Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/25BE41Cickovic

WEB APLIKACIJA ZA PRETRAGU WINDOWS LOGOVA

WEB APPLICATION FOR SEARCHING WINDOWS EVENT LOGS

Anđela Čičković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *Rad pruža pregled teorijskih osnova u oblast logova događaja na Windows operativnom sistemu. Takođe, u okviru ovog rada implementirana je i detaljno opisana aplikacija za prikupljanje, pretragu i filtriranje logova događaja unutar lokalne mreže.*

Ključne reči: *Log događaja, Windows, prikupljanje, pretraga, filtriranje*

Abstract – *This paper provides an overview of the theoretical foundations about event logging on Windows operating system. An application for collecting, searching and filtering event logs in local network has been implemented and described as part of this paperwork, as well.*

Keywords: *Event log, Windows, collecting, searching, filtering*

1. UVOD

Ubrzani razvoj računarskih sistema i svjetska zavisnost od tehnologije u današnje vrijeme nose sa sobom značajan napredak, ali i brojne izazove. Računarski sistemi postali su kompleksniji, obimniji i brži, pa se i broj procesa koji se odvijaju u njima eksponencijalno povećao. Samim tim, stvorila se potreba za efikasnijim metodama praćenja i upravljanja sistemima, pa je logovanje događaja (engl. *event logging*) u sistemu postalo izuzetno važno [1].

Logovi predstavljaju važan resurs u savremenim računarskim sistemima. Oni vjerno prikazuju redoslijed aktivnosti u radu sistema, odnosno, prikazuju istoriju događaja koja može pomoći da se isprati bilo koja aktivnost u sistemu [2]. Logovi vrše evidentiranje informacija o neuspjelim pokušajima pristupa sistemu, neovlašćenim radnjama, greškama, upozorenjima, uspješnim radnjama ili bilo kojoj sumnjivoj aktivnosti koja bi mogla ugroziti sistem [3]. Kroz logove se može vršiti nadzor rada sistema, što dalje omogućuje sprovođenje analize, optimizacije, te ranog otkrivanja potencijalnih problema i bezbjednosnih rizika [4]. Uslijed složenosti i obima današnjih sistema, količina logova koja se generiše brzo raste, pa efikasno upravljanje logovima i analiza istih predstavljaju veliki izazov. Tu na scenu stupaju aplikacije za pretragu i filtriranje logova. Ove aplikacije omogućavaju sistemskim administratorima da efikasno pretražuju, filtriraju i analiziraju logove, što im omogućava brzo reagovanje na potencijalne probleme i sigurnosne prijetnje i rizike [5].

Kao dio ovog rada izvršeno je istraživanje i analiza procesa logovanja događaja na računarima koji koriste *Windows* operativni sistem. Takođe, implementirano je softversko rješenje za prikaz, pretragu i filtriranje logova sa bilo kojeg računara u lokalnoj mreži.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Goran Sladić, red. prof.

2. WINDOWS EVENT LOGS

Već u uvodnom poglavlju su navedeni neki od najznačajnijih pojmova koji će se višestruko pominjati dalje u radu. Zbog toga, bitno je na samom početku razumjeti svaki od njih, njihovu međusobnu povezanost i namjenu.

Događaj (engl. *event*) predstavlja promjenu stanja sistema, pri čemu neki događaji predstavljaju redovne sistemskih aktivnosti (na primjer: uspješna prijava korisnika), dok drugi mogu da predstavljaju nepredviđene promjene koje su se desile u sistemu (na primjer: neuspješna instalacija drajvera) [6].

Evidencija događaja omogućava bilježenje svih događaja u operativnom sistemu, kako bi se na taj način osigurala kontrola pristupa i sigurnost sistema. Događaji se bilježe u **logove događaja** (engl. *event logs*) [7].

Logovi događaja, u daljem tekstu samo logovi, su fundamentalni fajlovi operativnog sistema, koji u svakom trenutku pružaju mogućnost uvida u sve promjene stanja sistema [8].

Počevši od *Microsoft Windows Vista* verzije operativnog sistema, *Microsoft* koristi *Windows Event Log* sistem za logovanje, koji događaje dijeli u dvije kategorije:

1. Windows logs – kategorija namijenjena čuvanju događaja koji se odnose na cijeli sistem. Unutar ove kategorije razlikuje se pet tipova događaja:

- **Application** – događaji na raspolaganju korisničkim aplikacijama, kako bi bilježile događaje od značaja za cijeli sistem;
- **Security** – događaji vezani za bezbjednost sistema i upotrebu sistemskih resursa;
- **System** - sistemski događaji, odnosno događaji u različitim dijelovima operativnog sistema;
- **Setup** – događaji nastali prilikom instalacije *Windows* operativnog sistema;
- **Forwarded Events** - događaji koji su proslijeđeni sa drugih servera, ukoliko je proslijeđivanje događaja konfigurisano.

2. Application and Services logs – kategorija logova namijenjena čuvanju događaja izazvanim aplikacijama ili

komponentama aplikacije, a koji se ne odnose na cijeli sistem [9].
Tema ovog rada jesu *Windows* logovi. Stoga, u daljem tekstu će se podrazumijevati da, kada se pominju logovi, to se odnosi na prethodno opisane tipove *Windows* logova, bez *Forwarded events* tipa.
Kada je u pitanju format zapisa logova koristi se **EVTX** (*Windows Extensible Markup Language (XML*) *Event Log*) format. Ovaj format podrazumijeva čuvanje podataka u *XML* formatu, što omogućava visok nivo detaljnosti prilikom pregledanja logova od strane *third party* softvera. Za opisivanje logova koriste se *property*-ji, a najznačajniji i najčešći u upotrebi, predstavljeni su u tabeli 1.

Tabela 1. *Property*-ji *kojima se opisuju logovi*

Naziv property-ja	Opis
Source	Naziv softvera (program ili komponenta sistema) koji je zabilježio događaj.
Event ID	Broj koji jedinstveno identifikuje tip događaja
Level	Nivo ozbiljnosti/važnosti događaja
User	Naziv korisnika u čije ime se događaj desio
Operational Code	Numerička vrijednost koja identifikuje aktivnost ili specifičan dio aktivnosti koji se odvijao u trenutku događaja
Log	Naziv fajla (dnevnika događaja) u kome je događaj zabilježen
Task category	Identifikuje kategoriju aktivnosti *event publisher*-a
Keywords	Skup kategorija ili oznaka koje se mogu koristiti za filtriranje ili traženje događaja
Computer	Naziv računara na kome se događaj desio
Date and Time	Datum i vrijeme nastanka događaja u dnevniku događaja

Kada je u pitanju nivo ozbiljnosti događaja pomenut u tabeli 1, razlikuje se pet različitih nivoa, a to su:

• **Information** – obavještenje, identifikuje promjenu u aplikaciji ili nekoj komponenti aplikacije, kao na primjer uspješan završetak neke operacije, pokretanje servisa, kreiranje resursa i slično.

• **Verbose** - detaljno obavještenje, koristi se kod događaja koji imaju detaljne informacije o sistemskoj ili aplikativnoj aktivnosti.

• **Warning** – upozorenje, ukazuje na neočekivanu aktivnost (potencijalni problem) koja bi mogla rezultirati ozbiljnijim problemom ako se ostavi bez nadzora.

• **Error** – greška, ukazuje na problem koji se dogodio i koji može uticati na funkcionalnosti van aplikacije ili komponente koje su prouzrokovale događaj.

• **Critical** – kritična greška, ukazuje na problem koji se desio i od koga se aplikacija ili komponenta na kojoj se isti dogodio ne mogu automatski oporaviti.

3. WINDOWS EVENT VIEWER

Windows Event Viewer jeste alat koji dolazi zajedno sa *Windows* operativnim sistemom i nudi različite funkcionalnosti kojima se obezbjeđuje praćenje i analiza logova. Ključne funkcionalnosti ovog alata su: pregled logova, filtriranje logova, pregled detalja događaja, kreiranje i primjena filtera, pregled logova sa udaljenih računara, i slično. Ovaj alat pruža jasan i pregledan prikaz logova na računaru i daje mogućnost praćenja performansi računara. Međutim, kao najveći nedostatak ovog alata smatra se kompleksnost prilikom kreiranja filtera, posebno za korisnike koji nemaju dovoljno znanja o radu sistema i logovima događaja. Kako bi se kreirao filter, alat zahtjeva od korisnika poznavanje pravila, strukture i formata *XPath* izraza, kao i ograničenja koje ima *Windows Event Log* vezano za iste.

4. MODEL SISTEMA ZA PRETRAGU LOGOVA

Ovaj sistem treba da omogući korisnicima bogat korisnički interfejs, tako da, brzo i lako mogu imati uvid u logove ne samo jednog, nego svih računara u lokalnoj mreži. Pored toga, ovaj sistem treba da omogući korisnicima jednostavnu i efikasnu pretragu i filtriranje logova, bez potrebe za poznavanjem strukture, formata i ograničenja *XPath query*-ja i sličnih tehničkih ograničenja.

4.1. Aktivnosti korisnika sistema

U sistemu za prikaz i pretragu logova postoji tačno jedan tip korisnika koji može da vrši sljedeće aktivnosti:
1. Korisnik koji pristupa sistemu može da pregleda i odabere jedan od dostupnih računara u mreži za koji želi da prikaže logove.
2. Korisnik ima tabelarni prikaz logova za odabrani računar u mreži.
3. Korisnik može da filtrira podatke prikazane u tabeli, po datumu i vremenu, po tipu događaja, po identifikatoru događaja, po ključnim riječima, po korisnicima, po izvoru događaja, te po nivou ozbiljnosti događaja.
4. Korisnik ima mogućnost da koristi opciju napredne pretrage, te na taj način kreira dodatne uslove za filtriranje logova, koji nisu dostupni kroz standardne opcije filtera.
5. U sklopu tabelarnog prikaza logova, korisnik može da odabere bilo koji log i prikaže detalje o istom.
6. Korisnik ima opciju da odštampa logove koji se trenutno prikazuju u tabeli.
7. Takođe, korisnik može da preuzme logove prikazane u tabeli u *CSV* formatu.

8. Dodatno, korisnik može da konfiguriše izgled tabele, može da mijenja kolone koje se prikazuju, te da mijenja kompaktnost, odnosno visinu redova.

4.2. Arhitektura sistema

Osnovni dijelovi sistema su jedna klijentska i jedna serverska aplikacija.

Serverska aplikacija se može posmatrati kao agent za prikupljanje logova sa ostalih računara u lokalnoj mreži. Po zahtjevu klijenta, server uspostavlja sesiju i prikuplja podatke sa odgovarajućeg računara u mreži.

Klijentska aplikacija predstavlja korisnički interfejs za uspostavljanje komunikacije sa serverom i obavljanje svih aktivnosti opisanih u prethodnom odjeljku (4.1).

5. IMPLEMENTACIJA

5.1. Serverska aplikacija

Serverska aplikacija napisana je u *C#* programskom jeziku, upotrebom *.NET* 6 radnog okruženja (engl. *framework*). Sastoji se od dva projekta, *Host* i *Logic*. *Host (5.1.1.)* predstavlja *REST* servis, zadužen za uspostavljanje komunikacije sa klijentom. Drugi projekat, *Logic*, zadužen je za realizaciju poslovne logike sistema , koja je u najvećoj mjeri implementirana u servisima, *NetworkDeviceService (5.1.2.)* i *EventLogService (5.1.3.)*.

5.1.1. Host

Host osluškuje zahtjeve klijenta na četiri *endpoint*-a, koji se koriste za:

1. dobavljanje logova (*url*: *„api/EventLogs“*)

2. filtriranje logova (*url*: *„api/EventLogs/filter“*)

3. dobavljanje opcija za filtriranje logova (*url*: „/api/EventLogs/filterOptions/{eventType}“)

4.dobavljanje dostupnih računara u lokalnoj mreži (*url*: *„api/NetworkDevice/availableHosts”*)

Zahtjevi pristigli na ove *endpoint*-e dalje se proslijeđuju odgovarajućim servisima koji predstavljaju dio *Logic* projekta (5.1.2.). Kada je u pitanju autentifikacija, aplikacija je podešena tako da autentifikuje korisnika koristeći *Negotiate SSP* (*Security Support Provider*).

5.1.2. Logic – NetworkDeviceService

NetworkDeviceService vrši analizu mreže i vraća informacije o dostupnim računarima u mreži. Analiza mreže vrši se uz pomoć alata *nmap* (*Network Mapper*). To je alat otvorenog koda, namijenjen za brzo skeniranje velikih mreža i reviziju sigurnosti (engl. *security auditing*). Iako se najčešće koristi za reviziju sigurnosti, ovaj alat je našao primjenu kod mnogih sistemskih i mrežnih administratora za svakodnevne, rutinske zadatke, kao što su popisivanje mrežnog inventara i nadgledanje *uptime*-a računara i servisa. Rezultat *nmap* skeniranja predstavlja popis skeniranih uređaja sa dodatnim informacijama, pri čemu količina i format dodatnih informacija zavise od proslijeđenih parametara. U slučaju ove aplikacije, osim dostupnih računara u mreži, nijedna druga informacija nije neophodna, pa se ni ne pozivaju komande za njihovo generisanje.

5.1.2. Logic - EventLogService

EventLogService namijenjen je rukovanju logovima. Ovaj servis uspostavlja komunikaciju sa ostalim računarima u mreži sa kojih je neophodno prikupiti podatke. Komunikacija između ovih računara odvija se upotrebom klasa, koje će biti opisane dalje u tekstu, a koje se nalaze unutar *System.Diagnostic.Eventing.Reader namespace*-a.

EventLogQuery predstavlja upit, koji definiše koji log podaci treba da budu vraćeni kao rezultat operacije čitanja logova. Koristeći ovu klasu moguće je postaviti različite filtere i uslove pretrage.
Prilikom kreiranja objekta ove klase, proslijeđuju se dva parametra. Prvi parametar može da bude ili putanja do fajla gdje se nalaze logovi nad kojima je potrebno izvršiti upit ili naziv sistemskog log fajla nad kojim je potrebno izvršiti upit. Pomoću drugog parametra, enumeracije *PathType,* definiše se da li je prvi parametar putanja do fajla (*FilePath*) ili naziv fajla (*LogName*). U slučaju ovog rješenja upiti se vrše isključivo and sistemskim log fajlovima. Prilikom kreiranja upita, moguće je proslijediti i treći parameter, koji predstavlja *query* string, na osnovu koga je moguće izvršiti filtriranje upita.

EventLogSession predstavlja klasu uz pomoć koje se uspostavlja sesija kako bi se izvršio pristup *Event Log* servisu na lokalnom računaru ili računaru u mreži. Prilikom kreiranja objekta ove klase, kao parametar se prosljeđuje naziv računara sa koga je potrebno pročitati logove. Po *default*-u se koristi *Windows* autentifikacija prilikom pristupanja udaljenom računaru, odnosno, šalju se korisničko ime i lozinka korisnika koji poziva ovu metodu.

EventLogRecord jeste klasa kojom se predstavljaju logovi događaja. Svaki objekat ove klase nosi podatke o vremenu kada je log zabilježen, nazivu računara na kojem je zapisan događaj, identifikacionom broju korisnika, identifikacionom broju događaja i još mnoge druge. Osnovne metode ove klase su *FormatDescription(),* koja vraća opis događaja u formatu prikazanom u sistemskom logu događaja i *ToXml()* koja vraća *xml* zapis događaja. Obzirom da ova klasa sadrži veliki broj *property*-ja, objekti ove klase se prije slanja klijentu mapiraju na objekte klase *EventLogDto* upotrebom *Automapper*-a.

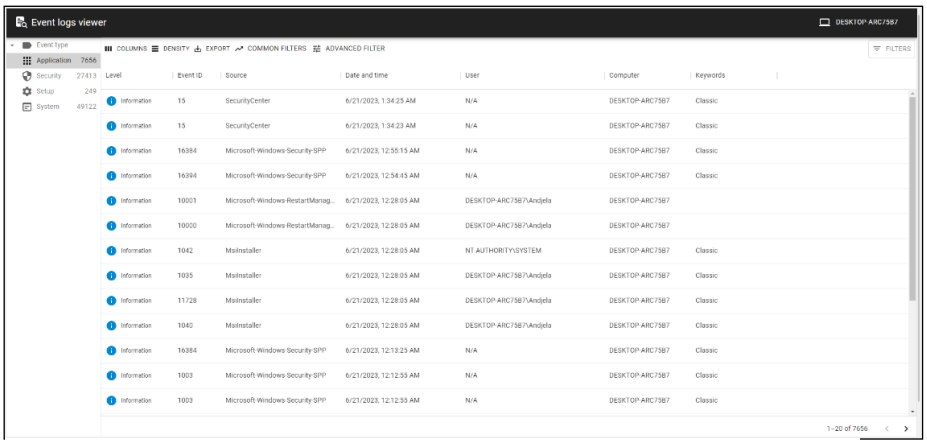
EventLogReader klasa omogućuje čitanje logova. Kao rezultat operacije čitanja vraća se objekat klase *EventRecord*. Prilikom kreiranja objekta klase *EventLogReader* konstruktoru se proslijeđuje kreirani upit (objekat klase *EventLogQuery*).
Pored ovih klasa, kako bi se implementirale neophodne funkcionalnosti, kreirane su dvije pomoćne klase i to *EventLogQueryBuilder* i *EventRecordExtensions*.

EventLogQueryBuilder klasa koristi se za izgradnju prethodno pomenutog *query* stringa (upita). Kreiranje upita vrši se na osnovu pravila za kreiranje *XML* upita uz pomoć *XPath* izraza.

EventRecordExtensions klasa sarži *extension* metodu *MatchesAll* za klasu *EventRecord*. Ova metoda koristi se da provjeri da li objekat klase zadovoljava kriterijume napredne pretrage. Potreba da se implementira ova metoda, proistekla je istovremeno kao rezultat zahtjeva za implementacijom napredne pretrage sa jedne strane, i ograničenja kada su u pitanju *XPath* upiti, sa druge strane.

5.2. Klijentska aplikacija

Klijentska aplikacija implementirana u *React*-u predstavlja alat uz pomoć koga korisnik može brzo, lako i jednostavno da pregleda i filtrira logove sa računara unutar mreže. Sve funkcionalnosti korisnik obavlja na jednoj stranici, izgrađenoj od više komponenti. Na slici 1. prikazana je stranica koja se prikazuje korisniku odmah nakon pokretanja aplikacije.



Slika 1 *Početna stranica aplikacije*

Centralni dio stranice predstavlja tabela sa log podacima. Po *default*-u, u tabeli se prikazuju *Application* logovi računara na kome je pokrenuta serverska aplikacija.

Na lijevoj strani se nalazi meni u okviru koga je moguće promijeniti tip događaja (*EventType*) čiji se logovi prikazuju.

U gornjem desnom uglu *header*-a, ispisan je naziv računara čiji logovi se prikazuju u tabeli, a klikom na isti otvara se padajući meni u okviru koga je moguće promijeniti *host* računar.

Takođe tabela ima i svoj *toolbar*, koji ima sljedeće opcije: promjena kolona koje se prikazuju u tabeli (*COLUMNS*), promjena kompaktnosti redova (*DENSITY*), *export* sadržaja tabele (*EXPORT*), brzo filtriranje (*COMMON FILTERS*), te opcija naprednog filtriranja (*ADVANCED FILTER)*

6. ZAKLJUČAK

U radu je opisana struktura i osnovna namjena *Windows* logova događaja. Implementirano je web rješenje koje omogućuje prikaz, pretragu i filtriranje logova, sa bilo kojeg računara u mreži. Sve prethodno navedeno omogućeno je korisniku bez potrebe za promjenom okruženja, instalacijom dodatnog softvera ili prelaskom sa jednog računara na drugi.

Na ovaj način se značajno ubrzava proces analize i otkrivanja problema, što su ključne aktivnosti u procesu povećanja efikasnosti i dostupnosti računarskih sistema. Neki od mogućih koraka u daljem razvoju i unaprjeđenju rješenja bi mogli biti:

• Omogućiti korisniku da grupiše uslove napredne pretrage uz pomoć logičkih operatora "*and*" i "*or*", te uvesti mogućnost negiranja nekog uslova uz pomoć operatora "*not*".

• Na korisničkom interfejsu dodati grafički prikaz statistika i izvještaja za logove.

• Omogućiti korisniku *import* i *export* filtera;

• Omogućiti dvosmjernu komunikaciju, odnosno, omogućiti *real-time* osvježavanje logova u tabeli ili bar uvesti sistem notifikacija.

7. LITERATURA

[1] Nicoleta Stanciu , " *Importance of event log management to ensure information system security*", Academy of Economic Studies, Bucharest, 2013.

[2] Zhuangbin Chen, Jinyang Liu Wenwei Gu, Yuxin Su, Jieming Zhu, Yongqiang Yang, Michael R. Lyu *" Deep Learning-based System Log Analysis for Anomaly Detection",* Januray 2022.

[3] GSEC Practical Assignment, *„Importance of Event Logging",* SANS institute, 2003

[4] https://www.blumira.com/what-are-event-logs-and-why-do-they-matter/ (pristupljeno u junu 2023.)

[5] Marcello Cinque, Raffaele Della Corte, Antonio Pecchia, *„Contextual filtering and prioritization of computer application logs for security situational awareness"*, 2020

[6] Risto Vaarandi, *„Tools and techniques for event log analysis",* Tallin Universitz of Technology, 2005

[7] Lei Zeng, Yang Xiao, Hui Chen, Bo Sun and Wenlin Han, *„Computer operating system logging and security issues: a survey",* 2016

[8] Vanja M. Korać, *„Digitalna forenzika u funkciji zaštite informacionog sistema baziranog na Linux i Windows platformama",* Univerzitet u Beogradu, 2014

[9] https://www.odseknis.akademijanis.edu.rs/wp-content/plugins/vtspredmeti/uploads/1585731723ARM%20Predavanje%2010%202016.pdf?script=lat (pristupljeno u junu 2023.)

Kratka biografija:

Anđela Čičković rođena je u Trebinju 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike – Informaciona bezbjednost odbranila je 2023.god.

kontakt: andjela.cickovic@gmail.com