ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

DIPLOMAMUNKA

OE-NIK Hallgató neve: **Szakács Péter Barnabás**
2021/22/2 Hallgató törzskönyvi száma: T/008707/FI12904/N

Óbudai Egyetem
Neumann János Informatikai Kar
Biomatika és Alkalmazott Mesterséges Intelligencia Intézet

DIPLOMAMUNKA FELADATLAP

Hallgató neve:	**Szakács Péter Barnabás**
Törzskönyvi száma:	T/008707/FI12904/N
Neptun kódja:	[redacted]

A diplomamunka címe:

Online munkaszolgáltatások auditja
Audit of online work services

Intézményi konzulens:	Dr. Póser Valéria
Külső konzulens:	Pfeiffer Szilárd

Beadási határidő: 2022. május 15.

A feladat

Az otthoni munka terjedésével egyre fontosabb lesz a cég rendszereinek megvédése külső és belső veszélyektől. Egy hálózati tűzfal segítségével lehetséges a munkafolyamat során használt szolgáltatások megfigyelése és irányítása. A fontos megfigyelendő szolgáltatások közé tartoznak a legnépszerűbbek, mint a Google által nyújtott, és az online találkozókat, és munkát lehetővé tevő eszközök.
A hallgató feladata a szakdolgozatban már kiépített Zorp tűzfalat használó rendszer frissítése, a meglévő Google szolgáltatások kiegészítése úgy, hogy irányítható legyen a felhasználók tevékenysége. Továbbá online munka szoftverek felkutatása, a hivatalos API vizsgálata, és implementálása a meglévő rendszerbe.

A diplomamunkának tartalmaznia kell:

- a meglévő Zorp rendszer frissítését,
- a már beépített Google szolgáltatások irányítását az API segítségével,
- online munkát lehetővé tevő szolgáltatások vizsgálatát,
- a kiválasztott szolgáltatások tervezését és implementációját a meglévő rendszerbe hivatalos API segítségével,
- a változtatások, vizsgálat, tervezés és implementáció részletes dokumentációját.

ÓBUDAI EGYETEM * Neumann János Informatikai Kar * PH. 1. Budapest

.....
Dr. Eigner György
mb. intézetigazgató

A diplomamunka elévülésének határideje: **2024. május 15.**
(OE TVSz 55.§ szerint)

A diplomamunkát beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 20.. Május 04.

Szakàcs Péter

hallgató aláírása

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Szakács Péter Barnabás .... Neptun kód: [redacted] ..... Tagozat: Mérnökinformatikus MSc

Telefon: [redacted] Levelezési cím (pl: lakcím): [redacted] .....

Szakdolgozat / Diplomamunka[1] címe magyarul:

Online munkaszolgáltatások auditja.....

Szakdolgozat / Diplomamunka[2] címe angolul:

Audit of online work services .....

Intézményi konzulens: Dr. Póser Valéria .....

Külső konzulens: Pfeiffer Szilárd .....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.02.26.	Alapok megbeszélése	
2.	2021.03.22.	Szolgáltatás megbeszélés	
3.	2021.04.15.	Zorp megbeszélés, egyeztetés	
4.	2021.05.14.	Dokumentum ellenőrzése	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.05.14 .....

.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Szakács Péter Barnabás....

Neptun kód: [redacted].....

Tagozat: Mérnökinformatikus MSc

Telefon: [redacted]

Levelezési cím (pl: lakcím): [redacted].....

Szakdolgozat / Diplomamunka[1] címe magyarul:

Online munkaszolgáltatások auditja.....

Szakdolgozat / Diplomamunka[2] címe angolul:

Audit of online work services .....

Intézményi konzulens: Dr. Póser Valéria.....

Külső konzulens: Pfeiffer Szilárd.....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.07.21.	Implementáció elkezdése	
2.	2021.08.13.	Naplózás megbeszélése	
3.	2021.11.12.	Tesztek megvizsgálása, tiltásról megbeszélés	
4.	2021.12.06.	A dolgozat felépítésének átbeszélése	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.12.08.....

.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Szakács Péter Barnabás

Neptun kód: [redacted]

Tagozat: Mérnökinformatikus MSc

Telefon: [redacted]

Levelezési cím (pl: lakcím): [redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

Online munkaszolgáltatások auditja

Szakdolgozat / Diplomamunka[2] címe angolul:

Audit of online work services

Intézményi konzulens: Dr. Póser Valéria

Külső konzulens: Pfeiffer Szilárd

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.01.11.	Tiltás kezdésének megbeszélése	
2.	2022.02.02.	Tiltás bemutatása	
3.	2022.02.24.	Programjavítások bemutatása	
4.	2022.04.27.	Program áttekintése, jóváhagyás	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2022.05.10

.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

Tartalom

ÖSSZEFOGLALÓ A TÉMÁRÓL

Az otthoni munka fontosságával egyre több biztonsági kérdés merül fel. Egyre fontosabb lesz a belső veszélyek előzetes elhárítása, nem csak a cég alkalmazottainak megfigyelése. Valamint új nagy szintű veszély az online konferencia szolgáltatások, mint például a Zoom, Microsoft Teams, és Google Meet. A megfigyelésükhöz lehetséges eszköz egy tűzfal, aminek előnye, hogy a munkavállalók belső tevékenységeit is lehet vele követni.

A szakdolgozatomban használt Zorp tűzfal, és az abban írt modul továbbfejleszthető, hogy alkalmas legyen az új követelmények elvállalására.

A feladat az eddig figyelt szolgáltatások megvizsgálása, hogy biztos legyek benne, hogy minden a legfrissebb verziót naplózza, és utána eldönteni, hogy a szolgáltatáson belül melyik műveleteket akadályozza meg a tűzfal.

Az online konferenciát lehetővé tevő szolgáltatások közül a feladat megvizsgálni a népszerűbbeket, valamint a beépítésük lehetőségét a tűzfal moduljába. Kiválasztani a megfelelőket, és a hivatalos dokumentáció segítségével implementálni a megfigyelést a modulba.

1. A PROBLÉMA FELVEZETÉSE

A COVID-19 vírus terjedésével egyre inkább előtérbe került a távoli, otthoni munka, és az azt támogató szolgáltatások fontossága. A drasztikus növekedés mellett a jövőre nézve is marad az otthoni munka, ezt jelzi az is, hogy a cégek fontolgatják, hogy bizonyos pozíciókat áthelyeznek végleg távolira. Viszont ennek köszönhetően hirtelen sokkal fontosabb lett a cég biztonsága.[1]

Az egyik legnagyobb veszély a cégekre nézve a saját felhasználói. Akár direkt, akár tudatlanul, az alkalmazottak nagy veszélynek teszik ki a munkahelyüket személyes, otthoni eszközök használatával. Ilyen veszélyek például a vírusok/káros programok munkahelyi rendszerbe kerülése, adatok kiszivárgása, vagy nem engedélyezett felhasználók bejutása a cég belső rendszerébe.[1]

Fontosabb veszélyek közé sorolják a fájlmegosztást és az üzenetküldést, de a jelenlegi helyzet miatt a legkockázatosabb eszközök közé került a videó konferenciát lehetővé tevő szolgáltatások. Ezeknek a biztosítására léteznek eszközök, mint például tűzfalak és virtuális magánhálózatok (VPN), de a változások miatt a bevált módszerek már nem feltétlenül megfelelőek.[1]

A „Google szolgáltatások auditja” nevű előző munkámban a Zorp hálózati tűzfallal dolgoztam, de az ott létrehozott, a Google szolgáltatásait naplózó modul már nem alkalmas a jelenlegi helyzet kezelésére. Naplózza az elterjedt Google szolgáltatások használatát, de nem előz meg semmit. Otthoni munkával ez már nem megfelelő, mivel semmi nem akadályozza meg a munkavállalót abban, hogy egyszerűen figyelmen kívül hagyja a biztonsági alkalmazottak figyelmeztetését. Fizikailag már nem lehet megkeresni az alkalmazottat, és mire a főnökét figyelmeztetjük, addigra a kár már megtörtént.

Ennek a javítása a cél a modul bővítésével, mivel a Zorp tűzfal képes a rajta átmenő forgalom megakadályozásra is.

2. A KORÁBBI BSC-S SZAKDOLGOZATOM ÖSSZEFOGLALÓJA

Először szükséges egy összefoglaló a korábbi munkámról, a szakdolgozatomról, mivel ez a folytatása lesz. Nem fogom annyira részletesen leírni, mint az előző dokumentációban, de a cél az, hogy egy alap kép legyen róla.

A platform, amin dolgoztam a szakdolgozatom során, a Zorp tűzfal. Ez egy proxy tűzfal, amit egy Ubuntu virtuális szerverre telepítettem, és beáll a böngésző és az internet közé, figyelve a rajta átmenő forgalmat. A tűzfal viselkedését Pythonban írt modulokkal határoztam meg.

A szakdolgozatom során egy ilyen modult hoztam létre, ami a Google által nyújtott szolgáltatásokból ötöt figyelt: Google Drive, Calendar, Gmail, People, és Docs. Ezeknek az API-jait vizsgáltam.

Ezek sok olyan műveleteket tartalmaztak, ami számomra nem volt érdekes, így szűrtem a szükséges műveleteket. Ezeket naplóztam a saját modulomban, aminek a neve AuditModule.py volt. Ebben a modulban a legfontosabb osztály a *LoggerClass* volt, ahol a tényleges szétválogatás és naplózás történt a HTTP request headerje alapján. A modul tesztelését a böngészőn elérhető, Google által nyújtott eszközzel tettem meg.

A Zorp rendszer négy fő részből áll:

- Zóna: IP alhálózatok meghatározására használják. Ez a tényleges vállalati hálózatok szempontjából lényegtelen, mert a Zorp saját lista alapján kezeli a zónákat. Amíg több alhálózat egy zónában van, addig a tűzfal szerint az egy alhálózat. Azonos alhálózat nem lehet két külön zónában.
- Szabály: Zónák alapján meghatározza, hogy ki milyen szolgáltatáshoz férhet hozzá, milyen műveletet hajthat végre. Ezt a paramétereken keresztül kell megadni, és nem a legelső, hanem a legpontosabb szabályt fogja alapnak venni. Ezt a legszűkebb hozzáférési lista, és a legrészletesebb paraméterlista határozza meg.
- Szolgáltatás: A szolgáltatás határozza meg, hogy a forgalomban lévő csomaggal mi történik. Lehet visszautasítás és elfogadás is, illetve képes meghívni egy proxy-t külön vizsgálatra is. Több fajta szolgáltatás létezik, de engem csak a Service érdekel, ami alkalmazásszinten továbbítja a csomagot, így ideális szűréshez, naplózáshoz és tiltáshoz.

- Proxy: Minden forgalommódosításra, megfigyelésre használt eszközök. Több van belőle, pl. FTP, SMTP, de engem csak a HTTP érdekel. C++-ban íródtak, de Pythonnal ki lehet őket bővíteni. Én az előző munkámban proxykat használtam.

3. A MEGLÉVŐ SZOLGÁLTATÁSOK ELLENŐRZÉSE

Korábbi szakdolgozatom bemutatását követően rátérnék a jelenlegi munkámra, amiben a Microsoft Teams és a Zoom funkcióit fogom elemezni a korábban használt Google API-okon kívül. De előtte szükséges újra elemezni a szakdolgozatomban használt Drive, Gmail, Calendar, Docs és People szolgáltatásokat a változások dokumentálása érdekében.

A szolgáltatások felügyelése és irányítása attól függ, hogy a Zorp tűzfal a fontos metódusokat figyelje. Ha nem egyezik az API-ban hívott metódussal, akkor nem fog működni se a monitorozás, se a kontroll.

Ez az egész diplomamunka alatt egy fontos szempont lesz, mivel a Google hajlamos gyakran frissíteni az alkalmazásait, kikényszerítve a tűzfal változtatását.

A Google különböző szolgáltatásai HTTP requestek alapján dönti el, hogy melyik műveletet hívta a felhasználó. A requestet felépítő HTTP metódusból és URI-ból egyértelműen el lehet dönteni a metódust. Az adatlekéréseket és a GET metódusokat továbbra se figyelem, mert biztonsági szempontból ezek a metódusok nem kritikusak.[2]

A továbbiakban a Google esetén feltételezem, hogy a hívott URI eleje a https://www.googleapis.com. Ezt a dokumentumban később már nem fogom kiírni.[2]

3.1. DRIVE

Az első API, aminek a változásait meg fogom vizsgálni, az a Google Drive. A vizsgált műveleteknek frissnek kell lenniük, ha normális működést szeretnék elérni. A Drive esetében az egyik legnagyobb változás a 2. verzióból a 3.-ra váltás, de ezen kívül vannak új műveletek is, amit dokumentálnom kell.

3.1.1. FÁJLOK

Az első a fájlok között a *delete* metódus, ami a régi /drive/v2internal/files/*fileId* helyett az újabb /drive/v3/files/*fileId* URI-t használja. A HTTP metódus nem változott, továbbra is DELETE. A piros színnel jelölt, dőlt betűstílussal írt szavak változók, amik a tényleges használat során nem a dokumentumban leírt szó lesz, hanem egy másik érték. Ez a továbbiakban is így lesz, de nem fogom feltüntetni minden alkalommal.[2]

Az *emptyTrash* metódus is változott, a régi /drive/v2internal/files/trash helyett az új /drive/v3/files/trash URI-t használja, a HTTP metódus továbbra is DELETE. Az *untrash* metódus már nem létezik a legfrissebb verzióban.[2]

Az *export* a /drive/v2internal/files/*fileId*/export URI helyett az újabb drive/v3/files/*fileId*/export-at használja.[2]

A *copy* szintén a régi /drive/v2internal/files/*fileId*/copy helyett a /drive/v3/files/*fileId*/copy-t használja a fájlok másolásához.[2]

3.1.2. KOMMENTEK

A modulban alapból a v3-as verzió van beépítve a kommentek figyeléséhez, így nem kell változtatni ezen a részen.[2]

3.1.3. VÁLASZOK

A válaszoknál történt frissítés a régi verzióról az újra, így itt is át kell írni megfelelő metódusokat.

Ilyen a *create*, aminél a HTTP metódus ugyanaz, de a régi /drive/v2internal/files/*fileId*/comments/*commentId*/replies URI-t lecserélték /drive/v3/files/*fileId*/comments/*commentId*/replies-ra.[2]

A *delete* és *update* metódusok ugyanazt a linket használják, amit frissíteni kell /drive/v2internal/files/*fileId*/comments/*commentId*/replies/*replyId*-ról /drive/v3/files/*fileId*/comments/*commentId*/replies/*replyId*-ra.[2]

3.1.4. FÁJLVERZIÓK

A fájlverziók módosítását és törlését frissíteni kell. Az HTTP metódus különbözik a kettőnél, de nem változott, viszont az URI igen. /drive/v2internal/files/*fileId*/revisions/*revisionId*-t kell kicserélni /drive/v3/files/*fileId*/revisions/*revisionId*-ra.[2]

3.1.5. ENGEDÉLYEK

Az engedélyek egy olyan rész, amit a szakdolgozatban írt modulban nem figyeltem, de fontos, ezért hozzáadom a megfigyelt műveletek listájához.

A *create* metódus létrehoz egy hozzáférést egy adott fájlhoz vagy drive-hoz. Post igét használ, és az URI-ja /drive/v3/files/*fileId*/permissions.[2]

A *delete* törli az Id-vel jelzett hozzáférést. DELETE HTTP metódust küld /drive/v3/files/*fileId*/permissions/*permissionId* linkkel.[2]

Frissítés az *update* metódussal történik, amihez PATCH /drive/v3/files/*fileId*/permissions/*permissionId* requestet kell küldeni.[2]

3.2. GMAIL

A következő Google API a Gmail. Ezt is meg kell vizsgálni új metódusok miatt, a munka teljessége érdekében.

A Gmail API-ja nagyrészt ugyanaz maradt, de van benne pár új funkció, amit érdemes auditálni, mert potenciális biztonsági kockázatot jelent egy vállalat számára.

Az alap URI https://gmail.googleapis.com, ezt nem fogom feltüntetni az egyedi példákban.[3]

3.2.1. BEÁLLÍTÁSOK

A beállítások között figyelnem kell azokat a frissítéseket, amik veszélyesek. Ilyen az *updateAutoForwarding*, ami frissíti az automatikus email tovább küldést. PUT /gmail/v1/users/*userId*/settings/autoForwarding a request amit küld, ahol a userId az email cím, ami küldte a kérést.[3]

Az Imap és Pop frissítése az *updateImap* és *updatePop* metódusokkal történik, és mindkettő a PUT igét használja. A linkek az Imap számára /gmail/v1/users/*userId*/settings/imap, és a Popnak /gmail/v1/users/*userId*/settings/pop.[3]

A vakáción beállítható automata választ az *updateVacation* metódussal kell frissíteni. Ez is PUT-ot használ, és az URI-ja /gmail/v1/users/*userId*/settings/vacation.[3]

A beállítások része még a sendAs email címek módosítása. Itt egy új metódus van, a *verify*. A request-je /gmail/v1/users/*userId*/settings/sendAs/*sendAsEmail*/verify. A sendAsEmail az ál-email cím, ami a vevőnek fog feltűnni küldőként.[3]

3.2.2. SZÁLAK

A szálak egy új gmail funkció, ami összesíti az összefüggő emaileket egy folytonos beszélgetéssé, az egyszerűbb átláthatóság miatt. Ezekhez ugyanúgy tartoznak címkék, mint az emailekhez, de velük ellentétben nem lehet szálakat létrehozni, csak leveleket beleszúrni.[4]

Négy metódus érdekes számomra. Az első a *delete*, ami végleg, visszavonthatatlanul törli az id által adott szálat. A requestje DELETE /gmail/v1/users/*userId*/threads/*id*.[5]

Módosítani a szálhoz rendelt címkéket a *modify* metódussal kell, ami POST igét használ, és az URI-ja /gmail/v1/users/*userId*/threads/*id*/modify.[5]

A kukába helyezés, és onnan kivevés egy hasznosabb funkció, mint az azonnali törlés. A *trash* és *untrash* metódusok POST-ot használnak, és a behelyezés linkje /gmail/v1/users/*userId*/threads/*id*/trash, a kivétel linkje /gmail/v1/users/*userId*/threads/*id*/untrash.[5]

3.3. CALENDAR

A Calendar API vizsgálata során arra az eredményre jutottam, hogy a Google naptárja, és a használatát segítő API, a Calendar API a Drivehoz hasonlóan a harmadik verzióján van, de mivel ezt már a szakdolgozatom elkészítése előtt frissítették, ezért nem szükséges itt semmilyen művelet. Már a harmadik verzió van implementálva.

3.4. PEOPLE

A következő vizsgálandó szolgáltatás, a People API egy új, fejlesztés alatt álló modern API, amit a régi Contacts API leváltására találtak ki. A metódusok nem változtak, viszont lehetővé tették a csoportos műveleteket.[6]

Az URI-k első része a https://people.googleapis.com, ezt a továbbiakban nem fogom feltüntetni.[6]

A csoportos létrehozáshoz POST /v1/people:batchCreateContacts HTTP requestet kell elküldeni.[6]

Tömegesen törölni ugyanolyan POST igével elküldött /v1/people:batchDeleteContacts metódussal lehet, és frissíteni a szintén POST igés /v1/people:batchUpdateContacts kéréssel lehet.[6]

3.5. DOCS

A Google Docs API-ja három metódusból áll, a lekérésből, frissítésből és létrehozásból. Egyiken se módosítottak semmit, így ezen nem kell dolgoznom.[7]

4. ONLINE KONFERENCIA SZOLGÁLTATÁSOK

A következő fejezetben rátérek az online munkaszolgáltatások vizsgálatára, hogy rendesen tudjam implementálni a naplózást és a tiltást a szolgáltatásokra. Ezekből a kettő fontos a Microsoft Teams és a Zoom., amiket a következő szekciókban fogom megvizsgálni.

A szempontok ugyanazok, amiket eddig használtam. Azok a műveletek lesznek dokumentálva, amik nem adatlekérő műveletek, és módosítanak valamit, legyen az naptár törlése, vagy beszélgetéscsatorna létrehozása.

4.1. MICROSOFT TEAMS

Az online munkát lehetővé tévő eszközök közül az egyik legnépszerűbb a Microsoft Teams. A konferenciák mellett támogat fájlmegosztást, hozzászólásokat, levelezést, és még egyéb szolgáltatásokat, ami egy népszerű eszközzé teszi a jelenlegi távoli oktatásban és munkában. Alkalmazások hozzárendelését is engedélyezi a különböző csapatok számára. Emiatt egy olyan munkavállaló, aki kárt akar okozni a vállalat rendszerében, nagyon sok eszközt kap a kezébe.[8]

Az egész rendszer a Microsoft Graph API-t használja, így a http linkek kezdete https://graph.microsoft.com/v1.0. Ezt a továbbiakban csak akkor fogom feltüntetni, ha valahol változik.[8]

A két fő rész API ami érdekel az a Users és a Teamwork. A Users monitorozása lehetővé teszi a munkavállalók megfigyelését, és maga a Microsoft Teams a Graph API Teamwork részét használja. A dokumentációban minden szükséges megtalálható ez alatt, így csak ezt a kettőt fogom figyelni.[9]

4.1.1. USERS

4.1.1.1. USER

Egy felhasználó létrehozásához POST /users requestet kell küldeni.[10]

A törléshez szükséges az azonosítója a felhasználónak, amit a Microsoft id-ként, illetve user-id-ként jelöli. Ezt a rendszer ki fogja cserélni a megfelelő értékkel, így ezt a szokásos piros színnel, dőlt betűvel jelölöm. A küldött request a DELETE /users/*user-id*.[11]

A frissítés ugyanazt az URI-t használja, de PATCH-et használ.[12]

4.1.1.2. APP ROLE

A felhasználók szerepét különböző alkalmazásokban lehet eltávolítani, és hozzáadni. A hozzáadás requestje POST /users/*user-id*/appRoleAssignments.[13]

A törléshez kell a szerep id-je, így a küldött kérés DELETE /users/*id*/appRoleAssignments/*id*.[14]

4.1.1.3. CALENDAR

A naptárak létrehozásánál a Microsoft már több linket is tud használni egy requesthez, így azokat is figyelni kell, viszont különféle kategóriákban megegyeznek a kérések, így azt többször nem fogom megemlíteni a dokumentumban. Mindkettő a POST HTTP metódust használja, de az URI-k /me/calendars, és /users/*user-id*/calendars.[15]

A naptárcsoportok létrehozásához szintén POST HTTP metódus kell, de a linkek /me/calendarGroups, és /users/*user-id*/calendarGroups.[16]

Az események létrehozásához POST HTTP metódust használ az összes, de hat különböző URI-t lehet hívni. Lehet a felhasználón keresztül a /me/events, /users/*user-id*/events-el, a naptáron keresztül a /me/calendar/events, /users/*user-id*/calendar/events, /me/calendars/events, és /users/*user-id*/calendars/events-el.[17]

4.1.1.4. CONTACT

A kapcsolatokhoz a kapcsolat létrehozása érdekel. Ezt lehet az alap mappában, illetve a felhasználó által létrehozott mappában. POST HTTP metódussal négy URI-t használ, az alap mappához /me/contacts, illetve /users/*user-id*/contacts kell, a saját mappához pedig /me/contactFolders/*id*/contacts, és /users/*user-id*/contactFolders/*id*/contacts.[18]

4.1.1.5. MAIL

Az üzenetek létrehozásánál először egy piszkozat készül, amit egy külön metódussal kell elküldeni. [19]

A levél létrehozásához POST HTTP metódussal kell küldeni négy URI-t, ami attól függ, hogy a felhasználó melyik mappába akarja helyezni. Az alapba a /me/messages, és /users/*user-id*/messages. Saját mappába a /me/mailFolders/*id*/messages és /users/*user-id*/mailFolders/*id*/messages link kell. [20]

Az üzenet küldés után az elküldött elemekbe megy alapból. Itt nincs mit állítani, így csak két metódus létezik. POST /me/sendMail és /users/*user-id*/sendMail.[21]

4.1.1.6. TEAMWORK

A csapatmunkánál a felhasználó számára kirendelt alkalmazásokról van szó. Itt lehet hozzáadni, frissíteni, illetve törölni. Alkalmazás hozzáadásához POST /users/*user-id*/teamwork/installedApps request kell. [22]

Frissítéshez POST /users/*user-id*/teamwork/installedApps/*id*/upgrade-et szükséges küldeni, és alkalmazás eltávolításához DELETE /users/*user-id*/teamwork/installedApps/*id*.[23][24]

4.1.2. TEAMWORK

4.1.2.1. TEAM

A Team a csapatok és tagjaik vezérlésére használt. Létrehozni csapatot alapok nélkül a POST /teams requesttel kell, és egy meglévő csoportból pedig a /groups/*id*/team URIval. Frissítéshez a PATCH /teams/*id*-t kell küldeni, ahol az id a csapat azonosítója.[25] [26] [27]

Törlés máshogy működik, mivel a Teams a Microsoft 365 csapatokon alapul, ezért közvetlenül azt kell törölni a DELETE /groups/*id*-vel.[28]

A Teamhez tagot hozzáadni a POST /teams/*id*/members-el kell, frissíteni pedig PATCH /teams/*id*/members/*membership-id*-vel, ahol a membership-id a szükséges tag azonosítója. A törléshez az URI megegyezik, de a HTTP metódus a DELETE.[29] [30] [31]

A csapatot lehetséges archívumba helyezni, ha már nem szükséges. Ilyenkor nem lehet már módosítani, üzenetet küldeni, de a tartalma megmarad, és visszavonható a folyamat, ha még is kellene a Team. Az archiválás a POST /teams/*id*/archive-al történik, és az archiválás visszavonása a POST/ teams/*id*/unarchive requesttel.[32] [33]

Lehetséges a csapatot, vagy valamelyik részét klónozni. Ez a művelet a hozzárendelt Microsoft 365 csoportot is klónozza. POST /teams/*id*/clone a szükséges request.[34]

4.1.2.2. GROUP

A Teamshez hozzárendelt csoportot is lehet vezérelni, így azt is figyelem.

Létrehozni a POST /groups-al lehet, frissítéséhez pedig szükséges az ID, így a request PATCH /groups/*id*. Törléshez a PATCH HTTP metódust le kell cserélni DELETE-re. [35] [36] [37]

A csoporthoz tagot hozzáadni más módszerrel kell mint a Teamhez. A requestje POST /groups/*id*/members/$ref. A tag eltávolításához szükséges az azonosító, így a requestje DELETE /groups/*id*/members/*id*/$ref. [38] [39]

A csoportoknak vannak tulajdonosai. Egyet hozzáadni a POST /groups/*id*/owners/$ref-el kell, és eltávolítani a DELETE /groups/*id*/owners/*id*/$ref-el.[40] [41]

Lehetséges véglegesen törölni egy csoportot a DELETE /directory/deletedItems/*id*-vel. Ez jelenleg csak csoportokra, felhasználókra és alkalmazásokra. Szükséges hozzá adminisztrátori jogkör, de ugyanúgy veszélyes a vállalatra nézve.[42]

4.1.2.3. CHANNEL

A csatornák a csapattársakkal folytatott beszélgetések helye, és általában minden projektnek van külön csatornája. [43]

Egy csatorna létrehozásához a POST /teams/*id*/channels requestet kell küldeni. Frissíteni PATCH /teams/*id*/channels/*id*-vel lehet, a törléshez pedig a PATCH HTTP metódust le kell cserélni DELETE-re.[44] [45] [46]

A csatornához tagot hozzáadni a POST /teams/*id*/channels/*id*/members-el kell. Frissíteni a PATCH /teams/*id*/channels/*id*/members/*id* requesttel kell, és a törléshez az előzőekhez hasonlóan a PATCH-et le kell cseréni DELETE-re.[47] [48] [49]

Lehetséges az üzenet küldést is monitorozni, de az nagy részt nem szükséges, mert ha bárki újat hozzá akarnának adni a csatornához azt a tűzfal figyelni fogja, legyen legális az vagy nem.

4.1.2.4. CHAT

Egy chatet létrehozni két ember, vagy egy ember és egy alkalmazás között POST /chats requesttel kell. Frissíteni PATCH /chats/*id*-vel kell.[50] [51]

Egy felhasználót a chathez hozzáadni POST /chats/*id*/members-el kell, eltávolításhoz pedig kell az azonosító, így a requestje DELETE /chats/*id*/members/*id*-vel lehet.[52] [53]

4.1.2.5. APP CATALOG

Az alkalmazásokat ki lehet adni egy katalógusba, nyilvánossá tenni. Ez elég komoly problémát tud jelenteni, ha nem figyelik, hogy ki tette.

Kiadni a POST /appCatalogs/teamsApps requesttel kell. Lehetséges olyat is kiadni, amihez kell először egy review, de ez a tűzfal szempontjából nem jelent semmi különbséget.[54]

Frissíteni a POST /appCatalogs/teamsApps/*id*/appDefinitions-el lehet a már kiadott alkalmazást.[55]

Törölni két félét lehet. A már kiadott alkalmazásokat a DELETE /appCatalogs/teamsApps/*id* requesttel lehet, és a még nem jóváhagyott alkalmazásokat a DELETE appCatalogs/teamsApps/*id*/appDefinitions/*id*-vel.[56]

4.1.2.6. APP

Az alkalmazásokat három részre lehet szétosztani. Csapatban található alkalmazások, személyhez hozzárendelt alkalmazások és chaten található alkalmazások. A személyes alkalmazásokat nem fogom itt megvizsgálni, mert a 3.1.1.6-os részben már megtettem.

A csapathoz rendelt alkalmazások telepítéséhez a POST /teams/*team-id*/installedApps requestet kell használni. Frissítéshez POST /teams/*team-id*/installedApps/*app-installation-id*/upgrade, és törléshez DELETE /teams/*team-id*/installedApps/*app-installation-id* szükséges.[57] [58] [59]

Chathez hozzáadni az alkalmazást a POST /chats/*chat-id*/installedApps-al kell, frissíteni pedig POST /chats/*chat-id*/installedApps/*app-installation-id*/upgrade requesttel. Törölni egy alkalmazást a DELETE /chats/*chat-id*/installedApps/*app-installation-id* kéréssel lehet.[60] [61] [62]

4.1.2.7. CALENDAR

A naptáraknál az események figyelése a fő cél, és az azokon történt változtatások.

A létrehozáshoz két URI verzió létezik, POST /groups/*id*/events és POST /groups/*id*/calendar/events.[63]

Frissítéshez elég sok URI létezik, de mind a PATCH HTTP metódust használja. Ami engem érdekel az a csoportos naptárak frissítése, ami a /groups/*id*/events/*id* és /groups/*id*/calendar/events/*id* használatával történik. A törlés ugyanezeket az URI-kat használja, de a DELETE HTTP metódussal.[64] [65]

Lehet 3 MB-ig mellékletet hozzáadni egy eseményhez. Ehhez az összes request a POST HTTP metódust használja, de az URIból sok van, és szituációfüggő. A felhasználó alap mappájában

lévő eseményhez hozzáadni a /me/events/*id*/attachments, /users/*id*/events/*id*/attachments, /me/calendar/events/*id*/attachments és /users/*id*/calendar/events/*id*/attachments-el kell.[66]

A felhasználó alap naptárcsoportjához tartozó eseményhez a /me/calendars/*id*/events/*id*/attachments és /users/*id*/calendars/*id*/events/*id*/attachments URIk használatával lehet hozzáadni mellékletet.[66]

Egy meghatározott naptárcsoporthoz tartozó eseményhez mellékletet a /me/calendargroups/*id*/calendars/*id*/events/*id*/attachments és /users/*id*/calendargroups/*id*/calendars/*id*/events/*id*/attachments linkek segítségével lehet hozzáadni.[66]

4.2. ZOOM

A Zoom a Microsoft Teamshez hasonlóan egy online konferencia eszköz, hasonló funkciókkal, de telefon támogatással, amit lehetséges az APIn keresztül vezérelni. Az API a REST architektúrán épül, mint az eddigi API-k, így ugyanazokat a HTTP requesteket kell összegyűjteni.[67]

Az összes URL kezdő része https://api.zoom.us/v2/. Ezt a továbbiakban nem fogom feltüntetni.[67]

4.2.1. ACCOUNTS

A Zoom csak mester accountoknak engedi, hogy kezeljenek al-accountokat. De veszélyt okozhat, ha egy mester account rossz kézbe kerül, ezért ezeket is figyelni kell.[68]

Egy al-account létrehozásához a POST /accounts requestet kell küldeni. Lehetséges egy al-accountot leválasztani a mester accountról a DELETE /accounts/*id* kéréssel. Ez az accountot nem törli, de nem lesz összekötve a volt mester accounttal a továbbiakban. [68] [69]

Lehetséges egy mester accountról rákényszeríteni beállításokat az alárendelt accountokra a PATCH /accounts/*id*/lock_settings kéréssel. Ez rossz kezekben nagy problémát tud okozni egy cégnek.[70]

Egy al-account tulajdonosát át lehet állítani egy másik felhasználóra. Ezt meg tudja tenni egy mester account a PUT /accounts/*id*/owner-el.[71]

4.2.2. CHAT CHANNELS

Csatornákat lehet account és felhasználó szinten kezelni.[72]

Felhasználó szintű csatornát létrehozni a POST /chat/users/*id*/channels kéréssel lehet. A név frissítése a PATCH /chat/channels/*id*-vel történik, és törölni ugyanazzal az URL-el kell, de DELETE HTTP metódussal. [72] [73] [74]

Egy csatornában lehet egy vagy több tag. Ezeket a DELETE /chat/channels/*id*/members/*id* kéréssel lehet törölni. Egy felhasználó a POST /chat/channels/*id*/members/me requesttel tud belépni az azonosított csatornába, és kilépni ugyanazzal az URL-el tud, de DELETE HTTP metódussal.[75] [76] [77]

Egy account szintű csatornát csak frissíteni és törölni lehet. Frissíteni a PATCH /chat/users/*id*/channels/*id*-vel történik, és törlés ugyanezzel az URL-el, csak DELETE-el.[78] [79]

Meghívni egyszerre öt tagot lehet egy POST /chat/users/*id*/channels/*id*/members kéréssel, de csak a felhasználó contact listájáról. Egy tag törléséhez kell az id, így a request DELETE /chat/users/*id*/channels/*id*/members/*id* lesz.[80] [81]

4.2.3. CLOUD RECORDINGS

A konferenciákról készített rögzítéseket lehet egyesével, illetve egyszerre törölni. Egyszerre a DELETE /meetings/*id*/recordings-al kell, és egyesével szükséges az ID, így a kérés DELETE /meetings/*id*/recordings/*id* lesz.[82] [83]

A rögzítés megtekintéséhez regisztrálni kell egy embert, ezt a POST /meetings/*id*/recordings/registrants kéréssel lehet megtenni.[84]

4.2.4. GROUPS

A Zoomban a csoportok accountok alá vannak rendelve. Ezeket a csoportokat a POST /groups kéréssel lehet létrehozni. Frissíteni az account alatt található csoportot a PATCH /groups/*id*-vel lehet, törölni pedig a DELETE /groups/*id*-vel.[85] [86] [87]

Egy csoporthoz tagokat a POST /groups/*id*/members requesttel lehet hozzáadni. Tagokat frissíteni illetve törölni egyszerre nem lehet, egyesével kell. Frissíteni PATCH /groups/*id*/members/*id*-vel kell, és törölni ugyanazzal az URL-el, de DELETE HTTP metódussal.[88] [89] [90]

4.2.5. MEETINGS

Konferenciákat egy felhasználó naponta maximum százszor tud elindítani API requesten keresztül, POST /users/*id*/meetings használatával. Frissítéshez és törléshez szükséges az ID, így az URL /meetings/*id* lesz, és a kulcsszavak PATCH és DELETE. Egy konferencia státuszát a PUT /meetings/*id*/status requesttel lehet frissíteni.[91] [92] [93] [94]

Egy résztvevő regisztrálásához POST /meetings/*id*/registrants kérést kell küldeni, és ezt elfogadni, törölni vagy tagadni a PUT /meetings/*id*/registrants/*id* kéréssel kell.[95] [96]

A konferencia közben a felvételt a PATCH /live_meetings/*id*/events-el lehet vezérelni.[97]

4.2.6. PHONE

A Zoomot telefonon is lehet használni, de külön fel kell állítani hozzá egy accountot. Ezt elindítani APIn keresztül a POST /accounts/*id*/phone/setup requesten keresztül kell.[98]

Egy accounthoz meg lehet osztani a voicemail hozzáférést mással a POST /phone/users/*id*/settings/*type* kéréssel, ahol a type a beállítás típusa. A frissítés és a törlés is ugyanezt az URL-t használja, csak PATCH és DELETE kulcsszavakkal.[99] [100] [101]

Telefonszámot hozzárendelni egy accounthoz a POST /phone/users/*id*/phone_numbers kéréssel kell, törölni pedig a DELETE /phone/users/*id*/phone_numbers/*number*-el. A number a telefonszám. A hívási naplót a DELETE /phone/users/*id*/call_logs/*id*-vel kell törölni.[102] [103] [104]

Egy telefon eszközt hozzáadni az accounthoz a POST /phone/devices kéréssel kell. Frissíteni a PATCH /phones/devices/*id*-vel lehet, és a törlés ugyanezt az URL-t használja DELETE HTTP metódussal.[105] [106] [107]

Meg lehet osztani egy telefonvonalat egy adott csoporttal a POST /phone/shared_line_groups-al. Törölni ezt a csoportot a DELETE /phone/shared_line_groups/*id*-vel lehet.[108] [109]

Egy tagot hozzáadni egy adott csoporthoz a POST /phone/shared_line_groups/*id*/members kéréssel lehet. Törölni lehet egyszerre több embert vagy csak egyet. Egy embert a DELETE /phone/shared_line_groups/*id*/members/*id*-vel lehet, és a csoportos törlés URL-je ugyanaz mint a hozzáadás, csak DELETE HTTP metódust használ.[110] [111] [112]

4.2.7. ROLES

A Zoom szerepkörei döntik el, hogy egy felhasználónak milyen jogok járnak, és mihez fér hozzá. Egy ilyen szerepkört létrehozni a POST /roles requesttel kell, és frissíteni a PATCH /roles/*id*-vel. A törlés URL-je ugyanaz, de DELETE HTTP metódust használ.[113] [114] [115]

Hozzárendelni egy szerepkört egy felhasználóhoz a POST /roles/*id*/members-el kell, és eltávolítani a DELETE /roles/*id*/members/*id*-vel.[116] [117]

4.2.8. ROOMS

A videókonferenciákhoz használt Zoom szobákat a POST /rooms-al lehet létrehozni, és törölni a DELETE /rooms/*id*-vel.[118] [119]

A szoba helyét meg lehet változtatni a hierarchiában, PUT /rooms/*id*/location kéréssel. Ezeknek a lefoglalása, illetve felszabadítása a PATCH /rooms/*id*/events-el történik.[120] [121]

4.2.9. USERS

A felhasználói fiókok védelme az egyik kulcs biztonsági szempont, különösen a Zoomnál, ahol az accountok függenek tőle.

Felhasználói fiókot létrehozni a POST /users kéréssel lehet. Törölni a DELETE /users/*id*-vel.[122] [123]

Egy Zoom asszisztens képes beállítani meetingeket, és a konferenciát vezetni, ha nem ér rá a „főnöke”. Egy usert beállítani asszisztensként a POST /users/*id*/assistants requesttel kell. Törölni többet egyszerre ugyanezzel az URL-el, de DELETE HTTP metódussal. Egy asszisztens törléséhez kell az ID, és ilyenkor a kérés DELETE /users/*id*/assistants/*id* lesz.[124] [125]

Egy felhasználót lehet deaktiválni, illetve aktiválni a PUT /users/*id*/status kéréssel.[126]

A jelszót át lehet állítani a PUT /users/*id*/password-el. Emailt frissíteni szintén PUT HTTP metódussal, de /users/*id*/email-el lehet.[127] [128]

Átváltani a hozzárendelt accountot a PUT /accounts/*id*/users/*id*/account-al lehet átállítani.[129]

4.2.10. WEBINARS

A webinarok is konferenciák, amiket létrehozni egy felhasználó a POST /users/*id*/webinars kéréssel tud maximum tízezer főig. Törölni a DELETE /webinars/*id*-vel lehet.[130] [131]

Egy webinar állapotát a PUT /webinars/*id*/status-al lehet változtatni.[132]

A webinarban az előadó többek között képes képernyőt megosztani, videót és képet küldeni. Hozzáadni többet a POST /webinars/*id*/panelists kéréssel lehet. Tömeges törlés URL-je a létrehozással megegyezik, de a HTTP metódus a DELETE. Egyedüli törléshez szükséges az ID, így a request DELETE /webinars/*id*/panelists/*id*-vel lehetséges.[133] [134] [135]

Egy webinarhoz a regisztrálás a POST /webinars/*id*/registrants kérés segítségével lehetséges. Maximum harminc főig lehetséges tömegesen is regisztrálni a POST /webinars/*id*/batch_registrants-al. Ezeket el lehet fogadni, törölni vagy letiltani a PUT /webinars/*id*/registrants/status-al.[136] [137] [138]

4.3. GOOGLE MEET

A harmadik munkaszolgáltatás, amit implementálásra gondoltam, az a Google Meet. De jelenleg ennek a szolgáltatásnak nincs saját API-ja, és más APIban sincs implementálva, így nem tudom ezt a szolgáltatást implementálni.

5. SZOLGÁLTATÁSOK IGÉNYEBEVÉTELÉNEK LETILTÁSA

A Zorp tűzfal az előző projektemben is felhasznált moduláris tűzfal. A tűzfal alap felépítésével nem fogok ebben a dokumentumban foglalkozni, mert már megvizsgáltam, és nem történt változás a számomra érdekes területen.

A tűzfallal kapcsolatban a fő kérdés a műveletek letiltása.

A projekt egyik célja a nem engedélyezett műveletek tiltása a cég alkalmazottainak. Itt felmerül az a kérdés, hogy kinek tiltsa le a műveleteket a Zorp, a Google/Microsoft/Zoom felhasználói jogosultságoktól függetlenül.

Két opció létezik. Az első az, hogy a Zorpra bízom az autentikációt, és én csak azt mondom meg, hogy melyik műveleteket kell tiltania a tűzfalnak. Ez az egyszerűbb megoldás, de nem nyújt megfelelő biztonságot. Ideális esetben lenne elválasztás a különféle szintek között, mint például menedzser, alkalmazott és adminisztrátor. Néhány szolgáltatásban be van ez építve, de nem feltétlenül biztonságos megoldás. Erre viszont van a Zorpnak megoldása.

A másik opció a jogosultságkörönként szétosztás. Ezt bonyolultabb megvalósítani, mint az első verziót, de sokkal biztonságosabb. Ha elkülönítem a jogköröket úgy, hogy az átlag felhasználók nem férnek hozzá azokhoz a műveletekhez, ami igazán nagy kárt tud okozni, akkor sokkal csökkent a fokozatosan védendő felhasználók száma.

Ehhez több lehetőségem van, az autentikációtól kezdve az IP-ig. Az autentikáció bonyolult, és alapból nem is a modulomban történik meg, így azt az opciót inkább hagyom.

Egyszerűbb az IP-k alapján dolgozni. Ehhez lehetne közvetlenül az IP-t vizsgálni, de a Zorpnál lehetséges egyszerűbben megvalósítani ezt a Zónákkal.

A zónák alhálózatonként működnek. Amíg egy IP cím a zónában van, addig a hálózat része. Ezekhez be lehet állítani jogokat, illetve alzónákat is létre lehet hozni.

A zónákat szabályok segítségével lehet használni, amik megadják, hogy egy adott zónából kik, milyen proxyhoz férnek hozzá.

A proxy-k vezérlik a Zorpot, és határozzák meg a modul pontos instance-ét, hogy milyen paraméterekkel induljanak el. Részletesebben az előző munkámban már kifejtettem, és itt nem változik, így nem fogom kifejteni.

Ezek segítségével fogom megvalósítani a letiltást a különböző csoportoknál.

Érdemes megjegyezni, hogy a Zorpnak van grafikus verziója is, és ott lehet manuálisan beállítani felhasználói csoportokat, de ez a lehetőség nekem nem nyitott, mert a command line-os verzióval dolgozok.

5.1. A LETILTANDÓ METÓDUSOK ELDÖNTÉSE

A következő szekció az „Egészségügyi informatikai rendszerek biztonsága” nevű tárgy során lett megírva a diplomamunka kiegészítésére Dr. Pitlik László segítségével.

A cél egy olyan lista létrehozása, amit fel lehet használni a jövőben a modulhoz, és ami meghatározza a három kategóriához a megfelelő letiltandó műveleteket a Google, Microsoft és Zoom API-jaiból.

5.1.1. A PROBLÉMA

Meg kell határoznom a naplózott műveletek értékét, és az alapján sorbarendeznem őket, hogy kategóriákra tudjam osztani. Ez a három kategória az én esetemben, amivel megegyeztem a külső konzulensemmel, az az adminok, a menedzserek és az átlag felhasználók.

A rendszer, amibe ezeket a műveleteket kell majd besorolni, az admin kategóriába tartozó felhasználókat teszi meg minden joggal rendelkezőnek. A menedzser és az átlagos felhasználó viszont nem tehet meg mindent, bizonyos műveleteket tiltanom kell.

Ezt kell megtennem a műveletek értéke alapján, és ezek az értékek meghatározása a feladat.

5.1.2. POTENCIÁLIS MEGOLDÁSI ALTERNATÍVÁK

Természetesen létezik más megoldási módszer, mint például saját döntésem szerint kiválasztani egy vagy két műveletet mindegyik APIból, és azokat hozzárendelni a felhasználó kategóriákhoz a működés demonstrálásának.

Egy másik opció a véletlenszám generátor használata, és a véletlenszámok alapján szétosztani a műveletek. De ez egy valódi cégben elég nagy problémákat tud okozni, és nem reális a valódi működésnél.

5.1.3. ADATOK ÉS MÓDSZEREK

Valós helyzetben rendelkezésemre állna egy vállalattól naplófájlok, amik alapján el tudnám dönteni, hogy mégis melyik műveletet használják a legtöbbet, melyikből volt probléma, és melyik okozta a legtöbb kárt.

Mivel én absztrakt céggel dolgozom, ezért ez számomra nem áll rendelkezésre, kénytelen vagyok a legjobb tudásom szerint kategóriákat felállítani, és ezekben meghatározni az értékeket, amik alapján el fogom dönteni a műveletek helyezkedését.

Először is a meglévő műveleteket kategóriába rendeztem, mivel sok olyan van, aminek a célja ugyanaz, csak például más APIban található. Ilyen a Microsoft és a Gmail esetén a levélküldés. A célja ugyanaz, csak a szolgáltató cég más, ezért ezeket egy csoportba rendezem, és egy műveletként fogom kezelni.

Az egyik része a megoldásnak a megfelelő feltételek meghatározása, ami alapján nem csak azt lehet beazonosítani, hogy melyik a legveszélyesebb művelet, és ez által az admin kezére kerüljön, hanem azt is bele kell számolni, hogy a tűzfal egy vállalat mindennapi működését figyelné. Emiatt meg kell határozni pozitív, enyhítő attribútomokat is.

Ezek a negatív és pozitív attribútumok a következők:

Pozitív, enyhítő attribútumok:

- Művelet szükségessége átlag munkához: mennyire lehet szükséges a vizsgált művelet egy vállalat mindennapi működéséhez, a nem informatikához kapcsolódó munka elvégzésére.
- Művelet gyakorisága átlag munkánál: mennyire gyakran fordulhat elő a művelet a vállalat mindennapi munkájában.
- Visszaállíthatóság lehetősége: potenciális kár vagy sérülés esetén mennyire lehet visszaállítani az adott műveleteket a korábbi, sérületlen állapotukba, a kárt kijavítva.

Negatív, veszélyességi attribútumok:

- Bizalmasság sérülése: mennyire okozhatja a művelet bármilyen felhasználó, fájl, vagy bármi más olyan kézre kerülését, akinek egyébként nem kellene hozzáférnie. Szükségesnek tartom, mivel az informatikai biztonságban az alapkövetelmények egyike.

- Sértetlenség sérülése: mennyire változtathat meg a művelet kihasználása egy fájlt/felhasználót, vagy bármilyen más erőforrást. Szükségesnek tartom, mivel az informatikai biztonságban az alapkövetelmények egyike.
- Rendelkezésre állás sérülése: mennyire akadályozhatja meg a művelet kihasználása a fájl/felhasználó vagy bármilyen más erőforrás elérhetőségét. Szükségesnek tartom, mivel az informatikai biztonságban az alapkövetelmények egyike.
- Kár mértéke: milyen súlyos a kihasználás után okozott kár. Szükségesnek tartom, mert segít a segít a súlyosság pontos meghatározásában.
- Csoportmunkára hatás: mennyire hat a cégen belül egy csoport munkájára az adott művelet kihasználása. Szükséges, mert a több embert érintő problémák még több kárt tudnak okozni.
- Bekövetkezés után kijavítás nehézsége: A kár megtörténése után mennyire bonyolult kijavítani. Fontosnak tartom, mert a művelet veszélyessége attól is függ, hogy mekkora kárt tud okozni, és ezt a kárt mennyire egyszerű kijavítani.
- Potenciál visszafordíthatatlan kárra: mennyire valószínű, hogy a kár visszafordíthatatlan lesz, például egy belső fájl kijut az internetre. Ezt fontosnak tartom, mert ez a típusú kár tudja potenciálisan a legnagyobb kárt okozni a vállalatnak.

A műveletek értékeléséhez az attribútumhoz egy 1-től 5-ig tartó skálát használtam, egy korábbi munka miatt, ahol ez egy jól működő skála volt.

A skála értéke attól függ, hogy pozitív-e az attribútum, vagy negatív. Pozitív esetén növekvő skálát használtam, a nagyobb érték azt jelenti, hogy fontosabb/szükségesebb az adott művelet.

Negatív esetén minél nagyobb volt az érték, annál komolyabban kell venni az adott attribútumnál a műveletet.

A létrejött táblázatot ki kell tölteni. Utána meg kell határozni azt az értéket, ami alapján sorrendbe rendezem a műveleteket.

Ehhez két módszert használtam.

5.1.3.1. ÁTLAGOLÓ MEGOLDÁS

Az átlagoló megoldáshoz összegeztem külön a pozitív és a negatív értékeket, és átlagoltam őket.

Ez után a negatív értékből kivontam a pozitív értéket az adott művelethez, hozzáadva ötöt, hogy kiküszöböljem a negatív értékek által okozott hibát, majd a kategória meghatározásához a maximum érték százalékait vettem. 85% feletti értékű műveleteket csak az admin tudja használni, 85% és 65% között csak a menedzser és az admin fér hozzá, az alatt pedig mindenki.

5.1.3.2. JÓSÁG ÉRTÉK, Y0

A jóság értékes meghatározáshoz az Y0-t, a jóság értéket egymillióra kell állítani, és ezt módosítani.

Ebben segített Dr. Pitlik László, akinek a módszertana szerint a meglévő műveletekhez egy külön táblázatba a pozitív és negatív attribútumokhoz hozzá lettek rendelve az adott sorszám a többi művelethez képest.

Ebből egy táblázat létrehozása után a https://miau.my-x.hu/myx-free/coco/beker_y0.php oldalon található Solver ad az oda helyezett táblázatra és az Y0-ra egy becslést.

Ezt az eredményt a dokumentumba való másolás után össze lehet hasonlítani a többféle megoldással, az előző átlagolóval is, és a számszerinti sorrendbe rendezett műveletek táblázatának átlagával is. Ezt természetesen ellenőrizni kell a becslés Y0-tól való eltérés, a delta szorzásával. Ha nagyobb, mint egy, akkor megfelelő.

Ezt a becslést felhasználva létrehoztam az előző megoldáshoz hasonló kategóriákat. Az adminoknak elérhető műveletek továbbra is 85%os határon maradnak, és a menedzser 65%, illetve teszteltem a menedzsert 50%os értékkel is.

Ezeknek a határoknak a pontos értékei az Y0 becsült értékével számolva:

	Határérték
Admin	999525 pont
Menedzser 65%	999748 pont
Menedzser 50%	999916 pont

1. táblázat Határértékek

5.1.4. EREDMÉNYEK

A létrehozott táblázat a következőképpen néz ki a pozitív attribútumok esetén:

Művelet	Művelet szükségessége átlag munkához	Művelet gyakorisága átlag munkánál	Visszaállíthatóság lehetősége
Komment és válasz készítése	4 pont	5 pont	3 pont
Komment és válasz módosítása	3 pont	5 pont	3 pont
Komment és válasz törlése	3 pont	4 pont	1 pont

2. táblázat Pozitív attribútumok kitöltve, minél zöldebb annál jobb

A negatív attribútumokból több létezik, és szükséges az irány, ami megadja, hogy az 5 a jóság szempontjából pozitív, vagy negatív dolog. Ez lehet 0 vagy 1.

Ilyenkor az összegzés az átlag veszélyességi érték és az átlag jóság érték különbsége.

A Solverrel létrehozott megoldás, és a sorrend táblázat értékei a következők:

	Bizalmasság sérülése	Sértetlenség sérülése	Rendelkezésre állás sérülése	Kár mértéke	Csoportmunkára hatás	Bekövetkezés után kijavítás nehézsége	Potenciál visszafordíthatatlan kárra	Művelet szükségessége átlag munkához	Művelet gyakorisága átlag munkánál	Visszaállíthatóság lehetősége	Y0	naiv2	naiv1	Csak negatív	opt1
Komment és válasz készítése	72 pont	22 pont	1 pont	17 pont	28 pont	42 pont	87 pont	33 pont	1 pont	77 pont	1000000 pont	38 pont	-1,9 pont	2,1 pont	1000173 pont
Komment és válasz	43 pont	22 pont	1 pont	17 pont	28 pont	42 pont	87 pont	79 pont	1 pont	77 pont	1000000 pont	40 pont	-1,7 pont	2 pont	1000156 pont
Komment és válasz törlése	1 pont	22 pont	87 pont	17 pont	28 pont	90 pont	43 pont	79 pont	24 pont	128 pont	1000000 pont	52 pont	-0,5 pont	2,1 pont	999996 pont

3. táblázat Sorrendbe rendezett műveletek táblázata

Ebben a táblázatban a naiv2 az első 10 attribútum átlaga, míg a naiv1 az átlagoló megoldással számított érték. A csak negatív csak a kockázati értékeket veszi figyelembe. Ezeket lehet ellenőrizni korrelációval.

5.1.4.1. FŐ KÉRDÉS: MELYIK JOBB?

A megoldások hatásosságának ellenőrzésére a hozzáférés kategóriákat érdemes figyelni. Megvalósítottam ezt az átlagoló megoldásra, és a két menedzser határral rendelkező jóság értékes megoldásra is.

Összesen három megoldást kell értékelnem: az átlagoló megoldást, a jóságértékes, szigorúbb 50%-os menedzserhatárral, és a jóságértékes 65% menedzserhatárral.

A megoldások összehasonlítása után arra jutottam, hogy a jóságértékes menedzser 50%-os határértéke túlságosan szigorú, és tilt olyanokat, amik szükségesek az átlag felhasználónak, mint például fájlok módosítása. Ez túlságosan sok fölösleges munkát adna a menedzsereknek és az adminoknak.

A naiv átlagoló megoldás szintén nem megfelelő, ugyanis az meg túlságosan engedékeny némely művelettel, bár már közelebb áll az ideális megoldáshoz.

A számomra legjobb megoldás a 65%os jóságérték. Ez elég szigorú, de meghagyja a szükséges műveleteket, amik egy vállalat napi munkájához szükséges.

5.1.5. VITA A BECSLÉS HATÉKONYSÁGÁRÓL

Mivel saját értékekkel dolgozok, és nincs valós, ezért szükséges ellenőrizni az értékeim hitelességét és konzisztenciáját. A tesztet Dr. Pitlik László hajtotta végre, egy harmadik, ellenőrző fél szimulálásaként.

A teszthez az általam megadott adatok lettek összehasonlítva egy véletlenszám-generátorral. A generátor által adott adatokra is megtörtént a jóságérték-elemzés.

A különböző attribútumokra külön-külön megtörtént a gyanú elemzés, ami az adott attribútum értéknek megnövelésével történt.

Ez után szintén el kell végezni erre az Y0-s jóságtesztet, ahol a jóságérték a súlyozott attribútum.

Az ez alapján született becsült értékből le kell vonni 10000-et, és el kell osztani 1000-el, így megkapva a becsült értéket. Ezt össze kell hasonlítani az eredeti értékkel, ami az eredeti érték és a becsült érték különbségét osztva az eredeti értékkel. Ez egy eltérési százalékot ad, amit lehet előjelesen és anélkül számítani.

Ezeknek az összes attribútum százalékainak összesítésével létre lehet hozni egy táblázatot az összes előjeles/előjel nélküli eltérésre, létrehozva a gyanú táblázatot.

Ebből meghatározható, hogy hol végeztem esetlegesen hibát, vagy rossz becslést.

Ezen kívül meghatározható az egész attribútumra a becslés és a súlyozott attribútumok korrelációja alapján az egész attribútumra a becslés pontossága, és az, hogy ez jobb-e, mint a véletlenszám-generátor.

Lehetséges csoportokra bontani a műveleteket. Ez a dokumentumban a létrehozás, frissítés, törlés, módosítás, és egyéb kategóriák. Ez alapján létrehozhatók mutatók, amikről megvizsgálható, hogy melyik kategória a legszigorúbb, és hasonló mértékek.

5.1.6. KÖVETKEZTETÉSEK

A jóság értékes vizsgálat alapján létre tudtam hozni egy megfelelő listát, ami alapján el tudtam dönteni, hogy melyik művelet lesz elérhető mindenkinek, csak a menedzsereknek és az adminoknak, illetve csak az adminoknak.

A konzisztencia teszt ellenőrzi, hogy az értékek mennyire megfelelőek, és hogy esetleg hol lehet javítani a feladatban.

Mivel a létrehozott értékek a konzisztencia teszt alapján nagyrészt megfelelőek, ezért a 65%-os jóságértékkel létrehozott hozzáférés listát fogom használni a tiltáshoz.

A tiltást definiáló értékeket, és a konzisztencia tesztet tartalmazó táblázatokat mellékeltem.

6. RENDSZERTERV

Egy új rendszer megtervezésére nincs szükség, mivel az előző projektben létrehozott tűzfal továbbra is megfelel a célnak, csak fejleszteni kell rajta. Bár ideális lenne egy külön szerver gépet használni, de erre a viszonylag kevés vizsgálandó szolgáltatásra továbbra is megfelel a virtuális gép.

Az előző projektem az API hívásokat a HTTP metódus/URI különbségek alapján azonosította, és logolta, hogy milyen művelet történt.

A Python nyelven írt modult kell módosítanom úgy, hogy a letiltást is képes legyen elvégezni. Az új megfigyelés implementálásához nem szükséges nagy változás, hanem a meglévő rendszert fogom kiterjeszteni az új szolgáltatásokra.

Bár a Zorp képes rá, a letiltás megvalósításához nem feltétlenül lesz szükséges a beérkező üzenet felbontása. Mivel továbbra is API hívások alapján teszek különbséget, így lehetséges a meglévő rendszert átalakítani úgy, hogy az azonosított metódust átengedje vagy tiltsa.

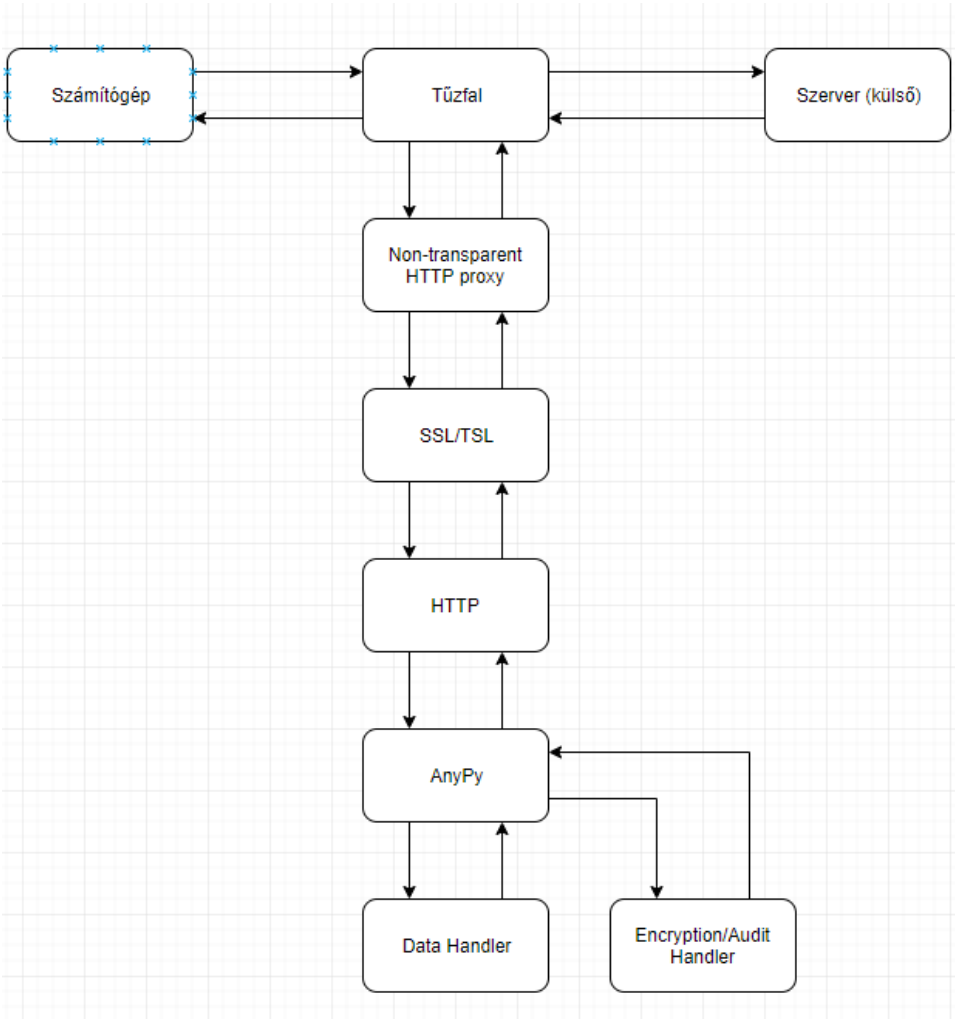
A sebesség ebből a szempontból továbbra is fontos, de nem túlnyomóan, mivel a Zorpban a Python modult átkonvertálni a C++ nyelven írt Zorp Core számára érthető programmá sokkal több idő, mint a modulban elvégzett műveletek, és én ezt nem tudom módosítani.

7. IMPLEMENTÁCIÓ

7.1. ÁLTALÁNOS JELLEMZŐK

Maga a Zorp tűzfal nem változott a szakdolgozatom megírása óta. Továbbra is egy Ubuntu virtuális gépen fut, ami a szükséges követelményeknek megfelel. Egy valódi környezetben elengedhetetlen a tényleges külön szerver, mivel ezt a rendszert nem egy klienre, hanem sokkal többre tervezték. Nekem csak egy klienssel kell dolgoznom, és a modul a virtuális géptől függetlenül mozdítható, így nem akadály vele dolgozni.

A rendszer továbbra is a következő ábra alapján működik, és kezeli az adatokat:



1. ábra Zorp rendszer működése

A tűzfal beáll a szerver és a számítógép közé a böngésző proxy beállítását kihasználva, és megvizsgálja a bejövő és kimenő adatokat az éppen futó modulok alapján.

Az ábrán látható folyamaton megy keresztül egy adatcsomag, amíg el nem jut az AnyPy modulig. Ebből következik a Data Handler, illetve az Encryption/Audit Handler. Az én modulom nem használ Data Handlert, mivel az főleg akkor kell, ha fel akarjuk bontani az adatcsomagot.

Nekem nem ez a célom, mivel az audithoz az előző feladatban nem volt szükséges az adatok felbontása. Meg lehetett határozni a meghívott műveletet a URL és link kombinációból. A kiterjesztett Google, illetve a teljes Zoom és a felhasznált Microsoft API-knál is ez a helyzet.

Az adatok titkosításához mindenképpen szükséges a külön Data Handler modul, de az adatfolyam korlátozásához nem feltétlenül, mert az megoldható anélkül is.

Ezért a továbbiakban se fogom külön modulban felbontani az adatokat, hanem tovább fejlesztem a meglévő AuditModule.py-t, ami az előző munkámban létrehozott modul.

Ebben a modulban megtalálható az ábrán található összes szükséges lépés, ami szükséges a Zorp megfelelő működéséhez.

7.2. A MODUL RENDSZERE

A Zorp egy modult instance-okon keresztül futtat. Ezt kell meghívni az indítóparancsban, illetve több, nem egymástól függő modul esetén egy külön fájlból meghívni az összeset. Nekem az egy modul miatt a parancs elég.

Magát a modult a vim szövegszerkesztővel írom, mint ahogy előzőleg is.

A nálam custom_audit_instance néven futó instance meghatároz egy service-t, ami megadja, hogy melyik Non-transparent HTTP proxy induljon el, az ábra alapján. Ezt a service-t egy dispatcherben hívja meg a program, ami még a service-en kívül meghatározza, hogy melyik IP címre csatlakozzon, és melyik portot figyelje.

Az IP cím változását figyelni kell egy valós rendszernél, de nekem a virtuális gép miatt a változásoktól függetlenül az 192.168.56.1-es IP cím a cím, ami a VirtualBoxot összeköti a valós géppel.

7.2.1. CUSTOMNONTRANSPARENTHTTPPROXY

A meghívott alap Non-transparent HTTP proxy általam megvalósított verziója, CustomNonTransparentHttpProxy, egy „config” metódust használ, ami a Zorp konstruktora. Ez meghívja a szülő osztály (az alap, sablon NonTransparentHttpProxy) konstruktorát, megadja, hogy melyik osztály fogja a tényleges műveletet végezni (esetemben LoggerClass), és tovább engedi az összes requestet, mivel a szakdolgozatban nem volt szükséges megakadályozni semmilyen adatfolyamot.

A jelenlegi feladat miatt ezen muszáj változtatni.

7.2.2. CUSTOMHTTPSPROXY

A LoggerClass meghívja a CustomHTTPSProxy-t, ami a tényleges HTTPS üzenetet bontja fel, és megadja, hogy pontosan milyen típusú metódusoknál végezzen vizsgálatot a program. Nekem ez a POST, PATCH, PUT és DELETE, mivel ezek végeznek módosításokat a figyelt szolgáltatásokban. Ez továbbra is így van, még a kiterjesztéssel is, így nincs szükség változtatni.

7.2.3. CUSTOMSSLPROXY

A CustomHTTPSProxy meghívja a CustomSSLProxy konstruktorát, ami megadja az EncryptionPolicy-t és meghívja az alap HttpProxy konstruktorát. Itt már nincs szükség saját metódusra, megfelel az alap HttpProxy.

7.2.4. ENCRYPTIONPOLICY

Az EncryptionPolicy adja meg a tűzfalnak azt, hogy milyen titkosítást, és azonosítást használjon. Nekem a kétoldalú titkosítás van beállítva, és ez továbbra is megfelel a célnak.

A Zorp egy certificate-et állít elő megadott kulcsból, illetve egy megbízható és nem megbízható listából. Ezek továbbra is megfelelnek a célnak. Az összes szükséges fájl a */etc/zorp/keybridge* mappában található meg. A kulcs a key.pem fájl, a jelszó pedig *passphrase*. Ez nem biztonságos, és valódi környezetben nem lenne megfelelő, de tesztelés céljából jó lesz. A megbízható lista a ZorpGPL_TrustedCA.cert.pem, illetve a hozzá szükséges kulcs a ZorpGPL_TrustedCA.key.pem. Ehhez hasonlóan a nem megbízhatóak listája a ZorpGPL_UnTrustedCA.cert.pem, és a hozzá tartozó kulcs a ZorpGPL_UnTrustedCA.key.pem.

Ezekre a .pem fájlokra szükség lesz a tesztelés során a virtuális gépen kívül, mivel meg kell őket adni a böngészőnek elfogadható, illetve nem elfogadható certificate-ként.

7.2.5. KZORP

A Zorphoz tartozik egy kzorp néven futó felület, de az nekem a szakdolgozat során nem állt rendelkezésemre, és továbbra se lesz szükséges, így az ahhoz tartozó változó, a config.options.kzorp_enabled továbbra is FALSE-ra van állítva.

7.3. LOGGERCLASS

A LoggerClass a tényleges auditáló osztály, amiben az összes metódus megtalálható. De mivel a tiltás nem naplózó funkció, ezért átneveztem WorkerClass-ra a tisztább használat kedvéért.

A fejlesztést két szinten valósítom meg.

Az első a meglévő szolgáltatások frissítése, és az újak beépítése a meglévő rendszerbe. Itt a már meglévő metódusokat is módosítani kell, hogy rendesen működjen a Zoom és a Microsoft API-jával.

A második a megakadályozás felépítése. Ennek akkor érdemes nekikezdeni, ha az első szint legalább már kész van, még ha nincs is tökéletesre letesztelve.

Fordítva túl sokat kellene módosítani a meglévő kódon, ami a megakadályozást irányítja, és csak sok problémát okozna.

7.3.1. EXAMINETHECALLEDMETHOD

Az examineTheCalledMethod meghatároz változókat, amikkel az implementáció további részében fogok foglalkozni.

A négy változó:

1. a hívott POST, PUT, PATCH, DELETE-et azonosító *method*, ami az előző munkámban, a szakdolgozatomban *copied_method* volt. A másolt változót töröltem, mert nem szükséges.
2. a *request_url_file*, ami a linkek elejét levágja, így nem kell dolgoznom a https:// és a .com közötti részekkel. Itt is a *copied_url_file*-t töröltem, mert nem szükséges.

3. *split_url*, ami a ’/’ jeleknél szétválasztja a *request_url_file*-t, így meg tudom vizsgálni, hogy egy adott szó melyik. Ezt behelyeztem a *self.session*-be azért, mert egyrészt nem konvertálódik C++ kóddá, valamint Pfeiffer Szilárd szerint jobban hasonlít tényleges, valódi Zorp kódra.
4. és a *len(self.session.split_url)*, ami a nevéből adódóan a *self.session.split_url* elemszámát adja meg. Ez az előző munkámban a *split_url_word_count* volt.

Ezek a változók tökéletesen megfelelnek számomra a továbbiakban is, így itt nem történt változás.

A következő rész eldönti, hogy a Google szolgáltatások közül melyikkel kell foglalkoznia éppen a rendszernek. Ez az előző szakdolgozatban elég volt, mivel nem érdekelt más cégtől szolgáltatás, de mivel Zoommal és Microsofttal is dolgozom már, így ez nem egy működő megoldás.

A külön Google szolgáltatásokat továbbra is meg kell határozni, de nem rögtön ebben a metódusban. Helyette először el kell dönteni, hogy egyáltalán melyik cég szolgáltatásait vizsgálom a jelen iterációban.

Ezt viszonylag egyszerű ellenőrizni, hiszen az összes URL eleje egyértelműen meghatározza, hogy melyik céggel dolgozom.

Ezt ki lehet nyerni a *url* változóból, amit a WorkerClass változóként kap meg.

Három URL-t kell megkülönböztetni, de nem mindegyik API használja ugyanazt a kezdetet.

A megvizsgált URL-ek:

1. A Microsoft csak egy linket használ az összes szolgáltatására a Graph API miatt, ez a https://graph.microsoft.com.
2. A Zoom esetében is csak egy API van, ezért itt is csak a https://api.zoom.com linkre kell figyelni.
3. A Google esetében lesz bonyolult a vizsgálat. Itt az alap a https://googleapis.com, de ez nem elég, mivel több APIval dolgozok, és több saját linkkel dolgozik. Ezért figyelni kell a további linkeket is:
    a. https://www.googleapis.com
    b. https://gmail.googleapis.com, a Gmail API számára.

c. https://people.googleapis.com, ami a People API elérésére szükséges
d. https://docs.googleapis.com, ez a Docs API-nak kell.

A megfelelő cég azonosítása után a modul meghívja a megfelelő metódust, ami tovább viszi a vizsgálatot az API, vagy az API rész azonosítására.

Ezek a metódusok:

1. Microsoft esetében az *examineMicrosoft()*,
2. Zoom esetében az *examineZoom()*,
3. és Google esetében az *examineGoogle()*.

Mindegyik metódushívás elé szükséges egy self-et berakni, így például a Google-t vizsgáló metódus teljes hívása a *self.examineGoogle()*.

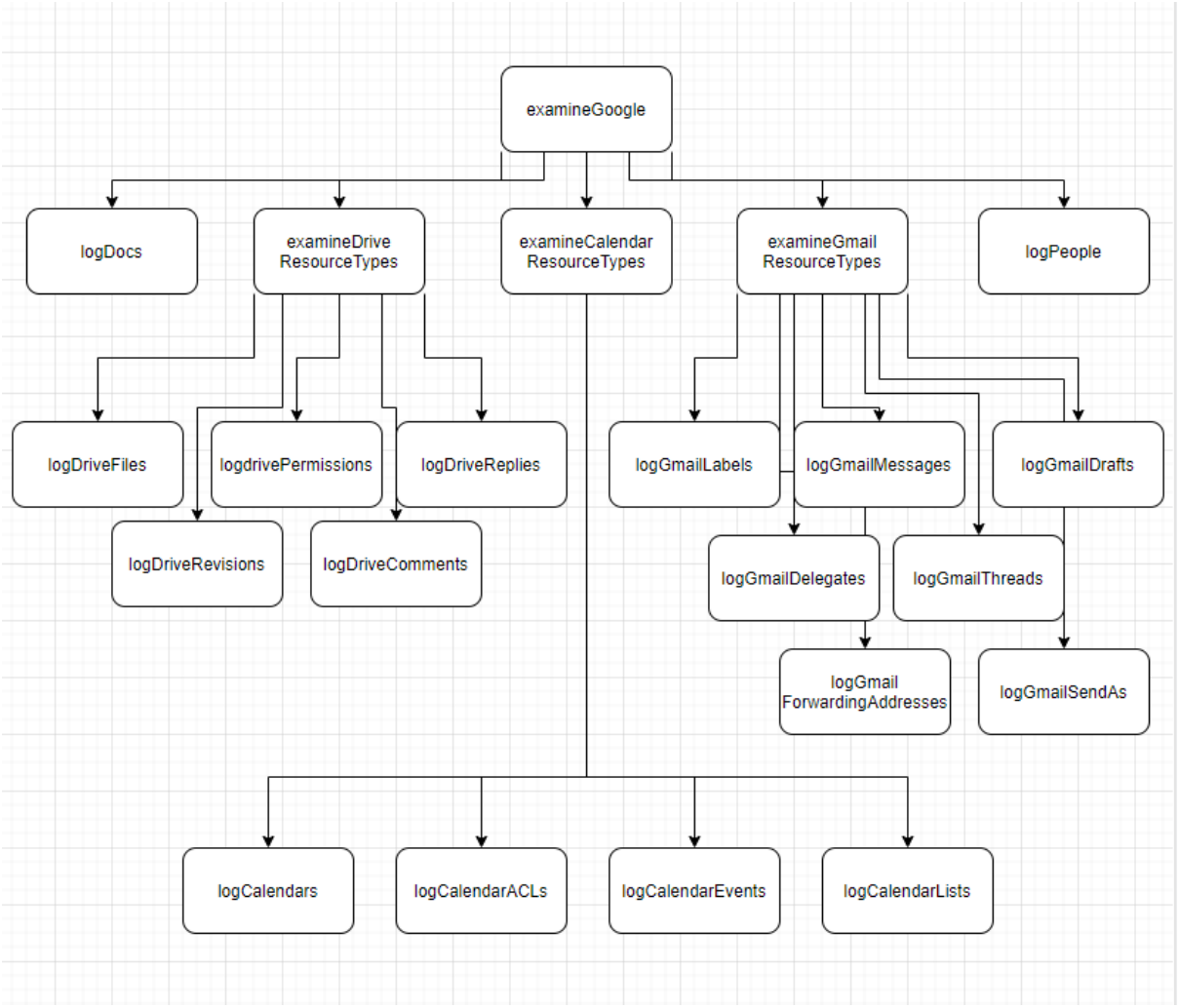
Mind a három metódus, illetve a továbbiakban vizsgált összes metódus a WorkerClass része, de ha ez nem így van, akkor azt külön meg fogom említeni.

Az eredeti metódusban ezek után következik egy mindent lefedő *return HTTP_REQ_ACCEPT*, ami tudatja a tűzfallal, hogy a jelenleg vizsgált kéréssel minden rendben van, és mehet tovább. Ez már nem megfelelő, mivel vannak olyan kérések, amiket vissza kell majd utasítani.

7.3.2. EXAMINEGOOGLE

Az új *examineGoogle()* metódus tartalma a régi *examineTheCalledMethod()* Google szolgáltatásokat vizsgáló részét. Itt nem történt semmi változtatás, mert nem adtam hozzá új API-t a vizsgálandókhoz.

A metódusok kapcsolata egymáshoz a következő ábrán látható:



2. ábra Google metódusok kapcsolata

A Google szolgáltatásokat vizsgáló metódusoknál csak azokat fogom kiírni, ami változott, és azon belül is csak a változásokat.

7.3.2.1. LOGDRIVEPERMISSIONS

A *logDrivePermissions()*-be az *examineDriveResourceTypes()* új feltételén keresztül jut el a program, ami a többihez hasonlóan ellenőrzi, hogy egy megadott szó a URLben a „permissions”.

A metódus azonosítja a kérést, és megfelelően naplózza.

A logoláshoz továbbra is a proxyLog-ot használom. Ez a Zorp saját funkciója, ami automatikusan kiír extra információt a felhasználó számára egy külön képernyőn, amit a Zorp

alapból mutat. Ilyen extra információ többek között a sessionID és az idő, így nem kell külön azokat is kiírnom, hanem rögtön ki van írva a többi mellé.

A kiírt információ mennyiségét is lehet állítani szintekkel, 1 és 9 között. Nekem négyre van állítva, Pfeiffer Szilárd javaslata szerint.

7.3.2.2. LOGPEOPLE

A People API-hoz a tömeges műveleteket egyszerű ellenőrizni azzal, hogy figyelem, hogy mire végződik a hívott kérés URL-je. Ezeket az elejére helyeztem el, hogy elkerüljem az ütközést a többi hasonló módon vizsgált kéréssel.

7.3.2.3. EXAMINEGMAILRESOURCETYPES

Ennél két módosítás történt. A hat szónál hosszabbak közé elhelyeztem a thread-ek figyelését, ami meghívja a logGmailThreads() metódust.

A második a kategóriába nem helyezhető beállítások, mint például a vakáció és az autoForwarding. Ezeket nem tudtam külön metódusba rakni, ezért elhelyeztem a végére, közvetlen naplózásra. Mindegyik hat szónál hosszabb, ezért nem lesz belőle probléma.

7.3.2.4. LOGGMAILTHREADS

Ez megfigyeli a thread-eken végzett műveleteket. Szétválasztani szintén a kérés URL végének figyelésével, illetve a *method* változó tartalmával történik.

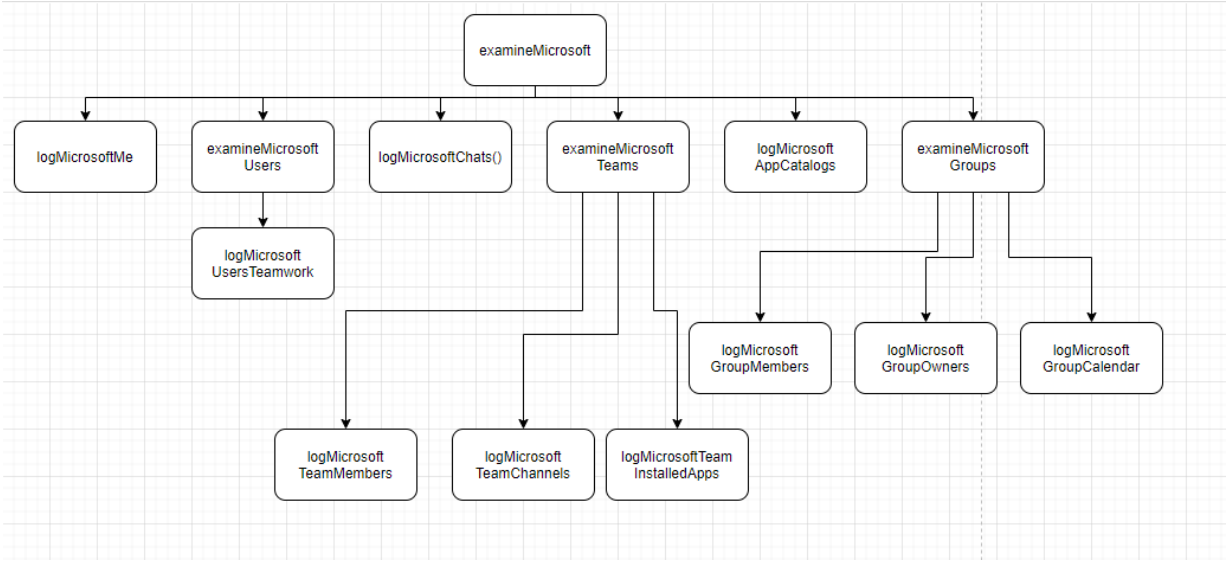
Azonosítás után naplózza a kiválasztott műveletet.

7.3.3. EXAMINEMICROSOFT

A többi szolgáltatással ellentétben a Graph API-nál nem lehet egyértelműen különbséget tenni az API különböző részei között, ezért muszáj az URL-ek szerint felosztanom a feladatot.

Ez a naplózás szempontjából semmi különbséget nem jelent, de a kódolás szempontjából nem lesz olyan egyértelműen átlátható, mint a másik kettő cég API-jainál.

A metódusok kapcsolatát a következő ábrával ki lehet fejezni:



3. ábra Microsoft metódusok kapcsolata

A vizsgálást a szokásos módon teszem, figyelem az URL elejét, és az alapján eldöntöm, hogy melyik metódust kell meghívni.

A /v1.0/me, /v1.0/chats és /v1.0/appCatalogs esetében nincs annyi külön URL, amit vizsgálni kellene, ezért ott rögtön naplózó metódusokat hívok meg. Ezek a logMicrosoftMe(), logMicrosoftChats(), és logMicrosoftAppCatalogs.

A /v1.0/users, /v1.0/teams és /v1.0/groups esetében szét kell választanom, hogy mit logolok, mert túl sok fajta kérés jöhet be az egyértelmű azonosításhoz.

A végén a Microsoft csoport végleges törlését egyszerűen logolom, mivel különbözik a többitől, és nincs értelme saját metódust csinálni neki.

7.3.3.1. LOGMICROSOFTME

A Graph API-ban a /v1.0/me-vel kezdődő URL-eket egyszerű naplózni. A linkek utolsó szava egyértelműen meghatározza, hogy milyen műveletet szeretne elvégezni a felhasználó. Ezeket figyelem, és proxyLog segítségével naplózom.

Fontos megjegyezni, hogy ez nem a Google, ahol egy API-t/API részét egyértelműen logolom. Itt több, különböző API rész található meg egybe helyezve.

7.3.3.2. LOGMICROSOFTAPPCATALOGS

Ez egy egyszerű metódus, ami egy alkalmazás hozzáadását, törlését és frissítését figyeli, és nem ütközik össze az API más részével.

7.3.3.3. LOGMICROSOFTCHATS

A chat részen végrehajtott műveleteken kívül még szükséges figyelni a chathez hozzárendelt alkalmazásokat, és az azokon végrehajtott műveleteket is. Ez a rész nagyban hasonlít a *logMicrosoftAppCatalogs()* metódusra.

7.3.3.4. EXAMINEMICROSOFTUSERS

Az *examineMicrosoftUsers()* metódus a /v1.0/users kezdetű URL-eket elemzi.

Egy része megegyezik a /v1.0/me kezdetű URL-ekkel, így azt vizsgálni egyszerű. Ugyanazt a feladatot látják el, így át lehet másolni a *logMicrosoftMe()* metódus megfelelő részét. A proxyLog üzeneteken se kell módosítani, mivel az egészen annyi változik, hogy melyik felhasználón végződik el a művelet.

Ezen kívül még vannak a felhasználói fiókon végzett műveletek, a szerepkörök és a teamwork. Ebből csak a teamworknek szántam külön metódust, mivel a többinek nincs annyi művelete, hogy megérje.

7.3.3.5. LOGMICROSOFTUSERSTEAMWORK

A teamworknek külön létrehozott metódus, ami az alap műveleteket figyeli és naplózza. Itt nincs átfedés az API különböző részei között.

7.3.3.6. EXAMINEMICROSOFTTEAMS

Ez a metódus a Microsoft Teams csapatain végzett műveleteket figyeli.

Itt se csinálok külön metódust az olyan műveletek számára, amit nem lehet egyértelműen összefűzni más, hasonló műveletekkel.

A három külön naplózó metódus, amit szétválasztottam a fő Teams API résztől:

1. *logMicrosoftTeamMembers()*, a csapat felhasználóin végzett alap műveleteket naplózza. Ezek a hozzáadás, frissítés és törlés.
2. *logMicrosoftTeamInstalledApps(),* a korábban a chat részben látott alkalmazásokat vezérlő műveletek a csapat számára.

3. *logMicrosoftTeamChannels(),* a csatornákat vezérlő API rész. Ez hosszabb, mint a többi, de csak azért, mert mellette a csatorna tagjain végzett alap műveleteket is figyeli a program.

7.3.3.7. EXAMINEMICROSOFTGROUPS

A csoportok a Microsoft esetében különböznek a csapatoktól, így a feladatban is külön metódus kell nekik.

Az alap műveletek mellett figyelni kell a csapat kialakítását a csoportból.

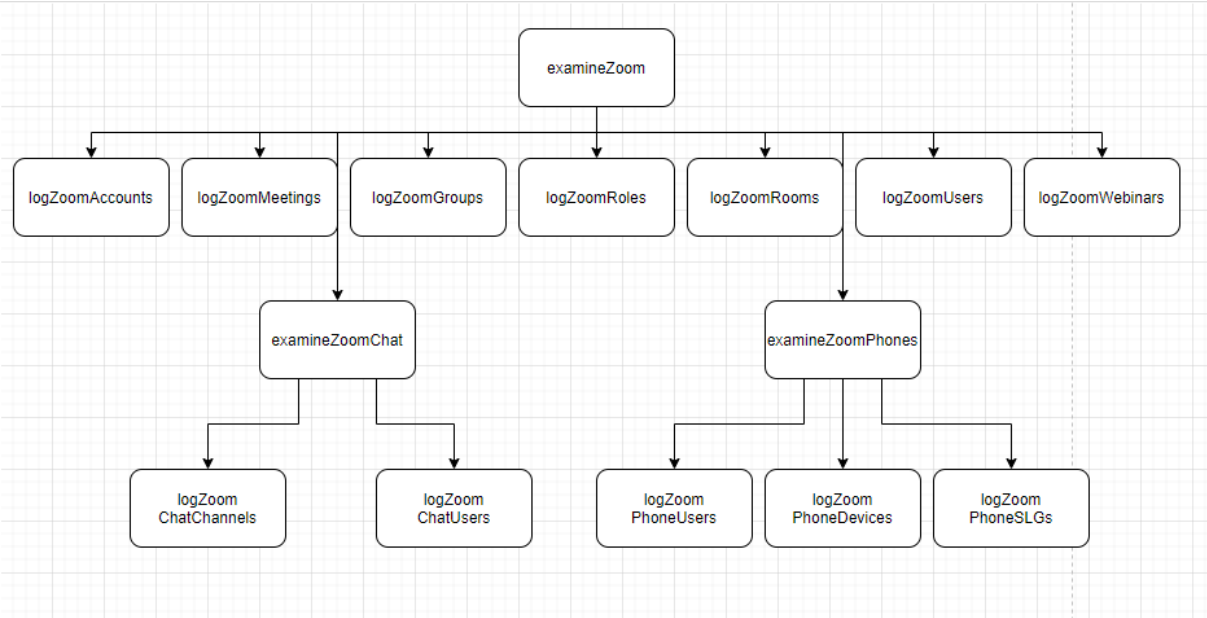
Itt is szintén három részt figyeltem meg külön metódusban. Ezek:

1. *logMicrosoftGroupCalendar(),* ezt két feltételből lehet elérni a URL-ekben található különbségek miatt. Ez a csoport eseményeit figyelő metódus.
2. *logMicrosoftGroupMembers(),* a csoport tagjait vezérlő műveleteket figyeli. Itt csak két művelet található, egy tag hozzáadása a csoporthoz, illetve a csoport törlése.
3. *logMicrosoftGroupOwners(),* a csoport tulajdonosai számára. Szintén csak törölni és hozzáadni lehet.

7.3.4. EXAMINEZOOM

A Microsoft Graph APIjával ellentétben a Zoomnak több API-ja van, de ebből engem csak a Zoom API érdekel. Ez jól átlátható, és a kódot is fel lehet osztani részekre úgy, hogy egyértelműen átlátható legyen.

Nem lesz az a keveredés és átfedés műveletek között, mint a Microsoftnál volt. Sokkal inkább hasonlít a Google által működtetett API-kra.



4. ábra Zoom metódusok kapcsolata

A Google-el ellentétben ez az API nem akkora, mint abban az összes együtt, ezért nem kell mindegyiket külön felbontani. Elég a kettő nagyobb API részt felbontani könnyebben vizsgálható részekre. Ezek:

1. *examineZoomChat(),* ami a Zoom Chat funkciói alatt összefogott műveleteket bontja fel külön kategóriákra,
2. *examineZoomPhones(),* ez a Zoom összes telefonos funkcióit osztja fel egyszerűbben auditálható részekre.

A többi funkció nem annyira megkülönböztethető, vagy nincs rá szükség, hogy külön felbontsam. Emiatt elég egy egyszerű naplózó modul az összes számára. Ezek a funkciók:

1. *logZoomAccounts(),* a Zoom fiókokat kezelő API rész,
2. *logZoomMeetings(),* a Zoom találkozókat naplózó metódus,
3. *logZoomGroups(),* a Zoom csoportjait vizsgálja. Ezek hasonlítanak a Microsoft esetében már megvizsgált csapatokra,
4. *logZoomRoles(),* a Zoom szerepköreit figyelő metódus, ami a fiókok és a felhasználók számára hasznos,
5. *logZoomRooms()*, a Zoom szobákat logoló metódus,
6. *logZoomUsers(),* a Zoom felhasználókat figyeli, amik a Zoom fiókhoz rendelhetőek, és

7. *logZoomWebinars(),* a Zoom konferenciáit naplózó metódus.

Ezeken kívül még egy külön műveletet érdemes vizsgálni, aminek fölösleges külön metódust készíteni. Ez az élő találkozó közbeni felvétel módosítása, amit azonnal naplózok a proxyLog segítségével.

7.3.4.1. EXAMINEZOOMCHAT

A Zoom chat funkcióiból kettőt kell vizsgálnom, a chat csatornákat és a felhasználói szintű csatornákat, amik a Zoom felépítéséből magasabb szinten találhatóak.

1. A csatornákat a *logZoomChatChannels()* metódus naplózza,
2. a felhasználói szintű csatornákat pedig a *logZoomChatUsers().*

7.3.4.2. EXAMINEZOOMPHONES

A Zoom telefonos integrációját az API három fő részen keresztül kezeli, ezt a három részt kell megvizsgálnom.

1. a *logZoomPhoneUsers()* a telefonnal rendelkező felhasználókat figyeli meg,
2. a *logZoomPhoneDevices()* magukat az eszközöket,
3. és a *logZoomPhoneSLGs()* a Shared Line Groupokat.

7.3.4.3. LOGZOOMACCOUNTS

A fiókok figyelése nagyrészt olyan műveletek naplózásából áll, amit nem a fiók indított, hanem a fiókon hajtanak végre. Ilyen például a telefon fiók hozzárendelése a felhasználóhoz, ami a dokumentációban a Phones alatt található, de nekem ide kell rendelnem.

Ezen kívül megtalálható a szokásos létrehozás és törlés. Nagyrészt a *request_url_file* végződéséből azonosítja a program a megfelelő műveletet.

7.3.4.4. LOGZOOMCHATCHANNELS

A chat csatornák vizsgálata inkább a benne található felhasználók kezeléséről szól, és az egyik fő feladat megkülönböztetni a több törlési műveletet.

Ebben a URL-ek hossza és a küldött HTTP metódus segít.

7.3.4.5. LOGZOOMCHATUSERS

A nevével ellentétben nem a felhasználókat, hanem a felhasználói szintű csatornákat vizsgálja ez a metódus. A Zoom e között és a fiók szintű csatornák között komoly különbséget tesz, ezért én is külön vizsgálom.

Itt csak két törlés művelet van, de a hosszuk alapján szintén egyértelműen meg lehet őket különböztetni.

7.3.4.6. LOGZOOMMEETINGS

A meetingeket naplózó metódus hasonlít az eddigiekhez struktúrában és módszerben is.

A különbség az, hogy itt vannak olyan részek, amiknek nem írtam saját metódust, mert nem érte meg, hanem különféle feltételek alapján több részre bontottam.

Ilyen például a felvételeket vizsgáló rész. Ez lett volna az egyetlen, aminek saját metódust lehetett volna írni, de nem tettem meg, hanem elválasztottam a vizsgálat többi részétől.

Így csak akkor kezdi el vizsgálni ezeket, ha biztos, hogy felvételekről van szó. Hasonlóan a külön metódusokhoz.

7.3.4.7. LOGZOOMGROUPS

Ebben a metódusban jobban kihasználom a különféle művelet URL-ek hosszát, mivel azok alapján hasonló módon a *logZoomMeetings()*-hez több részre van bontva.

Ezek a részek nem annyira nagyok, mint ott, és belül csak az elküldött szavakat vizsgálja.

7.3.4.8. LOGZOOMROLES

Itt a szokásos törlés, frissítés, létrehozás műveleteken kívül egy olyan rész található, amit fel kellett bontanom az egyszerű azonosítás érdekében.

7.3.4.9. LOGZOOMROOMS

Itt nincs semmilyen külön rész, összesen négy műveletet naplóz a már ismert feltételek alapján, ismert módszerekkel.

7.3.4.10. LOGZOOMUSERS

A felhasználók a Zoom rendszerében sokkal több hatalommal rendelkeznek, mint az egyszerű fiókok, ezért több műveletet is el tudnak végezni.

Az asszisztensek naplózásához külön részt hoztam létre, hogy ne kelljen az összes műveletet megvizsgálni minden alkalommal.

7.3.4.11. LOGZOOMWEBINARS

A webinarok esetében is van kettő olyan rész, aminek nem éri meg külön metódust írni, de érdemes felbontani az átláthatóság és a sebesség miatt is.

Ez a két rész a panelistek, és a registrantok körüli műveletek.

7.3.4.12. LOGZOOMPHONEUSERS

Ez a rész a telefonok felhasználói fiókjait naplózza, és az azon végrehajtható műveletet. A létrehozás nem ebben a metódusban van naplózva, hanem a *logZoomAccounts()*-ban, de a nevétől eltérően nincs alárendelve másnak, hanem tudja a saját beállításait kezelni, nem kell egy mester fiók hozzá.

Itt nincsenek külön részre bontva a műveletek.

7.3.4.13. LOGZOOMPHONEDEVICES

Az alap törlés, frissítés, létrehozás műveleteken kívül itt nincs mit logolni.

7.3.4.14. LOGZOOMPHONESLGS

Hasonlóan a korábbi *logZoomChatChannels()* metódushoz, itt is a több törlési művelet okoz problémát, de a *len(self.session.split_url)* változó segítségével ezt meg lehet oldani egyértelmű megkülönböztetéssel.

7.4. MEGAKADÁLYOZÁS

Az megakadályozáshoz az alap terv az, hogy ha megvan a művelet, akkor a naplózás után ellenőrzöm a zónát, és beállítok egy változót, ami alapján a program eldönti, hogy tovább lehet-e engedni a kérést.

A Zorp esetén a tiltás nem ’igen’ vagy ’nem’ kérdése. Négy üzenetet lehet használni, ami eldönti a kérést.

Ez a négy:

- *HTTP_REQ_ACCEPT*, ez tovább engedi a kérést semmilyen probléma nélkül.

- *HTTP_REQ_REJECT*, visszadobja a kérést, és nyitva tartja a kapcsolatot további kommunikációra.
- *HTTP_REQ_ABORT*, ami a tiltással együtt lezárja a kapcsolatot.
- *HTTP_REQ_POLICY*, ez meghív egy metódust az adott kapcsolathoz. A modulban korábban használtam, de most nem lesz rá szükség.

A két tiltó közül a *HTTP_REQ_REJECT*-et fogom használni. A példakörnyezetben, amit használok, nem érdemes lezárni a kapcsolatot.

A *self.error_info* string beállításával lehet üzenni a felhasználónak, hogy miért nem engedi tovább a Zorp a műveletet.

A zónákat meghatározni az InetZone metódussal lehet, ami a Zorp Core-ban található.

Változónak meg lehet adni a name-et, a nevét, illetve az addrs, ami a zóna IP tartománya. Ezzel lehet maszkkal egy adott részt lefoglalni. A másik opció a Zone() metódus, amihez csak a zárójelen belül oda kell írni a zóna nevét, illetve az addrs-t még szögletes zárójelben.

Az alap, ami kell az az internet, a minden. Ez a 0.0.0.0/0 IP tartományt használja.

Ezen kívül három zónát használok a példámhoz. Az első az adminisztrátorok, nekik lesz hozzáférésük minden művelethez.

A managerek a második, ők a legtöbb művelethez hozzáférnek, de a komolyabb műveletekhez, mint például Microsoft csoportok törlése, nem fognak.

A harmadik a users, ők az átlag felhasználók. Sok művelethez hozzáférnek, de olyanokhoz, amik nagyobb kárt tudnának okozni, nem fognak. Ilyen a Google Drive fájlok végleges törlése.

A zónák működése miatt egy IP mindig abba a zónába fog tartozni, ami a legrészletesebb. Így, ha meghatározom zónába az előfordulható IP-ket, akkor mindig oda fogja berakni az internet helyett.

A zónák tényleges IP címe nem fontos, egy valós cégnél úgy is változna. Három kell nekem, és mivel a rendszer a 192.168.56.0/24-ben van, ezért a fő zóna, amit éppen használok, a 192.168.56.0/25 lesz.

7.4.1. VÁLTOZÁSOK A PROXYBAN

Mivel több felhasználó kategóriával dolgozok, ezért kénytelen vagyok valamilyen szabályrendszert bevezetni, ami azonosítja, hogy mégis melyik felhasználó próbál műveletet végrehajtani, és neki milyen jogosultságai vannak.

Erre két megoldást fogok bemutatni, mivel a munkám során ezekkel kísérleteztem.

7.4.1.1. A DISPATCHER SZÉTOSZTÁSA

Az első megoldáshoz módosítani kell a modult működtető proxy-t. A szakdolgozatomban egy ilyen volt, a *CustomNonTransparentHttpProxy*. Ez határozza meg a WorkerClass-t használandó osztályként.

Ezt fel tudom használni arra, hogy megkülönböztessem a felhasználó típusokat egymástól. Az osztály átírása, és két ugyanolyan osztály behozatala szükséges hozzá.

Ezek a felhasználótípusoktól függően:

- *CustomNonTransparentHttpProxyA,* jelzi az admin felhasználót. A kód többi része nem változik. Ugyanúgy meghívja a szülő osztály *config* metódusát, megadja a WorkerClass-t használandó osztálynak, és tovább küldi az összes beérkező forgalmat ellenőrzésre.
- *CustomNonTransparentHttpProxyM,* ami a menedzsereket jelzi, 'm'-re módosítja a zóna jelzőjét tároló változót.
- *CustomNonTransparentHttpProxyU*, ami a felhasználók szintjét kezelő proxy. A zóna jelzője 'u'.

A proxykat el is kell indítani, és a zónákkal összepárosítani, viszont azt már nem ezekben az osztályokban kell megtenni.

Ahhoz, hogy a tiltás működjen, át kellett írnom a meglévő, korábban a szakdolgozatban készült részeket.

A *custom_audit_instance*-en belül létrehozok még az előzőkhez hasonlóan két Service-t, és az előzőt módosítom úgy, hogy mindegyik Service elindítson egy-egy Proxy-t, ami az előző szekcióban volt definiálva. Ezek a service-ek:

- a *custom_audit_service_admin* indítja a *CustomNonTransparentHttpProxyA*-t,

- a *custom_audit_service_manager* indítja a *CustomNonTransparentHttpProxyM*-t,
- és a *custom_audit_service_user* indítja a *CustomNonTransparentHttpProxyU*-t.

A Zónák felvitelén túl kénytelen voltam a modult elindító metódus, a Dispatcher változtatásán. Az, ami a szakdolgozatban íródott, csak egy proxy elindítására képes, viszont nekem több proxyval kell dolgoznom ebben a megoldásban.

Itt meg kell említenem a Zorp egy funkcióját, a Rule-t. Ez a szabály össze tud kötni egy Service-t és egy zónát, ha definiáltam. Automatikusan működik, és a meghatározás után nincs más dolgom vele. Ezt megvizsgáltam, de nem használtam végül.

Ehhez rendelkezésemre áll a CSZoneDispatcher, ami zónák alapján hív meg Service-eket. Hasonló a korábban használt Dispatcherhez.

A Service-eket a beérkező és a cél zóna alapján hívja meg. Nekem a cél zóna mindegy, ezért oda '*' kerül, ami jelzi, hogy az bármi lehet.

A zónák alapján a következő Service-ek lesznek meghívva:

- az *admins* zóna alapján a *custom_audit_service_admin*,
- a *managers* zóna alapján a *custom_audit_service_manager*,
- és a *users* zóna alapján a *custom_audit_service_user*.

Minden más egyezik. Ezek után ki kell találnom egy módszert, hogy a zóna típusát tovább küldjem a tényleges dolgozó osztályba.

Ilyen opció például a globális változó, ami működik, de nem szép megoldás. Egy másik opció a szülő proxy meghívása, és az alapján a változó meghívása. De van jobb módszer.

7.4.1.2. EGY EGYSZERŰBB MEGOLDÁS

Egy másik megoldást is lehet használni, amihez ugyanúgy kellenek a fő osztályban történt változások, de nem szükséges az összes változás a Dispatcherben.

Zónákat ugyanúgy létre kell hozni, mert valami alapján meg kell különböztetni a beérkező kapcsolatokat. Ezek ugyanazok lesznek, mint az előző megoldásban.

A különbség abban áll, hogy nem szükséges szétosztanom a Dispatchert, hanem maradhat az eredeti. Ez nem osztja fel a beérkező forgalmat kategóriákra, de ez nem probléma, mivel ezt meg tudom tenni később a fő naplózó és tiltó osztályban.

7.4.2. WORKERCLASS VÁLTOZTATÁSOK

Létezik egy *session* osztály, amiben minden egyes külön kapcsolat adatai megtalálhatóak. Ebben benne van a forrás és a cél IP címe, portja, és hasonló információk, valamint a zóna. Ezt a *self.session.server_zone* változóból lehet megtudni.

Mivel a zóna alapból nem elemezhető, ezért a változóból ki kell venni a zóna nevét, ami a zóna definícióban találhatóval ellentétben a *self.session.server_zone.name*-ben található. A továbbiakban ezt fogom ellenőrizni a zóna azonosításáért.

Ezen kívül létrehozok egy *self.session.needtoblock* boolean változót, ami megadja, hogy szükséges-e az adott műveletet blokkolni. Ez alapból hamis. Ezt is a sessionbe helyezem el, ugyan is a Zorp felépítése miatt a *self.session*-ben található adatok nem lesznek átkonvertálva C++-ra, gyorsítva a programot.

Ezt az új változót vizsgálom az *examineTheCalledMethod* végén. A kérdés nélküli tovább engedést töröltem, és már csak akkor engedi tovább a folyamatot a metódus, ha a *self.session.needtoblock* változó hamis.

Ha igaz, akkor a metódus a HTTP_REQ_REJECT választ adja, és ehhez az „Access denied on your level.” hibaüzenetet csatolja a felhasználó számára.

Ahhoz, hogy eldöntse a program, hogy kell-e blokkolni egy műveletet, két féle megoldást használok. Először is nem minden műveletnél teszem fel ezt a kérdést, hanem csak azoknál, amit kell blokkolni.

A blokkolandó műveleteknél azt ellenőrzöm, hogy valaki milyen kategóriába kerül. Viszont nem nézem egyenként végig, hanem egyszerűsítek.

Az adminok mindenhez hozzáférnek, így őket el lehet hanyagolni. A userekre és a managerekre kell figyelnem csak.

A tiltás felépítése alapján két kategóriát lehet meghatározni egy műveletre. Vagy akkor kell blokkolni ha user hívta meg, vagy akkor ha manager hívta meg. Az utóbbinál viszont garantált, hogy a usernek is blokkolni kell, így azt meg lehet oldani más módon.

A userek esetén ellenőrzöm, hogy a zóna neve ’users’, tehát felhasználó van bejelentkezve. Ha igen, akkor a *self.session.needtoblock* változót igazra állítom, ami tiltja majd a műveletet.

Managerek esetén nem azt ellenőrzöm, hogy valaki ’users’ vagy ’managers’ zónából érkezett, hanem azt, hogy valaki nem ’admins’. Ezt azt jelenti, hogy a felhasználó nem admin, így nem szabad hozzáférést adni neki a hívott művelethez. Azt, hogy melyik megoldást kell ebből a kettőből használni, azt a korábban létrehozott tiltólistából döntöttem el.

Ennek a blokkolási módszernek az is az előnye, hogy nem kell minden egyes alkalommal mind a három zónatípust megvizsgálni, hanem egy feltételvizsgálatból megoldható az egész feladat. Ez egyszerűsíti a kódot, és valamennyivel gyorsítja is a modult.

7.5. SEBESSÉG

Egy sok klienst kiszolgáló tűzfal esetében a sebesség egy fontos szempont. Nekem is figyelembe kellett venni a modul írása során, főleg a jelentős növekedés miatt. A korábbi szakdolgozatban használt sebességnövelési módszerek itt is érvényesek.

A Zorp felépítése miatt van egy nagy sebesség akadály, amit nem tudok kiküszöbölni, mert a tűzfal beépített része. Ez a két nyelv közötti konvertálás. A Zorp Core és a modulok két különböző nyelven íródtak. A Core C++ nyelven készült, és a modulok Pythonnal. E között a két nyelv között a konvertálás sokkal több időt vesz el, mint amit én a modullal tudnék nyerni.

A tűzfal működését nem tudom befolyásolni, így muszáj volt a modulon belül segíteni a sebességen.

A vizsgálat során a sok if/elif feltétel felbontása különböző szintekre, illetve a Google, Microsoft és Zoom szolgáltatások felbontása pont ezt a célt szolgálja.

A sok if/elif nem a leghatékonyabb megoldás, egy Switch jobb lenne. Ilyen a Pythonban nincs, helyette lehet Dictionary-t használni. Ez nem megfelelő, mivel több változót figyelek egyszerre, és az alapján azonosítok műveleteket.

Még egy sebességnövelési módszer a minél kevesebb feltétel vizsgálat. Egy request_url_file.endswith() hatékonyabb, mint három-négy különböző feltétel, ami figyeli a lehetséges programhibákat, és meg is határozza, hogy melyik műveletet akarja végrehajtani a felhasználó. Még segítség is, hogy egy .endswith() vagy .startswith() esetében nem kell figyelnem arra, hogy a program egy nem létező értéket akar-e lekérni a *self.session.split_url*-től.

A Zorp párhuzamosít automatikusan, nem kell nekem megvalósítani a modulban. Minden híváshoz külön szálat használ. Észleli a számítógépen található logikai processzorok számát, és azt használja limitnek.

7.6. FORRÁSKÓD

Az előző feladathoz hasonlóan a forráskód a Zorp szerveren található, a virtuális gépen.

Maga a szerver egy alapbeállításokkal rendelkező telepítés, és ezen ebben a feladatban sem változtattam. Emiatt a feladat egésze megtekinthető az AuditModule.py python modulban. Ez megtalálható mellékletként az eredeti formátumban, illetve szövegfájlként is.

8. TESZTELÉS

Az implementáció végrehajtása után szükséges az elvégzett munkát tesztelni, és a hibás funkciókat kijavítani. A tesztelést is két szintben fogom végezni. Az első a naplózás tesztelése lesz, a második pedig a tiltás.

8.1. ZORP

A tűzfalat be kell állítani a böngésző és a számítógép közé.

Böngészőnek eredetileg egy friss Mozilla Firefoxot akartam használni. Azon teszteltem a szakdolgozatban vizsgált műveleteket, így kevés az esély rá, hogy a más böngésző miatt probléma lesz. Itt be kell állítani proxy-nak a virtuális gépet a szerverrel, és az elfogadott tanúsítványok közé berakni a Zorp saját tanúsítványát. Ez a virtuális gépből WinSCP segítségével megszerezhető.

Ez már nem opció, mivel a Mozilla a biztonság miatt átállt egy központi CA-ra. Emiatt muszáj a Microsoft Edge-et használnom. Itt ugyanazt kell csinálni, mint a Firefoxban, de a proxy-t nem lehet külön beállítani, hanem a számítógép proxyját kell állítani. Ezt 192.168.56.254-es IP-re állítottam, 3128-as portra.

A felállításhoz elindítom az Oracle VirtualBoxon a Zorpot. Valószínűleg működik más virtuális gép programmal is, de én ezt használom.

A szerver elindulása belépek, a felhasználónév és a jelszó is balasys. Nem biztonságos, és egy valódi környezetben meg kellene változtatni, de nekem tesztelés céljából megfelel. A szükséges admin hozzáférés „sudo su” paranccsal szintén a balasys jelszót használja.

A működéshez nem szükséges, de nekem teszteléshez muszáj a PuTTY kliens beléptetése a tűzfalra. A belépési folyamatot itt is megismétlem.

Elindítom a tűzfalat a „/usr/bin/zorp –F –l –v 4 --user zorp --group zorp –p /etc/zorp/AuditModule.py –a custom_audit_instance” paranccsal. A -v a verbosity, ami megadja, hogy milyen részletesen írja ki a program a kapcsolaton történteket. Ez nekem minimum négynek kell lennie, mivel azt állítottam be a proxyLogoknál, de az ajánlott a hat, vagy akár a maximum kilenc.

8.2. NAPLÓZÁS

A naplózás teszteléséhez szükséges lesz egy tesztelő környezet, amin következmény nélkül tudok API kéréseket végezni. Ezt általában az adott cég valamilyen formában biztosítja.

Ezeknek a felderítése és működésre bírása az első feladat, mert nem minden szolgáltatásnál egyszerű.

8.2.1. GOOGLE

A Google teszteléséhez a szakdolgozatban használt módszert fogom igénybe venni.

A Google által üzemeltetett API Explorer 2019 vége felé megszűnt, helyette az API Reference oldalon lehet külön a műveleteket tesztelni, a megfelelő művelet oldalára navigálva. Ehhez szükséges egy Google fiók megfelelő engedélyekkel, de azt e-mailen keresztül lehet adni.

A régi, direkt erre kitalált Google fiókot fogom használni, ami más célt nem szolgál, ez a zorpthrowaway@gmail.com.

A tesztelendő művelet weboldalán a „Try it!” gombra kattintva előjön a felület, amit a teszteléshez fogok használni. Itt meg lehet adni különféle paramétereket a kéréshez. Ez számomra lényegtelen, mivel a program a paraméterektől függetlenül érzékeli a műveletet. Ezzel véd a potenciális támadások ellen is.

Be fogok írni a szükséges paraméterekhez egy lényegtelen szöveget, a többit pedig figyelmen kívül hagyom.

Az új műveletek mellett meg fogom vizsgálni a régiket is, kontroll miatt. Így biztosítva lesz, hogy működik minden része a programnak. Ezen kívül még olyan kéréseket is fogok tesztelni, amire a tűzfalnak nem szabad reagálnia.

Az Execute gombra kattintás után meg kell jelennie a PuTTY kliensen a művelethez tartozó üzenetnek.

8.2.1.1. ELSŐ TESZTELÉS

A tesztelés során a korábban megvalósított műveletek működtek, de voltak problémák az új résszel.

Az első probléma az volt, hogy a Google a tesztelésnél módosított a url-en, és kell egy extra „content” szó az elejére. Így a link https://content.googleapis.com lesz, illetve a gmail és a többi https://content-gmail.googleapis.com.

A fő hiba a Gmail Threads részénél volt, ott rossz feltételt vizsgáltam, a hatodik helyett az ötödiket kell. Ezt áthelyeztem a megfelelő helyre.

A másik probléma a Drive Permissions-nél volt, mivel nem raktam be az első vizsgáló feltételhez kivételnek a korábbi „revisions” és „comments” mellé. Emiatt fájlként észlelte az engedélyeket.

A Pop, Imap és egyéb beállítások, illetve a People API batch műveletei működtek.

8.2.1.2. MÁSODIK TESZT

A javításokkal a Threads esetén a törlés megjavult, de a többi nem jó, mert a *split_url* nem megfelelő elemét vizsgáltam. Ezt átírtam 7-re.

A Permissions esetén beírtam a kivételek közé, és így már megfelelően működött.

8.2.1.3. HARMADIK TESZT

A Threads jó elemét vizsgálva már megfelelően működik a rendszer Google része.

8.2.2. MICROSOFT

A Google régi API Explorer-éhez hasonlóan a Microsoft is üzemeltet egy online tesztelési szolgáltatást a Graph API számára.

A Graph Explorer segítségével le lehet tesztelni a szükséges metódusokat. A számomra érdekes POST, és egyéb módosító műveletek teszteléséhez be kell jelentkezni egy Microsoft fiókba. Nekem ez az egyetemi e-mail fiók lesz, az tesztelésre megfelel.

Be kell állítani a megfelelő HTTP metódust, verziót, és a URL-t. Ebből a verzió fix, csak az 1.0-át használom. A többi a tesztelendő művelettől függ.

A megfelelő feltételek beállítása után futtatni kell a „Run query” gombra kattintással.

8.2.2.1. TESZTEK

A Microsoft Graph rész tesztelését a Google-el ellentétben részekben fogom végezni, mivel sok van belőle.

A kategóriák az implementációban megvalósított szétválasztások, például /users és /me.

A /users részbe gond nélkül beért a metódusba. A Teamwork működött, de az alap patch/delete és AppRoleAssignments nem. Itt az volt a hiba, hogy a *len(self.session.split_url)* rászámít egy extrát a hosszra, de a *self.session.split_url* nem. A javítás után már nem futott IndexErrorra.

A /me-n sokat nem kellett tesztelni, mivel ugyanaz, mint a /users első része, és az gond nélkül működött. Itt se volt vele probléma.

AppCatalogs-nál nem írtam oda a self-et a *self.session.split-url* elé néhány helyen, amit a Python elvár. Pótlás után már működik. A létrehozó metódus nem működött, mert teamsApp-ot írtam teamsApps helyett.

A /teams résznél egyedül a membersnél rontottam el a PATCH/DELETE-et, azon kívül minden működött.

Group esetében semmilyen hibát nem találtam, minden működött gond nélkül.

Chat-nél sem találtam problémát.

8.2.3. ZOOM

A Zoom teszteléséhez nincs egyszerű online eszköz, mint az előző két esetben.

A hivatalos oldal azt ajánlja, hogy a Postman programmal teszteljem.

Ehhez szükséges a Postman online felülete és az API importálása egy workspace-be.

Sajnos ez a megoldás számomra nem megfelelő, mivel nem a szükséges linket küldi el, hanem egy saját linket, ez a https://orion-http.gw.postman.co/vl/request. Ez a Zorp számára nem megfelelő, így más megoldást kell keresnem.

Ez a megoldás a „Felhőszámítási rendszerek” nevű tárgy keretében született meg.

8.2.3.1. ALTERNATÍV TESZTELÉS: JENKINS ÉS CURL

A cURL egy olyan eszköz, amit a Zoom tesztelésére potenciálisan fel lehet használni. Ezt a saját számítógépemen teszteltem, amin a Zorp virtuális gép fut, de az nem volt megfelelő.

Az említett tárgy keretében egy Jenkins szervert hoztam létre, ami cURL használatával képes automatizálni a tesztelést nem csak a Zoom esetében, hanem a Microsoft és a Google API-knál is.

A Jenkins egy tesztautomatizáló környezet, aminek külön Ubuntu szervert hoztam létre. Ez szintén egy virtuális gépen fut, de nincs semmilyen proxy beállítva rá, ami a tűzfalra mutatna.

A Jenkins szerverbe lépéshez szükséges felhasználónév jenkins, és a jelszó szintén jenkins. Ez nem egy biztonságos megoldás, de a tesztelés céljára megfelel.

A programot, magát a Jenkinst böngészőben kell megnyitni, az IP-n és a hozzárendelt porton. Nekem ez a 192.168.56.102:8080, de ezt lehet állítani a Jenkins fájljai között.

Az online felületen több Freestyle Projectet hozok létre, ami engedi, hogy használjam a shellt cURL parancsok kiadására. A több projekt azért van, hogy egyenként tudjam tesztelni a program részeit, például külön teszt a *logZoomUsers()* és a *logZoomPhoneUsers()* metódusoknak.

Viszont a rendes cURL parancsok előtt össze kell kötni a Jenkinst a tűzfal szerverrel. Ehhez lehetséges parancsokat kiadni a cURL parancsok előtt. Ezt megtettem minden egyes Freestyle Projectre.

Ezek a parancsok:

- export HTTP_PROXY=balasys:balasys@192.168.56.254:3128
- export HTTPS_PROXY=balasys:balasys@192.168.56.254:3128
- export FTP_PROXY=balasys:balasys@192.168.56.254:3128.

A proxy kialakítása után kiadom a cURL parancsokat, amik a következő struktúrát használják: curl -k --request POST/PUT/PATCH/DELETE --url „Teljes url link, pl. https://api.zoom.us/v2/users/id” --header Authorization: Bearer valamitoken.

A Bearer token normális működéshez szükséges lenne, és meg is lehetne szerezni, de a Zorp akkor is érzékeli a parancsot ha hibás a token. Ez engedélyezi a tesztelést, így nem fogok törődni a Bearer tokennel a továbbiakban.

Egy projekthez hozzáadom az összes szükséges tesztet az adott részhez. Ezt megismételtem az összes Zoom műveletre, létrehozva egy teszteléshez használható, egyszerű rendszert.

8.2.3.2. TESZTEK

A tesztelést a többivel hasonlóan részekre vettem, viszont a benne lévő Zoom műveleteket egyszerre futtattam. Az accountok esetén találtam egy hibát, amiben a *self.session.split_url*-ből

egyel nagyobb értéket próbáltam kiszedni. Ez rossz kiírást és hibát is okozott, de a Zorp képes kihagyni egy nem létező változót, és működik tovább a program.

```
Jan 14 14:02:17 core.debug(4): (svc/custom_audit_instance/custom_audit_service:1/custo
Traceback (most recent call last):
  File "/etc/zorp/AuditModule.py", line 77, in examineTheCalledMethod
    self.examineZoom()
  File "/etc/zorp/AuditModule.py", line 273, in examineZoom
    self.logZoomAccounts()
  File "/etc/zorp/AuditModule.py", line 311, in logZoomAccounts
    proxyLog(self, "zoom.accounting", 4, "Account '%s' deleted", self.split_url[4])
IndexError: list index out of range
```

1. Képernyőkép Zorp hibát dob a split_url-re

Ez a hiba az egész Zoom résznél jelen volt, és emiatt ezt át kellett írni. Valamint az account törlésének vizsgálata során a feltételhez egyel többet írtam, mint amit kellett volna, így azt is javítanom kellett.

Ez után teszteltem a Roles-t, Chat channels-t, Chat users-t, és a Devices-t. Ezekben nem találtam problémát.

A következő probléma a Phone users-nél jelentkezett. A telefonszám módosítás vizsgálata jó volt, de a többit nem érzékelte. Ezt a *len(self.session.split_url)* vizsgálatánál rontottam el, mert az egyenlő (==) helyett nagyobb vagy egyenlő-t kellett volna használnom (>=). Javítás után a probléma megoldódott.

A Groupsnál és a Meetingsnél ugyanaz a probléma jelentkezett. A műveletek adott részét, pl. Groupsnál a members, rossz helyen vizsgáltam, így az oda tartozó műveleteket más vizsgálat megtalálta és megállította. Itt azokat a részeket át kellett helyezni, de nem kellett a teljes vizsgálást átírni. Áthelyezés után működött.

A Rooms és a Phone Shared Line Groups (SLGs) jó volt, ahhoz nem kellett nyúlnom.

Users esetében ugyanabba a probléma futottam bele, mint a Groupsnál és a Meetingsnél, azt is át kellett rendezni, valamint az assistants feltételnél a változó túl nagy volt, oda egyel kevesebbet kellett adni. Utána működött.

Utolsónál a Webinarsnál a .endswith(„/registrants”)-t elgépeltem, nem kell a ’/’ jel. Utána már a batch_registrants-t is tudja érzékelni.

8.3. A TILTÁS TESZTELÉSE

A tiltás teszteléséhez a korábban létrehozott Jenkins infrastruktúrát fogom használni. A Zoom teszteléséhez használt projektek mellé létrehoztam egy Microsoft és egy Google projektet, amivel azokat a műveleteket vizsgálom, amiket tiltok.

A naplózás tesztelése során már megtörténtek a tesztek ezekre az API-kra, így nem tartottam szükségesnek, hogy azokat is teszteljem, amiket nem tiltok, és amit nem módosítottam.

A tesztelés során mind a három felhasználó kategóriát tesztelni kell, a usereket, menedzsereket és adminokat. Ezek a modulban három különböző zónában találhatóak, így ezt nekem is meg kell valósítani.

Lehetséges a Jenkins szervert mozgatni, vagy klónozni más IP tartományba, ezzel áthelyezve egy másik zónába, de én egy egyszerűbb módszert fogok használni.

A modulon belül változtatni fogom, hogy a Jenkins szerver által használt 192.168.56.102-es IP cím éppen melyik zónában van. Ezt az IP tartomány átírásával fogom megtenni, hogy a 192.168.56.0/25-ös tartomány a megfelelő zónában legyen.

A többi zóna valós környezetben fontos lenne, és ez a módszer sok másik rendszert zavarna, de tesztelés szempontjából a célnak megfelel.

8.3.1. USERS

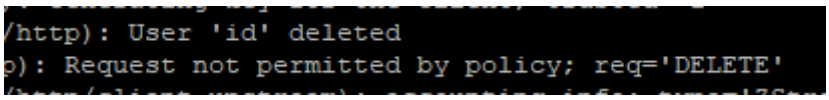
Az átlagos felhasználókat tartalmazó zóna tesztelése egyszerű. Itt a cél az, hogy minden művelet, ami tiltásra került, legyen letiltva. Az átlagos felhasználó számára az összes ilyen művelet elérhetetlen.

Mint a Zoom tesztelésénél, itt is felállítottam a két virtuális gépet, elindítva a tűzfalat. Áttekinthetőség kedvéért a tűzfalat Putty-on indítottam el, mivel ott végig lehet nézni a teljes teszt output-ot, míg VirtualBox-ban csak a legvégét. Ez után elindítottam egyesével a teszteket, mindig megnézve az eredményt.

Ez volt az első teszt, így voltak problémák. Az első probléma a korábban leírt CSZoneDispatcher-es megoldással volt, mivel pár próbálkozás kellett hozzá, hogy megfelelően megadjam a paramétereket.

Ez a megoldás sikerült, de inkább más, jobb megoldást választottam. A következő tesztek során a *self.session*-ben lévő változók használása volt a cél. Ez is sikerült pár próbálkozás után.

Miután rendesen felállt a rendszer, megnéztem a megfelelő tesztek eredményét. A tiltás felhasználó része működött, példának a Microsoft Graph API-ban egy felhasználó végleges törlése látható, és az, hogy nem engedi át a kérést a Zorp.



2. Képernyőkép Felhasználó törlése tiltva van

Ezzel sikerült a Users zóna tesztelése.

8.3.2. ADMINS

Az Admins zónával sokat nem érdemes foglalkozni, mivel itt ugyanaz a helyzet, mint az előzőnél.

Az implementáció miatt csak azt kell ellenőrizni, hogy minden művelet átmegy-e. Ha igen, akkor a tesztelés ezen része kész.

Zónamódosítást kellett végrehajtanom, hogy a 192.168.56.0/25-ös tartomány benne legyen az Admins zónában. Ez megtörtént, és egy rendszerújraindítás után újra végrehajtottam a megfelelő projekteket a Jenkins szerveren.

Itt is ugyanazt kaptam, mint a Users esetén, hogy a tiltás admin része megfelelően működik.

8.3.3. MANAGERS

A Managers zóna igényli a legtöbb figyelmet. Ebben már nem olyan egyszerű a helyzet, hogy az összeset tiltja vagy engedélyezi. Itt már egy lista alapján kell dolgoznom, hogy minden művelet megfelelően tiltva, vagy engedélyezve.

Ismét végrehajtottam egy zónamódosítást, majd egyesével lefuttattam a Jenkins projekteket. Az eredményeket a tiltási táblázattal ellenőriztem. Itt már találtam eltérést, átengedést miközben tiltás kellett volna, és fordítva. Ezeket javítottam.

A tiltás tesztek végrehajtása után a modul késznek tekinthető, mivel a naplózást már korábban teszteltem.

9. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A modult, mint az előző munkámban, itt is lehetséges továbbfejleszteni. A módszer hasonló, mint a szakdolgozatomban.

Ebben a szekcióban alapul lehet venni ezt a munkát, és hogy ez milyen módon bővíti a korábbi modul funkcionalitását.

Az első opció a felhasznált API-k növelése. Ez megtörtént a jelenlegi munkámban, de a Microsoft Graph API és a Zoom API bevonása is csak nagyon kis százaléka a teljes potenciálnak. Az interneten található API-k nagy részét lehetne implementálni a modulba.

Némileg csökkentve ezt a kört, példának a Microsoft Graph API kihagyott részeit hoznám fel. Ebben a Microsoft által szolgáltatott funkciók komoly része megtalálható olyan állapotban, hogy implementálható lenne.

Egy másik szolgáltatás, amit említettem, az a Google Meet. Ez fontos lenne, és szintén egy elég sokat használt online konferenciaszolgáltatás, de a Google nem adott ki rá nyilvános API-t, amivel vezérelhetni lehetne. Emiatt kénytelen vagyok kihagyni ezt a szolgáltatást a modulból.

A másik továbbfejlesztési módszer a funkcionalitás növelése. Ez az előző modulban a tiltás volt, amit a jelenlegiben meg is valósítottam. Egy újabb funkciója a titkosítás.

Ehhez egy módszer a feltöltött fájlok titkosítása, hogy esetleges belső anyag szivárgása esetén ne legyen mindenki által olvasható és felhasználható a dokumentum.

Még egy opció a válaszkódok figyelése, és azok naplózása, illetve az alapján a visszaküldött hibák keresése.

Illetve lehetséges a jelenlegi modulom implementálása több Zorp modul összességébe, hogy együtt működjön velük.

10. ÖSSZEFOGLALÁS

10.1. MAGYAR NYELVŰ ÖSSZEFOGLALÁS

A feladatom egy tiltó és naplózó modul létrehozása volt, ami az előző, alapszakos szakdolgozatomban létrehozott modult felhasználja, és kibővíti. A bővítéshez benne kell lennie online munkaszolgáltatásoknak, mivel az utóbbi időben nagyon megnövekedett az online, otthoni munka, és az ehhez szükséges szoftverek használata.

Felhasznált API-knak a Microsoft Graph API Teamsre vonatkozó részét, illetve a Zoom API-t választottam. Elemzés után kiválasztottam, és leírtam a számomra szükséges műveleteket. A tiltás elvégzéséhez ezeket a műveleteket kategorizáltam és elemeztem a veszélyt. Ezt az elemzést alapul vettem a tiltás programozásánál, három felhasználó kategóriát létrehozva hozzá: Felhasználó, Menedzser, Admin.

Az implementációt két fázisban végeztem el. Az első során a korábbi modul struktúráját alapul véve, de kibővítve bevezettem a két új API-t a modulba, hogy legyenek naplózva. Az első fázisban létrehozott naplózó struktúrát kibővítettem a tiltás implementációjával azokon a műveleteken, amiken az elemzés során ezt szükségesnek véltem.

A tesztelést szintén két fázisban végeztem el. Az elsőben az API-k szolgáltató cégek által nyújtott eszközöket használtam a modul tesztelésére, ellenőrizve az összes művelet korrekt naplózását. A Zoom API során akadt probléma, így kénytelen voltam egy Jenkins szervert létrehozni a cURL használatához, ami automatizálta a tesztelés egy részét. A második fázisban ezt a szervert használtam fel a tiltás tesztelésére, kibővítve a többi API-ban található tiltandó műveletekkel.

Végeredményként a projekt során létrehoztam egy olyan modult, ami egy Zorp tűzfalba illesztve képes védeni egy vállalat belső munkáját a felhasználók munkájának naplózásával és kéretlen műveletek tiltásával. Mindezzel biztonságosabbá téve a dolgozók munkáját cégen belül, illetve otthonról is.

10.2. ENGLISH SUMMARY

My task was to create a blocking and auditing module, which uses and expands the module created for my thesis during my BSc studies. The expansion has to include online work services, since online work from home rapidly increased lately, including the use of softwares neccessary for such work.

I chose to use the Zoom API, and the Microsoft Graph API's part that is relevant to Teams. After analyzing it, I chose and noted down the functions neccessary to me. To carry out the blocking, I categorized these functions, and examined the dangers. This examination was used as a base to program the blocking, creating three categories: User, Manager, Admin.

Implementation was done in two phases. In the first, I took the previous module's structure as a base, but expanding on it, I introduced the two new API's into the module, so they were being logged. I expanded on this structure in the next phase, implementing the blocking on the functions that I thought neccessary during the examination.

Testing was also done in two phases. During the first phase, I used the tools provided by the companies that provide the APIs to test the module, checking the correct logging of all the tasks. This caused a problem with the Zoom API, which forced me to utilize cURL, and to create a Jenkins server to automate part of the testing. I used this server in the second phase to test the blocking, expanding it with the functions to be blocked from the other APIs.

The end result of the project was a module, which, when inserted into a Zorp firewall, can protect the internal work of a company by logging the employees' work, and blocking the unwanted functions. This makes the work of employees safer both from withing the company, and from home.

IRODALOMJEGYZÉK

[1]Bayern, M.: 84% of businesses will increase remote work after COVID-19, despite security concerns (https://www.techrepublic.com/article/84-of-businesses-will-increase-remote-work-after-covid-19-despite-security-concerns/), utoljára megtekintve: 2022. 04. 28.

[2]Google: API Reference (https://developers.google.com/drive/api/v3/reference), utoljára megtekintve: 2022. 04. 28.

[3]Google: Gmail API (https://developers.google.com/gmail/api/reference/rest), utoljára megtekintve: 2022. 04. 28.

[4]Google: Managing Threads (https://developers.google.com/gmail/api/guides/threads), utoljára megtekintve: 2022. 04. 28.

[5]Google: People API (https://developers.google.com/people/api/rest), utoljára megtekintve: 2022. 04. 28.

[6]Google: Google Docs API (https://developers.google.com/docs/api/reference/rest), utoljára megtekintve: 2022. 04. 28.

[7]Microsoft: Microsoft Teams API Overview (https://docs.microsoft.com/en-us/graph/teams-concept-overview), utoljára megtekintve: 2022. 04. 28.

[8]Microsoft: Use the Microsoft Graph API to work with Microsoft Teams (https://docs.microsoft.com/en-us/graph/api/resources/teams-api-overview?view=graph-rest-1.0), utoljára megtekintve: 2022. 04. 28.

[9]Microsoft: Create user (https://docs.microsoft.com/en-us/graph/api/user-post-users?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[10]Microsoft: Delete user (https://docs.microsoft.com/en-us/graph/api/user-delete?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[11]Microsoft: Update user (https://docs.microsoft.com/en-us/graph/api/user-update?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[12]Microsoft: Grant an appRoleAssignment to a user (https://docs.microsoft.com/en-us/graph/api/user-post-approleassignments?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[13]Microsoft: Delete an appRoleAssignment granted to a user (https://docs.microsoft.com/en-us/graph/api/user-delete-approleassignments?view=graph-rest-1.0&tabs=http), utoljára megtekinve: 2022. 04. 28.

[14]Microsoft: Create Calendar (https://docs.microsoft.com/en-us/graph/api/user-post-calendars?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[15]Microsoft: Create CalendarGroup (https://docs.microsoft.com/en-us/graph/api/user-post-calendargroups?view=graph-rest-1.0), utoljára megtekintve: 2022. 04. 28.

[16]Microsoft: Create Event (https://docs.microsoft.com/en-us/graph/api/user-post-events?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[17]Microsoft: Create Contact (https://docs.microsoft.com/en-us/graph/api/user-post-contacts?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[18]Microsoft: Create Message (https://docs.microsoft.com/en-us/graph/api/user-post-messages?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[19]Microsoft: Send mail (https://docs.microsoft.com/en-us/graph/api/user-sendmail?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[20]Microsoft: Install app for user (https://docs.microsoft.com/en-us/graph/api/userteamwork-post-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[21]Microsoft: teamsAppInstallation:update (https://docs.microsoft.com/en-us/graph/api/userteamwork-teamsappinstallation-upgrade?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[22]Microsoft: Uninstall app for user (https://docs.microsoft.com/en-us/graph/api/userteamwork-delete-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[23]Microsoft: Create team (https://docs.microsoft.com/en-us/graph/api/team-post?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[24]Microsoft: Create team from group (https://docs.microsoft.com/en-us/graph/api/team-put-teams?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[25]Microsoft: Update team (https://docs.microsoft.com/en-us/graph/api/team-update?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[26]Microsoft: Delete group (https://docs.microsoft.com/en-us/graph/api/group-delete?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[27]Microsoft: Add member to team (https://docs.microsoft.com/en-us/graph/api/team-post-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[28]Microsoft: Update member in team (https://docs.microsoft.com/en-us/graph/api/team-update-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[29]Microsoft: Remove member from team (https://docs.microsoft.com/en-us/graph/api/team-delete-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[30]Microsoft: Archive team (https://docs.microsoft.com/en-us/graph/api/team-archive?view=graph-rest-1.0), utoljára megtekintve: 2022. 04. 28.

[31]Microsoft: Unarchive team (https://docs.microsoft.com/en-us/graph/api/team-unarchive?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[32]Microsoft: Clone team (https://docs.microsoft.com/en-us/graph/api/team-clone?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[33]Microsoft: Create group (https://docs.microsoft.com/en-us/graph/api/group-post-groups?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[34]Microsoft: Update group (https://docs.microsoft.com/en-us/graph/api/group-update?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[35]Microsoft: Delete group (https://docs.microsoft.com/en-us/graph/api/group-delete?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[36]Microsoft: Add member (https://docs.microsoft.com/en-us/graph/api/group-post-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[37]Microsoft: Remove member (https://docs.microsoft.com/en-us/graph/api/group-delete-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[38]Microsoft: Add group owner (https://docs.microsoft.com/en-us/graph/api/group-post-owners?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[39]Microsoft: Remove owner (https://docs.microsoft.com/en-us/graph/api/group-delete-owners?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[40]Microsoft: Channel resource type (https://docs.microsoft.com/en-us/graph/api/resources/channel?view=graph-rest-1.0), utoljára megtekintve: 2022. 04. 28.

[41]Microsoft: Create channel (https://docs.microsoft.com/en-us/graph/api/channel-post?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[42]Microsoft: Patch channel (https://docs.microsoft.com/en-us/graph/api/channel-patch?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[43]Microsoft: Delete channel (https://docs.microsoft.com/en-us/graph/api/channel-delete?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[44]Microsoft: Add member to channel (https://docs.microsoft.com/en-us/graph/api/channel-post-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[45]Microsoft: Update member in channel (https://docs.microsoft.com/en-us/graph/api/channel-update-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[46]Microsoft: Remove member from channel (https://docs.microsoft.com/en-us/graph/api/channel-delete-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[47]Microsoft: Send chatMessage in a channel (https://docs.microsoft.com/en-us/graph/api/channel-post-messages?view=graph-rest-1.0), utoljára megtekintve: 2022. 04. 28.

[48]Microsoft: Create chat (https://docs.microsoft.com/en-us/graph/api/chat-post?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[49]Microsoft: Update chat (https://docs.microsoft.com/en-us/graph/api/chat-patch?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[50]Microsoft: Add member to chat (https://docs.microsoft.com/en-us/graph/api/chat-post-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[51]Microsoft: Remove member from chat (https://docs.microsoft.com/en-us/graph/api/chat-delete-members?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[52]Microsoft: Publish teamsApp (https://docs.microsoft.com/en-us/graph/api/teamsapp-publish?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[53]Microsoft: Update teamsApp (https://docs.microsoft.com/en-us/graph/api/teamsapp-update?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[54]Microsoft: Add app to team (https://docs.microsoft.com/en-us/graph/api/team-post-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[55]Microsoft: teamsAppInstallation: upgrade (https://docs.microsoft.com/en-us/graph/api/team-teamsappinstallation-upgrade?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[56]Microsoft: Remove app from team (https://docs.microsoft.com/en-us/graph/api/team-delete-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[57]Microsoft: Add app to chat (https://docs.microsoft.com/en-us/graph/api/chat-post-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[58]Microsoft: teamsAppInstallation: upgrade (https://docs.microsoft.com/en-us/graph/api/chat-teamsappinstallation-upgrade?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[59]Microsoft: Uninstall app in a chat (https://docs.microsoft.com/en-us/graph/api/chat-delete-installedapps?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[60]Microsoft: Create event (https://docs.microsoft.com/en-us/graph/api/group-post-events?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[61]Microsoft: Update event (https://docs.microsoft.com/en-us/graph/api/event-update?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[62]Microsoft: Delete event (https://docs.microsoft.com/en-us/graph/api/event-delete?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[63]Microsoft: Add attachment (https://docs.microsoft.com/en-us/graph/api/event-post-attachments?view=graph-rest-1.0&tabs=http), utoljára megtekintve: 2022. 04. 28.

[64]Zoom: Introduction to Zoom API (https://marketplace.zoom.us/docs/api-reference/introduction), utoljára megtekintve: 2022. 04. 28.

[65]Zoom: Create a sub account (https://marketplace.zoom.us/docs/api-reference/zoom-api/accounts/accountcreate), utoljára megtekintve: 2022. 04. 28.

[66]Zoom: Disassociate a sub account (https://marketplace.zoom.us/docs/api-reference/zoom-api/accounts/accountdisassociate), utoljára megtekintve: 2022. 04. 28.

[67]Zoom: Update locked settings (https://marketplace.zoom.us/docs/api-reference/zoom-api/accounts/updateaccountlocksettings), utoljára megtekintve: 2022. 04. 28.

[68]Zoom: Update the account owner (https://marketplace.zoom.us/docs/api-reference/zoom-api/accounts/updateaccountowner), utoljára megtekintve: 2022. 04. 28.

[69]Zoom: Create a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/createchannel), utoljára megtekintve: 2022. 04. 28.

[70]Zoom: Update a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/updateuserlevelchannel), utoljára megtekintve: 2022. 04. 28.

[71]Zoom: Delete a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/deleteuserlevelchannel), utoljára megtekintve: 2022. 04. 28.

[72]Zoom: Remove a member (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/removeauserlevelchannelmember), utoljára megtekintve: 2022. 04. 28.

[73]Zoom: Join a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/joinchannel), utoljára megtekintve: 2022. 04. 28.

[74]Zoom: Leave a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels/leavechannel), utoljára megtekintve: 2022. 04. 28.

[75]Zoom: Update a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels-account-level/updatechannel), utoljára megtekintve: 2022. 04. 28.

[76]Zoom: Delete a channel (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels-account-level/deletechannel), utoljára megtekintve: 2022. 04. 28.

[77]Zoom: Invite channel members (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels-account-level/invitechannelmembers), utoljára megtekintve: 2022. 04. 28.

[78]Zoom: Remove a member (https://marketplace.zoom.us/docs/api-reference/zoom-api/chat-channels-account-level/removeachannelmember), utoljára megtekintve: 2022. 04. 28.

[79]Zoom: Delete meeting recordings (https://marketplace.zoom.us/docs/api-reference/zoom-api/cloud-recording/recordingdelete), utoljára megtekintve: 2022. 04. 28.

[80]Zoom: Delete a meeting recording file (https://marketplace.zoom.us/docs/api-reference/zoom-api/cloud-recording/recordingdeleteone), utoljára megtekintve: 2022. 04. 28.

[81]Zoom: Create a recording registrant (https://marketplace.zoom.us/docs/api-reference/zoom-api/cloud-recording/meetingrecordingregistrantcreate), utoljára megtekintve: 2022. 04. 28.

[82]Zoom: Create a group (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/groupcreate), utoljára megtekintve: 2022. 04. 28.

[83]Zoom: Update a group (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/groupupdate), utoljára megtekintve: 2022. 04. 28.

[84]Zoom: Delete a group (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/groupdelete), utoljára megtekintve: 2022. 04. 28.

[85]Zoom: Add group members (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/groupmemberscreate), utoljára megtekintve: 2022. 04. 28.

[86]Zoom: Update a group member (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/updateagroupmember), utoljára megtekintve: 2022. 04. 28.

[87]Zoom: Delete a group member (https://marketplace.zoom.us/docs/api-reference/zoom-api/groups/groupmembersdelete), utoljára megtekintve: 2022. 04. 28.

[88]Zoom: Create a meeting (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingcreate), utoljára megtekintve: 2022. 04. 28.

[89]Zoom: Update a meeting (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingupdate), utoljára megtekintve: 2022. 04. 28.

[90]Zoom: Delete a meeting (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingdelete), utoljára megtekintve: 2022. 04. 28.

[91]Zoom: Update meeting status (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingstatus), utoljára megtekintve: 2022. 04. 28.

[92]Zoom: Add meeting registrant (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingregistrantcreate), utoljára megtekintve: 2022. 04. 28.

[93]Zoom: Update registrant's status (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/meetingregistrantstatus), utoljára megtekintve: 2022. 04. 28.

[94]Zoom: Use in-Meeting recording controls (https://marketplace.zoom.us/docs/api-reference/zoom-api/meetings/inmeetingrecordingcontrol), utoljára megtekintve: 2022. 04. 28.

[95]Zoom: Set up a Zoom Phone account (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/setupaccount), utoljára megtekintve: 2022. 04. 28.

[96]Zoom: Set up shared access (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/addusersetting), utoljára megtekintve: 2022. 04. 28.

[97]Zoom: Update shared access (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/updateusersetting), utoljára megtekintve: 2022. 04. 28.

[98]Zoom: Remove shared access (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/deleteusersetting), utoljára megtekintve: 2022. 04. 28.

[99]Zoom: Assign phone number to user (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/assignphonenumber), utoljára megtekintve: 2022. 04. 28.

[100]Zoom: Unassign phone number (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/unassignphonenumber), utoljára megtekintve: 2022. 04. 28.

[101]Zoom: Delete a user's call log (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone/deletecalllog), utoljára megtekintve: 2022. 04. 28.

[102]Zoom: Add a device (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-devices/addphonedevice), utoljára megtekintve: 2022. 04. 28.

[103]Zoom: Update a device (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-devices/updateadevice), utoljára megtekintve: 2022. 04. 28.

[104]Zoom: Delete a device (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-devices/deleteadevice), utoljára megtekintve: 2022. 04. 28.

[105]Zoom: Create a shared line group (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-shared-line-groups/createasharedlinegroup), utoljára megtekintve: 2022. 04. 28.

[106]Zoom: Delete a shared line group (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-shared-line-groups/deleteasharedlinegroup), utoljára megtekintve: 2022. 04. 28.

[107]Zoom: Add members to a shared line group (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-shared-line-groups/addmemberstosharedlinegroup), utoljára megtekintve: 2022. 04. 28.

[108]Zoom: Unassign members of a shared line group (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-shared-line-groups/deletemembersofslg), utoljára megtekintve: 2022. 04. 28.

[109]Zoom: Unassign a member from a shared line group (https://marketplace.zoom.us/docs/api-reference/zoom-api/phone-shared-line-groups/deleteamemberslg), utoljára megtekintve: 2022. 04. 28.

[110]Zoom: Create a role (https://marketplace.zoom.us/docs/api-reference/zoom-api/roles/createrole), utoljára megtekintve: 2022. 04. 28.

[111]Zoom: Delete a role (https://marketplace.zoom.us/docs/api-reference/zoom-api/roles/deleterole), utoljára megtekintve: 2022. 04. 28.

[112]Zoom: Assign a role (https://marketplace.zoom.us/docs/api-reference/zoom-api/roles/addrolemembers), utoljára megtekintve: 2022. 04. 28.

[113]Zoom: Update role information (https://marketplace.zoom.us/docs/api-reference/zoom-api/roles/updaterole), utoljára megtekintve: 2022. 04. 28.

[114]Zoom: Unassign a role (https://marketplace.zoom.us/docs/api-reference/zoom-api/roles/rolememberdelete), utoljára megtekintve: 2022. 04. 28.

[115]Zoom: Add a Zoom Room. (https://marketplace.zoom.us/docs/api-reference/zoom-api/rooms/addaroom), utoljára megtekintve: 2022. 04. 28.

[116]Zoom: Delete a Zoom Room (https://marketplace.zoom.us/docs/api-reference/zoom-api/rooms/deleteazoomroom), utoljára megtekintve: 2022. 04. 28.

[117]Zoom: Change a Zoom Room's location (https://marketplace.zoom.us/docs/api-reference/zoom-api/rooms/changezrlocation), utoljára megtekintve: 2022. 04. 28.

[118]Zoom: Check-in or check-out of a Zoom Room (https://marketplace.zoom.us/docs/api-reference/zoom-api/rooms/checkinrooms), utoljára megtekintve: 2022. 04. 28.

[119]Zoom: Create users (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/usercreate), utoljára megtekintve: 2022. 04. 28.

[120]Zoom: Delete a user (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userdelete), utoljára megtekintve: 2022. 04. 28.

[121]Zoom: Add assistants (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userassistantcreate), utoljára megtekintve: 2022. 04. 28.

[122]Zoom: Delete user assistants (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userassistantsdelete), utoljára megtekintve: 2022. 04. 28.

[123]Zoom: Delete a user assistant (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userassistantdelete), utoljára megtekintve: 2022. 04. 28.

[124]Zoom: Update user status (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userstatus), utoljára megtekintve: 2022. 04. 28.

[125]Zoom: Update a user's password (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/userpassword), utoljára megtekintve: 2022. 04. 28.

[126]Zoom: Update a user's email (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/useremailupdate), utoljára megtekintve: 2022. 04. 28.

[127]Zoom: Switch a user's account (https://marketplace.zoom.us/docs/api-reference/zoom-api/users/switchuseraccount), utoljára megtekintve: 2022. 04. 28.

[128]Zoom: Create a webinar (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarcreate), utoljára megtekintve: 2022. 04. 28.

[129]Zoom: Delete a webinar (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinardelete), utoljára megtekintve: 2022. 04. 28.

[130]Zoom: Update webinar status (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarstatus), utoljára megtekintve: 2022. 04. 28.

[131]Zoom: Add panelists (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarpanelistcreate), utoljára megtekintve: 2022. 04. 28.

[132]Zoom: Remove panelists (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarpanelistsdelete), utoljára megtekintve: 2022. 04. 28.

[133]Zoom: Remove a panelist (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarpanelistdelete), utoljára megtekintve: 2022. 04. 28.

[134]Zoom: Add a webinar registrant (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarregistrantcreate), utoljára megtekintve: 2022. 04. 28.

[135]Zoom: Perform batch registration (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/addbatchwebinarregistrants), utoljára megtekintve: 2022. 04. 28.

[136]Zoom: Update registrant's status (https://marketplace.zoom.us/docs/api-reference/zoom-api/webinars/webinarregistrantstatus), utoljára megtekintve: 2022. 04. 28.