SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1177

Izrada kibernetičke vježbe za zdravstveni sektor upotrebom alata OpenEx

Vatroslav Jakopec

Zagreb, srpanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 1177

Pristupnik:	**Vatroslav Jakopec (0036535110)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Stjepan Groš

Zadatak: **Kibernetička vježba za zdravstveni sektor ostvarena upotrebom alata OpenEx**

Opis zadatka:

Kibernetičke vježbe bitan su dio izgradnje sposobnosti obrane tvrtki. Svrha kibernetičkih vježbi je uvježbavanje timova koji brane neki informacijski sustav kako bi u slučaju stvarnih napada reagirali što bolje i efikasnije te na taj način umanjili štetu koja nastaje organizaciji. Kibernetičke vježbe mogu se provoditi na tehničkoj, taktičkoj, operativnoj i strateškoj razini. Za svaku razinu koriste se različite metode provođenja vježbi, od kojih su najčešće "table-top" vježbe za upravljačku razinu te kibernetički poligoni za tehničku razinu. U sklopu završnog rada potrebno je definirati kibernetičku vježbu za upravljačku razinu neke zdravstvene organizacije. Vježbu je potrebno implementirati i provoditi korištenjem alata otvorenog koda OpenEx. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

SADRŽAJ

1. Uvod

Kibernetičke vježbe važne su za izgradnju i testiranje obrambenih sposobnosti tvrtke od napada. Njihov je cilj uvježbavanje i pripremanje timova koji svakodnevno koriste računalnu opremu i informacijske sustave na mogućnost stvarnog napada. Kroz provođenje vježbe mogu se identificirati snage i slabosti u sustavu ili propisanim procedurama firme u bezopasnom i često neformalnom okruženju. Vježbe se mogu provoditi na tehničkoj, taktičkoj, upravljačkoj i strateškoj razini. Najčešći tip su tzv. *tabletop* vježbe (engl. *Tabletop Exercise*, TTX) na upravljačkoj razini[6].

U sklopu ovog rada definira se i provodi jedna kibernetička vježba na upravljačkoj razini za firmu u zdravstvenom sektoru. Kao alat za pomoć pri izvođenju vježbe koristi se program otvorenog koda OpenEx. OpenEx je relativno nov i slabo istražen alat koji u trenutku pisanja ovog rada ima oskudnu dokumentaciju i malo javno dostupnih primjera korištenja. U tom smislu, ovaj rad služi i kao priručnik za planiranje i provođenja vježbi u OpenEx-u.

Poglavlje *Kibernetičke vježbe* je pregled osnovnih i općenitih informacija o kibernetičkim vježbama: koje vrste postoje, tko sve sudjeluje u vježbi, od čega se sve vježba sastoji te kako se obrađuju podaci prikupljeni tijekom vježbe. U poglavlju *OpenEx* prolazi se kroz aplikaciju za izradu i provođenje kibernetičkih vježbi OpenEx. Uz upute za instalaciju i objašnjene su osnove tehnologije *docker* na kojoj je zasnovano pokretanje aplikacije. U detalje su opisane mogućnosti i funkcionalnosti aplikacije po pojedinim stranicama. Dan je osvrt na korisnost aplikacije te na koje se sve načine trenutno može iskoristiti s obzirom da je i dalje u razvoju. U poglavlju *Razvijena vježba* opisana je kibernetička vježba za zdravstveni sektor napravljena u sklopu ovog rada te njezina implementacija u OpenEx.

2. Kibernetičke vježbe

U ovom poglavlju nalazi se općenit pregled bitnih pojmova o kibernetičkim vježbama. Dan je uvod u kibernetičke vježbe gdje je ukratko opisano što su i koja im je svrha. Nakon toga su navedene vrste i podjele. Objašnjeni su svi dionici u životnom ciklusu vježbe, od planiranja i provođenja do obrade prikupljenih podataka.

2.1. Uvod u kibernetičke vježbe

Kibernetičke vježbe su simulacije mogućih kibernetičkih napada na firmu koje se provode radi testiranja i unaprijeđena znanja te spremnosti zaposlenika u slučaju stvarnog napada. Jedna od glavnih značajki takvih vježbi je da se provode u kontroliranoj i bezopasnoj okolini. Ovisno o okolnostima, vježba može biti više ili manje formalna. Vježbe su dizajnirane tako da potiču ljude iz različitih dijelova tvrtke da razmišljaju o svojim postupcima, poboljšavaju komunikaciju s ostalim kolegama i izmjenjuju ideje.

Moguće je da postoje nepoznate slabosti u propisanim procedurama i nedoumice oko odgovornosti u kritičnim situacijama, što dovodi do usporene komunikacije. Kroz vježbu i simuliranje napada, ovakvi problemi često postaju vidljivi i lakši za identifikaciju, što omogućuje da se uvedu potrebne promjene, rješenja i popravci. Provođenje vježbe može naučiti zaposlenike da prepoznaju sumnjive situacije i pokazati važnost pravilnog reagiranja na iste. Kako su najčešći napadi oni koji iskorištavaju ljudsku slabost i grešku, pravilna edukacija o sigurnosnim postupcima omogućava većem broju zaposlenika da aktivno sudjeluju u zaštiti sustava i suzbijanju napada. Visoka razina spremnosti i svijesti zaposlenika o mogućnosti kibernetičkog napada ne može spriječiti sam napad, ali ima potencijal ubrzati oporavak od napada te ublažiti posljedice.

Posljedice pretrpljenog napada mogu biti razne. Ovisno o vrsti napada, moguće je da procure razni povjerljivi podaci o firmi ili njenim klijentima. Klijenti mogu biti poslovni partneri ili pojedinci koji koriste usluge koja firma pruža. U svakom slučaju, ovisno o prirodi napada, može doći do curenja povjerljivih informacija, što će zasigurno stvoriti probleme i klijentima. Takve situacije često za sobom povlače i pravne

posljedice. Nadležne institucije i oni klijenti čiji su podaci procurili napadom moraju biti obaviješteni o tome što se dogodilo. Nadležne institucije mogu zahtijevati inspekcije infrastrukture firme te propisati novčane kazne. Klijenti u nekim slučajevima mogu tužiti firmu za štetu koju su oni pretrpjeli uslijed napada. Sve ovo može biti težak udarac za reputaciju tvrtke i njezino financijsko stanje. Očigledno postoji širok spektar mogućih posljedica i problema koji dolaze tek nakon što napad završi. Provođenjem dobro dizajniranih kibernetičkih vježbi sve se ove posljedice nastoje minimizirati.

2.2. Vrste kibernetičkih vježbi

Kibernetičke vježbe mogu se provoditi na tehničkoj, taktičkoj, upravljačkoj i strateškoj razini.

Taktičke vježbe služe evaluaciji i unaprijeđenu tehničkih vještina osoblja koje je direktno zaduženo za kibernetičku sigurnost. U ovim vježbama, igrači aktivno odgovaraju na simulaciju konkretnog napada, često modeliranog po pravom iz prošlosti. Koriste se alati za detektiranje i analizu prijetnji. Vještine koje se ovim vježbama unaprijeđuju mogu biti analiza zloćudnog programa, penetracijsko testiranje, nadgledanje mreže, detekcija incidenta i ostale slične vještine.

Tehničke vježbe bave se infrastrukturom i informacijskim sustavom firme. Testira se otpornost mrežne infrastrukture, programske potpore i uređaja na slabosti i napade. Pregledavaju se sigurnosne postavke kao kontrola pristupa, autentifikacija, računalni sustav za otkrivanje napada.

Upravljačke vježbe fokusiraju se na poboljšanje koordinacije, komunikacije i internih procesa u odgovoru na napad. Testira se način na koji različiti timovi komuniciraju, kako se dijele informacije, spremnost na incident, upravljanje resursima, učinkovitost internih procesa i protokola u kritičnim situacijama.

Strateške vježbe bave se donošenja dugoročnih odluka u firmi koje se ne mogu donijeti u kratkom roku.

HSEEP navodi pet vrsta vježbi za kibernetičku sigurnost[6]. Te vrste su seminar, radionica, *tabletop* vježba, igra i funkcionalna vježba. Ovih pet vrsta se može razvrstati u diskusijske (engl. *discussion-based*) ili operativne (engl. *operation-based*). Operativne vježbe fokusiraju se na detaljno simuliranje tehničkih aspekata napada. Raspravljačke vježbe fokusiraju se na hipotetske scenarije, raspravu o postupcima i o donošenju odluka. Seminar (engl. *seminar*) je rasprava dizajnirana da igrače upozna s novim ili promijenjenim planovima, politikama i internim procesima i protokolima. Radionica (engl. *workshop*) je poput seminara, ali je cilj stvoriti određeni plan. U

tabletop vježbi (engl. *tabletop exercise*, TTX) igrači iz različitih timova raspravljaju o hipotetskom scenariju u opuštenoj okolini te se evaluira proces donošenja odluka. Igra (engl. *game*) je simulacija u kojoj postoje barem dva tima u natjecateljskom okruženju. Postoje pravila i određeni cilj. Funkcionalna vježba (engl. *functional exercise*, FE) je operativna vježba koja ocjenjuje koordinaciju, zapovijedanje i kontrolu.

Ostatak ovog rada bavi se vježbama na upravljačkoj razini.

2.3. Autori, voditelji i sudionici vježbe

Jedan od najbitnijih faktora za provođenje uspješne vježbe je vješt tim za planiranje vježbe (engl. *Exercise Planning Team*). Tim za planiranje vježbe odgovoran je za skoro svaki aspekt vježbe. Tim nadzire, dizajnira, razvija, i najčešće se bavi provođenjem i evaluacijom vježbe. To uključuje određivanje konkretnih ciljeva za vježbu, smišljanje scenarija i pisanje dokumentacije.

Sudionici vježbe koji su potrebni za provedbu su igrači, promatrači, moderatori vježbe i sakupljači podataka[6]. *Igrači* (engl. *players*) su zaposlenici firme. Tijekom vježbe, najčešće su raspoređeni u nekoliko smisleno oblikovanih timova. Reagiraju na predstavljenu situaciju onako kako bi da su se stvarno našli u njoj. Odluke donose na temelju svog stručnog znanja, znanja internih protokola i politika tvrtke te iskustvom s dodatnih obuka. *Promatrači* (engl. *observers*), kao što ime kaže, promatraju vježbu i ne sudjeluju u raspravama i donošenju odluka. To su najčešće zaposlenici za koje nema mjesta u vježbi ali im je moderator vježbe dopustio promatranje. *Moderator vježbe* (engl. *facilitator*) je osoba koja je dobro upoznata s vježbom, najčešće iz samog tima za planiranje vježbe. Poželjno je da je moderator stručna osoba u području sigurnosti ili na neki način povezana sa sektorom u kojem se provodi vježba te ako se koriste dodatni alati za provođenje vježbe, da ima dobro razumijevanje tih alata. Njihov je glavni posao zadržavanje fokusa diskusije u planiranom smjeru, tj, spriječiti igrače da previše skrenu s teme. To rade kroz komunikaciju s timovima. U slučaju da primijete da su igrači izgubili fokus na ono što je bitno, moderatori ih mogu nekim smjernicama vratiti na pravi put. Do skretanja s puta može doći iz više razloga. Moguće je i da se neki aspekti vježbe mogu krivo interpretirati, pogotovo u prvim pokušajima provođenja vježbe. U svakom slučaju, moderatori su tu da minimiziraju nepotrebno skretanje s puta i držanje fokusa na onome što je bitno za vježbu. Sakupljači podataka (engl. *data collectors*) promatraju igrače i ne sudjeluju u donošenju odluka i raspravama. Njihov je posao dokumentirati postupke igrača i pratiti propisane metrike tijekom provođenja vježbe. Ti podaci su bitni za analizu i evaluaciju nakon vježbe, a to što ih bilježe

sakupljači podataka omogućava igračima da nesmetano odrađuju svoje zadatke.

2.4. Struktura vježbe

Na kibernetičku vježbu ne bi se trebalo gledati kao test u smislu stresne situacije. Bolje je na vježbu gledati kao priliku da firme i organizacije pregledaju i isprobaju svoje planove, politike i procedure, poboljšaju koordinaciju i samopouzdanje, otkriju slabosti i vježbaju timski rad. Ne postoji popis točnih ciljeva svake pojedine vježbe, već firma ili organizacija sama mora donijeti odluku o uspješnosti i korisnosti provođenja vježbe, te jesu li rezultati zadovoljavajući i ciljevi postignuti. Jedna konkretna vježba ima sljedeću strukturu [6]:

Vježba započinje uvodnim plenumom (engl. *opening plenum*). To je orijentacijski sastanak u kojoj moderatori vježbe uvode igrače u simulirani scenarij, objašnjavaju pravila i generalno način provođenja vježbe. Navode se timovi te njihove uloge i dužnosti. Objašnjavaju se bilo kakvi dodatni alati, u slučaju da se koriste. Ukratko, u uvodnom plenumu igračima su dane sve potrebne informacije za sudjelovanje u vježbi čak ako ne znaju ništa o vježbama unaprijed.

Nakon uvoda, slijede tzv. interaktivne sjednice (engl. *interactive sessions*). Pojedini timovi se sada moraju nalaziti na zasebnim mjestima za rad. Tamo će članovi zasebnih timova voditi dijalog i raspravljati o postupcima u vježbi. Rasprava počinje nakon što se iznese prvi *inject*. *Inject* je dio scenarija koji se na neki način iznosi igračima, koji bi trebao potaknuti igrače na neku akciju i komunikaciju ili diskusiju. Na *inject* se može gledati kao na prepričavanje dijela fiktivne priče koja se događa u vježbi. *Inject* će uglavnom biti realan događaj ili podražaj, nešto s čime bi se igrači mogli susresti na radnom mjestu. Evo primjera jednog *injecta* u konvencionalnoj TTX vježbi:

Vaš računalni sustav za otkrivanje napada dojavio je pokušaj autentifikacije grubom silom pomoću protokola ssh prema nekom internom poslužitelju firme. Poznata je IP-adresa računala koje je izvor ovog pokušaja. Primijetite da ta IP-adresa pripada računalu jednog od vaših zaposlenika.

Nakon što se ovo iznese igračima, njihov je red da rasprave što bi dalje radili s tim informacijama koje su upravo dobili. *Inject* može i ne mora biti za sve igrače. U nekoj IT firmi, prethodno spomenuti primjer bio bi prikladan za tim za računalnu siguzrnost (engl. *IT Security*), a manje prikladan za tim za korisničke usluge (engl. *Customer Services*). Naravno, informacije iz *injecta* možda i jesu vrlo bitne za tim za korisničke usluge ili bilo koji drugi tim, ali u stvarnom radnom okruženju, samo bi tim

za informacijsku sigurnost izravno saznao tu informaciju. Sada je na članovima tog tima da vode raspravu o daljnjim postupcima. Njihova rasprava mogla bi teći ovako:

Kontaktirajmo osobu čija je IP-adresa, možda je kolega slučajno nešto napravio što je sustav ovako protumačio. Provjerimo za svaki slučaj taj poslužitelj za neobične aktivnosti. Uzima li sustav u obzir interne IP-adrese unutar firme? Je li javno dostupna adresa tog poslužitelja?

Moderator vježbe mogao bi se uključiti u njihovu raspravu s dodatnim upitima ako misli da je potrebno:

Dojavljujete li ovo ikome odmah, ili ćete prvo sami malo istražiti što se dogodilo?

Ovime moderator vježbe može timu skrenuti pažnju na ono što je bitnije u vježbi, ako misli da su igrači malo skrenuli s puta. Dalje se odvija komunikacija između timova, te svaki tim onda odrađuje svoje akcije i vodi raspravu, onako kako misle da je potrebno. Moderatori ostavljaju igračima predviđeno vrijeme za njihovu raspravu i akcije, nakon čega će doći novi *inject*. *Injecti* mogu i ne moraju biti strogo tempirani. Ništa ne brani moderatorima da iznose *injecte* onako kako misle da je najbolje, čak ako je to drugačije od onoga što je u planu vježbe. U nekim slučajevima, ako moderatori misle da je tako najbolje, mogu izbaciti ili na mjestu osmisliti nove *injecte*, da se prilagode igračima i vježbi. Dobri moderatori vježbe će primijetiti takve prilike i prilagoditi se situaciji i igračima da stvore bolje iskustvo vježbe.

2.5. Nakon vježbe

Odmah nakon vježbe, slijedi *hot wash*[6], tijekom kojeg igrači daju povratne informacije. Bitno je da se to odradi odmah nakon vježbe, dok su dojmovi i sjećanja još svježa. Osim toga, ovo je prilika za sakupljače podataka da dopišu podatke koje iz bilo kojih razloga tijekom vježbe nisu bili zapisani u cijelosti. Igračima se mogu podijeliti i upitnici s pitanjima gdje se ocjenjuje zadovoljstvo pojedinim aspektima vježbe. Nakon *hot washa* s igračima, moderatori vježbe i sakupljači podataka kratko raspravljaju o provedenoj vježbi.

2.6. Prikupljanje i obrada podataka

Prije provođenja vježbe bitno je definirati koji podaci su bitni i trebali bi se zapisivati, a koji ne[6]. Dobra definicija podataka koji će se pratiti olakšava prikupljanje i obradu te osigurava da će povratna informacija biti korisna. Za lakše zapisivanje podataka

potrebno je da moderator igre i sakupljači podataka budu dobro upoznati s procesima unutar firme te odnosom među pozicijama zaposlenika. S tim znanjem, moderatori mogu bolje odrediti što točno sakupljači trebaju pratiti. Ono što se može uzeti kao opće pravilo i pratiti za svaku vježbu je ime igrača koji je donio pojedinu odluku, koja je ta odluka bila, zašto je napravljena baš ta odluka i kako je tim reagirao na tu odluku.

Cilj analize podataka je izvući neke jasne zaključke iz provedene vježbe. Sakupljači podataka analiziraju svoje bilješke o raspravama koje su igrači vodili i njihove postupke. Ovo se uspoređuje s postojećim planovima. U ovom trenu će se jasno moći utvrditi koliko su se tijekom igre pratili postojeći planovi. U slučaju odstupanja od planova, iz podataka bi se trebalo moći zaključiti što je dovelo do takvog razvoja situacije. Bitno je odrediti zašto se nešto odigralo onako kako je. Kada se to analizira, može se osmisliti kako rješiti identificirane probleme.

3. OpenEx

OpenEx je aplikacija otvorenog koda koja se može koristiti kao alat pri izvođenju kibernetičkih vježbi[4]. OpenEx omogućava planiranje *injecta* u vremenu, automatizira prikupljanje povratnih informacija od igrača, pomaže organizaciji timova i dozvoljava prijenos vježbi u zip arhivi kako bi se lakše dijelile. *OpenEx* razvija francuska tvrtka Filigran[2]. Filigran pruža usluge obavještajnog rada o kibernetičkim prijetnjama te programsku potporu za planiranje i provođenje kibernetičkih vježbi. Na njihovoj internetskoj stranici postoji demo verzija aplikacije. Demo verzija služi za pregled mogućnosti aplikacije i da se stekne dojam o tome kako to sve izgleda i funkcionira. Ne može se koristiti za provođenje i planiranje vježbi. Nema javno upogonjene instance OpenEx aplikacije na kojoj se mogu koristiti sve njene funkcionalnosti. Javno upogonjena je samo demo verzija s minimalnim mogućnostima preglda sučelja aplikacije. Oni koji žele koristiti potupnu verziju aplikacije morati će je sami upogoniti. U sljedećem potpogtlavlju opisan je postupak pokretanja OpenEx-a. Dane su teoretske osnove potrebne za razumjevanje tehnologije pomoću koje se OpenEx pokreće popraćene praktičnim primjerima. Zatim su detaljno opisane funkcionalnosti aplikacije.

3.1. Priprema sustava za pokretanje

Moguća su dva načina pokretanja OpenEx-a. Jedna je opcija pokrenuti ga iz izvornog koda. Druga opcija je pokretanje putem *docker* kontejnera. Ta opcija detaljno je pokrivena u ovom radu. Bitno je razumjeti osnove o sustavu docker i kontejnerizaciji prije nego što se započne postupak instalacije.

3.1.1. Docker

Virtualizacija je tehnika emulacije više računala na jednom fizičkom računalu, zvanom *domaćin* (engl. *host*). Virtualizacija dijeli resurse *host* računala virtualnim računalima. Iz perspektive virtualiziranih računala, ona su međusobno izolirana kao i od računala

na kojem se izvode. Virtualizacija ima više primjena. Jedna od njih je korištenje aplikacija u izoliranom okruženju ili neovisno o operacijskom sustavu računala domaćina. Najraširenija tehnika virtualizacije je pokretanje virtualnih strojeva korištenjem programske potpore kao što su Oracle VM VirtualBox ili VMware. Virtualni stroj programski emulira sve bitne dijelove računala: memoriju, procesor, grafičku karticu, prostor za pohranu, te ako je virtualizacija na razini operacijskog sustava, virtualizira se i sama jezgra operacijskog sustava.

Docker je skup programa koji omogućavaju virtualizaciju na razini procesa. Ti procesi nazivaju se *docker* kontejneri, a tehnika virtualiziranja i izoliranja procesa zove se kontejnerizacija. Pokretanje *docker* kontejnera znatno je manje zahtjevno što se tiče računalnih resursa u usporedbi s pokretanjem virtualnih strojeva, što je velika prednost ako se koriste na pravi način. Tzv. *docker* slike (engl. *docker image*) služe kao predložak za pokretanje kontejnera. Postoje javni repozitoriji *docker* slika i mogućnost stvaranja i održavanja privatnog. Najveći repozitorij *docker* slika je *Docker Hub*. Na *Docker Hub* bilo tko može objaviti svoje *docker* slike. U većini slučajeva, ovdje se mogu pronaći sve potrebne *docker* slike.

Docker je dostupan na svim raširenijim operacijskim sustavima. Na Linux, BSD i MacOS sustavima može se instalirati putem upravitelja paketa. Za Windows sustave postoji instalacijska datoteka. Za pokretanje dockera na Windows sustavu potrebno je imati konfiguriran WSL (*Windows Subsystem for Linux*).

Slijedi jedan primjer korištenja dockera. Na lokalno će se računalo dohvatiti *docker* slika i pokrenuti kontejner u kojem je pokrenut upravitelj bazom podataka PostgreSQL. U naredbeni redak može se napisati naredba:

```
~ » docker pull postgres
```

Ovom naredbom docker sa *Docker Huba* preuzima sliku *postgres* na lokalno računalo. Izlaz naredbe izgleda ovako:

```
Using default tag: latest
latest: Pulling from library/postgres
f03b40093957: Pull complete
(...)
4ae692d11ad3: Pull complete
Digest: sha256:31c9342603866f29206a06b77c8fed48b3c...
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Sljedećom naredbom može se uvjeriti da je slika uistinu preuzeta. Nareba prikazuje listu svih *docker* slika na lokalnom računalu i neke njihove značajke poput identifikatora i količine zauzetog prostora:

```
~ » docker image ls
REPOSITORY   TAG      IMAGE ID      CREATED      SIZE
postgres     latest   0c88fbae765e  4 days ago   379MB
```

Sada se sliku može pokrenuti:

```
$ docker run \
    -d --name pg-kontejner \
    -e POSTGRES_USER=korisnik \
    -e POSTGRES_PASSWORD=tajna_lozinka \
    postgres
```

Ovom naredbom pokreće se kontejner i daje mu se ime *pg-kontejner*. Zastavicom *-d* odabire se opcija pokretanja kontejnera u odvojenom načinu rada (engl. *detached mode*), što znači da ga se pokreće u pozadini te neće zauzimati naredbeni redak. Zastavicom *-e* definiraju se parovi ključeva i vrijednosti za varijable okoline (engl. *environment variables*). Na kraju, definira se koju sliku docker koristi za pokretanje kontejnera, u ovom slučaju *postgres*. Nakon izvođenja ove naredbe upravitelj bazom podataka PostgreSQL stvarno je pokrenut na portu 5432 te se na njega može spojiti i koristiti ga. Za zaustavljanje i brisanje kontejnera, potrebno je saznati njegov identifikator (stupac `CONTAINER ID`):

```
$ docker ps
CONTAINER ID  IMAGE     COMMAND                  CREATED
0d541f02ab2a  postgres  "docker-entrypoint..." 13 min...
```

Pa ga se potom može zaustaviti te ako ga više ne želimo koristiti, može ga se i obrisati:

```
$ docker stop 0d541f02ab2a
0d541f02ab2a
$ docker rm 0d541f02ab2a
0d541f02ab2a
```

Postoje bolji načini za upravljanje većim brojem *docker* slika istovremeno ali ovo dobro ilustrira što se događa i kako docker pokreće kontejnere. OpenEx koristi nekoliko kontejnera istovremeno, pa se koristi i efikasniji način za pokretanje. Koristi se

program *docker-compose* i konfiguracijska datoteka `docker-compose.yaml`. U ovoj datoteci su u *yaml* formatu definirani svi kontejneri koji se pokreću, te njihove pripadne varijable okruženja, portovi koje će zauzeti, način perzistencije memorije, međusobna ovisnost kontejnera i slični podaci. Dobra je praksa držati varijable okruženja u zasebnoj datoteci. Docker će ih automatski potražiti u datoteci `.env` ako se nalazi u direktoriju iz kojega se pokreće. Varijable u `.env` datoteci kao parovi ključeva i vrijednosti odvojeni znakom '=' i zapisane u zasebne retke. Primjer jednog retka te datoteke je `SPRING_MAIL_PORT=25`.

3.1.2. Poslužitelj elektroničke pošte

Filigran je priredio `docker-compose.yml`[5] datoteku kojom se pokreće OpenEx i sve potrebno za rad same aplikacije, ali to nije sve što nam treba. OpenEx se koristi u kombinaciji sa poslužiteljem elektroničke pošte kako bi ostvario neke svoje funkcionalnosti. Taj poslužitelj nije osiguran u `docker-compose` datoteci, već ga se treba zasebno postaviti. Većina firmi ima svoje poslužitelje elektroničke pošte pa se za ovaj dio instalacije ne trebaju posebno brinuti. Za ovu instalaciju, koristi se dodatan *docker* kontejner s poslužiteljem elektroničke pošte[1]. Za spajanje na poslužitelj potrebno je instalirati klijentsku aplikaciju koja podržava IMAP i SMTP protokole, što je većina klijentskih aplikacija za elektroničku poštu. U nastavku se koristi aplikacija Mozilla Thunderbird.

3.2. Instalacija

U sklopu ovog rada pripremljena je `docker-compose.yml` datoteka koja pokreće OpenEx i poslužitelj elektroničke pošte. Napisane su i skripte koje pomažu pri upravljanju računima u OpenEx aplikaciji i na poslužitelju pošte. Sve je javno dostupno na Git repozitoriju local_openex_setup[3]. U nastavku teksta pretpostavlja se korištenje tog repozitorija.

Prije pokretanja ičega, potrebno je podesiti nekoliko konfiguracijskih datoteka. Prva je `.env` datoteka, kojom ćemo definirati lozinke i korisnička imena za MinIO i PostgreSQL bazu podataka koje OpenEx koristi te neke postavke poslužitelja elektroničke pošte. Potrebno je kopirati `.env.sample` datoteku u `.env` te onda u datoteci `.env` promjeniti varijable označene sa *ChangeMe*. Nakon toga, upisuju se potrebni podatci u `players.txt` datoteku. Ovdje se definiraju pojedini igrači te njihove odgovarajuće mail adrese. U prvom redu datoteke navedeni su stupci u koje se upisuju

podaci o igračima poput imena i prezimena. Nije potrebno ispuniti svaki podatak za svakog igrača. Nužna je jedino lozinka koja će se koristiti za elektroničku poštu i barem neki podatak iz kojeg se može generirati adresa elektroničke pošte. Ako se zadaje vrijednost za stupac *email*, ne piše se *@domena*, skripta će to sama dodati.

Za lakše dodavanje računa u Thunderbird klijent, može se urediti `/etc/hosts` datoteka na Unix sustavima. Pandan ovoj datoteci na Windows sustavima je datoteka: `C:\Windows\System32\drivers\etc\hosts`. U ovu datoteku može se dodati linija `127.0.0.1 .openex.local`. Thunderbird će pretpostaviti adresu poslužitelja pomoću domene mail adrese. Ovako ćemo to moći ostaviti na `.openex.local`, te će operacijski sustav znati da se zapravo radi o `localhost`.

Sada se iz korijenskog direktorija repozitorija može pokrenuti cijeli sustav naredbom:

```
$ docker-compose up -d
```

Pri prvom pokretanju naredbe, pokreće se preuzimanje svih potrebnih *docker* slika. Ovo može potrajati nekoliko desetaka sekundi. Kada se naredba izvrši, može se provjeriti je li sve u redu naredbom:

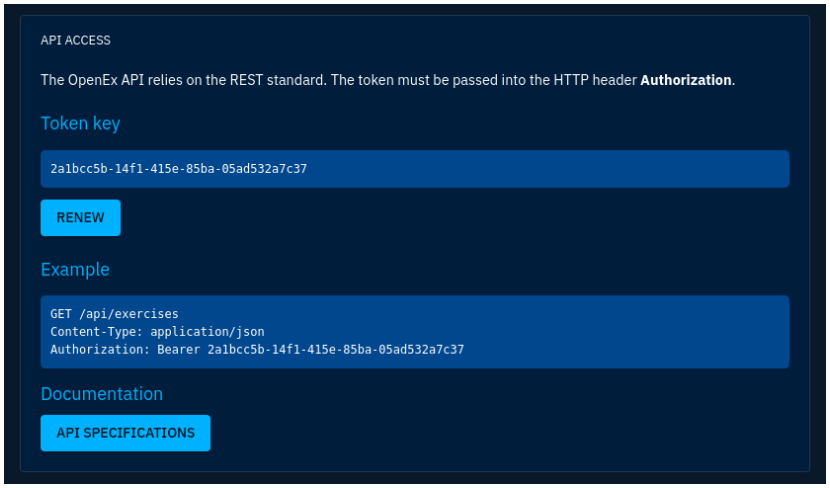
```
$ docker-compose ps
```

```
Name                                    Command                  State
-------------------------------------------------------------------------------
local_openex_setup_mailserver_1   /usr/bin/dumb-init -- supe ...   Up (healthy)




local_openex_setup_minio_1        /usr/bin/docker-entrypoint ...   Up (healthy)
local_openex_setup_openex_1       /usr/bin/tini -- java -jar ...   Up
local_openex_setup_pgsql_1        docker-entrypoint.sh postgres    Up
```

Slika 3.1: Dio izlaza naredbe "docker-compose ps"

U ispisu ove naredbe trebali bi vidjeti imena pokrenutih kontejnera, njihova stanja i zauzete portove (izostavljeno zbog širine). Moguće je da stanje kontejnera poslužitelja pošte bude *Unhealthy*, no to je u redu, i promijenit će se jednom kada se dodaju neki računi. Prije pokretanje skripte za dodavanje igrača, potrebno je prijaviti se kao administrator u OpenEx aplikaciju. Sama OpenEx aplikacija trebala bi već biti dostupna na portu 8080. Pristupa joj se internetskim preglednikom na adresi `localhost:8080`.

Zadani podaci za prijavu su adresa *admin@openex.io* i lozinka *admin*. U gornjem desnom kutu stranice, nalazi se stranica profila korisnika. Na dnu te stranice je ključ za pristup API-u pokrenute instance OpenEx-a:



Slika 3.2: Ključ za API

Potrebno je kopirati taj ključ te ga postaviti za vrijednost varijable `API_KEY` u skripti `add_player_openex.sh`. Sada se može pokrenuti skripta za dodavanje igrača:

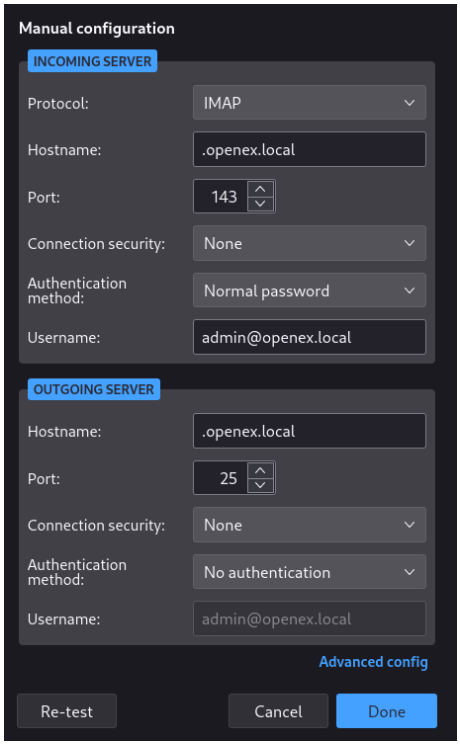
```
$ bash setup_players.sh
```

Skripta će podacima iz `players.txt` datoteke stvoriti korisnike u poslužitelju pošte pomoću skripte koja dolazi uz njega i u OpenEx aplikaciji putem API-a. Nakon što se skripta izvrši, bitno je pričekati da u izlazu naredbe `docker-compose ps` svi kontejneru budu *Up* ili *Up (healthy)*.

Sada se stvoreni računi igrača mogu koristiti za prijaviti u poslužitelj elektroničke pošte. Pregled svih uspješno dodanih računa u poslužitelj radi se naredbom:

```
$ bash mail_setup.sh email list
```

U Mozilla Thunderbird klijentkoj aplikaciji sada se može prijaviti nekim od stvorenih računa. Nakon unosa osnovnih podataka (ime, adresa, lozinka), potrebno je odabrati opciju *Configure manually*. Stavljaju se sljedeće postavke:



Slika 3.3: Postavke za poslužitelj elektroničke pošte

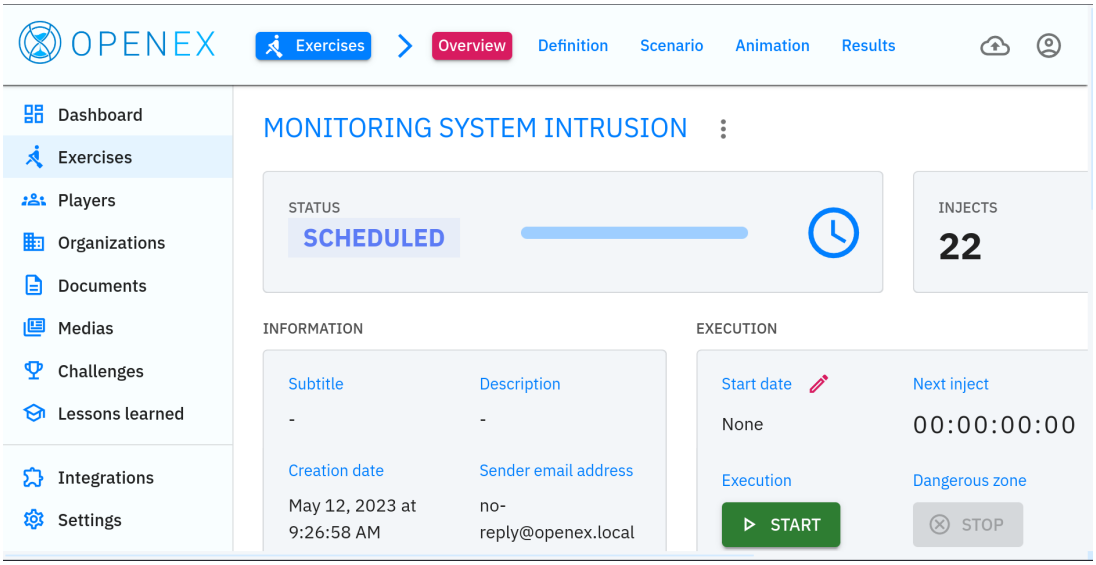
U slučaju da `/etc/hosts` nije uređen po uputama, adresu poslužitelja se na `localhost`. Pri stvaranju računa, Thunderbird upozorava da poslužitelj ne koristi kriptografiju te da nije siguran. Ovo je u redu, jer služi samo za testiranje i demonstraciju, te nije u produkcijskom okruženju. Nakon što dodate jedan račun, korisno je pokušati poslati jednu poruku pomoću tog računa, kao test je li sve u redu. Ako je, poruka bi trebala biti dostavljena, te se može nastaviti s dodavanjem ostalih računa.

Za gašenje cijelog sustava koristi se naredba `docker-compose down` pokrenuta u direktoriju gdje se nalazi `docker-compose.yml` datoteka. Perzistencija podataka je implementirana, tako da će na idućem pokretanju sustava sve će ostati onakvo kakvo je bilo pri gašenju.

3.3. OpenEx Aplikacija

Ako nisu promijenjeni, zadane vjerodajnice za prijavu u OpenEx su *admin@openex.io* i lozinka *admin*.

Pri vrhu prozora uvijek će se nalaziti zaglavlje, u kojem se na krajnje desnoj strani nalazi gumb za uvoz vježbe u platformu te izbornik za stranicu profila te odjavu iz sustava. S lijeve strane prozora uvijek će se nalaziti izbornik na kojem se bira jedan od prikaza, svaki od kojih je detaljno objašnjen u sljedećim poglavljima.



Slika 3.4: Izbornik za vježbe u OpenEx-u

3.3.1. Uvoz i izvoz vježbi

OpenEx ima mogućnost uvoza i izvoza napravljenih vježbi. Aplikacija ih pakira u zip arhive, da se mogu na praktičan način dijeliti. Opcija uvoza gotove vježbe se nalazi u zaglavlju aplikacije s desne strane (*Import an exercise*). Izvoz vježbe se provodi u pregledu određene vježbe. Pokraj imena vježbe nalazi se izbornik u kojem je jedna od opcija izvoz vježbe (*Export*). Može se odabrati uz vježbu izvesti i igrače koji su zapisani kao sudionici. To znači da će se pri uvozu ove vježbe u drugi sustav uvesti i ti igrači. Ovo generalno nije poželjno, jer postoji dobro osmišljeno rješenje na prilagodbu konkretne vježbe na različite timove i igrače, što će se detaljnije objasniti kasnije.

Na nekoliko mjesta u aplikaciji gdje postoji neka lista, kao što je lista vježbi ili lista *injecta*, postoji mogućnost izvoza te liste u CSV (*Comma Separated Values*) formatu.

3.3.2. Profil trenutnog korisnika

Krajnje desno u zaglavlju aplikacije nalazi se stranica profila trenutnog korisnika sustava (slika 3.4). Ovdje se mogu mijenjati razni korisnički podaci. Postoji polje za unos broja telefona, mobitela i PGP javnog ključa. U OpenEx se kasnije u razvoju planira dodati funkcionalnost koja koristi te podatke, u vrijeme pisanja ovoga postoji samo mogućnost pohrane.

Na dnu stranice nalazi se ključ za pristup API-ju konkretne instance OpenEx-a. Ovo je već bilo korišteno tijekom instalacije. Ključ je potrebno poslati u svakom HTTP zaglavlju upućenom API-ju. Kompletna specifikacija API-ja također se nalazi na dnu stranice.

3.3.3. Oznake

Kroz gotovo cijeli sustav koriste se oznake (engl. *tags*). Oznakama se objektima unutar sustava mogu filtrirati i grupirati. Objekti mogu biti igrači, vježbe, dokumenti, organizacije i skoro sve drugo što postoji unutar sustava aplikacije. Oznaku određuju dodijeljeno ime i boja.

3.3.4. Upravljačka ploča

Nakon uspješne prijave u aplikaciju, otvara se glavna stranica aplikacije, njezin *dashboard*. Na prvo otvaranje *dashboarda*, sva polja za prikaz će biti prazna, te će se postepeno puniti dodavanjem igrača na platformu te planiranjem i provođenjem vježbi.

U polju *Next injects to send* prikazani su nadolazeći *injecti* u vježbama koje se provode. Odbrojava se do njihovog slanja te su sortirani po preostalom vremenu do okidanja. Ostala polja ne zahtijevaju dodatna objašnjenja i napomene.

3.3.5. Izbornik za vježbe

Odabirom *Exercises* na izborniku lijevo dolazimo na popis postojećih vježbi, kao što se vidi na slici 3.4. Odabirom stvaranja nove vježbe ("+" u donjem desnom kutu) otvara se forma s osnovnim podacima o vježbi. Ovi podaci mogu se promijeniti kasnije. Odabirom neke vježbe, prikazuje se stranica pregleda (engl. *overview*). Zaglavlje aplikacije sada se popunilo novim opcijama za navigaciju koje se odnose na vježbu. Ovdje se može izvesti vježba u zip arhivu jednom od opcija u izborniku pokraj imena vježbe. Kako je izbornik za vježbe centralni dio OpenEx aplikacije i ima najviše detalja i funkcionalnosti, slijedi opis pojedinih podizbornika unutar njega.

Pregled

Pregled (*overview*) vježbe ima nekoliko zanimljivih funkcionalnosti. Polje *Execution* ima opcije za pokretanje, pauziranje, zaustavljanje i resetiranje vježbe. Ove opcije se pojavljuju ovisno o njihovoj dostupnosti. Na primjer, vježba se može pauzirati tek nakon što je pokrenuta. Aplikacija tijekom provođenja vježbe automatski prati razne statistike i status o vježbi. Ovo sve ostaje vidljivo nakon što je vježba zaustavljena ili je došla do kraja. U ovom kontekstu, resetirati vježbu znači obrisati sve te podatke, što omogućava novo pokretanje vježbe. Prije resetiranja, podaci koji se brišu mogu se izvesti iz aplikacije. U ovom se polju također može postaviti vrijeme početka izvođenja vježbe. U tom slučaju, vježba će započeti automatski u zakazano vrijeme i dalje se odvijati prema svom rasporedu. Druga opcija je ručno pokretanje vježbe.

Polje *Control* omogućava testno pokretanje vježbe (*Dryrun*) i slanje provjere komunikacije (*Comcheck*) igračima. Testnim pokretanjem vježbe, određenim korisnicima će na mail doći injecti redom kao u vježbi, ali u kratkim vremenskim razmacima, a ne stvarno kako su vremenski planirani. Ovo omogućava provjeru pravilnog redoslijeda, te testiranje svih *injecta*. U provjeri komunikacije igračima će na mail doći poveznica klikom na koju mogu potvrditi da im je mail stigao. Ovo je bitno, jer skoro sva komunikacija u OpenEx-u ide mailom. Za provjere komunikacije definira se rok do kada igrači mogu odgovoriti na njih, te publike (*Audiences*) kojima je provjera upućena. U polju *Settings* može se postaviti zaglavlje teksta svih mailova koji se šalju igračima, te tzv. *no-reply* adresa koju sustav koristi za slanje mailova. Na dnu stranice nalaze se polja koja prikazuju pojedina testna pokretanja i provjere komunikacije, te se odabirom na njih može pregledati status.

Definiranje vježbe

Definition stranica služi definiciji svih objekata potrebnih za stvaranje *injecta*. To su *Audiences*, *Media pressure* i *Challenges*. Izbornik se nalazi s desne strane.

Publika (*Audiences*) predstavlja jedan tim igrača unutar vježbe. Dodavanjem publike, ona se samo stavlja u listu. Zatim je potrebno odabrati tu publiku, te dodati igrače u nju. U daljnjim koracima planiranja vježbe, planira se s obzirom na publike, a ne igrače. Time je jedino potrebno postavljanje uvezene vježbe dodjeljivanje konkretnih igrača u odgovarajuće publike.

Ideja iza *Media pressures* je dočarati kako se u javnosti vide događaji iz vježbe. Mogu se stvoriti TV kanali, internetski blog ili novine. Zatim se može napisati objava na blogu, TV reportaža ili novinski članak. U iste se mogu dodavati slike i dokumenti.

Ove različite vrste se funkcionalno nikako ne razlikuju. Igračima se na mail šalju poveznice na pojedine medijske objave.

Izazovi (*Challenges*) su OpenEx-ov mehanizam za klasičan CTF (*Capture The Flag*) zadatak. U CTF zadacima igrači na neki način traže skrivene tzv. zastave, koje su najčešće dugi niz znakova, te kada ih nađu, mogu ih unijeti u polje za unos kao dokaz da su riješili zadatak. Challenge se definira izvan same vježbe, a dodaje se u neku vježbu tek u koraku stvaranja *injecta*. Ovdje, u koraku definicije moguć je samo pregled već dodanih izazova.

Scenarij vježbe

Stranica scenarija služi za dodavanje *injecta* u vježbu. Za svaki *inject* definira se nakon koliko se vremena okida u odnosu na početak vježbe. Svaki *inject* može se konfigurirati: kojim publikama je namijenjen, naslov i tekst maila kojim dolazi, priloženi dokumenti, kriptira li se mail. Opcije vezane uz prikupljanje podataka biti će razjašnjene kasnije u radu.

Injecti su ovdje poredani uzlazno po vremenu okidanja. Njima se tijekom provođenja vježbe može dodatno upravljati tako da ih se onemogući (*Disable*), ili okine u vrijeme prije onoga u rasporedu (*Trigger now*). Postoji i mogućnost isprobavanja pojedinog *injecta* (*Try the inject*). Isprobavanje *injecta* prikazati će status o uspješnosti (ili neuspješnosti) slanja *injecta*. Očekivano je da trenutnom korisniku isprobani *inject* dođe mailom, ako je uspješno okinut.

Ako je korisnik upoznat s izazovima i medijima u OpenEx-u, onda je iz imena svakog tipa *injecta* jasno čemu služi i što će poslati igračima. Bitno je napomenuti da je *Manual* vrsta *injecta* jedina vrsta koja se ne šalje mailom. U vrijeme okidanja, OpenEx neće ništa napraviti, već je na voditeljima vježbe da obrade taj *inject*, kako god oni to zamislili.

Provođenje vježbe

Animation sadrži s desne strane izbornik nekoliko različitih stranica bitnih tijekom izvedbe vježbe. Na dnu izbornika postoji opcija *Chat*, koja u vrijeme pisanja ovog rada nije implementirana.

Stranica *Timeline* ima vremensku crtu na kojoj su po publikama prikazani planirani *injecti*. Ispod se nalazi popis *injecta* koji će tek doći na red, te popis procesuiranih injecta. Za procesuirane injecte može se pogledati izvještaj o statusu s podacima o uspješnosti.

Mails postoji kao mogućnost slanja improviziranih *injecta*, *injecta* koji nisu unaprijed isplanirani. Tijekom izvođenja vježbe, ovdje se može dodati novi mail, te će se otvoriti forma slična onoj za kreiranje *injecta*, koji se onda odmah šalje.

Stranica *Validations* služi da se *injecti* s očekivanom reakcijom igrača mogu ocijeniti. Tijekom kreiranja *injecta*, definira se očekuje li se reakcija igrača. Nakon što se *inject* okine, netko iz tima za provođenje vježbe može dati ocjenu timu baziranu na njihovoj reakciji. Ovo je jedan oblik prikupljanja podataka tijekom vježbe.

Stranica s dnevnikom vježbe (*Exercise logs*) vrlo je jednostavna. Ovdje se može dodati zapis u dnevnik, vidljiv svim korisnicima koji mogu pristupiti vježbi. Uz zapis se prikazuje i autor te vrijeme pisanja.

Rezultati

Results s desne strane izbornik nekoliko različitih stranica koje sadrže informacije o provedenoj vježbi.

Statistics prikazuje veliki broj grafova i dijagrama. Svi su objašnjeni svojim imenom i ne zahtijevaju dodatne napomene.

Lessons learned se koristi za prikupljanje i obradu podataka direktno od igrača. Nakon provedene vježbe, može im se poslati upitnik na koji će odgovoriti putem forme unutar OpenEx aplikacije. Sva pitanja u formi su oblika: kliznik za ocjenu od 0 do 100, tekstualno polje za pozitivan komentar i tekstualno polje za negativan komentar.

Reports služi zapisivanju izvještaja. Izvještaj je snimak svih ili nekog podskupa podataka iz *Lessons learned* i/ili *Statistics*. Neki podaci na stranicama rezultata se brišu resetiranjem ili novim pokretanjem vježbe, ali izvještaji perzistiraju.

3.3.6. Igrači

Odabirom *Players* na lijevom izborniku dolazimo do liste postojećih igrača u sustavu. Igraču osim što se zapisuju osnovne informacije, može pripadati nekoj organizaciji i državi. Također mu se mogu zapisati broj telefona i PGP javni ključ. Bitno je razlikovati entitete igrača i korisnika unutar sustava OpenEx. Korisnici imaju mogućnost prijave u sustav te gledanja i uređivanja vježbi, dok igrači to nemaju. Igrači samo sudjeluju u vježbi. Ali, zbog nekih funkcionalnosti, kao što je testno pokretanje vježbe, svaki dodani korisnik u sustav pojaviti će se i kao igrač. Obrat ne vrijedi.

3.3.7. Organizacije

Entitet organizacije u OpenEx-u služi kao još jedan način grupiranja i filtriranja igrača. U slučaju da u sustavu postoji veći broj vježbi za različite firme, ovo postaje korisno. Osim igrača, i korisnici mogu imati pripadnost nekoj organizaciji. Ovo je korisno jer se dozvole čitanja i pisanja po postojećim vježbama mogu definirati na razini organizacije.

3.3.8. Datoteke

Na stranici *Documents* u sustav OpenEx mogu se dodavati dokumenti koji se tijekom stvaranja vježbe mogu koristiti kao privitci za *injecte*. Dokument može biti bilo koja datoteka.

3.3.9. Mediji

OpenEx podržava stvaranje TV kanala, internetskih blogova i izdavača novina. Za iste se mogu stvarati medijske objave. Različiti tipovi medija se funkcionalno nikako ne razlikuju. Sve tipove medija se može urediti što se tiče njihovog izgleda, od primarnih i sekundarnih boja do postavljanja logotipa, a pojedinim medijskim objavama može se po volji definirati i ime autora te broj komentara i *lajkova*. Sve ove opcije imaju estetsku svrhu, te nemaju funkcionalan utjecaj. Medijske objave se mogu slati kao *inject*.

3.3.10. Izazovi

Izazovi (engl. *challenges*) su OpenEx-ov mehanizam za klasične CTF zadatke. Izazovu se može definirati broj bodova, maksimalan broj pokušaja, i naravno, proizvoljan broj zastava. Izazovi se također mogu slati kao *inject*, u kojem će igrači u formu unositi zastave koje su našli, te za njih dobivati bodove.

3.3.11. Lekcije

Lekcije (engl. *lessons learned*) služe kao predložak za stvaranje upitnika na kraju vježbe. Lekcije mogu sadržavati više kategorija pitanja. Pitanja se onda dodaju u te kategorije. Stvorene lekcije se koriste kao predložak koji se može primijeniti za upitnike koji će se slati igračima nakon vježbi. Na svako pitanje se može odgovoriti

ocjenom od 0 do 100, pozitivnom i negativnom povratnom informacijom. Pitanja bi trebalo napraviti tako da ovaj način odgovaranja ima smisla.

3.3.12. Postavke

U postavkama se može nekoliko stvari. Na stranici *Parameters* prikazana je verzija sustava te njegovih komponenti, te neke estetske postavke. Stranica *Users* koristi se za upravljanje korisnicima. Korisnik se svojim mailom i lozinkom može prijaviti u sustav. Za rad na vježbama, korisnik mora biti administrator. U suprotnom će ga se tijekom prijave tretirati kao igrača, a mogućnost prijave igrača u OpenEx u vrijeme pisanja ovoga nije implementirana. Grupe se koriste za upravljanje dozvola korisnika. Za grupe se može definirati kojim vježbama imaju pristup pisanja i čitanja, te se u grupe onda dodaju korisnici. Na stranici *Tags* se na razini sustava mogu definirati oznake.

3.4. Planiranje i provođenje vježbi

U vrijeme pisanja ovog rada, ne postoji službena dokumentacija za korištenje alata OpenEx. Čak ni proces instalacije nije detaljno dokumentiran. Osim što ne postoji službena dokumentacija, ne postoje ni neslužbeni izvori informacija o ovom alatu. Usprkos tome što je besplatan i otvorenog koda, nema javno dostupnih izrađenih vježbi od strane korisnika OpenEx-a. Slijede neki zaključci o mogućim načinima korištenja OpenEx-a bazirani na funkcionalnim mogućnostima aplikacije.

3.4.1. Ideje i mogućnosti

U vrijeme pisanja, jedini način komunikacije sustava OpenEx i igrača je putem elektroničke pošte. Način komunikacije u tom smjeru je jasan, ali postavlja se pitanje na koji način bi igrači trebali odgovarati na *injecte* te na koji način je zamišljena komunikacija između timova. OpenEx nema svoje rješenje na ovaj problem. Jedino što OpenEx može je jednosmjerna komunikacija mailom s igračima. To ostavlja na voditeljima vježbe da riješe kako će se odgovarati na injecte. U duhu mogućnosti OpenEx-a, najviše smisla ima da se sva komunikacija između timova odrađuje mailom. U tom slučaju, voditelj kao administrator poslužitelja elektroničke pošte ima pregled nad svom komunikacijom i točnim vremenima. Ovaj pristup ne zahtijeva da su svi timovi u fizički istoj prostoriji. Nadalje, ovaj pristup ne zahtijeva ni da su igrači unutar istog tima u istoj prostoriji, ali u tom slučaju je upitna moguća kvaliteta rasprave među igračima. Bitno je da igrači neometano mogu unutar tima raspravljati o postupcima. Ovo su sve mogućnosti koje ovise o pojedinim vježbama i skupu ljudi koji sudjeluju. OpenEx nikako ne sprječava ni jedan od ovih odabira, ali je određivanje koji je najbolji način provođenja vježbe uz OpenEx zadatak je koji nema jedno točno rješenje.

Nisu sve vrste kibernetičkih vježbi jednako dobro izvedive u OpenEx-u. Mogućnosti koje OpenEx pruža čine se najprikladnije za operativne i strateške vježbe. Mogao bi se koristiti i za taktičke ili tehničke vježbe, ali u tom slučaju su potrebni kibernetički poligoni odvojeni od OpenEx-a, jer ih OpenEx ne može simulirati niti ikako integrirati.

OpenEx se može vrlo dobro koristiti kao platforma za planiranje vježbi koje se ne moraju nužno provoditi kroz OpenEx, već na neki drugi način. U ovom smislu, OpenEx ima mnoge prednosti: definicija timova, način za planiranje *injecta* po vremenskoj crti, pridruživanje vježbama upitnike za igrače i mogućnost kolaborativnog rada većeg broja ljudi. U ovom pogledu, najveći je plus to što OpenEx nudi mogućnost uvoza i izvoza, te time dijeljenja vježbi. Čitanje vježbe učitane iz OpenEx-a također može biti

pomoć i podsjetnik pri provođenju vježbe na klasičan ili neki drugi način.

3.4.2. Nedostaci

U vrijeme pisanja ovog rada, OpenEx je daleko od gotovog proizvoda spremnog za korištenje. Mnoge funkcionalnosti još uopće ne postoje, kao što je aktivna interakcija igrača sa OpenEx-om. Igrači nikako ne mogu pristupiti OpenEx-u, osim pri odgovaranju na upitnik. Ako se korisnik koji nije administrator pokuša prijaviti u sustav, dočekat će ga prazna stranica na kojoj samo piše *Player dashboard*, što implicira da će i igrači u budućim verzijama OpenEx-a moći pristupiti aplikaciji.

Od već implementiranih funkcionalnosti, opcije dostavljanja *injecta* bi trebale biti robusnije. Svaki *inject* u OpenEx-u mora imati definirano vrijeme kada se okida. Tijekom provođenja vježbe, pojedini *inject* se može otkazati ili okinuti ranije, te se može usred provođenja promijeniti u koje se vrijeme *inject* okida, ali ne postoji opcija za *inject* da se okine ručno u vrijeme kada moderator to želi, što je veliki i očiti propust, te veliko ograničenje za planiranje. Moguće je to zaobići na neki način, na primjer, postaviti da se *inject* okida u vrijeme daleko od zamišljenog trajanja vježbe, pa ga ručno okinuti ranije u željeno vrijeme. Ovo je ružno rješenje na problem koji je mogao biti lako riješen dodavanjem opcije da se *inject* okida isključivo ručno.

Jos je jedan problem što su *injecti* strogo linearni. Ne postoji mogućnost definiranja bilo kakve međuovisnosti *injecta*, što je također veliko ograničenje za planiranje vježbe.

4. Razvijena vježba

U sklopu rada, pripremljena je jedna kibernetička vježba za zdravstveni sektor na operativnoj razini. Vježba je stvorena s OpenEx-om na umu, što dolazi s nekim pretpostavkama o načinu provođenja. Provođenje je zamišljeno tako da su svi timovi u jednoj prostoriji, ali dovoljno fizički odvojeni da igrači mogu nesmetano raspravljati unutar svog tima. Komunikacija između timova oko donesenih odluka, postupaka, i dojavljivanja informacija odvija se elektroničkom poštom. Komunikacija sa svima izvan timova u vježbi također je putem maila. Igrači te mailove šalju na adresu koju je moderator zamislio za tu svrhu. Moderator je također u prostoriji s igračima, te može lako pratiti kako se situacija razvija. Brzina reakcije igrača na neke *injecte* bit će bitna. Vježba nije idealna te neki scenariji i događaji u vježbi nisu realni u pravim uvjetima bolnice, što s medicinske strane, što s tehničke. Cilj vježbe nije da bude što spremnija za pravo izvođenje, već istražiti i pokazati kako bi se implementirala u OpenEx.

Vježba je za veliku javnu bolnicu. Bolnica koristi sustav za automatsko praćenje vitalnih znakova koji je spojen na elektronički sustav za praćenje podataka o pacijentima. Napadač koristi otkrivenu ranjivost nultog dana u automatskom sustavu za praćenje kako bi lažirao podatke koje sustav dojavljuje i tako nakon nekog vremena onemogućio normalan rad bolnice. Cilj vježbe je provjeriti lanac komunikacije između timova i izvršavanje svih legalnih obveza u slučaju ovakvog napada.

Vježba je inspirirana drugim primjerom vježbe iz [6].

4.1. Timovi

Vježba je zamišljena za njegovateljski tim, tim doktora, bolnički IT tim i menadžment tim. Njegovateljski tim sastoji se od medicinskih sestri i tehničara. Članovi ovog tima su oni koji imaju najviše direktnog kontakta s pacijentima. Tim doktora uključuje specijaliste s raznih područja medicine. Bolnički IT tim su zaposlenici zaduženi za održavanje cijele računalne infrastrukture bolnice. U stvarnom slučaju, bolnički IT zaposlenici bili bi zaduženi samo za održavanje računala opće svrhe. Za održavanje

specijalnih uređaja kao što su oni u sustavu za automatsko praćenje vitalnih znakova, u realnom slučaju bi za to bila zadužena firma koja ih proizvodi, čiji zaposlenici nisu stalno prisutni u ustanovama gdje se koriste, već dolaze na zahtjev. Menadžerski tim sastoji se od nekoliko različitih članova: odvjetnika ili nekog drugog stručnjaka za pravo, ravnatelja ustanove te stručnjaka za odnose s javnošću.

4.2. Injecti

Slijedi opis osmišljenih *injecta* te kratki opis kako su implementirani u OpenEx. Uz većinu *injecta* koji se šalju kao mail, pripremljen je i *manual inject* koji sadrži prijedloge poticajnih pitanja koje moderator može postaviti ako misli da je potrebno.

Uvodna riječ

Pozdrav,
Radite u velikom gradu u javnoj zdravstvenoj ustanovi, bolnici. Postoji veliki broj različitih zavoda i operacijskih sala s dovoljno velikim brojem stručnog osoblja, medicinskih sestri, tehničara, doktora specijalista, kirurga. Bolnica je opremljena modernom računalnom opremom koja olakšava svakodnevne procese. Koristi se EHR (Electronic Health Records) sustav s bazom podataka o pacijentima, liječenju, povijesti bolesti, propisanim lijekovima itd. U EHR sustav integriran je sustav za praćenje pacijenata, tako da se podaci o tlaku, količini kisika u krvi i slično automatski pohranjuju za pacijente kojima se mjere. Sustav se također koristi za praćenje vitalnih znakova tijekom operacije. EHR je u bolnici već dugi niz godina. Doktori i ostalo osoblje odlično su se upoznali s time. Sustav za praćenje pacijenata je relativno nov, koristi se 3 mjeseca, ali zbog jednostavnog i intuitivnog dizajna brzo su se pohvatale barem njegove osnovne funkcionalnosti. Svaki pojedini sustav za praćenje sastoji se od centralnog računala koje šalje podatke u ERH i uređaja za mjerenje kojih može biti jedan ili više.

Ovaj *inject* u OpenEx-u realiziran je kao mail s više primatelja. Svaki tim zaprimi uvodnu riječ. Ne očekuje se nikakva reakcija timova na ovaj *inject*.

Mala greška

Obaviješteni ste sustavom da su pacijentu koji je zadržan u bolnici nakon srčanog udara mjerenja krvnog tlaka opasno visoka. Odmah ga idete provjeriti, no čovjek se čini smiren, naizgled mu se ništa lose nije dogodilo u zadnjih nekoliko minuta.

Ovaj događaj je posljedica napadačevog testiranja otkrivene ranjivosti u uređaju za automatsko praćenje.

Ovaj *inject* realiziran je kao mail za njegovateljski tim. Ovdje se već očekuje komunikacija timova. Njegovateljski tim može pozvati doktore da dodatno pregledaju pacijenta. Može se početi sumnjati na računalnu opremu i uključiti IT tim u raspravu. Poticajna pitanja:

- Kako reagirate na ovo?
- Vjerujete li pacijentu da je uredu?
- Radite li ručno neka mjerenja?
- Dojavite li ikome što se dogodilo?
- Ostaje li netko uz pacijenta?

Novi krivi znakovi

Sustav pošalje obavijest da pacijent koji boluje od KOPD-a (kronična opstruktivna plućna bolest) ima neobično visok broj udisaja po minuti. Pri pregledu pacijenta, ponovo, izgleda i osjeća se uredu. Za svaki slučaj provjerite mu ostale vitalne znakove, te primijetite nešto zanimljivo: otkucaji srca su mu viši od onoga što piše u EHR sustavu. Primijetite da mu zapis otkucaja srca postepeno raste zadnjih 3 sata. Iako je sve u normalnim granicama, ne poklapa se s onim što je stvarno izmjereno.

Otkrivene nedosljednosti u podacima ponovo su djelo napadačevog testiranja mogućnosti nad sustavom. Ovaj put je napadač smanjio iznos vitalnog znaka sa povišene razine na normalnu.

Ovaj *inject* je također realiziran kao mail za njegovateljski tim. Očekivanja od igrača su slična kao u prošlom *injectu*: nastavak komunikacije timova oko primjerenog događaja i sumnja na računalnu opremu.

Operacija

Izvodite operaciju srčane premosnice. Operacija dobro krene, ali nakon reza na prsima, sustav za praćenje dojavljuje da pacijentu naglo pada tlak u krvi. Prema tome, trebalo bi mu brzo dati odgovarajuće lijekove.

Napadač ne manipulira ovim podacima, već su stvarni. Od igrača se očekuje relativno brz odgovor s njihovim postupkom. Ovisno o brzini odgovora i postupku koji odaberu, postoje dva ishoda iz ove situacije. Ideja ovog *injecta* je potencijalno navesti

igrače na krivi postupak zbog stvorene sumnje u računalnu opremu. *Inject* je realiziran kao mail za njegovateljski i doktorski tim.
Poticajna pitanja:

- Što poduzimate?
- Vjerujete li mjerenjima?
- Provjeravate li ručno?
- Dajete li lijek?
- Koliko brzo odluke moraju biti donesene?

Operacija - dobar odabir

Pacijentu stvarno padne tlak tijekom operacije, te su mu lijekovi koje ste mu dozirali potencijalno spasili život, u svakom slučaju, to je bila dobra odluka. Ostatak operacije prođe uspješno.

Ovaj *inject* služi kao potvrda dobre odluke timova. *Inject* se okida ako igrači donesu dobru odluku u zadovoljavajućem vremenskom okviru. Na voditelju je vježbe da odredi jesu li ti uvjeti ispunjeni ili ne.
Poticajna pitanja:

- Kome dojavite što se dogodilo (ako ikome)?
- Zašto ste postupili kako jeste?
- Je li se nešto moglo bolje poduzeti?

Operacija - krivi odabir

Pacijentu stvarno padne tlak tijekom operacije, ali ste potrošili previše vremena na raspravljanje i čekanje. Po ostalim vitalnim znakovima zaključite da pacijent pada u komu zbog nedostatka kisika u krvi. Sprječavanje daljnjih posljedica postaje novi prioritet. Uspješno ste spriječili da pacijent padne u komu, ali operacija sada mora biti otkazana, te se pacijenta mora zadržati na promatranju.

Ovaj *inject* je potvrda loše ili spore odluke igrača. Sada će sigurno i menadžerski tim biti uključen u komunikaciju, ako još nije bio. Moguće su legalne posljedice ovog događaja.

Novinski članak - krivi odabir

Član obitelji javlja iskustvo iz bolnice...

Tijekom rutinske operacije srčane premosnice, pacijent skoro umro na stolu. Usprkos svoj tehnologiji, doktori su jedva na vrijeme dali lijekove za sniženi krvni tlak koji je ustupio tijekom operacije. Pacijent će biti zadržan nekoliko dana zbog promatranja. Što se dogodilo? Otkazuju li doktori ili njihova oprema?

Ovaj *inject* je realiziran kao medijska objava, novinski članak. Svim igračima na mail dolazi poveznica na članak. Jedino se očekuje reakcija menadžerskog tima što se tiče legalnih postupaka i komunikacije s javnošću.

Pravi napad

Odjednom svi sustavi za praćenje počnu alarmantno javljati različite kritične vitalne znakove. Bolnicom zavlada buka. Ne znate gdje biste počeli gledati ni kako reagirati. Panično počnete obilaziti pacijente i brzo primijetite da brojevi koje sustavi dojavljuju nisu realni ni mogući: krvni tlakovi 10 puta veći od normalne vrijednosti, otkucaji srca nekoliko puta u minuti, nepostojeći kisik u krvi... Ubrzo shvatite da je sustav zasigurno napadnut. Kako stvari stoje, svi podaci koji se dobivaju automatizirano trenutno ne vrijede. Svi poslovi se trenutno moraju obavljati ručno. Iznimno je teško pravovremeno reagirati na sve potencijalne probleme s obzirom na broj pacijenata.

Inject je realiziran kao mail za IT, njegovateljski i doktorski tim. Sada se provjerava spremnost na ovakvu situaciju. Koji su, ako uopće postoje, postupci zamišljeni za ovakvu situaciju.

Poticajna pitanja:

- Kako se nosite s ovom situacijom?
- Jeste li spremni na ovako nešto?
- Kome dojavite što se dogodilo?

Posljedice

Za vaš incident i probleme koji se događaju na široko se čuje. U kontakt stupi policija u suradnji s ministarstvom zdravstva. Informirani ste o posljedicama koje slijede. Zahtijevaju detaljan popis osjetljivih informacija koje su procurjele. Zanima ih koji su vam daljini planovi s radom, kako ćete spriječiti što se dogodilo te koliko će trajati oporavak.

Inject je mail za menadžerski tim. Bitno je u ovakvoj situaciji znati količinu štete koja se dogodila. Očekuje se od tima menadžera da pruža što bolji odgovor na pitanja kao što su: uzrok napada, količinu procurenih osjetljivih informacija i plan oporavka.

Poticajna pitanja:

– Imate li dovoljno podataka da odgovorite na sva pitanja?

– Što ćete im reći?

– Kako će kazna utjecati na rad bolnice?

– Kako će pad reputacije utjecati na rad bolnice?

Novinski članak - napad

Bolnica pretrpjela hakerski napad
Haker je provalio u sustav najveće bolnice u gradu. Otkazala je sva informatička oprema. Normalan rad se neće vratiti danima. Fokus je samo na najpotrebitijima, ali u ovim uvjetima, i to preplavljuje trenutnu količinu radnika. Tko je krivac za ovo? Je li haker prepametan? Je li sigurnost bolnice preslaba? Ni policija ni bolnički IT tim i dalje nisu identificirali napadača, a i daleko je od rješenja problema. Znači li to cijeli set nove opreme? I bitnije, kada će se stanje vratiti na normalno?

Za kraj vježbe, jedan novinski članak koji pokazuje kako je javnost doživjela događaje iz vježbe. Na mail svim igračima dolazi poveznica na članak.

5. Zaključak

Kibernetičke vježbe su ključne za unaprjeđenje sigurnosti i zaštitu podataka u svakoj firmi, pa tako i u firmama zdravstvenog sektora koje posjeduju brojne osjetljive i privatne podatke velikog broja ljudi. Vježbe imaju nekoliko važnih koristi, uključujući pripremu osoblja za krizne situacije, podizanje svijesti o sigurnosti i dobrih praksi te poboljšanje internih protokola i postupaka.

Planiranje i provođenje ozbiljnih kibernetičkih vježbi zahtijeva vrijeme, resurse i tehničke vještine. U tom kontekstu, alat otvorenog koda OpenEx može biti korisna pomoć pri planiranju i provođenju vježbi. Iako OpenEx ima korisne i dobro implementirane funkcionalnosti, važno je napomenuti da je još uvijek u razvoju i da u trenu pisanja ovog rada ima i puno nedostataka. Daleko je od gotovog proizvoda.

Unatoč tome, OpenEx se može koristiti kao dodatni alat ili podrška pri planiranju i provođenju kibernetičkih vježbi. S obzirom na daljnji razvoj, OpenEx ima potencijal postati izvrsna platforma za planiranje i provođenje vježbi na velikim razinama. Njegove funkcionalnosti i prilagodljivost mogu omogućiti organizacijama da simuliraju realistične scenarije, identificiraju ranjivosti i poboljšaju svoju pripravnost za kibernetičke prijetnje.

Važno je naglasiti da kibernetičke vježbe trebaju biti redovite i kontinuirane aktivnosti. Samo jedna vježba nije dovoljna da se osigura sigurnost. Potrebno je razvijati kulturu sigurnosti, educirati osoblje o najnovijim prijetnjama i praksama te redovito provjeravati i ažurirati sigurnosne protokole i postupke.

Kako bi se osigurala cjelovita sigurnost, osim kibernetičkih vježbi, također je važno ulagati u sigurnosnu infrastrukturu, primjenjivati najbolje prakse, koristiti pouzdane sigurnosne alate i redovito nadograđivati sustave. Samo cjelovit pristup sigurnosti može zaštititi zdravstvene ustanove i osigurati njihov neometen rad.

LITERATURA

[1] docker-mailserver. URL https://github.com/docker-mailserver/docker-mailserver.

[2] Filigran. URL https://www.filigran.io/en/.

[3] lolcal_openex_setup. URL https://github.com/boolwinkle/local_openex_setup.

[4] Filigran. Openex. . URL https://github.com/OpenEx-Platform/openex.

[5] Filigran. Openex - docker. . URL https://github.com/OpenEx-Platform/docker.

[6] U.S. Department of Homeland Security. Office of Infrastructure Protection. Cyber tabletop exercise for the healthcare industry facilitator and planner guide. 2013. URL https://www.hsdl.org/c/view?docid=789781. Zadnje pristupljeno: 05.28.2023.

Izrada kibernetičke vježbe za zdravstveni sektor upotrebom alata OpenEx

Sažetak

Kibernetičke vježbe ključan su faktor u sigurnosti poduzeća. Ovo je posebno bitno za firme u čijim su rukama potencijalno ljudski životi, kao što je zdravstveni sektor. Planirati i provoditi kibernetičke vježbe zahtjevan je proces. Korisno je imati na raspolaganju alate za pomoć to procesu. OpenEx je alat otvorenog koda koji omogućava planiranje, provođenje i dijeljenje napravljenih vježbi. OpenEx još nije dovršen proizvod i ima mnoge nedostatke, ali korisno je biti upoznat s funkcionalnostima koje nudi i kako ih koristiti.

Ključne riječi: Kibernetičke vježbe, OpenEx, zdravstveni sektor, inject, docker, igrači, moderator vježbe

Creating cyber security exercises for the health sector using the OpenEx tool

Abstract

Cyber exercises are a key factor in a company's security. This is especially important for companies that could potentially save lives, such as those in the healthcare sector. Planning and conducting cyber exercises is a demanding process. It is very useful to have tools to assist in that process. OpenEx is an open-source tool for planning, conducting, and sharing exercises. OpenEx is still in development, and has many flaws, but it is useful to know its capabilities and how to use them.

Keywords: Cyber exercises, OpenEx, healthcare sector, inject, docker, players, exercise facilitator