SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 737

VIZUALIZACIJA PODATAKA PRIKUPLJENIH TIJEKOM UPRAVLJANJA INCIDENTOM

Tihana Šupljika

Zagreb, lipanj 2022.

Sadržaj

1. Uvod 1
2. Sigurnosni incidenti 3
3. Vizualizacija prikupljenih podataka 6
3.1. Alat Timesketch 7
3.1.1. Karakteristike 8
3.1.2. Nedostaci 10
4. Metode vizualizacije u ASP.NET Core aplikaciji 12
4.1. Opis modela i biblioteke 12
4.2. Implementacija 13
4.2.1. Vizualizacija učestalosti odvijanja događaja 14
4.2.2. Vizualizacija vremenske crte 16
4.3. Budući rad 19
5. Zaključak 20
Literatura 21
Sažetak 23
Summary 24
Skraćenice 25

1. Uvod

Sa sve većim razvojem tehnologije, digitalizacijom poslovanja i povećanjem broja osoba koje svakodnevno koriste računalne sustave sve su prisutniji sigurnosni incidenti. Zaštita sustava pomaže u sprječavanju zlonamjernih događaja, ali se unatoč njenom postojanju mogu dogoditi incidenti kao posljedica njezinog neispravnog ili manjkavog korištenja te iskorištavanjem rupa u zaštiti sustava i samom sustavu. U slučaju kada dođe do incidenta, i on bude otkriven, potrebno je iskoristiti raspoložive resurse za prikupljanje, upravljanje i analizu informacija o odvijanju incidenta kao što su ljudi obučeni za upravljanje incidentima i alati za analizu incidenata. S obzirom na trajanje incidenta, broj napadnutih podsustava i/ili složenost napadnutog sustava, taj je postupak potencijalno složeni proces, ali je nužan za dobivanje uvida u incident, napadača i razvoj daljnje zaštite.

Tijekom odvijanja incidenta, a pogotovo nakon njegova završetka, prikupljene podatke koristimo kako bi dobili uvid u incident. Cilj nam je spoznati početak, način izvedbe, ciljeve, prouzročenu štetu i slične podatke o incidentu i napadaču. Stečeno znanje primjenjujemo na suzbijanje trenutačnog i sprječavanje budućih incidenata unaprjeđenjem postojećeg ili izradom boljeg sigurnosnog sustava te obrazovanjem korisnika sustava i ljudi odgovornih za zaštitu sustava. Kao pomoć pri ovom procesu mogu se koristiti raznovrsni alati za pregled i analizu podataka. Takvi alati mogu kategorizirati događaje incidenta, pružiti bolju preglednost podataka, ujediniti slične podatke, te ako imaju funkcionalnost vizualizacije prikupljenih podataka ili podataka izvedenih iz postojećih, pružiti određene vizualizacije radi lakšeg uvida.

U ovome je radu cilj objasniti što su incidenti, koje događaje smatramo incidentima, kako se oni mogu otkriti, te koji se postupci provode tijekom upravljanja incidentima i pokušaja sprječavanja njihove eskalacije. U kontekstu incidenata, objasnit će se važnost alata za analizu prikupljenih podataka, navesti koji su to podaci i staviti naglasak na vizualizaciju, pogotovo učestalosti određenih uzoraka, vremenske crte i odnosa između događaja. Pokazati će se jedan od alata za analizu zajedno s njegovim prednostima i nedostacima na temelju kojih će se potom opisati primjer aplikacije te dati postupak kojim korisnik samostalno može implementirati određene vizualizacije u vlastitim projektima kako bi analizirao i pregledao podatke koji se odnose na tijek događaja koji su se odvijali unutar pojedinog incidenta.

U prvom poglavlju rada ukratko su navedene općenite informacije o incidentima, njihovom otkrivanju i upravljanju. Zatim su u sljedećem poglavlju opisane svrhe alata za analizu, razlozi potrebe za vizualizacijom podataka, navedeni su primjeri vizualizacija korisnih u kontekstu incidenata te je opisan alat Timesketch sa svojim prednostima i nedostacima. U posljednjem je poglavlju dan opis ASP.NET Core aplikacije koja sadrži neke od spomenutih vizualizacija te postupak kojim se podaci mogu vizualizirati.

2. Sigurnosni incidenti

U današnje su doba računalni sustavi sveprisutni u ljudskim životima te su dio kako osobne, tako i poslovne svakodnevice većine svijeta. Osim što se koriste u osobnim životima, računala se također naveliko koriste u većini oblika poslovnog svijeta: u uredima, u bolnicama, za slanje elektroničke pošte, za nadziranje drugih sustava i slično. S obzirom na takvu sveprisutnost, očekivano je da postoje ljudi i organizacije čiji je interes iskorištavanje i napadanje računalnih sustava, pogotovo na sustave unutar poslovnog svijeta. Takve pojave smatraju se sigurnosnim incidentima te njihova učestalost raste s povećanjem digitalizacije.

Općenito, sigurnosne incidente možemo definirati kao „proboj sigurnosti sustava s ciljem nedozvoljenog pristupanja i narušavanja njegova integriteta i/ili dostupnosti“ [1], tipično u obliku sljedećih aktivnosti:

- pokušaj nedozvoljenog pristupa sustavu ili podacima
- nedozvoljeno korištenje sustava za obrađivanje ili pohranjivanje podataka
- izmjena *firmware*-a, *hardware*-a ili *software*-a bez dozvole vlasnika
- zloćudan prekid ili uskraćivanje usluga

Incident gledamo kao neočekivani ili neželjeni događaj, ili slijed događaja, koji može ugroziti rad sustava, organizacije i slično. Pod događajem smatramo pojavu u sustavu, usluzi ili vezi koja pokazuje na proboj sigurnosnih pravila, neuspjeh sigurnosnih zaštita ili do tada nepoznatu situaciju koja se čini kao prijetnja sigurnosti sustava [2]. Sigurnosni incident može uključivati niz aktivnosti – kompromitiranje korisničkih računa, *phishing* napadi, proboj sigurnosne stijene (*firewall*) tvrtke, krađa podataka i drugi postupci [3].

Napadač koji ima namjeru ući u sustav i napasti ga izvodi proizvoljan broj akcija na takozvanom lancu napada (*attack-chain*) kako bi proveo napad. Nije nužno provesti sve korake kako bi napad bio uspješan, ali napadač obično izvodi barem prvi od njih kako bi osigurao uspješnost. Prvi korak jest infiltracija tijekom koje napadač ulazi u mrežu kako bi dobio uporište u sustavu izvođenjem *phishing*-a, krađom identifikacije, pokretanjem zloćudnih programa ili nekog drugog sličnog postupka. Zatim izvodi izviđanje s ciljem skupljanja informacija o sustavu i planiranja daljnjih koraka. Nakon toga, napadač izvodi takozvano bočno kretanje (*lateral movement*) upotrebom ukradenih podataka i stečenog znanja o sustavu kako bi pronašao i ušao u druge dijelove sustava. Ovaj korak se izvodi dok napadač ne pronađe sve podatke koji ga zanimaju, s naglaskom na podatke potrebne za

unaprjeđenje razine identifikacije koju posjeduje, a koja je potrebna za ulazak u bitnije i strože čuvane dijelove sustava. Sljedeći korak je dolazak do cilja, odnosno dolazak do kritičnih dijelova ili podataka unutar sustava koji su od važnosti napadaču te krađa tih podataka, obično upotrebom zloćudnih programa, servisa u oblaku ili FTP-a postavljenog za napad. Posljednji korak u lancu jest očuvanje prisutnosti postavljenjem stražnjih vrata (*backdoor*) kako bi kasnije mogao ponovno ući u sustav. [3]

Bitan korak u upravljanju incidentima jest njihovo otkrivanje i suzbijanje. Kada je računalni sustav probijen, može proći 100 do 200 dana prije nego što prijetnja bude otkrivena te dodatnih 20 do 30 dana prije nego što bude suzbijena. Kada incident bude otkriven, za upravljanje njime je zadužena osoba ili tim koji ima znanje potrebno za pravilnu analizu i procjenu težine incidenta, otkrivanje koji su sve korisnici i dijelovi sustava pod rizikom te na kraju za čišćenje sustava kako bi se spriječili budući incidenti. [3]

Jedan od ključnih elemenata za pravovremeno otkrivanje incidenta jest postojanje izvora podataka koji se mogu iskoristiti kao primjereni resurs. U tu svrhu koriste se DNS zapisi za otkrivanje domena ili IP adresa s kojih potječu pokušaji ulaska u sustav, zapisi elektroničke pošte za identifikaciju adresa s kojih su poslani mogući *phishing* napadi, zapisi pristupa sustavu, kako fizičkih, tako i mrežnih, zapisi servisa operacijskog sustava i aplikacija za identifikaciju zloćudnih ili neuobičajenih aktivnosti, zapisi daljinskog pristupa (*remote access*) za prepoznavanje neuobičajenih adresa i vremena pristupa te *Web proxy* zapisi za identifikaciju HTTP odgovora i zloćudnog mrežnog prometa. Vjerojatnost otkrivanja povećana je uspostavljanjem pravilnih protokola, a može uključivati metode otkrivanja i sprječavanja napada, preporuka za prijavu i odgovore na prijave otkrivenih mogućih zloćudnih aktivnosti te resurse za otkrivanje i sprječavanje takvih aktivnosti. Dodatno se preporučuje praćenje i analiziranje korisničkih aktivnosti poput pretjeranog udvostručavanja i izmjenjivanja datoteka, nedozvoljenog ili pretjeranog korištenja prenosivih medija, spajanje nepoznatih uređaja na računalo unutar sustava, prebacivanje podataka na nedozvoljene servise te korištenje nedozvoljenih VPN-ova, sustava za prijenos podataka ili mreža za očuvanje anonimnosti. [2]

Suzbijanje incidenta, odnosno čišćenje sustava odnosi se na više mogućih postupaka, ovisno o vrsti i cilju incidenta. U slučaju curenja podataka potrebno je spriječiti daljnje curenje te obavijestiti i savjetovati korisnike čiji su podaci kompromitirani. Kada je riječ o zarazi zloćudnim programom, potrebno je što prije izolirati pogođene sustave i pregledati ih kako

bi se program mogao otkloniti ili kako bi se sustav obnovio ako se uspostavi da program nije moguće sigurno ukloniti. Nakon nedozvoljenog pristupa najbolje je uspostaviti sustav sprječavanja daljnjeg takvog pristupa i protokole za slučaj da se ponovi te izbaciti korisnika koji je nedozvoljeno pristupio sustavu. Dodatno je moguće dopustiti takav pristup ako je cilj upravljanja incidentom promatranje kako bi se ustanovili ciljevi napada. [2]

Tijekom svih koraka upravljanja incidentom, pravilno praćenje i bilježenje događaja osigurava da se pravovremeno mogu poduzeti potrebni koraci u slučaju novih incidenata i procijeniti moguće rizične aktivnosti. Ono obuhvaća sljedeće podatke čiji je integritet potrebno osigurati [2]:

- datum odvijanja događaja
- datum opažanja događaja
- opis događaja
- poduzeti koraci
- osoba/organizacija kojoj je događaj/incident prijavljen

Navedeni podaci, kao i svi oni koji su korišteni za otkrivanje incidenta, a koje je sustav zabilježio ili služe kao dokazi događaja, potrebno je naknadno analizirati kako bi se dobio uvid u cjelokupni tijek incidenta, cilj napadača, moguće nastale štete i slično. Analiza takvih podataka korisna je u naknadnim napadima, za unaprjeđenje sigurnosti sustava te za obučavanje osoblja čija je odgovornost očuvanje sigurnosti i/ili suzbijanje incidenata. Iako je moguća i ručna analiza, u sljedećem poglavlju naglasak će se staviti na automatiziranu analizu, tj. upotrebu alata za analizu, pregled i upravljanje prikupljenim podacima te posebice na njihovu vizualizaciju.

3. Vizualizacija prikupljenih podataka

Cilj bilo koje djelotvorne analize, a tako i analize sigurnosnog incidenta, tj. prikupljenih podataka jest otkriti koje je akcije napadač proveo i proučiti ih zajedno s pojavama koje su nastale na sustavu kao posljedica njihova izvođenja te donijeti zaključke. Veliku ulogu u analizi i donošenju zaključaka igra vizualizacija prikupljenih podataka. Vizualizacija podataka odnosi se na prevođenje podataka u vidljivi kontekst, kao što su mape ili grafovi, kako bi se podaci lakše razumjeli te kako bi se iz njih lakše izvukli zaključci. Glavni cilj vizualizacije u bilo kojem slučaju jest otkriti uzorke, kretanja (*trend*) i iznimke u velikim skupovima podataka. Općenito, vizualizacija je jedan od dijelova znanosti o podacima koji kaže da je podatke, nakon što se prikupe, procesiraju i modeliraju, potrebno vizualizirati kako bi se izveli zaključci. [4]

U današnje vrijeme sve je veći broj podataka kojima se rukuje i koje se analizira te je upravo zbog toga vizualizacija bitna. Pravilan prikaz uzoraka, kretanja i iznimaka isključuje suvišne podatke i stavlja naglasak na one ključne te s time pomaže u izvođenju ispravnih zaključaka i donošenju odluka na temelju podataka. Kako bi se podaci prikazali na ispravan način, potrebno je izabrati ispravan način vizualizacije. Odabirom neispravnih oblika, poruka koju želimo prenijeti njihovom uporabom neće biti uspješno prenesena ili se neće moći izvesti ispravni zaključci. Postoji veliki broj načina vizualizacije, najčešći od kojih su dijagrami, tablice, grafovi i mape kao što su površinski dijagrami, stupčasti dijagrami, grafovi distribucije, vremenske crte i drugi [5].

Kada je riječ o vizualizaciji podataka tijekom analize sigurnosnih incidenata, cilj je ustanoviti i prikazati učestalosti obzirom na vrijeme i/ili mjesto događaja, tijek događaja, odnosno smještaj događaja u vremenu i odnose, tj. povezanosti između događaja.

Učestalost podataka najčešće se ostvaruje izvođenjem potrebnih izračuna te potom prikazivanjem dobivenih uzoraka u obliku nekog od dostupnih dijagrama koji odgovara vrsti uzorka koji se želi prikazati. Najjednostavniji primjereni prikaz može se ostvariti upotrebom stupčastog dijagrama iz kojeg su brzo vidljiva različita vremena (na primjer dani kada su se odvijali događaji), razlike u broju događaja po vremenu njihova odvijanja, kada se najviše ili najmanje događaja odvijalo i ostali slični podaci [6].

Vremenska crta u kontekstu ovog rada predstavlja organizirani pogled na događaje koji su se odvijali na računalnom sustavu, gdje je događaj akcija provedena na sustavu od strane

samog sustava, korisnika ili nekog drugog okidača te koju je sustav prepoznao [7]. Cilj ovakvog pregleda podataka jest ustanoviti vremenski odnos između događaja, na primjer koji je događaj prethodio nekom drugom događaju, kako bi se otkrile moguće uzročno-posljedične veze između njih te na temelju toga izveli zaključci o tome što se i kada dogodilo i što je dovelo do određenih pojava na sustavu [8].

Nevremenske odnose između događaja, ako postoje, ipak prikazujemo drugim oblicima vizualizacije, najčešće povezanim grafovima. Ovakav graf čini skup čvorova koji predstavljaju željene kategorije, u ovome kontekstu atribute unutar događaja kao što je napadnuto računalo ili korisnik sustava, te linije koje predstavljaju odnose između čvorova. Prednost ovog oblika vizualizacije jest u pronalasku popularnosti određenog primjerka atributa, tj. učestalost njegovog pojavljivanja u incidentu što može ukazati na mogući cilj napada. [9]

Do nedavno, analize napadnutih sustava i podataka koji su prikupljeni tijekom upravljanja incidentom bile su izvođene ručnim pregledom. Iako se neki sustavi i podaci i dalje analiziraju na taj način, računalni su sustavi postali složeni te je ručna analiza takvih sustava postala naporna i neučinkovita. Kako bi se ubrzao proces analiziranja i otkrivanja ključnih informacija, počeli su se razvijati alati koji automatiziraju taj proces i/ili pružaju pregledniji uvid u prikupljene podatke. Takvi alati mogu izvući, filtrirati, kategorizirati i pružiti pregled, odnosno vizualizaciju podataka prikupljenih s napadnutih sustava i podatka prikupljenih tijekom upravljanja incidentima.

Postoji cijeli niz alata koji pružaju spomenute funkcionalnosti. Neki od njih služe isključivo izvlačenju ključnih podataka iz nečitljivih ili teško čitljivih zapisa, kao što je alat Plaso koji može stvoriti zapis vremenske crte [10]. Zatim postoje alati koji pružaju automatiziranu analizu i donošenje zaključaka na temelju zapisa kao što je alat Turbinia [11]. Konačno, postoje i alati poput alata Timesketch opisanog u nastavku rada koji daju bolji pregled podataka, određene vizualizacije, te posebice vremenske crte događaja.

3.1. Alat Timesketch

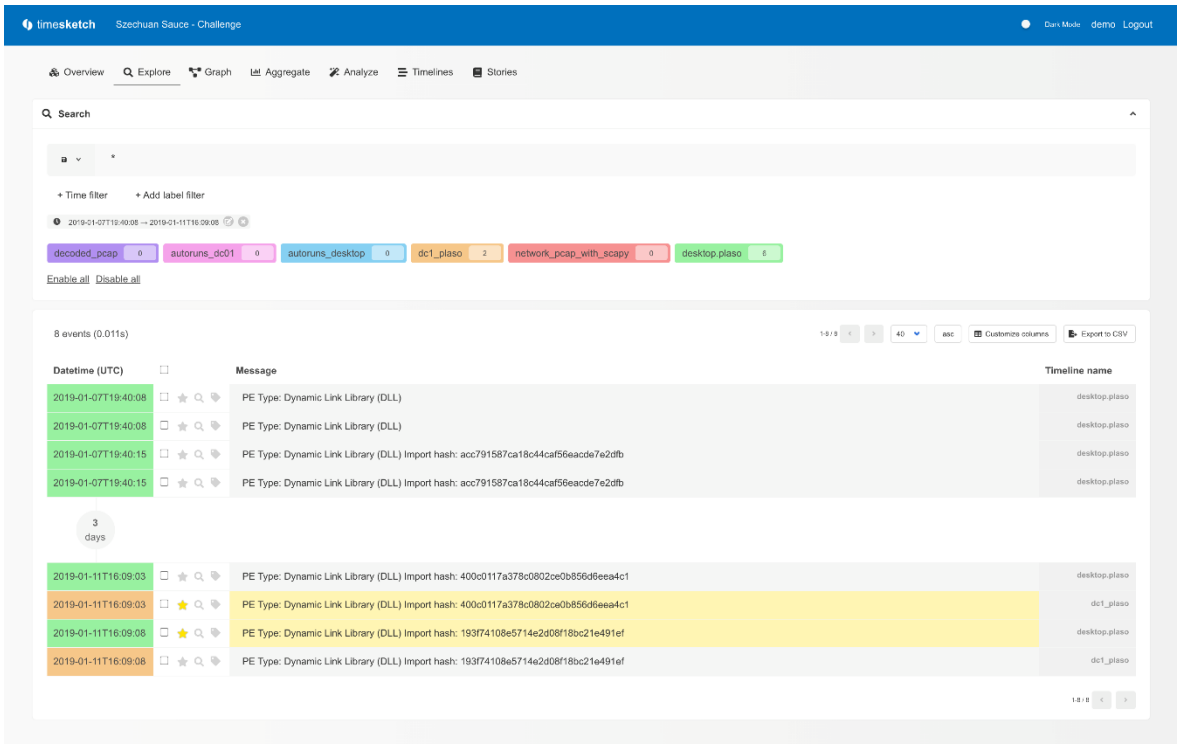
Timesketch je alat namijenjen analizi događaja, primarno onima koji su se odvijali na računalnim sustavima [12]. Korištenjem takozvanih „skica“ olakšana je organizacija i analiza podataka te im je dano značenje označavanjem i komentarima. Jedna skica

predstavlja jedan slučaj, tj. istragu. Svaka skica može imati više vremenskih crta i prikaza kroz koje je omogućeno pretraživanje, filtriranje i pregledavanje podataka. Ulaz alata predstavlja CSV ili JSONL datoteka s popisom događaja označenih sa sljedećim atributima:

- *message* – podatak o događaju
- *datetime* – datum događaja
- *timestamp_desc* – podatak o tome što autogenerirani *timestamp* označava(npr. „file created“)
- proizvoljni atributi koje korisnik može dodati kako bi bolje opisao događaj

3.1.1. Karakteristike

U odjeljku *Explore* prikazanom na slici Sl. 3.1 dana je vremenska crta u tabličnom obliku. Ovaj dio alata glavna je funkcionalnost Timesketch-a te pruža detaljan uvid u događaje poredane kronološkim redoslijedom s jasno naglašenim većim vremenskim razmacima (na slici vidljivo kao bjelina u tablici s istaknutim brojem dana koji su protekli između događaja). Dodatno, u jednom tabličnom prikazu moguće je vidjeti više međusobno isprepletenih vremenski crta istovremeno, gdje je svaka od njih označena vlastitom bojom. Pritiskom na pojedinu liniju u tablici proširuje se redak te se daju pojedinosti o događaju, tj. popisani su svi stupci s pripadnim vrijednostima koji su se nalazili u datoteci koju je unio korisnik. Nadalje, kao što je spomenuto u prijašnjem poglavlju, dvije funkcionalnosti koje želimo imati tijekom analize događaja su mogućnost pretraživanja i filtriranja podataka. Alat Timesketch nudi opciju pretraživanja u ovome odjeljku kroz primjenu običnog tekstualnog pretraživanja, logičkih operatora AND, OR i NOT, regularnih izraza i slično. Nakon što se pretraživanje izvede, moguće ga je i spremiti te je ono potom vidljivo i lako dostupno prilikom svakog sljedećeg pretraživanja. Filtriranje se također izvodi na ovome mjestu primjenom filtera na temelju komentara, atributa, već ugrađenih oznaka (*starred*), oznaka (*labels*) koje je korisnik samostalno dodao za obilježavanje događaja i tako dalje.

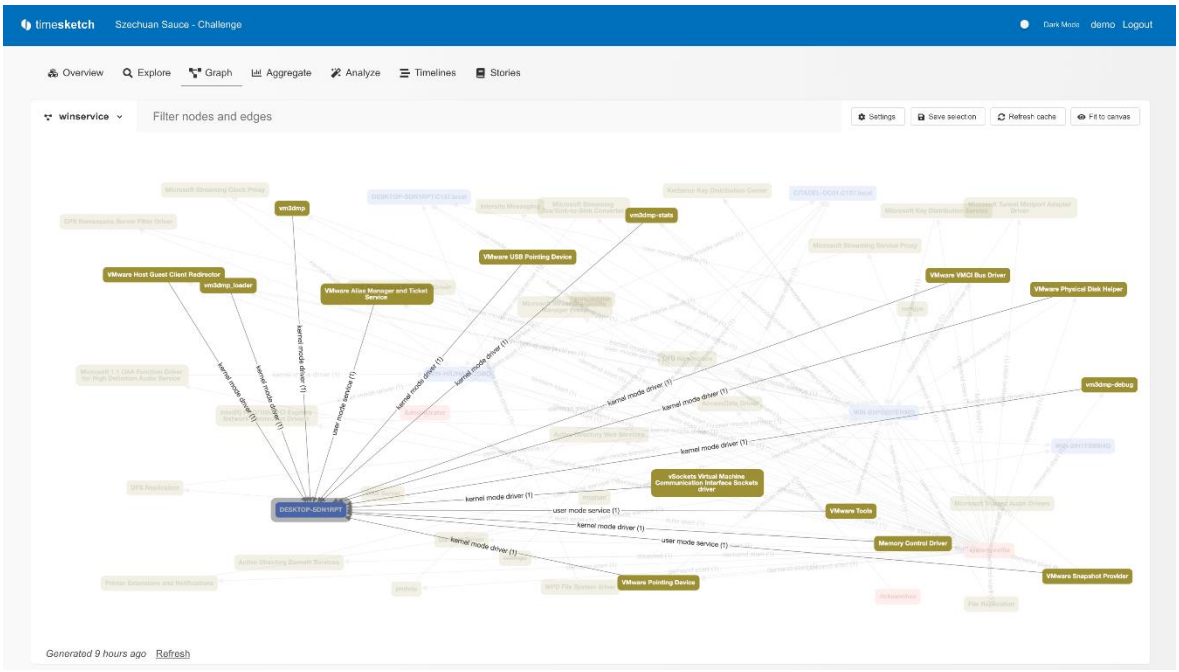


Sl. 3.1 Prikaz odjeljka *Explore* [12]

U sljedećem odjeljku dolazi do izražaja funkcionalnost vizualizacije gdje korisnik može generirati grafove povezanosti na temelju unesene vremenske crte. Izgled samog grafa ovisi o vrsti zapisa koji predstavlja vremensku crtu. Na slici Sl. 3.2 je tako prikazan primjer grafa stvorenog iz zapisa koji se odnosi na servise pokrenute na operacijskom sustavu Windows te je na njemu vidljiva povezanost između računala, korisnika koji je koristio računalo i servisa koji su se izvodili tijekom njegova korištenja.

Zatim, dodatno je pružena funkcionalnost agregacije podataka (odjeljak *Aggregate*) te s time vizualizacija u obliku dijagrama (kružni, stupčasti, linijski, tablični) koji omogućuju dodatne uvide na temelju učestalosti događaja s obzirom na polja po korisnikovom izboru.

U odjeljku *Analyze* korisnik može iskoristiti automatizirane analizatore podataka kako bi dobio nove podatke, odnosno kako bi dobio zaključke izvedene na temelju tih postojećih podataka. Tako na primjer analizator iz URL-a može izvući pojmove koji su pretraživani ili iz povijesti pretraživanja izdvojiti ona pretraživanja koja su izvedena izvan standardnog perioda aktivnosti. U zadnjem odjeljku (*Stories*) moguće je napisati narativni opis istraživanja događaja te ga ujediniti s podacima iz vremenske crte ili dijagramima koji su rezultat agregacije podataka.



Sl. 3.2 Prikaz generiranog grafa [12]

3.1.2. Nedostaci

S obzirom na opisane karakteristike alata, Timesketch je sveobuhvatan u analizi događaja sa svojom raznovrsnošću funkcionalnosti koje pruža. Ipak, unatoč tome su njegovi najkorisniji dijelovi inicijalno primjenjivi samo na specifične slučajeve ili ne pružaju oblik vizualizacije koji se očekuje kada se spominju pojmovi „vremenska crta“ i „vizualizacija“.

Dok odjeljak za stvaranje grafova generira korisne grafove, oni su trenutačno primjenjivi samo na tri oblika postojećih zapisa: Windows servisi, Windows prijave i povijest preuzimanja preglednika Google Chrome. Ako korisnik želi generirati grafove za vlastite zapise, mora samostalno napisati Python skriptu za izvlačenje ključnih podataka i njihovo povezivanje u graf. Dodatno, grafovi koji se generiraju za spomenute zapise su isključivo grafovi povezanosti te su korisni samo ako nas zanima odnos između pojedinih atributa događaja, odnosno u kojim se događajima pojavljuju isti atributi.

Isto vrijedi i za analizatore događaja. Timesketch pruža preddefinirane analizatore, ali su oni primjenjivi samo na postojeće vrste zapisa kao što je povijest pretraživanja te korisnik treba napisati vlastite ako želi koristiti ovu funkcionalnost nad drugim zapisima.

Kako se u ovome radu stavlja naglasak na vizualizaciju podataka, posebice na vremensku crtu, važno je reći da ovaj alat ne pruža vizualizaciju vremenske crte. Ona je prikazana samo

u tabličnom obliku te s velikim brojem događaja unutar jednog zapisa, pogotovo ako su događaji vremenski blizu jedan drugome, pojam vremenske crte u njegovu kontekstu gubi na značajnosti te taj prikaz postaje običan tablični popis. S time se korisniku otežava intuitivni uvid u vrijeme i učestalost odvijanja događaja te on više nema razloga koristiti ovaj alat kao preglednik vremenske crte.

U sljedećem poglavlju, osim što će se po uzoru na funkcionalnost agregacije podataka dati primjer vizualizacije temeljen na učestalosti događaja, opisat će se i jednostavan način implementacije vremenske crte u obliku grafa.

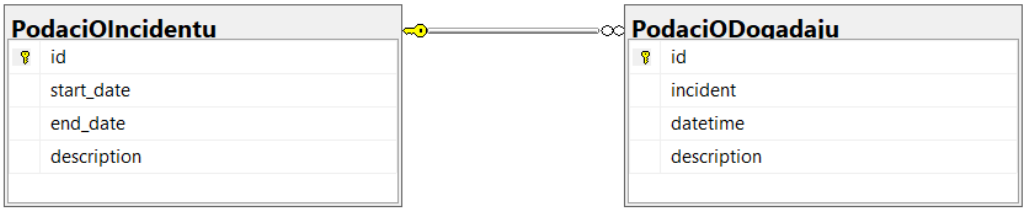
4. Metode vizualizacije u ASP.NET Core aplikaciji

Za izradu aplikacije korišten je ASP.NET Core razvojni okvir upotrebom jezika C# unutar alata Microsoft Visual Studio. Aplikacija je oblikovana kao MVC (*Model-View-Controller*) aplikacija. Modeli korišteni u aplikaciji, kao i sama baza podataka, implementirani su pomoću Microsoft SQL Servera, ali je moguće koristiti neku drugu bazu podataka. Kako bi se baza podataka mogla povezati s aplikacijom, potrebno je osigurati ispravnu poveznicu unutar datoteka *appsettings.json* i *Program.cs*. Prikupljanje podataka iz baze podataka ostvareno je *code-first* pristupom i korištenjem Entity Framework Corea. Za *frontend* aplikacije korišten je inicijalno dostupan izgled nastao stvaranjem projekta unutar razvojnog okruženja s ponekim izmjenama i dodacima za bolju preglednost funkcionalnosti aplikacije.

4.1. Opis modela i biblioteke

Glavni cilj ovog projekta jest pružiti jednostavne načine vizualizacije bez obzira na kontekst aplikacije, odnosno vizualizacija se može odnositi na bilo koji skup podataka, ali se zbog konteksta ovog rada korišteni podaci odnose na incidente i događaje koji njima pripadaju. S obzirom na to da se naglasak ne stavlja na složenost korištenih podataka i modela, koriste se što jednostavniji modeli kao što je prikazano na slici Sl. 4.1.

Prvi model `PodaciOIncidentu` opisuje incident te daje informacije o početku i kraju incidenta i kratki opis istoga. Svaki incident je jedinstven te može imati neograničen broj događaja koji su opisani modelom `PodaciODogadaju`. Događaji su povezani s incidentom kojem pripadaju preko `id`-a incidenta te imaju svoje vrijeme izvođenja (na primjer vrijeme stvaranje datoteke) i kratki opis. Za svrhe rada stvoreni su zapisi nekoliko incidenata, ali je samo jedan od njih popunjen s većim brojem događaja te korišten za testiranje i izvođenje aplikacije. Spomenuti zapisi bit će prikazani u sklopu potpoglavlja o implementaciji, a njihov sadržaj sadrži izmišljene podatke kada je riječ o datumu incidenta i događaja te privremene (takozvane *placeholder*) podatke za njihov opis pošto on nije bitan za izvršavanje ključnih dijelova aplikacije.

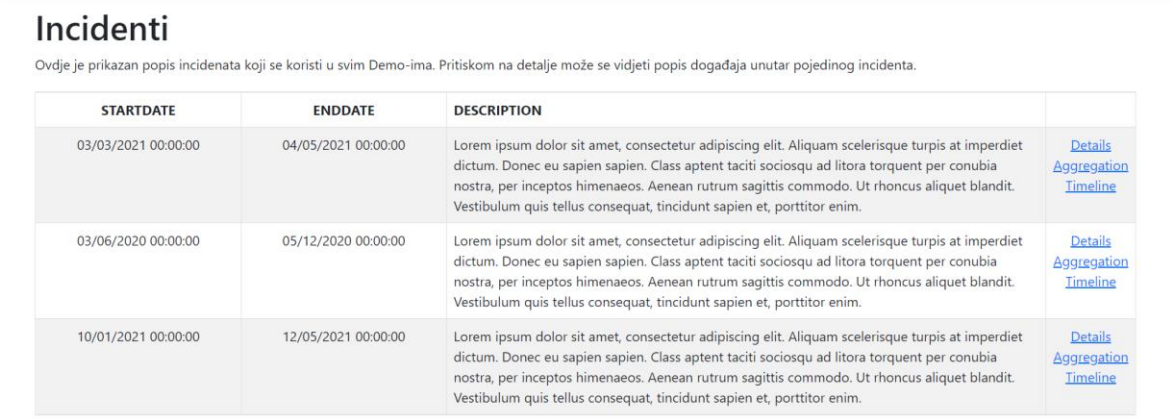


Sl. 4.1 Model podataka korištenih u aplikaciji

Nadalje, za implementaciju vizualizacije korištena je Javascript biblioteka Chart.js [13]. U sklopu biblioteke nalazi se osam vrsta dijagrama svakog od kojih je moguće urediti, animirati i prikazati prema vlastitim potrebama: površinski, stupčasti, mjehurasti, tortni, linijski, polarni, radarni i raspršeni s mogućnosti spajanja dva ili više dijagrama u jedan. Jedan od dijelova dijagrama koje je moguće prilagoditi jest interakcija s dijagramom, tj. s čvorovima dijagrama. Početna postavka dijagrama jest da se prelaskom pokazivača prikaže prozor (*tooltip*) s vrijednostima čvora, ali je ovu funkcionalnost moguće prilagoditi tako da se pritiskom na čvor dogodi nešto drugo, što je iskorišteno u ovoj aplikaciji kod implementacije vremenske crte. Biblioteka se u projekt uključuje upotrebom *npm*-a, preuzimanjem Github direktorija ili korištenjem CDN poveznice. Za prikaz dijagrama potrebno je napisati skriptu koja stvara dijagram i HTML oznaku *canvas* na mjestu gdje ga želimo smjestiti unutar pogleda (*View*).

4.2. Implementacija

Za potrebe uvida u podatke o incidentima i događajima, stvorene su podstranice koje u tabličnom obliku sadrže sve podatke o modelima. Za svaki incident su navedene poveznice na detalje o incidentu gdje se nalazi prikaz svih događaja koji mu pripadaju, te na podstranice koje sadrže agregacijski prikaz i vremensku crtu. Na slici Sl. 4.2 prikazan je tablični prikaz incidenata, a slično izgleda i podstranica s detaljima o incidentu.



Sl. 4.2 Tablični prikaz svih incidenata

Prije samog popunjavanja modela s podacima iz baze podataka, potrebno ih je prilagoditi za povezivanje s osima dijagrama. U tu svrhu dodani su modeli pogleda (*view models*) koji osim svih atributa koje sadrži osnovni model, također imaju atribut `Quantity` kojim se definira vrijednost na y-osi. Iako se modeli podataka korišteni za pojedini dijagram razlikuju, podatke u pogledu (*View*) pripremamo na isti način (jedina je razlika naziv atributa – kod modela za stupčasti dijagram korišten je naziv *Date*, a kod modela za vremensku crtu *Datetime*). Kako bi skripta mogla koristiti željene podatke, odnosno da bi ih ispravno pročitala prilikom povezivanja s osima, potrebno ih je serijalizirati u JSON oblik:

```
    var XLabels =
Newtonsoft.Json.JsonConvert.SerializeObject(Model.Select(x =>
x.Date).ToList());

    var YValues =
Newtonsoft.Json.JsonConvert.SerializeObject(Model.Select(x =>
x.Quantity).ToList());
```

4.2.1. Vizualizacija učestalosti odvijanja događaja

Za implementaciju vizualizacije učestalosti korišten je stupčasti dijagram iz biblioteke Chart.js. Kako bi se vizualizirali željeni podaci, potrebno je iz baze podataka skupiti broj događaja po svakom danu za koji se pojavljuje događaj što je izvedeno kroz funkcionalnosti Entity Framework Corea. Prikupljeni podaci se potom prosljeđuju u pogled (*View*) u kojem je smješten dijagram te gdje se prebacuju u valjani oblik prije nego što se koriste za stvaranje dijagrama.

Kao što je rečeno prethodno pri opisu Chart.js biblioteke, za prikaz dijagrama potrebno je imati skriptu koja stvara traženi dijagram. U njoj prvo definiramo vrstu dijagrama koja se želi koristiti, te predajemo podatke definiranjem osi dijagrama:

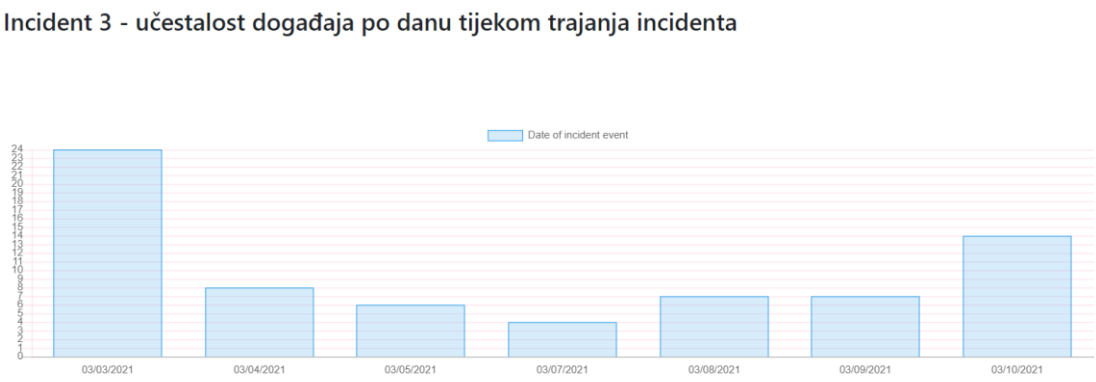
```
        var chartName = "chart";
        var ctx =
document.getElementById(chartName).getContext('2d');
        var data = {
           labels: @Html.Raw(XLabels),
           datasets: [{
              label: "Date of incident event",
              ...
              data: @Html.Raw(YValues)
           }]
        };
```

Kod 4.1 – isječak koda za definiranje podataka za dijagram

Zatim se definiraju opcije u kojima uključujemo osi dijagrama te željene opcije za svaku od njih kao što su izgled osi, raskorak između vrijednosti na osima, minimalne i maksimalne vrijednosti koje se žele prikazati na njima te drugi slični podaci koji za potrebe ovog dijagrama nisu od važnosti. Nakon što su definirani svi podaci i opcije koji se koristi za stvaranje dijagrama, sam dijagram definiramo sljedećim kodom:

```
     var myChart = new  Chart(ctx, { options: options, data: data,
type:'bar' });
```

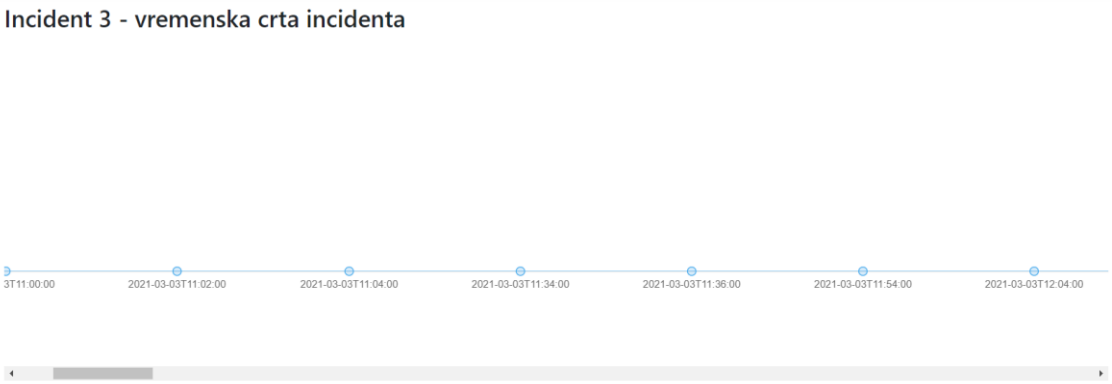
Dobiveni dijagram postavljen je na stranici pomoću oznake *canvas* kao što je prikazano na slici Sl. 4.3.



Sl. 4.3 Dijagram učestalosti događaja po danu

4.2.2. Vizualizacija vremenske crte

Iako postoje neke biblioteke ASP.NET Core okruženja, one imaju svoje mane ili zahtjeve koje ne bi bilo moguće izbjeći ili zadovoljiti u sklopu ove aplikacije. Stoga je za vremensku crtu prikazanu na slici upotrijebljen linijski dijagram biblioteke Chart.js. Kako bi bila osigurana ravnopravnost događaja, tj. da svi događaji budu na istoj razini, vrijednost y-osi za sve je događaje postavljena na nula.



Sl. 4.4 Vremenska crta implementirana upotrebom linijskog dijagrama

Prikupljanje podataka za vremensku crtu izvodi se na isti način kao kod njihova prikupljanja za prikaz detalja incidenta, a priprema za dijagram na ranije opisani način. Podaci dijagrama definirani su kao i za stupčasti dijagram, a jedina razlika je u tome da su čvorovi koji predstavljaju događaje povećani za potrebe olakšanog pritiska pokazivačem:

```
pointRadius: 5,
pointHoverRadius: 5
```

Dijagram se stvara kroz kod sličan onome za stupčasti dijagram u prethodnom potpoglavlju:

```
var myChart = new  Chart(ctx, { options: options, data: data, type:'line' });
```

Zatim, kako bi događaje bilo moguće smjestiti na crtu s preglednim razmacima i potom prolaziti kroz crtu korištenjem pomične trake (*scroll*), treba se postaviti sljedeće opcije:

```
responsive: true,

maintainAspectRatio: false
```

Za razliku od stupčastog dijagrama, kako bi željeni oblik pregleda bio funkcionalan, širinu i visinu dijagrama ne može se definirati unutar oznake *canvas* već je potrebno te atribute postaviti nad elementima *div* unutar kojih se nalazi dijagram, zajedno sa omogućavanjem scroll-a po x-osi.

```
<div class="box-body wrapper" style="height: 400px; width: 100%; position: relative; overflow-x: scroll;">

   <div class="chart-container" style="width: @(Model.Count * 200 + "px"); height: 300px;">

      <canvas id="chart"></canvas>

   </div>

</div>
```

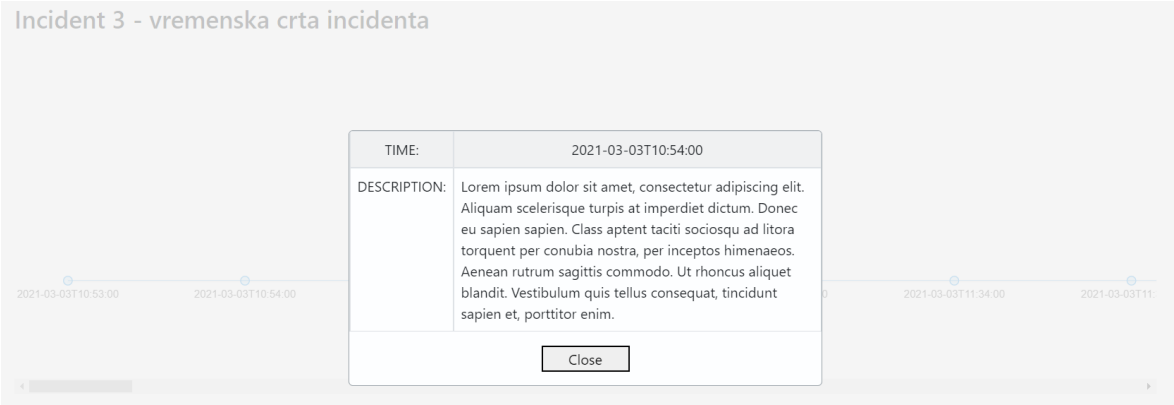
Kod 4.2 – Isječak koda koji prikazuje uključivanje dijagrama i vrijednosti atributa

U ovoj implementaciji odabran je prikaz podataka o događaju korištenjem modalnih prozora kao što je prikazano na slici Sl. 4.5. Zbog redundantnosti prikaza podataka, u opcijama je potrebno isključiti uobičajeni prikaz preko tooltipa:

```
tooltips: { enabled: false }
```

Dodatno je radi preglednijeg prikaza, odnosno kako bi vizualizacija što manje izgledala kao dijagram, a više poput vremenske crte, potrebno isključiti prikaz y-osi i mrežnih linija za obje osi na pripadajućim mjestima u opcijama:

```
display: false
```



Sl. 4.5 Modalni prozor s podacima o događaju

Kod ranijeg opisivanja biblioteke Chart.js, navedena je mogućnost izmjene interakcija s čvorovima dijagrama. Ova opcija korištena je kod vizualizacije kako bi se omogućilo iznad spomenuto prikazivanje podataka o događaju unutar modalnog prozora. Željenu funkcionalnost postavljamo definiranjem aktivnosti `onclick` u opcijama dijagrama. Kada se pokazivačem pritisne na čvor na dijagramu, poziva se funkcija unutar koje se uzima indeks odabranog čvora. Indeks čvora se potom koristi za pronalazak događaja u listi koja je prethodno serijalizirana. Modalni prozor prikazujemo postavljanjem atributa `display`.

```
onClick: function(c,i){
    e = i[0];
    var items = @Html.Raw(JsonConvert.SerializeObject(Model))
    var index = e._index;
    document.getElementById("time").textContent =
items[index].Datetime;
    document.getElementById("desc").textContent =
items[index].Description;
    document.getElementById("information").style.display =
"block";
}
```

Kod 4.3 – Isječak koda aktivnosti *onclick* za otvaranje modalnog prozora

4.3. Budući rad

Kao i svaki drugi projekt, i ovaj ima mjesta za unaprjeđenje. Do izražaja poglavito dolazi vremenska crta i praznina oko njenog prikaza. Korisno bi bilo implementirati mogućnosti približavanja i udaljavanja od pojedinog događaja ili skupa događaja jer bi s time korisnik mogao dobiti uvid u određene mogućnosti agregacijskih funkcionalnosti kao što je pregled grupiranosti događaja po vremenu događanja. Povezano s time jest poboljšanje preglednosti u obliku smanjenja raskoraka između događaja s obzirom na vremenski razmak između njih.

Jedna od jednostavnijih promjena bilo bi uključivanje više agregacijskih dijagrama koji bi pružili drugačije poglede na podatke. Dodatno, pomoću stupčastih dijagrama mogli bi se pružiti uvidi u kretanja događaja u ovisnosti o dobu dana, dana u tjednu, mjeseca i tako dalje.

Funkcionalnost koju ova aplikacija nema implementiranu u pogledu vizualizacije jest graf za prikazivanje povezanosti između događaja. U tu svrhu bilo bi potrebno prvo stvoriti složeniji model podataka. Naime, trenutačno nema podataka na temelju kojih bi se podatke moglo povezati jer oni nisu bili od važnosti za ovu aplikaciju. Primjer podataka koji bi se mogli nadodati su mjesto odvijanja događaja, kako fizičko (na primjer lokacija ureda u kojem se nalazi računalo ili računalni sustav), tako i virtualno (računalo u mreži), zatim korisnika koji je koristio sustav za vrijeme odvijanja događaja ili korisnika koji je vlasnik sustava, osoba ili organizacija kojoj je prijavljen zabilježeni događaj ili koja je istraživala događaj. Za prikaz novog oblika podataka i povezanosti između događaja bilo bi potrebno pronaći ili napraviti novu biblioteku koja bi pružala implementaciju povezanih dijagrama kao onog kojeg sadrži alat Timesketch jer Chart.js nema te mogućnosti.

5. Zaključak

Predočavanjem složenosti sigurnosnih incidenata i koje sve korake napadač treba izvesti kako bi uspješno izvršio napad, vidljiva je važnost pravilnog zabilježavanja događaja te analize tih zapisa i svih ostalih podataka prikupljenih tijekom upravljanja incidentom. Ključ unaprjeđenja računalnih sustava i njihove sigurnosti nalazi se u proučavanju podataka prikupljenih tijekom postojećih i svih budućih incidenata, a to je jedino moguće vođenjem pravilne evidencije te naknadnom analizom svih prethodno spomenutih podataka. Implementacijom novih saznanja, novi se napadi neće spriječiti, ali će biti otežani za izvođenje te potencijalno lakši za prepoznati i suzbiti.

Veliku ulogu u tom procesu igra vizualizacija podataka jer ona pruža uvid u vremenska i prostorna kretanja napadača te moguće iznimke u tim kretanjima koje osobe odgovorne za sigurnost sustava nisu prepoznale zbog prethodnog neznanja. Vizualizacija je općenito proces koji se sve češće koristi u današnjem vremenu jer zbog tehnološkog napretka društva raste broj podataka kojima treba upravljati i koje treba koristiti unutar privatnog i javnog svijeta.

Kroz prikaz jednostavne aplikacije dane su mogućnosti implementacije dvaju vizualnih uvida u prikupljene podatke. S jedne strane, spomenuta jednostavnost omogućuje da bilo tko može uključiti željene funkcionalnosti u svoj projekt, ali s druge strane znači da opisane vizualizacije ne pružaju najbolje uvide. U tome se pogledu aplikacija može značajno unaprijediti kako bi pružila bolje uvide, ali i kako bi uključivala funkcionalnosti koje trenutačno ne sadrži.

Literatura

[1] UK National Cyber Security Centre, *What is a cyber incident*, (2016, rujan). Poveznica: https://www.ncsc.gov.uk/information/what-cyber-incident; pristupljeno 25. ožujka 2022.

[2] Australian Cyber Security Centre, *Guidelines for Cyber Security Incidents*, (2022, ožujak). Poveznica: https://www.cyber.gov.au/acsc/view-all-content/advice/guidelines-cyber-security-incidents; pristupljeno 25. ožujka 2022.

[3] Hayes, K. A., *What is Incident Detection and Response?*, Rapid7, (2016, ožujak). Poveznica: https://www.rapid7.com/blog/post/2016/03/29/what-is-incident-detection-and-response/; pristupljeno 26. svibnja 2022.

[4] Brush, K, *Data visualization*, TechTarget, (2020, veljača). Poveznica: https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization; pristupljeno 25. svibnja 2022.

[5] *What Is Data Visualization? Definition, Examples, And Learning Resources*, Tableau. Poveznica: https://www.tableau.com/learn/articles/data-visualization; pristupljeno 25. svibnja 2022.

[6] Rodgers, T., *Which Type of Chart or Graph is Right for You?*, Tableau. Poveznica: https://www.tableau.com/learn/whitepapers/which-chart-or-graph-is-right-for-you; pristupljeno 25. svibnja 2022.

[7] Metz, J., *Pearls and pitfalls of timeline analysis*, Osdfir, (2021, listopad). Poveznica: https://osdfir.blogspot.com/2021/10/pearls-and-pitfalls-of-timeline-analysis.html; pristupljeno 15. travnja 2022.

[8] Disney, A., *The basics of timeline data visualization*, Cambridge Intelligence, (2020, studeni). Poveznica: https://cambridge-intelligence.com/the-basics-of-timeline-data-visualization/; pristupljeno 25. svibnja 2022.

[9] Guinel, S., *Building a Graph Visualization Tool*, Dataiku, (2020, listopad). Poveznica: https://blog.dataiku.com/building-a-graph-visualization-tool; pristupljeno 25. svibnja 2022.

[10] Plaso. Poveznica: https://plaso.readthedocs.io/en/latest/; pristupljeno 25. svibnja 2022.

[11] Turbinia. Poveznica: https://github.com/google/turbinia; pristupljeno 25. ožujka 2022.

[12] Timesketch. Poveznica: https://github.com/google/timesketch; pristupljeno 15. travnja 2022.

[13] Chart.js. Poveznica: https://www.chartjs.org/; pristupljeno 27. svibnja 2022.

Sažetak

Vizualizacija podataka prikupljenjih tijekom upravljanja incidentom

Sigurnosni incidenti postali su dio svakodnevice s tehnološkim napretkom društva. Kako raste njihova učestalost, ali i složenosti, potrebno je voditi ispravnu evidenciju događaja koji su se odvijali u sustavu i prikupljati podatke s dijelova sustava koji su bili pod utjecajem napadača i njegovih postupaka, a koji su korisni za analizu incidenta. Tijekom analize, nije potrebno samo pregledati podatke, već ih vizualizirati kako bi bilo moguće vidjeti uzorke, kretanja i iznimke u podacima te konačno izvesti zaključke koji mogu koristiti u unaprjeđenju sigurnosti sustava. Ovaj rad opisuje određene metode i oblike vizualizacije, prikazuje jedan od alata za analizu i vizualizaciju podataka te opisuje načine na koje se pojedine vizualizacije mogu implementirati u osobnim projektima.

Ključne riječi: sigurnosni incidenti, analiza podataka, vizualizacija podataka, dijagrami, vremenska crta, Timesketch

Summary

Visualization of data collected during incident response

Security incidents have with technological progress of society become a part of everyday life. As they become more frequent and their complexity grows, it is necessary to ensure proper records of events that have occured in the system and to collect useful data from parts of the system that were impacted by the attacker's actions. During the analysis, it is not only necessary to review the collected data, but to also visualize it with intention of getting a better insight into patterns, trends and exceptions which could lead to conclusions useful for improving the system's security. This thesis describes methods and types of visualization, provides insight into one of the tools for data analysis and visualization and provides ways of implementing certain types of visualization in personal projects.

Key words: security incident, data analysis, data visualization, diagrams, timeline, Timesketch

Skraćenice

FTP	*File Transfer Protocol*	mrežni protokol za prijenos datoteka
DNS	*Domain Name System*	sustav domenskih imena
IP	*Internet Protocol*	internetski protokol
HTTP	*Hypertext Transfer Protocol*	protokol za prijenos hiperteksta
VPN	*Virtual Private Network*	privatna virtualna mreža
CSV	*Comma-separated Values*	zarezom odvojene vrijednosti
JSON/JSONL	*Javascript Object Notation/Javascript Object Notation Lines*	zapis Javascript objekta/linijski zapis Javascript objekta
URL	*Uniform Resource Locator*	usklađeni lokator sadržaja
CDN	*Content Delivery Network*	mreža za dostavljanje podataka
HTML	*Hypertext Markup Language*	jezik za označavanje hiperteksta