      type: "string"
      description: "lab status in the server"
      enum:
      - "available"
      - "pending"
      - "error"
    xml:
      name: "Lab"
  Tag:
    type: "object"
  properties:
    id:
      type: "integer"
      format: "int64"
    name:
      type: "string"
```

```
xml:
  name: "Tag"
Order:
  type: "object"
  properties:
    id:
      type: "integer"
      format: "int64"
    labId:
      type: "integer"
      format: "int64"
    quantity:
      type: "integer"
      format: "int32"
    runDate:
      type: "string"
      format: "date-time"
    status:
      type: "string"
      description: "Order Status"
      enum:
        - "running"
        - "approved"
        - "scheduled"
    complete:
      type: "boolean"
xml:
  name: "Order"
User:
  type: "object"
  properties:
    id:
      type: "integer"
      format: "int64"
    username:
      type: "string"
    firstName:
      type: "string"
    lastName:
      type: "string"
    email:
      type: "string"
    password:
      type: "string"
    phone:
      type: "string"
    userStatus:
      type: "integer"
      format: "int32"
      description: "User Status"
xml:
  name: "User"
externalDocs:
  description: "Find out more about Swagger"
  url: "http://swagger.io"
```

Δ.1.2 HTML

Δείτε: <https://codeberg.org/DATE-A-A-S/dateaas/src/branch/main/api/html-client/index.html>

Δ.1.3 NodeJS

Δείτε: <https://codeberg.org/DATE-A-A-S/dateaas/src/branch/main/api/nodejs-client>

Γλωσσάρι

B

Bait-and-switch Τα μοτίβα δολώματος και εναλλαγής διαφημίζουν (Bait-and-switch patterns advertise) ένα δωρεάν (ή σε πολύ μειωμένη τιμή) προϊόν ή υπηρεσία που δεν είναι πλήρως διαθέσιμο ή διατίθεται σε μικρές ποσότητες. Αφού ανακοινωθεί η μη διαθεσιμότητα του προϊόντος, η σελίδα παρουσιάζει παρόμοια προϊόντα υψηλότερης τιμής ή χαμηλότερης ποιότητας.. βλέπε **Dark pattern**

C

Confirmshaming Το Confirmshaming χρησιμοποιεί τη ντροπή για να ωθήσει τους χρήστες να δράσουν. Για παράδειγμα, όταν οι ιστότοποι διατυπώνουν μια επιλογή για την απόρριψη ενός ενημερωτικού δελτίου ηλεκτρονικού ταχυδρομείου με τρόπο που ντροπιάζει τους επισκέπτες ώστε να αποδεχτούν. βλέπε **Dark pattern**

E

etcd Το **etcd** είναι ένας μόνιμος, ελαφρύς, κατανεμημένος χώρος αποθήκευσης δεδομένων κλειδιών-τιμών που έχει αναπτύξει η CoreOS. Αποθηκεύει αξιόπιστα τα δεδομένα διαμόρφωσης της συστάδας, αντιπροσωπεύοντας τη συνολική κατάσταση της συστάδας σε κάθε δεδομένη χρονική στιγμή. **3.4**

P

Pod Η βασική μονάδα χρονοπρογραμματισμού στο Kubernetes είναι ένα **pod**, το οποίο αποτελείται από ένα ή περισσότερα κοντέινερ που εγγυάται ότι βρίσκονται στον ίδιο κόμβο. Σε κάθε pod στο Kubernetes ανατίθεται μια μοναδική διεύθυνση IP εντός του cluster, επιτρέποντας στις εφαρμογές να χρησιμοποιούν θύρες χωρίς τον κίνδυνο σύγκρουσης. **3.4**

Privacy Zuckering Το "Privacy Zuckering" - που πήρε το όνομά του από τον συνιδρυτή του Facebook και διευθύνοντα σύμβουλο των Meta Platforms Mark Zuckerberg - είναι μια πρακτική που εξαπατά τον χρήστη ώστε να μοιράζεται περισσότερες πληροφορίες από αυτές που σκόπευε να μοιραστεί. Η Καλιφόρνια έχει εγκρίνει κανονισμούς που περιορίζουν αυτή την πρακτική από τις επιχειρήσεις στο νόμο περί προστασίας του απορρήτου των καταναλωτών της Καλιφόρνιας.. βλέπε **Dark pattern**

S

Swarm manager Ο διαχειριστής(manager) του Docker Swarm είναι ένα συστατικό του Docker Swarm, ενός εγγενούς εργαλείου ομαδοποίησης και ενορχήστρωσης για κοντέινερ Docker. Χρησιμεύει ως το επίπεδο ελέγχου για τη διαχείριση μιας συστάδας κόμβων Docker, συντονίζοντας την ανάπτυξη

και την κλιμάκωση εφαρμογών που περιέχουν κοντέινερ σε πολλούς κεντρικούς υπολογιστές. Ο διαχειριστής του Docker Swarm είναι υπεύθυνος για διάφορες εργασίες, όπως: **Διαχείριση συστάδας:** Χειρίζεται τη δημιουργία και τη διαχείριση ενός σμήνους Docker Swarm, επιτρέποντας στους κόμβους να ενταχθούν ή να αποχωρήσουν από το σμήνος δυναμικά. **Ενορχήστρωση υπηρεσιών:** Ο διαχειριστής συντονίζει την ανάπτυξη και την κλιμάκωση των υπηρεσιών, οι οποίες είναι οι ενθυλακωμένες εφαρμογές που εκτελούνται σε κοντέινερ Docker. Διανέμει τα κοντέινερ στους διαθέσιμους κόμβους με βάση τη διαθεσιμότητα των πόρων και τις εκτιμήσεις εξισορρόπησης φορτίου. **Υψηλή διαθεσιμότητα:** Ο διαχειριστής διασφαλίζει τη διαθεσιμότητα των υπηρεσιών με τον αυτόματο επαναπρογραμματισμό των "υπηρεσιών-σε-κοντέινερ" σε περίπτωση αποτυχίας ή αφαίρεσης κόμβων. Παρακολουθεί την υγεία των κόμβων και ανακατανέμει το φόρτο εργασίας ανάλογα με τις ανάγκες. **Εξισορρόπηση φορτίου:** Ο διαχειριστής εφαρμόζει μηχανισμούς εξισορρόπησης φορτίου για την κατανομή της εισερχόμενης κίνησης σε κοντέινερ που εκτελούνται σε διαφορετικούς κόμβους, βελτιστοποιώντας τη χρήση των πόρων και παρέχοντας ανοχή σε σφάλματα. **Ασφάλεια και έλεγχος ταυτότητας:** Επιβάλλει μηχανισμούς ελέγχου πρόσβασης και ελέγχου ταυτότητας για την προστασία της συστάδας Docker Swarm και των πόρων της. Πιστοποιεί τους κόμβους και επαληθεύει τις ταυτότητές τους για να διασφαλίσει την ασφαλή επικοινωνία εντός της συστάδας. Συνολικά, ο διαχειριστής του Docker Swarm λειτουργεί ως κεντρικό σημείο ελέγχου και συντονισμού για τη διαχείριση της ανάπτυξης, της κλιμάκωσης και της διαθεσιμότητας των εφαρμογών που περιέχουν κοντέινερ σε ένα σύμπλεγμα Docker Swarm. **3.4**

Swarm worker Ο εργάτης(worker) του Docker Swarm είναι ένα συστατικό του Docker Swarm, ενός εγγενούς εργαλείου ομαδοποίησης και ενορχήστρωσης για δοχεία Docker. Είναι υπεύθυνος για την εκτέλεση εφαρμογών που περιέχουν κοντέινερ ως μέρος μιας συστάδας Docker Swarm υπό την καθοδήγηση του διαχειριστή Swarm. Ο εργάτης του Docker Swarm εκτελεί τις ακόλουθες βασικές λειτουργίες: **Εκτέλεση των κοντέινερ:** Ο κόμβος εργάτης εκτελεί τα κοντέινερ Docker που αποτελούν τις υπηρεσίες που αναπτύσσονται στο σύμπλεγμα Docker Swarm. Παρέχει τους απαραίτητους πόρους και το περιβάλλον εκτέλεσης για την εκτέλεση των "υπηρεσιών-σε-κοντέινερ". **Εκτέλεση εργασιών:** Ο κόμβος εργάτης λαμβάνει οδηγίες από τον διαχειριστή Swarm για την εκτέλεση συγκεκριμένων εργασιών, όπως η εκκίνηση ή η διακοπή "υπηρεσιών-σε-κοντέινερ", η κλιμάκωση υπηρεσιών ή η ενημέρωση διαμορφώσεων. Εκτελεί αυτές τις εργασίες με την εκκίνηση ή τον τερματισμό των "υπηρεσιών-σε-κοντέινερ" ανάλογα με τις ανάγκες. **Ανακάλυψη υπηρεσιών:** Ο κόμβος εργάτης συμμετέχει στον μηχανισμό ανακάλυψης υπηρεσιών του Docker Swarm. Επικοινωνεί με άλλους κόμβους εργάτες και τον διαχειριστή για να ανακαλύψει υπηρεσίες και να λάβει τις απαραίτητες πληροφορίες για τη δρομολόγηση της κίνησης στα κατάλληλα κοντέινερ. **Εξισορρόπηση φορτίου:** Ο κόμβος εργάτης μπορεί επίσης να συμμετέχει στην εξισορρόπηση φορτίου, δεχόμενος την εισερχόμενη κίνηση και κατανέμοντάς την στα κοντέινερ που εκτελούνται στον εργάτη ή σε άλλους κόμβους της συστάδας. Αυτό βοηθά στη βελτιστοποίηση της χρήσης των πόρων και στην παροχή ανοχής σε σφάλματα. **Αναφορά πόρων:** Ο κόμβος εργάτης αναφέρει περιοδικά τη διαθεσιμότητα των πόρων του και την κατάσταση της υγείας του στον

διαχειριστή σμήνους. Οι πληροφορίες αυτές χρησιμοποιούνται από τον διαχειριστή για τη λήψη τεκμηριωμένων αποφάσεων σχετικά με την τοποθέτηση των "υπηρεσιών-σε-κοντέινερ" και την κατανομή των πόρων. **Συνοπτικά, ένας εργάτης του Docker Swarm είναι ένας κόμβος μέσα σε μια συστάδα Docker Swarm που εκτελεί κοντέινερ, εκτελεί τις εργασίες που του έχουν ανατεθεί, συμμετέχει στην ανακάλυψη υπηρεσιών και συμβάλλει στην εξισορρόπηση φορτίου. Διαδραματίζει κρίσιμο ρόλο στην κατανεμημένη εκτέλεση και διαχείριση των εφαρμογών με κοντέινερ που ενορχηστρώνει το Docker Swarm. 3.4**

Βιβλιογραφία

- Banerjee, Mishtu **and others** (2011). ?Quantifying Privacy Violations? in *Secure Data Management*: by editor Willem Jonker **and** Milan Petkovi . Berlin, Heidelberg: Springer Berlin Heidelberg, **pages** 1–17. isbn: 978-3-642-23556-6.
- Donenfeld, Jason (**january** 2017). ?WireGuard: Next Generation Kernel Network Tunnel? **indoi**: 10.14722/ndss.2017.23160.
- Dowling, Benjamin **and** Kenneth G. Paterson (2018). *A Cryptographic Analysis of the WireGuard Protocol*. Cryptology ePrint Archive, Paper 2018/080. <https://eprint.iacr.org/2018/080>. **url**: <https://eprint.iacr.org/2018/080>.
- Guardian, The (2018). *The Cambridge Analytica Files*. The Cambridge Analytica Files. <https://www.theguardian.com/news/2018/mar/22/facebook-gave-data-about-57bn-friendships-to-academic-aleksandr-kogan>. [Online; accessed 13.11.2022].
- Jumakhan, Haseebullah **and** Amir Mirzaeinia (**february** 2024). ?Wireguard: An Efficient Solution for Securing IoT Device Connectivity Regular Research Paper (CSCI-RTMC)? **in**.
- K.V, Charan, S. P Sharmila **and** A.S Manjunath (2014). ?Customizing AOSP for different embedded devices? **in** *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*: **pages** 259–264. doi: 10.1109/IndiaCom.2014.6828139.
- Krutz, Daniel E. **and others** (2015). ?A Dataset of Open-Source Android Applications? **in** *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*: **pages** 522–525. doi: 10.1109/MSR.2015.79.
- Luo, Qian **and others** (2022). ?Automatic Detection for Privacy Violations in Android Applications? **in** *IEEE Internet of Things Journal*: 9.8, **pages** 6159–6172. doi: 10.1109/JIOT.2021.3109785.
- Pingle, Bhargav, Aakif Mairaj **and** Ahmad Y. Javaid (2018). ?Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use? **in** *2018 IEEE International Conference on Electro/Information Technology (EIT)*: **pages** 0192–0197. doi: 10.1109/EIT.2018.8500082.
- Ponakala, Rajasekhar **and** Matthew N Dailey (2020). ?LineageOS Android Open Source Mobile Operating System: Strengths And Challenges? **in**.
- privacyinternational.org (2019a). *How Menstruation Apps Are Sharing Your Data*. **en**. **url**: <https://www.privacyinternational.org/long-read/3196/no-bodys-business-mine-how-menstruations-apps-are-sharing-your-data> (**urlseen** 09/09/2019).
- (2019b). *Research Methodology for Dynamic Analysis*. **en**. **url**: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%201%20Research%20Methodology%20for%20Dynamic%20Analysis.pdf> (**urlseen** 09/09/2019).
- (2019c). *Response from AppsFlyer*. **en**. **url**: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%206%20Response%20from%20AppsFlyer.pdf> (**urlseen** 09/09/2019).

- privacyinternational.org (2019d). *Response from CleverTap*. en. url: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%205%20Response%20from%20CleverTap.pdf> (**urlseen** 09/09/2019).
- (2019e). *Response from Facebook*. en. url: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%207%20Response%20from%20Facebook.pdf> (**urlseen** 09/09/2019).
- (2019f). *Response from Mi Calendario*. en. url: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%204%20Response%20from%20Mi%20calendario.pdf> (**urlseen** 09/09/2019).
- (2019g). *Response from Pinkbird*. en. url: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%203%20Response%20from%20Pinkbird.pdf> (**urlseen** 09/09/2019).
- (2019h). *Response from Plackal Tech*. en. url: <https://www.privacyinternational.org/sites/default/files/2019-09/Annex%202%20Response%20from%20Plackal%20Tech.pdf> (**urlseen** 09/09/2019).
- Regupathy, Rajaram (2014). ?Android Debug Bridge (ADB)? **in** *Unboxing Android USB: A Hands-On Approach with Real World Examples*: Berkeley, CA: Apress, **pages** 125–138. isbn: 978-1-4302-6209-1. doi: 10.1007/978-1-4302-6209-1_7. url: https://doi.org/10.1007/978-1-4302-6209-1_7.
- Song, Xin-long **and** Chung-Huang Yang (2017). ?Mobile Device Management System Based on AOSP and SELinux? **in** *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*: **pages** 417–420. doi: 10.1109/DSC.2017.93.
- Tran, Minh **and** others (2012). ?Tracking the Trackers: Fast and Scalable Dynamic Analysis of Web Content for Privacy Violations? **in** *Applied Cryptography and Network Security*: **by** editor Feng Bao, Pierangela Samarati **and** Jianying Zhou. Berlin, Heidelberg: Springer Berlin Heidelberg, **pages** 418–435. isbn: 978-3-642-31284-7.
- Xue, Lei **and** others (2019). ?NDroid: Toward Tracking Information Flows Across Multiple Android Contexts? **in** *IEEE Trans. Inf. Forensics Secur.*: 14.3, **pages** 814–828. doi: 10.1109/TIFS.2018.2866347. url: <https://doi.org/10.1109/TIFS.2018.2866347>.
- Yaacoub, Jean-Paul A. **and** others (2023). ?Ethical hacking for IoT: Security issues, challenges, solutions and recommendations? **in** *Internet of Things and Cyber-Physical Systems*: 3, **pages** 280–308. issn: 2667-3452. doi: <https://doi.org/10.1016/j.iotcps.2023.04.002>. url: <https://www.sciencedirect.com/science/article/pii/S2667345223000238>.