Osnove informacione bezbednosti

Projektni zadatak 16

Tim 13

- PR33/2020 Tatjana Kosić
- PR82/2020 Nikola Lazarević
- PR79/2020 Jovan Vukosavljević
- PR85/2020 Dejan Dobrilović

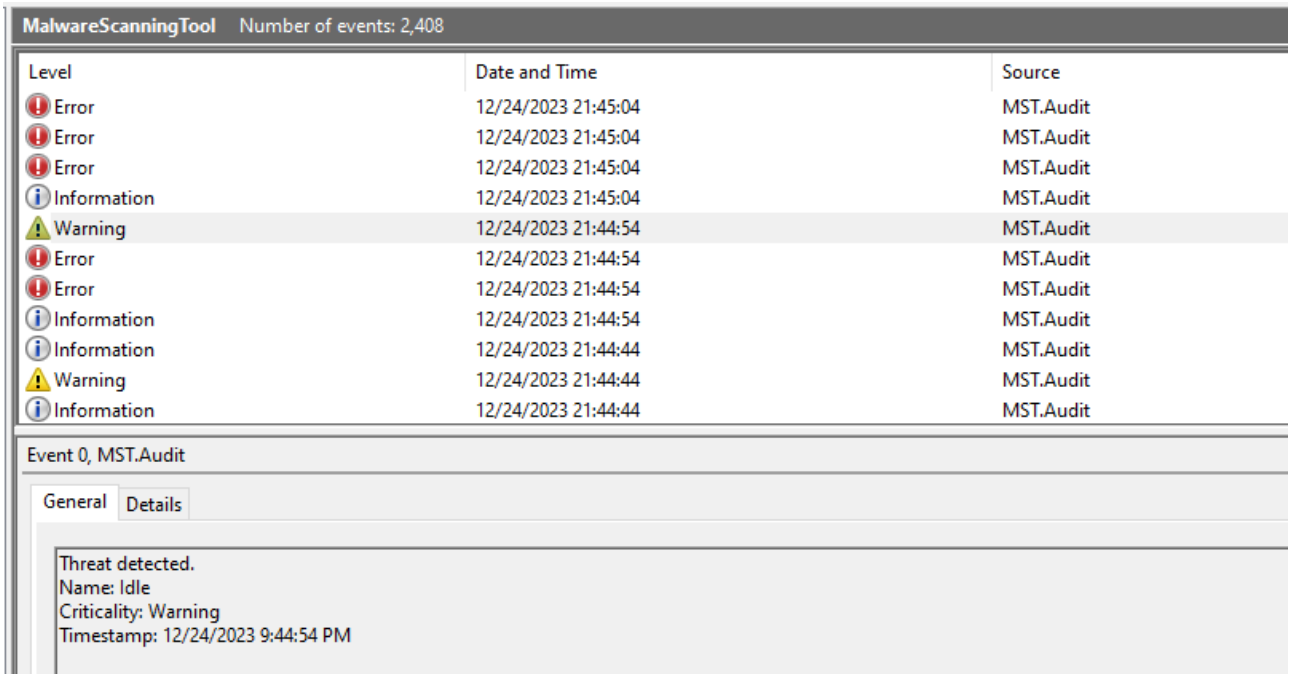
Opis projektnog zadatka

Projektni zadatak predstavlja aplikaciju koja ima zadatak da detektuje procese koji nisu dozvoljeni. Glavna komponenta, Malware Scanning Tool (MST), periodicno se aktivira i proverava listu dozvoljenih procesa. Dozvoljeni procesi su unapred definisani u okviru whitelist malware konfiguracije koja sadrži spisak dozvoljenih procesa i korisnika pod kojim dati procesi mogu biti pokrenuti.

Ukoliko MS detektuje takav neovlašćeni proces, šalje alarm Intrusion Detection System (IDS) komponenti. Alarm sadrži informacije o datumu i vremenu detekcije, naziv procesa i nivo kritičnosti. Nivo kritičnosti se definiše na sledeći način:

1. Information - ukoliko je prvi put detektovan neovlašćeni proces,
2. Warning - ukoliko je drugi put detektovan neovlašćeni proces,
3. Critical - ukoliko je treći put (ili veći broj puta) detektovan neovlašćeni proces.

MST, pored prosleđivanja alarma IDS komponenti, loguje i kritične događaje u svojoj log datoteci u Windows Event Log-u.



Slika 1. Windows Event Viewer

Tok podataka

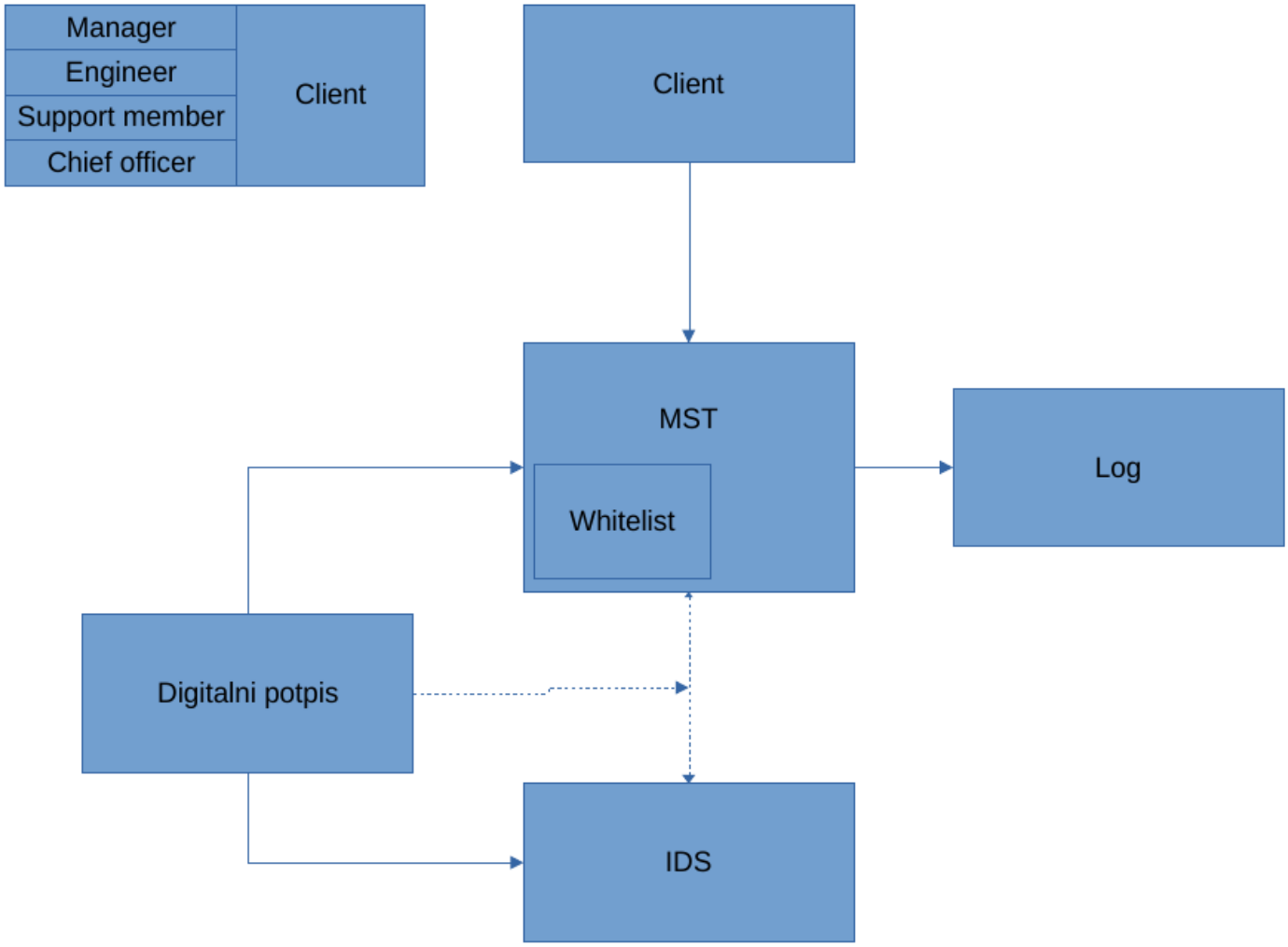
Komunikacija između MS i IDS komponente se uspostavlja preko sertifikata i za sve poruke koje se razmenjuju između njih implementirano je digitalno potpisivanje. Dodatno, na zahtev, IDS komponenta obezbeđuje proveru integriteta fajla gde loguje sve događaje prijavljene od strane MS komponente.

Mogućnost izmene whitelist malware konfiguracije implementirana je kroz poseban interfejs u okviru MS komponente tako da samo autorizovani korisnici (autentifikovani preko Windows autentifikacionog protokola, dok je autorizacija bazirana na RBAC modelu) imaju pravo da izmene konfiguraciju na sledeci način:

1. Manager - može da pročita konfiguraciju ali nema pravo da napravi izmenu,
2. Engineer - ima pravo da izmeni postojeće parametre u konfiguraciji, kao i da dodaje nove procese, ali nema pravo da briše postojeće procese,
3. Support Member - ima pravo da izmeni postojeće parametre u konfiguraciji, da dodaje nove procese kao i da briše postojeće,
4. Chief Officier - jedina grupa korisnika koja ima pravo da obriše konfiguracioni fajl.

Slika 2. Klijent UI sa listom mogućih opcija

Arhitektura projekta



Opis interfejsa i osnovne funkcionalnosti sistema

Program se sastoji iz 3 konzolne aplikacije: MalwareScanningTool, IntrusionDetectionSystem i Client. MalwareScanningTool i IntrusionDetectionSystem komuniciraju preko WCF-a (Windows Communication Foundation), dok za autentifikaciju koriste sertifikate. Da bi se obezbedio integritet i autentičnost poruke, kao i neporecivost od strane pošiljaoca da je upravo on poslao poruku, implementirano je digitalno potpisivanje koje se zasniva na kriptografiji javnog ključa zajedno sa heš algoritmom (SHA1).

Autorizacija klijenata bazirana je na RBAC modelu koji je zasnovan na permisijama koje se dodeljuju određenim ulogama. Pomoću klase CustomPrincipal koja implementira specifično ponašanje IPrincipal interfejsa i metode IsInRole, proveravaju se privilegije korisnika u aplikaciji.

Name	Value
Chief Officier	Read,Modify,Add,Remove,Create,Delete
Engineer	Read,Modify,Add
Menager	Read
Support Member	Read,Modify,Add,Remove

Slika 3. RolesConfigFile

MalwareScanningTool:

void Connect()	Uspostavljanje veze sa klijentom
void Disconnect()	Prekid veze sa klijentom
string CreateConfigurationFile()	Kreiranje konfiguracione datoteke
string DeleteConfigurationFile()	Brisanje konfiguracione datoteke
string AddEntry(ConfigurationEntry entry)	Dodavanje unosa u konfiguracionu datoteku

string ModifyEntry(ConfigurationEntry entry)	Modifikovanje postojećeg unosa konfiguracione datoteke
string DeleteEntry(int id)	Brisanje unosa na osnovu id-a
List<ConfigurationEntry> ReadConfigurationFile()	Čitanje konfiguracione datoteke

IntrusionDetectionSystem:

void SendMessage(string message, byte[] sign)	Slanje potpisane poruke
void UpdateIDS(Alarm alarm)	Ažuriranje alarma
string CheckFileIntegrity(string hash, byte[] sign);	Provera integriteta fajla

Data Model

ConfigurationEntry

int id	Identifikator procesa
string processName	Ime procesa
List<string> users	Lista korisnika koji imaju dozvolu da pokreću proces

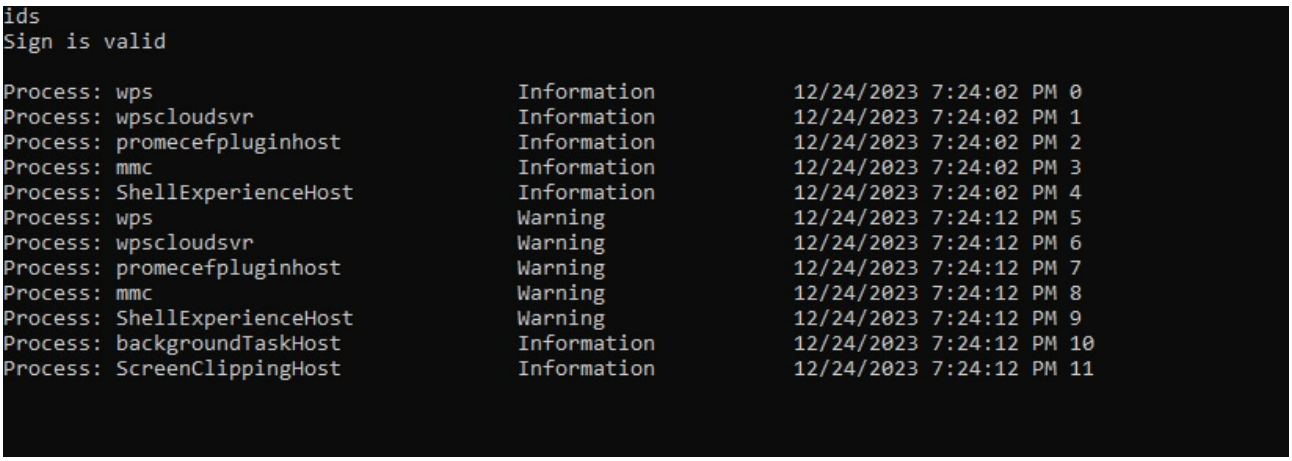
Alarm

int id	Identifikator alarma
string proccessName	Ime procesa koji je izazvao alarm
AlarmCriticality criticalityLevel	Nivo kritičnosti alarma
DateTime timestamp	Vreme kada je alarm izazvan

Threat

Process process	Proces koji je izazvao pretnju
int timesDetected	Broj prijavljivanja pretnje

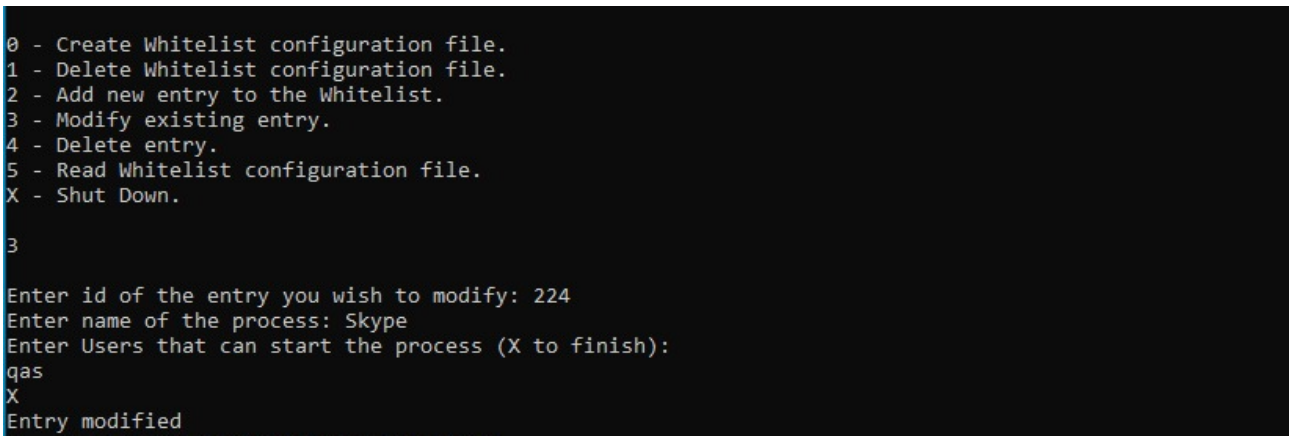
User Manual



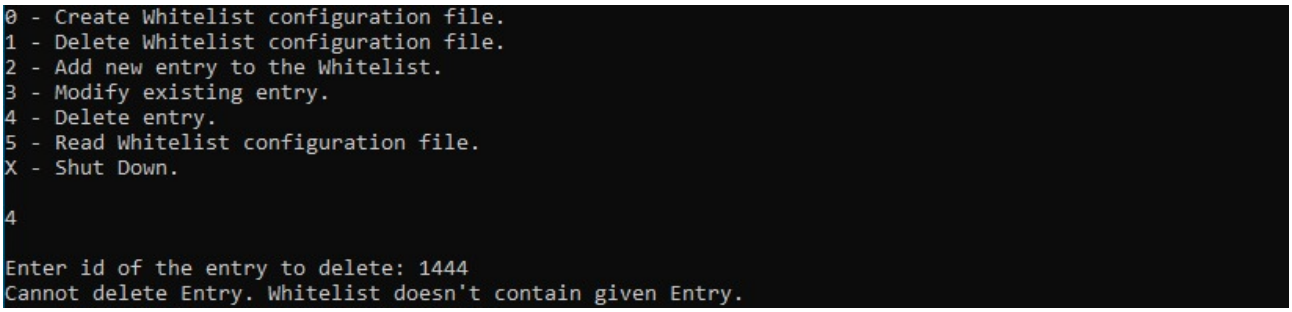
Izgled IDS prozora sa listom procesa

Izgled MST prozora sa informacijama

Klijent UI sa listom mogućih opcija



Primer uspešno izvršene operacija (modifikacija)



Primer neuspešno izvršene operacije (brisanje)

Korišćene tehnologije

Tehnologije koje su korišćene prilikom izrade projekta:

1. Okruženje - Visual Studio 2022
2. Programski jezik - C#, .NET Framework (Console Application, ClassLibrary)
3. WCF (Windows Communication Foundation) - frejmvork dizajniran da podrži razvoj distribuiranih sistema tamo gde servisi imaju udaljene potrošače (klijente).
4. Sertikifati - predstavlja digitalni identitet korisnika izdat od strane sertifikacionih tela (*certification authority, CA)* koje je odgovorno da verifikuje,izdaje i povlači sertifikate.
5. Digitalno potpisivanje - garantovanje integriteta, autentifikacije i neporecivosti.
6. RBAC (Role-Based Access Control) - model koji se koristi prilikom implementacije autorizacije u sistemima koji podrazumeva dodeljivanje uloga, autorizaciju uloga i ovlašćenje za dozvolu.

7. Auditing - odnosi se na proces praćenja, snimanja/logovanja, analize i izveštavanja o bezbednosnim događajima u sistemu.