KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Raimundas Stulpinas

Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės

Magistro darbas

Darbo vadovas
doc. dr. Algimantas Venčkauskas

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

KOMPIUTERIŲ KATEDRA

Raimundas Stulpinas

Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės

Magistro darbas

Recenzentas

prof. dr. R. Butleris
2012-05-23

Vadovas

doc. dr. A. Venčkauskas
2012-05-

Atliko

IFN-0/3 gr. stud.
Raimundas Stulpinas
2012-05-22

Kaunas, 2012

TURINYS

SUMMARY

Web and desktop applications threats are one of the most often e-crimes at present time. We fulfilled research about the most often risks and we have chosen to propose an embedded security solution for databases, because of it‘s importance in almost every web and desktop application, for secure saving and storing information.

Our goal is to create a flexible embedded system for securing web and desktop applications. These types of applications have high security threats when using databases, so we built an embedded intrusion detection system for disabling threats from *SQL* injections, stored *XSS* and *CSRF* attacks. That‘s a flexible embedded system for discovering and preventing malicious database attacks, with updatable rules base, events logging base and configurable attackers detection and registration base.

Keywords: embedded intrusion detection system, securing databases

LENTELIŲ SĄRAŠAS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Kompiuterinių sistemų atsiradimo ir vystymo tikslas – padėti žmogui įgyvendinti darbo ir poilsio veiksmus. Nebuvo manoma jog ateityje kompiuterinėmis sistemomis bei programiniais kodais bus manipuliuojama ir atakuojami vartotojai bei sistemos. Ta pati mąstysena išliko atsiradus internetui, tačiau šių dienų kompiuterinių sistemų ir jų vartotojų atakavimo tikimybė yra didžiulė, nes jau nekalbant apie kompiuterių vartotojų skaičių, interneto vartotojų yra milžiniškas kiekis. Natūralu jog didėjant vartotojų kiekiui, didėja ir galimų kenksmingų atakų skaičius. Interneto tinklas suteikia labai daug galimybių vartotojui: tiek tvarkyti banko sąskaitą, užsiregistruoti dantų klinikoje ar galų gale bendrauti socialiniuose tinkluose. Ši interneto pusė yra geroji, o blogoji yra ta, kad jame yra daugybė informacijos apie galimas atlikti atakas prieš taikomąsias sistemas. Be abejo, interneto tinklu ištisai keliauja daugybė žalingų veiksmų paleistų programišių, kurie siekdami naudos ar dėl tam tikrų interesų, bando pakenkti pasirinktoms aukoms ar taikomosioms sistemoms. Siekiant išvengti galimų kompiuterinių atakų, kuriamos įvairios apsaugos sistemos ir įrenginiai galintys atremti įvairias elektronines atakas.

Problema – taikomųjų sistemų saugos pažeidžiamumai.

Darbo tikslas – ištirti ir sukurti įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę.

Darbo uždaviniai:

- Išnagrinėti taikomųjų sistemų pažeidžiamumus, bei įterptinių priemonių naudojimą apsaugai nuo keliamų grėsmių;
- Pateikti įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės sprendimą, suteikiant taikomosioms sistemoms apsaugą nuo keliamų grėsmių;
- Realizuoti siūlomos įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinę realizaciją;
- Atlikti pasirinktus tyrimus, siekiant ištirti įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės galimybes.

Pirmoje darbo dalyje išnagrinėjus dabartinę padėtį taikomųjų programų grėsmių srityje pateikti šie rezultatai: išskyrėme pagrindines lokalių ir saityno taikomųjų sistemų populiariausias

atakas ir pažeidžiamumus, ištyrėme galimus apsisaugojimo būdus ir metodus tiek pačioms sistemoms, tiek sistemoms saugančioms duomenis – duomenų bazių valdymo sistemos.

Antroje darbo dalyje pateikėme reikalavimus įterptinei sistemai, atvaizdavome ir aprašėme projektuojamos įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės architektūrą bei veikimo principus. Veiklos procesus atvaizdavome pasitelkdami BPMN notacijos diagramas. Diagramos braižytos naudojantis *CASE* įrankiu *MagicDraw UML*, pritaikant *BPMN* ir *UML* notacijas.

Trečioje darbo dalyje pateiktas eksperimentinės realizacijos variantas, aprašytas konfigūravimas bei ypatumai. Eksperimentinė realizacija tinka tiek lokalioms, tiek saityno taikomosios sistemoms.

Ketvirtoje darbo dalyje pateikti eksperimentinės realizacijos tyrimo duomenys: atliktas duomenų bazių atakų atsparumo testas kuris parodė, jog naudojant eksperimentinės realizacijos sprendimą, *SQL* injekcijų atakos, išliekančios *XSS* ir *CSRF* atakos buvo aptiktos 100%, panaudojant tiek savo paruošus atakų šablonus, tiek specialų pažeidžiamumų analizavimo įrankį. Atliktas apkrovos ir streso testas parodė, jog vidutinis užklausų atsakymo laikas pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę padidėjo apie 5%.

Nauda ir privalumai:

- Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės sprendimas pasižymi tuo, kad yra taikomas duomenų bazių valdymo sistemų lygmenyje ir taip apsaugo realizuotų taikomųjų sistemų DB nuo kenksmingų DB atakų;
- Šis apsaugos principas ypač aktualus, kai kuriant taikomąją sistemą nebuvo atsižvelgta į reikiamą minimalų saugumo užtikrinimą, patikrinant įvedamus sistemos vartotojo duomenis;
- Konfigūruojamas kenksmingus veiksmus atlikusių vartotojų modulis;
- Nuolatos galima atnaujinti taisyklių bazės įrašus, siekiant apsaugoti nuo norimo srauto duomenų, kurie gali būti kenksmingi;
- Kadangi įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė yra duomenų bazės lygmenyje, tuomet ji pritaiko apsaugą visoms taikomosioms sistemoms besinaudojančioms ta pačia duomenų baze.

Šio darbo realizacijos rezultatai buvo praktiškai panaudoti siekiant užtikrinti saityno svetainių saugumą.

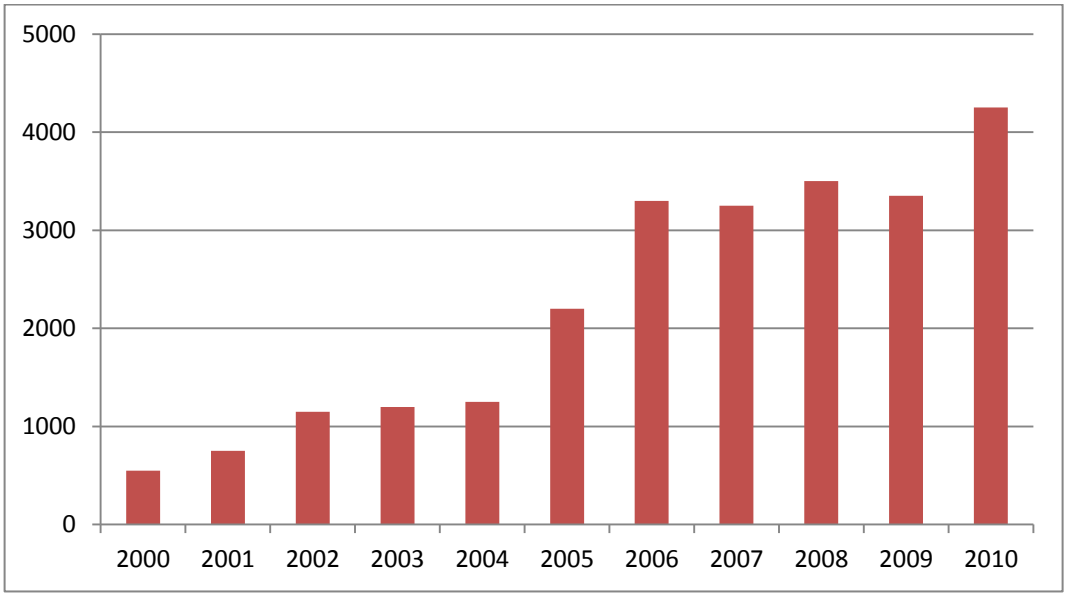
1. ĮSILAUŽIMŲ Į TAIKOMĄSIAS SISTEMAS ANALIZĖ

1.1 Analizės tikslas

Darbo tiklas – detaliai ištirti pagrindines saugumo grėsmes, paplitusias lokaliose ir saityno taikomosiose sistemose. Išnagrinėsime siūlomus apsaugai nuo grėsmių metodus bei sistemas.

1.2 Tyrimo sritis, objektas ir problema

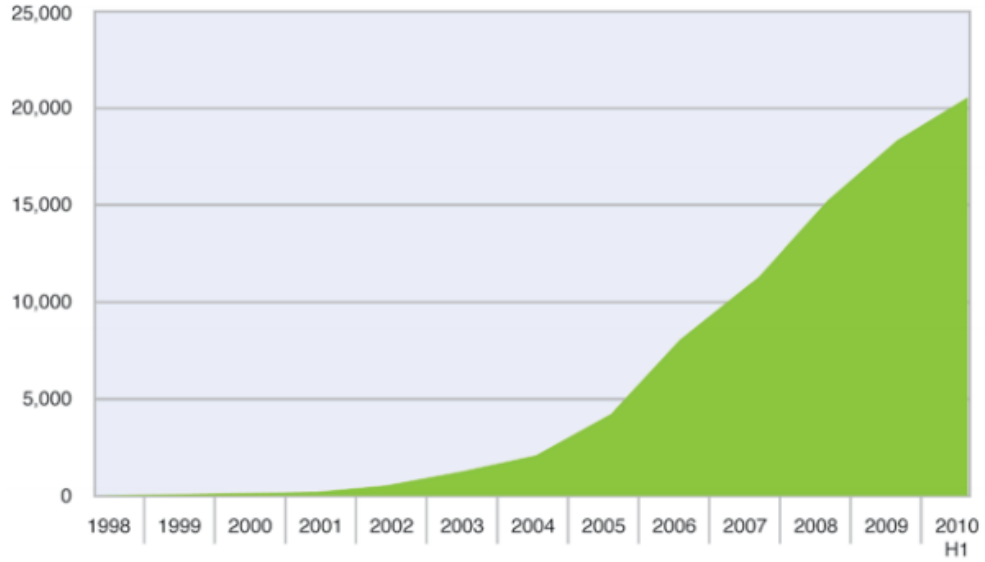
Šiuolaikinis gyvenimas nebeįmanomas be sukurtų įvairių tiek lokalių, tiek saityno sistemų. Plintant internetui, kompiuterinių atakų skaičius nuolatos kyla dėl įvairiausių priežasčių. Dabartinė statistika nurodo, jog šiuo metu pasaulyje yra 6,845,609,960 [5] interneto vartotojų, kas leidžia suprasti kiek yra potencialių atakuotojų tiek prieš saityno sistemas, tiek prieš taikomąsias sistemas. Vykdomų tinklinių kompiuterinių atakų skaičius kiekvienais metais vis didėja, o saugumui skiriamas dėmesys yra vis dar per mažas. Padėtis kiekvienais metais vis blogėja, anot IBM X-FORCE grėsmių analizių serviso, 2010 metų pirmąjį pusmetį buvo atrasti 4,396 nauji pažeidimai, tai yra 36% padidėjimas palyginus su tuo pačiu laikotarpiu 2009 metais [6] (*pav.* 1).



1 pav. Atskleistų atakų statistika 2000-2010 m. [6]

Gali atrodyti, kad dauguma atakų yra vykdoma prieš internetines sistemas, kadangi atakų tipai dažniausiai naudojami tie patys, yra pritaikomi daug paprasčiau negu lokaliose sistemose,

bei jų panaudojimo būdai yra plačiai aprašyti internetinėje terpėje. Tačiau, pagal X-FORCE atliktą statistiką, įvykdomų atakų procentas yra ganėtinai apylygis: prieš internetines sistemas yra atliekama kiek daugiau nei pusė visų atakų, t.y. 55%. 2 pav. pavaizduosime kokiais tempais didėjo saityno sistemų pažeidžiamumų kiekiai.



2 pav. Saityno sistemų pažeidžiamumų statistika 1998-2010 m. [6]

2010 metų pažeidžiamumų apžvalga [6]**:**

- didžiausią nerimą kelia atakuotojų gabumai, sėkmingai prasiskverbti pro gerai saugomus tinklus, apsaugotus pažangiausiomis technologijomis;
- atakuotojai vis dar randa kelius, kaip įterpti kenksmingą kodą per *JavaScript, Flash* ir *PDF* technologijas – auka atsidariusi *PDF* bylą, net nepastebi jog tuo metu suveikia byloje įterptas kenksmingas kodas. Nors ir dažnai siūloma atsisiųsti naujus pataisymus *Acrobat Reader* programai, siekiant pašalinti šį pažeidžiamumą, atakuotojai atranda vis kitus būdus ir kelius;
- naujų virsusų atsiradimas, kurie kuo toliau tuo labiau būna galingesni, t.y. gebantys atlikti didžiulę žalą;
- išlieka didelis šiukšlių (angl. *spam*) ir žvejojimo (angl. *phishing*) atakų populiarumas;
- žvejojimo atakų pagrindinis taikinys išlieka tas pats – finansinės institucijos.

Didžiausia problema yra tame, kad įvykdomų atakų skaičius vis didėja, kadangi sistemų kūrėjai ne visuomet atsižvelgia į galimas atakų grėsmes, dėl ko galimi nemaži nuostoliai, ypač sugebėjus prieiti prie naudojamos duomenų bazės, kurioje saugoma slapta, ne vieša ir jautri informacija.

1.3 Taikomųjų sistemų saugumo aspektai

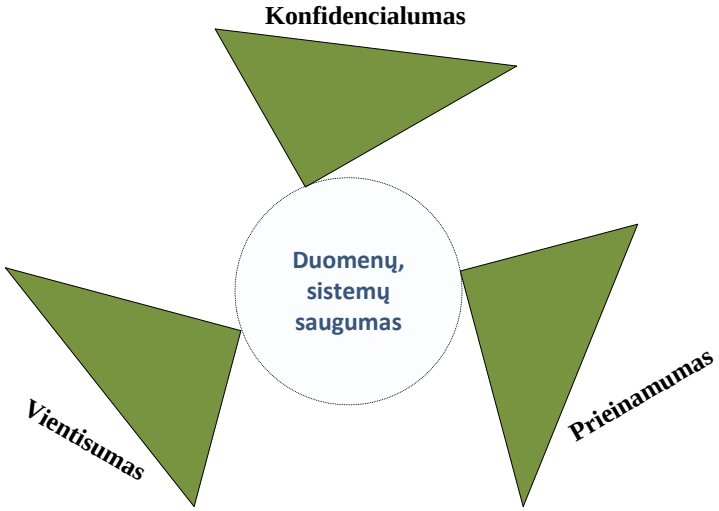
Norint užtikrinti saugumą kompiuterinėse sistemose, pradėjus vystytis informacijos saugai buvo pradėti naudoti saugumo aspektai, nors jie nėra detalūs, tačiau apibendrinantys pagrindinius saugos aspektus [3, 4].

Vientisumas. Vientisumo praradimas įvyksta tuomet, kuomet yra neautorizuotai modifikuojama informacija, t.y. neleistinas asmuo pakeičia tam tikrus duomenis. Tai turi būti leidžiama atlikti tiktai autorizuotiems vartotojams.

Prieinamumas. Turi būti užtikrinamas priėjimas prie informacijos be kliūčių vartotojui, tai ypač svarbu kritinėms sistemoms.

Konfidencialumas. Konfidencialumo praradimas įvyksta tuomet, kuomet nelegalių vartotojų yra perskaitoma informacija, kurie negali jos perskaityti. Tai ypač aktualu su labai jautria informacija.

Pateiktos trys pagrindinės kompiuterių sistemų saugumo savybes, kurias būtina užtikrinti siekiant sukurti saugumą (3 pav.). Tai yra įvykdoma įvairiais būdais, nuo fizinių iki elektroninių priemonių.



3 pav. Pagrindinės saugos savybės

3 paveiksle atvaizduoti pagrindiniai saugos informacijos aspektai, tai: vientisumas, konfidencialumas ir prieinamumas.

1.4 Taikomųjų sistemų išskyrimas

Taikomąsias sistemas galime išskirti į dvi grupes: saityno bei lokalias sistemas, kadangi jų pažeidžiamumai, galimos atakos, veikimo principai yra labai skirtingi. Apie kiekvieną grupę detaliau [1]:

Lokalios (angl. *desktop stand-alone*) sistemos. Norint jomis dirbti, nereikalingas internetas, talpinimas nutolusiame serveryje ar pasiekiamumas per kliento naršyklę, tačiau reikalingas diegimas kompiuteryje kuriame ji bus naudojama. Sistema gali būti pasiekiama vietiniame tinkle ir naudojama kelių vartotojų. Lokali taikomoji sistema naudojama nepriklausomai atskiruose kompiuteriuose tam, kad būtų galima redaguoti dokumentus, peržiūrėti video ar paklausyti audio bylas ir t.t. (*pvz.* Microsoft Office, VLC player, Adobe paketai).

Saityno (angl. *web*) sistemos. Norint naudotis šiomis sistemomis – būtinas interneto tinklas bei kliento naršyklė. Vartotojams šios sistemos suteikia daugiau funkcionalumo, patogumo bei efektyvumo, tačiau turi ir savo didžiausią privalumą ir trūkumą – gali būtų pasiekiamas didelio kiekio vartotojų.

Saugumas. Tarp šių sistemų vyrauja akivaizdus saugumo lygio skirtumas. Naudojant lokalias sistemas, yra sąlyginai maža tikimybė ją užkrėsti, kadangi turi būti sukurti specialūs programai tinkami virusai, arba panaudojamos atakavimo priemonės skirtos naudojamai operacinei sistemai – su saityno sistemomis yra kur kas paprasčiau. Saityno sistemos gali būti pažeidžiamos panaudojus visas tradicines populiariausias atakas, pateikiamas pagal OWASP [2] projekto surinktus duomenis. Saityno taikomosios sistemos pasiekiamos per naršyklę, todėl gali būti atakuojamos naudojantis tomis pačiomis kenksmingomis atakomis. Dauguma patyrusių sistemų kūrėjų tai žino, tačiau mažiau patyrę ne visuomet į tai atsižvelgia, kadangi labiausiai akcentuojamas funkcionalumas ir išvaizda, į saugumą dažniausiai atkreipiamas dėmesys jau po įvykusių atakų. Pvz.: internetinė bankininkyst, šioje srityje ypatingai turi būti užtikrintas saugumas, kadangi nuostoliai ir nepatogumai tiek klientams, tiek bankams būtų milžiniški.

1 lentelė. Taikomųjų sistemų privalumai

Saityno sistemos	Lokalios sistemos
Palaikymas	Saugumas
Prieiga	Prieinamumas
Funkcionalumas	Kaina

Dauguma lokalių sistemų palaipsniui yra keičiama į saityno sistemas, jeigu tai įmanoma, dėl šių pagrindinių savybių (1 lentelė) [1]:

- ✓ palaikymas – saityno sistema yra įdiegta vienoje vietoje, todėl ją prižiūrėti yra daug paprasčiau, nei atnaujinant sistemą kiekviename kompiuteryje;
- ✓ prieiga – prie saityno sistemos galima prieiti bet kuriuo metu iš bet kurios pasaulio vietos (būtina interneto prieiga), nepriklausomai kokiu kompiuteriu naudojamasi;
- ✓ funkcionalumas – saityno sąsaja vartotojui dažniausiai būna lanktesnė, išvaizdesnė, bei funkcionalesnė.

Lokalių sistemų pagrindinės savybės:

- ✓ saugumas – didesnis saugumas nei atviroje visuomenei prieigoje, prie kurios gali prieiti didžiulis skaičius potencialių atakuotojų su įvairiais atakų tipais;
- ✓ prieinamumas – tereikia turėti elektros kompiuterio maitinimui, nereikia rūpintis dėl kitų nebūtinų dalykų. Saityno sistemai būtina būti prieinamai, veikti greitai be trukdžių, tačiau šiuos punktus gali smarkiai paveikti įvairios pvz.: gamtos stichijos.
- ✓ Kaina – sistema nuperkama vieną kartą ir įrašoma į kompiuterį, kas dažniausiai yra daug pigiau, nei saityno sistemų kūrimas ir jų priežiūra.

Kaip matome, tiek taikomosios, tiek lokalios sistemos turi savo privalumus bei trūkumus, vartotojui suteikiamos visos galimybės rinktis kokią sistemą nori ir gali naudoti, tačiau sistemų kūrėjai turi kreipti didelį dėmesį kuriamų sistemų saugumo užtikrinimui.

1.5 Taikomųjų sistemų pažeidžiamumų analizė

Pažeidžiamumas – tai klaida programinėje įrangoje, kuri leidžia atakuotojui apeiti saugumo nustatymus. Už pažeidžiamumus yra atsakingi sistemų programuotojai, kadangi atakuotojai tiesiog pasinaudoja klaidomis, kurias dažniausiai lemia per mažas programuotojų saugumo žinių lygis. Terminas *informacijos atskleidimas* reiškia tai, jog atakuotojas gali pamatyti

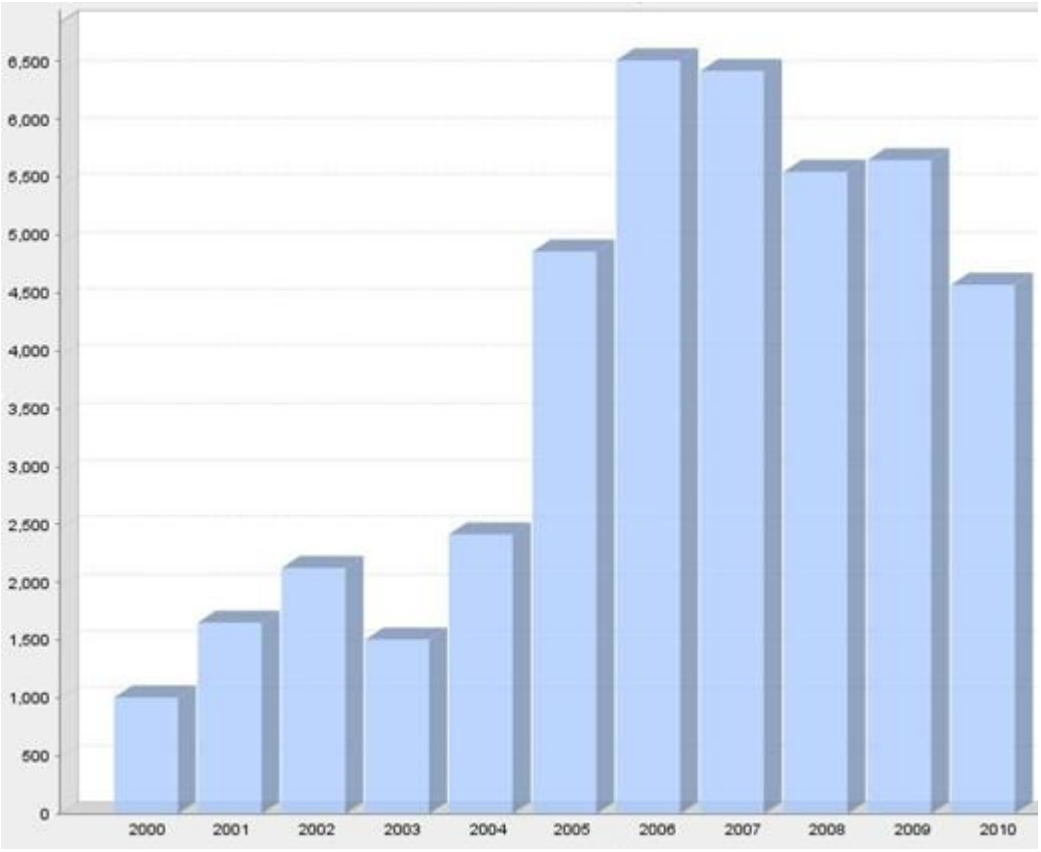
informaciją, kurios jis neturėtų matyti [7]. Paminėjome šiuos terminus, kadangi jie bus dažnai naudojami darbe.

Kūrimo, programavimo klaidos. Tai ko gero pagrindinė priežastis, dėl kurios yra įvykdomos vienokios ar kitokios atakos. Tai įvyksta dėl prastos saugos architektūros taikomojoje sistemoje, bei dėl programavimo klaidų. Šias problemas įtakoja tiek prastos kūrėjų žinios apie saugą, bei skubėjimas greitai atlikti užsibrėžtus projektus. Kadangi klaidos būna gana greitai aptinkamos, dažnai patyrus tam tikrus nuostolius, jas iškart siekiama pašalinti. Išvada tokia, jog skubėjimas programuojant palieka didelę tikimybę pažeidžiamumų atsiradimui programinėje įrangoje. Kūrimo proceso metu, būtina pilnai testuoti produktą, kas leistų sumažinti kompiuterinių atakų galimybes.

Logiškas mąstymas. Dažnai vartotojai yra apkvailinami pačiais primityviausiai būdais, pvz.: atsiunčiamas elektroninis laiškas iš nepažįstamo asmens be jokio logiško teksto ir su prisegtuku, kurį atsidarius auka būna apkrečiama atakuotojo nustatytais būdais, tas pats galioja ir parsisiunčiant nežinomas bylas iš interneto. Taigi, pirmiausia reikia blaiviai mąstyti prieš atliekant tam tikrus veiksmus prie kompiuterio, norint išlikti nenukentėjusiam.

1.5.1 Įsilaužimų į lokalias taikomąsias sistemas analizė

Su interneto atsiradimu, padidėjo ir skaičius individų, kurie naudoja internetinį tinklą kitų vartotojų, įstaigų atakavimui. Dauguma tinkslinių atakų, virusų ir kirminų yra galimi dėl pažeidžiamumų taikomosiose sistemose, kurios apdoroja nepatikimus duomenis atkeliaujančius tinklu. IT industrija kūrė ir kuria ugniasienes, antivirusines programas, programinių paketų atnaujinimus tam, kad sumažintų galimų atakų skaičių, labiausiai nuo išorinių. Tačiau tokios saugumo priemonės tik sumažina galimų atakų kiekį. Pagal CERT sudarytą statistiką, programinės įrangos pažeidžiamumai išlieka dideli (3 pav.).



4 pav. Programinės įrangos pažeidžiamumai 2000-2010m. laikotarpiu [8]

Programinės įrangos pažeidžiamumų pikas buvo 2006-2007m. (3 pav.). Tai įvyko dėl spartaus technologijų vystymosi, programų kūrimo gausos, vartotojų bendruomenės didėjimas. Viskas buvo daroma tam, kad būtų sukurtas veikiantis produktas, į saugumą nebuvo kreipiama per daug dėmesio, kas ir įtakojo tokius rezultatus. Pradėjus kelti saugumo problemą, po truputį pažeidžiamumai mažėjo, tačiau esant tokiam dideliam kiekiui programinių paketų, jų įvairių versijų, begalei vartotojų, vis pasitaiko įvairių sistemų, dažniausiai naudojamo programinio kodo, klaidų. Jos gali pasireikšti be specialaus vartotojo įsikišimo, taip pat vartotojas gali siekti surasti atitinkamus pažeidžiamumus ir juos išnaudoti savo tikslams, o jie gali būti labai žalingi. Dabartinius laikus galime vadinti saugumo era, kadangi metamos didelės pajėgos atakų aptikimų analizėms, statistikoms sudaryti, bei programiniais kodais ir įrenginiais siekiama užtverti galimus pažeidžiamumus.

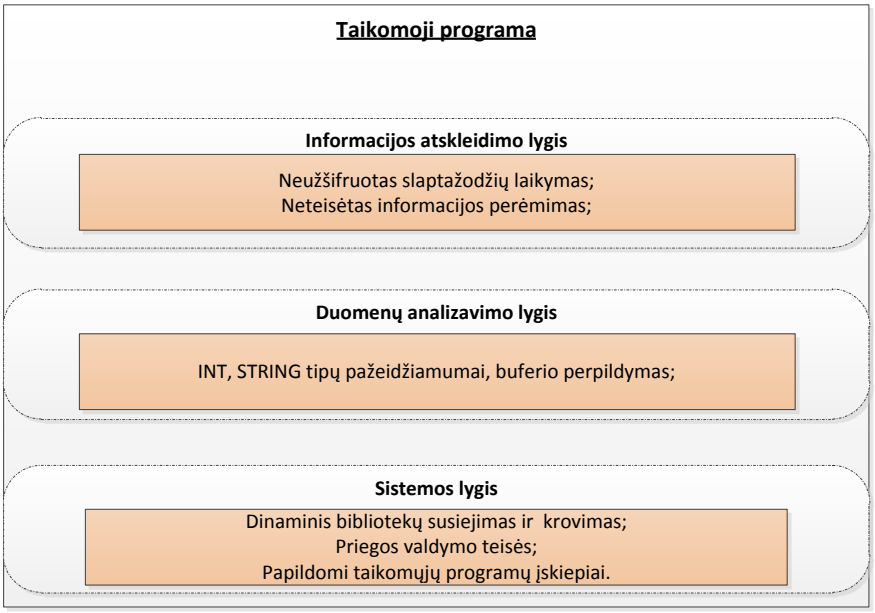
Aprašysime galimus atakų pobūdžius pagal jų tipus [7, 9]:

- ✓ **Sistemos lygio atakos**
  - ✓ **Prieigos teisės** – sistema susideda iš tam tikrų objektų, tokie kaip dažniausiai naudojamos bylos ir direktorijos. Šie objektai turi prieigos teises: skaitymo, rašymo ir vykdymo (angl. *read, write, execute*). Galimi atvejai: bloga konfigūracija, darbuotojo prieigos teisių išnaudojimas;
  - ✓ **Standartiniai ir silpni slaptažodžiai** – suteikiami ir naudojami silpni, standartiniai slaptažodžiai. Nurodoma, jog silpni slaptažodžiai yra viena iš didžiausių problemų informacinių technologijų pasaulyje. Naudojami standartiniai prisijungimo duomenys įdiegus sistemas, naudojami logiški žodžiai ar iš žodyno, kas leidžia piktavaliams lengvai patekti į privačias zonas;
  - ✓ **Skriptai** – leidžiama susiprogramuoti norimus priedus prie bazinės taikomosios sistemos, pvz. Microsoft Office. Įsilaužėlis įterpdamas savo skripto dalį, gali sukelti nemažas pasėkmes;
  - ✓ **Dinaminis susiejimas ir krovimas (angl. *dynamic linking and loading*)** – užkraunant taikomąją sistemą, tuo pačiu užkraunamos ir jai reikalingos įvairios bibliotekos. Galimas pažeidžiamumas yra bibliotekų užkrovimo vietoje (angl. *path*), kadangi atakuotojas gali pakeisti tikrąją bibliotekos bylą savuoju;
- ✓ **Duomenų analizavimas (angl. *data parsing*)**
  - ✓ **Buferio perpildymas (angl. *buffer overflow*)** – tai pažeidžiamumai, kurie sudaro didžiulę dalį išnaudojant programinių įrangų saugumo skyles. Veikimo principas yra toks, kad talpinimui sistemoje yra perduodama daugiau duomenų, negu buvo numatęs sistemos programuotojas. Kai tai įvyksta, galima pakeisti sekančius duomenis;
  - ✓ ***String* tipo formatavimo pažeidžiamumas** – kuomet vartotojo įvesta informacija yra atvaizduojama formatavimo funkcijomis C ir C++ kalbose, galima tikimybė jog bus įvesti formatavimo simboliai perskaityti atminčiai, „nulūžti“ programai ar įvykdyti kitas komandas. Kiekvienas *string* tipo atvaizdavimas, turėtų būti formatuojamas ‚*%s*‘ formatu;
  - ✓ ***Integer* tipo perpildymo pažeidžiamumas** – anksčiau kalbėjome apie buferio perpildymo pažeidžiamumą, kuomet atakuotojas gali užrašyti norimą informaciją atmintyje ir perimti programos valdymą. *Integer* tipo perpildymas įvyksta, kuomet

perduodamas mažesnis ar didesnis leistinas skaičius, tai vėlgi susiję su C ir C++ kalbomis.

- ✓ **Informacijos atskleidimas**
  - ✓ **Neužšifruotas slaptažodžių laikymas** – dauguma taikomųjų sistemų, kuriose reikalingas tam tikras prisijungimas, laiko užšifruotus slaptažodžius bylose, prieinamus tik administratoriams. Tačiau yra nemažai sistemų, kur ši jautri informacija laikoma atviru tekstu ir visiems prieinama;
  - ✓ **Laikinų bylų sukūrimas** – taikomosios sistemos atlikdamos tam tikrus veiksmus dažnai išsaugo tam tikrą informaciją laikinose bylose. Problema, kad bylos yra neištrinamos, prieinamos su paprastomis privilegijų teisėmis, leidžiant pamatyti tam tikrą informaciją;
  - ✓ **Paliekama informacija atmintyje** – dažna klaida yra ta, jog informacija yra paliekama atmintyje po taikomosios sistemos naudojimo. Tai leidžia atakuotojui peržiūrėti svarbią informaciją, tokią kaip slaptažodžiai ar šifravimo raktai.

Kaip matome, yra tikrai nemažai pažeidžiamumų, kuriuos galima panaudoti atakakuojant taikomąsias sistemas. Kiekvienas sistemų kūrėjas kurdamas sistemas, turėtų atsižvelgti į šiuos pažeidžiamumus, kadangi tai leistų užkirsti žinomus kelius galimoms atakoms.



5 pav. Lokalių sistemų lygių grėsmės

Pateikėme bendrą paveikslą (5 pav.), atvaizduojantį taikomųjų programų atitinkamų lygių atakas.

Taip pat naudojamos papildomos įvairios priemonės prieš norimas atakuoti taikomąsias sistemas, taip pat norint išgauti reikiamą informaciją, ar kitaip pakenkti aukai [10, 11]:

- ✓ Virusas (angl. *virus*) – programa ar programinio kodo dalis, kuri atlieka kenksmingus veiksmus, prisijungdama prie kitos programos;
- ✓ Kirminas (angl. *worm*) – programa ar programinio kodo dalis, kuri atlieka kenksmingus veiksmus galinti plisti savaime, nereikiant kitų programų;
- ✓ Trojos arklys (angl. *trojan horse*) – programa kuri slepiasi už kitų programų ir atlieka kenksmingus veiksmus. Vykdo veiksmus kai yra aktyvuojama;
- ✓ Galinės durys (angl. *back door*) – dažniausiai slaptažodis, žinomas tik atakuotojui, kuris leidžia pasiekti atakuojamą kompiuterį apeinant saugumo procedūras;
- ✓ Loginė bomba (angl. *logic bomb*) – kompiuterinio kodo dalis, kuri įterpiama į kompiuterį ir yra aktyvuojama destrukciniams veiksmams nurodytu laiku;
- ✓ Brutalios jėgos ataka (angl. *brute force attack*) – didžiulių kompiuterinių resursų reikalaujančios atakos, išbandančios visas įmanomas slaptažodžių atspėjimo kombinacijas;
- ✓ Paslaugos atjungimo ataka (angl. *denial of service*) – atakuotojas siunčia aukai tinklu didelius kiekius užklausų ir atakuotojo kompiuteriui neatlaikius užklausų kiekio, dažniausiai tampa neprieinamas;
- ✓ Paskirstyta paslaugos atjungimo ataka (angl. *distributed denial of service*) – aukų kompiuteriai užkrečiami kenksmingu kodu, kuriuo suformuojamas kompiuterinių „zombių“ tinklas, o užkrėstieji kompiuteriai vadinami botais. Atakuotojas naudoja šiuos botus, iš kurių būtų siunčiamas didžiulis kiekis užklausų į vienintelį aukos kompiuterį, kad jo teikiama paslauga būtų atjungta.
- ✓ Žvejojimo ataka (angl. *phishing attack*) – žvejojimo atakos naudoja apgaulę, siekiant sužinoti slaptažodžius, kitus svarbius duomenis imituojant tikrus el. laiškus ar kompiuterinės sistemos žinutes.

Kaip matome, yra daugybė būdų tiek apkvailinti vartotoją, tiek sistemas. Norint sumažinti šias galimybes, verta naudoti antivirusines sistemas, turėti įjungtą ugniasienę, nenaudoti įtartinų programų bei vengti įtartinų elektroninių laiškų.

Įsilaužimai į taikomąsias sistemas yra visiškai skirtingi negu į saityno sistemas. Apie jas kitame skyriuje.

1.5.2 Įsilaužimų į saityno taikomąsias sistemas analizė

Saityno taikomųjų sistemų puolimai pasižymi kitomis atakomis, tačiau jos yra labiau žinomos ir dažniau taikomos, bet ne visos sistemos pasiruošusios jas atremti. 55% visų atakų yra įvykdoma prieš internetines sistemas [6].

Aptarsime pagrindinius saityno atakų tipus, pasinaudodami OWASP projekto duomenimis, kuri užsiima internetinių sistemų kūrėjų informavimu apie labiausiai kritines internetinių sistemų saugumo skyles. Šis projektas išskiria 10 pačių populiariausių saityno taikomųjų sistemų atakų tipų 2010 metais, tačiau mes apžvelgsime 5 populiariausias iš jų. Kad kiekviena internetinė sistema būtų saugi, kūrėjai turėtų peržiūrėti pranešimus apie populiariausias atakas, kas leistų jiems kurti saugias saityno taikomąsias sistemas.

Apie kiekvieną atakos pobūdį plačiau [2]:

1. **Injekcijos** – tai galimos įterptinės atakos, kuomet nėra visai ar nepilnai tikrinami vartotojo įvesti duomenys, kas gali privesti prie kenksmingų veiksmų.

   Populiariausia injekcija – ***SQL***. Ją panaudojus, galima išgauti norimą informaciją iš duomenų bazių, ją peržiūrėti, pakeisti, ištrinti bei įrašyti [13].

   SQL injekcijos atakos pavyzdys – turime paieškos formą, kurioje įvedus norimo surasti vartotojo vardas, vykdoma paieškos užklausa:

   ```
   <form method="post" action="ieskotivartotoja.php">
   <input type="text" name="name">
   <input type="submit" value="Search" name="search">
   </form>
   ```

   Užklausa atrodo taip:

   ```
   string sql = "SELECT * FROM users WHERE username = '" + $username + "'";
   ```

   Atakuotojui įvedus į paieškos laukelį *admin' OR 1=1* --, būtų gaunama užklausa:

   ```
   SELECT * FROM users WHERE username = 'admin' OR 1=1 --';
   ```

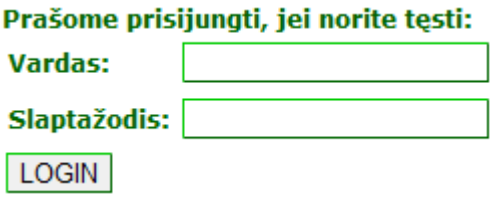
   Tokiu atveju atakuotojui būtų grąžinami visi lentelės įrašai, kadangi sąlyga *1=1* yra visada teisinga, o žymės -- nurodo užklausos pabaigą.

   Sprendimas. Norint apsisaugoti nuo *SQL injekcijų* atakų, vartotojo įvesti duomenys turėtų būti iš karto patikrinti ar nėra įvesta neleistinų simbolių ir tik tuomet atliekama užklausa.

2. **Cross-site-scripting (XSS) atakos** – tai gali įvykti, kuomet internetinė taikomoji sistema paima nepatikrintą informaciją ir siunčia naršyklei. XSS atakos leidžia piktavaliui įvykdyti įkeltus programinius skriptus aukos naršyklėje, taip perimadamas sesijas, slapukus, nukreipti vartotoją į kenksmingą sistemą [6, 12, 13].

   XSS atakos pavyzdys. Puslapyje esantis paprastas įvedimo laukelis su paieškos mygtuku. Jeigu saityno sistema nėra apsaugota nuo *XSS* atakų, tuomet įterpę žemiau esantį kodą į laukelį, gauname atitinkamą rezultatą 6 paveiksle.

```
<br><br>Prašome prisijungti, jei norite tęsti:<form action="destination.asp"><table><tr><td>Vardas:</td><td><input type=text length=20 name=login></td></tr><tr><td>Slaptažodis:</td><td><input type=text length=20 name=password></td></tr></table><input type=submit value=LOGIN></form>
```



6 pav. XSS atakos pavyzdys

Pasinaudojant tokiomis sistemos skylėmis, atakuotojas įterpęs tokį kodą, gali sužinoti aukų prisijungimo duomenis, taip pat piktavalis įterpęs šį kodą, gali perduoti nuorodą aukai, pvz:

```
http://testasp.vulnweb.com/Search.asp?tfSearch=%3Cbr%3E%3Cbr%3EPra%9Aome+prisijungti%2C+jei+norite+t%26%23281%3Bsti%3A%3Cform+action%3D%22destination.asp%22%3E%3Ctable%3E%3Ctr%3E%3Ctd%3EVardas%3A%3C%2Ftd%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dlogin%3E%3C%2Ftd%3E%3C%2Ftr%3E%3Ctr%3E%3Ctd%3ESlapta%9Eodis%3A%3C%2Ftd%3E%3Ctd%3E%3Cinput+type%3Dtext+length%3D20+name%3Dpassword%3E%3C%2Ftd%3E%3C%2Ftr%3E%3C%2Ftable%3E%3Cinput+type%3Dsubmit+value%3DLOGIN%3E%3C%2Fform%3E
```

Išskiriami du *XSS* atakų tipai [13]:

- **Neišliekantis** (angl. *reflected*) – kuomet kenksmingas kodas įvykdomas iškart su perduodama nuoroda ar atvaizduojama forma;
- **Išliekantis** (angl. *stored*) – kenksmingas kodas dažniausiai išsaugojamas duomenų bazėje, kuomet vartotojas atveria atitinkamą puslapį, prieš jį yra įvykdoma ataka.

Sprendimas. Norint išvengti *XSS* atakų, reikia vėlgi validuoti visą vartotojo įvestą informaciją.

3. **Autentifikacijos ir sesijų perėmimo atakos (ASPA)**. Tokių duomenų perdavimui turėtų būti naudojamas saugus perdavimo kanalas kaip *https*, slaptažodžiai turi būti stiprūs, sesijos ID turi būti ilgas ir sudėtingas, sesijos metu turėtų būti dažnai keičiamas, kad atakuotojui būtų sunkiau atspėti [2].
4. **Nesaugios tiesioginės nuorodos į objektus (angl. *Insecure direct object references, IDOR*).** Tarkime, kad atakuotojas yra autorizuotas sistemos vartotojas, kuris gali prieiti prie atitinkamų bylų įterpdamas kitos direktorijos ar bylos pavadinimus, kas leistų jam peržiūrėti norimą bylą. Taip įvyksta netikrinant vartotojo prieigos teisių [16].

   Sprendimas. Norint išvengti šios grėsmės, reikia vartotojams priskirti atitinkamas teises ir kiekvieną kartą tikrinti, ar vartotojas bando pasiekti jam leidžiamą matyti bylą.
5. **Cross-Site Request Forgery (CSRF) atakos.** Tai tokia ataka, kai aukai užkraunamas puslapis, kuriame yra žalinga užklausa. Toji užklausa leidžia identifikuotis puslapyje su aukos inicialais (per slapukus, per veikiančią sesiją) ir įvykdyti iš anksto nustatytis vaiksmus, pvz.: atsijungti, pakeisti duomenis, nupirkti prekę elektroninėje parduotuvėje, pervesti pinigus iš banko sąskaitos ir pan. Panagrinėkime atakos principo pavyzdį [17]:
   - atakuotojas pasižiūri kokia užklausa siunčiama jam įvykdžius bankinį pavedimą:
     ```
     GET http://bankas.lt/transfer.do?acct=BOB&amount=100 HTTP/1.1
     ```
   - atakuotojas aukai sukuria savo norimą užklausą:
     ```
     http://bankas.lt/transfer.do?acct=ATAKUOTOJAS&amount=10000
     ```
   - atakuotojas įdeda užklausą į galimas nuorodas:
     ```
     <a href="http://bankas.lt/transfer.do?acct=ATAKUOTOJAS&amount=10000">
     PERŽIŪRĖK KAS ČIA!</a>
     <img src="http://bankas.lt/transfer.do?acct=ATAKUOTOJAS&amount=10000"
     width="1" height="1" border="0">
     ```
   - aukai nuspaudus vieną iš nuorodų, automatiškai bus įvykdyta užklausa ir pervesti pinigai į atakuotojo sąskaitą (auka turi būti identifikuota sistemoje).

Sprendimas. Užkraunant puslapį turėtų būti sukuriama slapta žymė (angl. *token*), kuri tikrinama kiekvieno paspaudimo metu ir kurios nepasiektų atakuotojas, kas neleistų piktadariui įvykdyti kenksmingos užklausos, o vartotojas galėtų jaustis saugus.

Aptarėme penkias šiuo metu populiariausias atakas saityno taikomosiose sistemose ir norint apibendrinti jas, sudarėme lentelę nr. 2.

2 lentelė. Saityno atakų rizikų apibendrinimas

	Kriterijai			
Atakos tipas	**Panaudojimas**	**Paplitimas**	**Aptikimas**	**Poveikis**
Injekcijos	*Lengvas*	*Paplitęs*	*Vidutinis*	*Didelis*
XSS	*Vidutinis*	*Labai paplitęs*	*Lengvas*	*Vidutiniškas*
ASPA	*Vidutinis*	*Paplitęs*	*Vidutinis*	*Didelis*
IDOR	*Lengvas*	*Paplitęs*	*Lengvas*	*Vidutiniškas*
CSRF	*Vidutinis*	*Paplitęs*	*Lengvas*	*Vidutiniškas*

Lentelėje nr. 2 pateikiame penkių populiariausių saityno taikomųjų sistemų atakų tipų apibendrinimą pagal nurodytus kriterijus: panaudojimo, paplitimo, aptikomo bei poveikio.

1.5.3 Duomenų bazių apsaugos metodai

Panagrinėsime kelis apsaugos metodus, saugančius duomenų bazes nuo galimų populiariausių grėsmių. Duomenų bazės yra viena iš svarbiausių taikomųjų sistemų dalių, kuriose laikoma tiek svarbi, slapta, tiek viešai prieinama informacija. Atakuotojams dažnai pavyksta pasiekti slaptą ar tik autorizuotiems vartotojams skirtą informaciją duomenų bazėse, kadangi būna netinkamai pasirūpinta jų apsauga. Ypač svarbu užtikrinti duomenų bazių saugą saityno taikomosiose sistemose, kur tikimybė, kad bus pabandyta išgauti slaptą informaciją iš duomenų bazių, yra labai didelė.

DXJL.

Autorių D. Liwu, X. Ruzhi, J.Lizheng, L. Guangjuan [20] siūlomas duomenų bazių apsaugojimo būdas nuo *SQL* atakų, apima siunčiamų paketų tinklu filtravimą. Metodas analizuoja tinklu ir duomenų bazių protokolais keliaujančius paketus, išrenka ir analizuoja *SQL*

sąlygas. Metodo privalumas, toks, jog nereikia modifikuoti esamos internetinės taikomosios sistemos, o trūkumas jog analizavimas trunka ganėtinai lėtai, kadangi filtruojamas kiekvienas paketas.

XGD.

Autorių X. Ruzhi, G. Jian, D. Liwu [21] pasiūlytas metodas *SQL* atakų aptikimui. Siūlomo metodo architektūra susideda iš šių modulių:

- ✓ autentifikacijos – tikrinamos bandančio susijungti vartotojo prieigos teisės;
- ✓ įgaliotasis (angl. *proxy*) modulis – naudoja virtualios duomenų bazės technologiją, užslėpdamas tikrąjį duomenų bazės *IP* adresą ir prievadą;
- ✓ apsaugos nuo atakų modulis – aptinka ir nutraukia kenksmingas *SQL* atakas;
- ✓ jungties stebėjimo modulis – stebi visus susijungimus į duomenų bazę ir visus srautus realiu laiku;
- ✓ audito modulis – kaupia visą įvykių žurnalą, kurį esant reikalui galima peržiūrėti;
- ✓ paslaugų konfigūravimo modulis – paslaugų konfigūravimo galimybė.

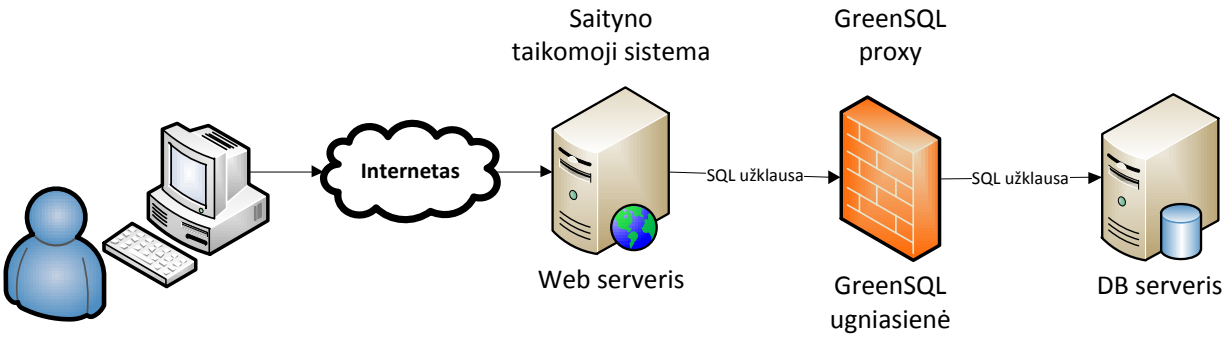
Šio metodo architektūra daug didesnė nei prieš tai aprašyto. Naudojami net šeši moduliai, atliekantys svarbias funkcijas, bei suteikiantys svarbią informaciją. Pasiūlytas metodas ganėtinai sudėtingas, reikalaujantis didesnių resursų.

RP.

Taip pat tinkamas Roko Paškevičiaus [22] pasiūlytas metodas, žiniatinklio užkarda – kuri apsaugo duomenų bazes nuo *SQL* injekcijų atakų. Tačiau pagrindinis šio pasiūlyto metodo trūkumas – priemonė turi būti įdiegta į kiekvieną saityno taikomąją sistemą.

GSQL.

GreenSQL duomenų bazės saugos sprendimas [27] – atviro kodo duomenų bazės ugniasienė, skirta apsaugoti duomenų bazę nuo *SQL* injekcijų atakų. GreenSQL veikia kaip įgaliotasis serveris (angl. *Proxy*) *SQL* komandoms (*žr.* 7 pav.) – palaiko MySQL ir PostgreSQL duomenų bazes. Šio sprendimo logika paremta įvertinant *SQL* komandų rizikos matricą, taip pat blokuojant administracines komandas (*DROP,CREATE ir t.t.*).



7 pav. GreenSQL architektūra

Kaip matoma iš 7 paveikslo, GreenSQL kviečia tikrąjį duomenų bazės serverį *SQL* komandų vykdymui, o saityno taikomoji sistema jungiasi prie GreenSQL serverio lyg jis būtų tikrasis duomenų bazės serveris.

Duomenų bazių apsauga yra ypatingai svarbi, kadangi tiek lokalios, tiek saitynos taikomosios sistemos saugo duomenis ne kur kitur, o duomenų bazėse. Todėl tam turi būti skirtas ypatingas dėmesys, siekiant apsaugoti tiek slaptus, tiek viešus duomenis – leižiant vykdyti atitinkamus veiksmus su duomenimis tam priklausantiems asmenims. Apsaugojimo priemonių apibendrinimas pateiktas 3 lentelėje.

3 lentelė. Duomenų bazių apsaugos priemonių apibendrinimas

Priemonės	Savybės
DXJL	Tinklu siunčiamų paketų filtravimas.
XGD	Saugumo vartai tarp duomenų bazės ir saityno serverio (angl. *web server*), užtikrinantys duomenų saugumą.
RP	Saityno užkarda skirta konkrečioms saityno sistemoms.
GSQL	Įgaliotasis serveris, filtruojantis keliaujantį srautą į duomenų bazę.

1.6 Įsilaužimų aptikimo ir prevencijos sistemos

Įsilaužimų aptikimas yra veiksmas, kuomet aptinkama žalinga informacija ar kodas tinkle, įrenginyje ar sistemoje. Tai gali būti įdiegta programinė įranga arba fiziniai įrenginiai stebintys keliaujantį srautą ir siekiantys perspėti vartotoją apie galimas atakas, tuo pačiu išsaugant šią informaciją bylose. Dažniausiai šios įterptinės sistemos yra taikomos tinklu keliaujančiam srautui nagrinėti ir aptikti galimą žalingą informaciją.

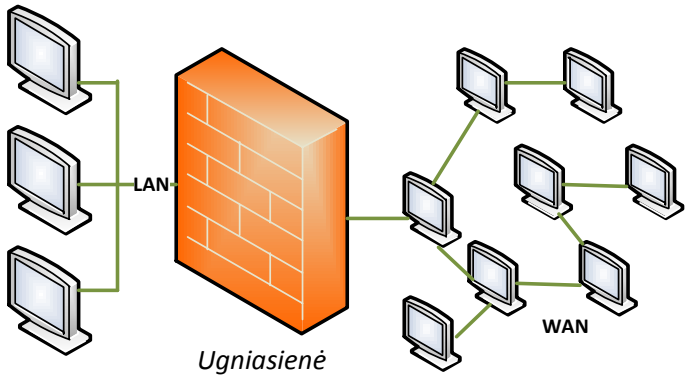
Galimi įsilaužimų aptikimo sistemų tipai [14, 15]:

- ✓ Tinklo įsilaužimų aptikimo sistemos – stebi tinklu keliaujančią informaciją visuose septyniuose OSI modelio sluoksniuose;
- ✓ Bevielio tinklo įsilaužimų aptikimo sistemos – analizuoja tinklo srautą, tuo pačiu sekant bandymus prisijungti prie prieigos taško;
- ✓ Tinklo veikimo anomalijų aptikimo sistemos – stebima ar yra nuokrypių nuo darbo profilių: stebimi vartotojo veiksmai, sekami atsiradę nuokrypiai nuo normalaus profilio;
- ✓ Kompiuterinės įsilaužimų aptikimo sistemos – įdiegiamos kiekviename kompiuteryje, sekamas ir analizuojamas taikomųjų sistemų veikimas, bylų pakeitimai, prisijungimai.

Mes patyrinėsime galimas lokalių ir saityno taikomųjų sistemų grėsmes ir pasiūlysime metodą joms aptikti. Mūsų pasiūlytas metodas nebus fizinis aparatas ar atskira programinė įranga, o jis bus papildoma, įterptinė programinio kodo dalis, susieta su jau esančia taikomąja sistema.

1.7 Apsaugos priemonės nuo grėsmių taikomosioms sistemoms

Užkardos (angl. *firewall*). Visų pirma reikėtų paminėti užkardų naudojimą, kadangi jos gali smarkiai sumažinti kompiuterių, taip pat ir taikomųjų sistemų riziką būti atakuotoms. Tai jau tinklinio lygmens apsauga, kuri padeda apsisaugoti nuo įsilaužėlių, virusų, kirminų, trojos arklių. Be užkardos kompiuteris tampa atviromis durimis visiems galimiems įsilaužimams [18].



8 pav. Užkardos pavyzdys [18]

Pagal nustatytas užkardos taisykles yra tikrinami ir blokuojami tie paketai, kurie nurodyti aprašytose taisyklėse. Kaip matome 8 paveiksle, pavaizduotas vietinis tinklas ir išėjimas į internetą. Ugniesienė pastatyta prieš vidinį tinklą, kad būtų galima kontroliuoti kiekviena įeinantį ir išeinantį paketą, kad būtų sumažinta tikimybė būti atakuotam ir apkrėstam kenksmingais veiksmais.

Antivirusinės. Antivirusinės programos skanuoja operatyviąją atmintį, kompiuterio kietąjį diską, prižiūri paleidžiamas programas ir elektroninio pašto laiškus, siekiant surasti kompiuterinius virusus. Galimi skanavimo tipai [19]:

- nurodomas skanavimas – antivirusinė pradeda skanuoti kai vartotojas jai tai nurodo;
- suplanuotas skanavimas – regulariai skanuojama nurodytu laiku dienos, savaitės ar mėnesio;
- užsikrovimo skanavimas – antivirusinė pradeda skanuoti, užsikrovus kompiuteriui;
- realaus laiko skanavimas – antivirusinė dirba veikiant kompiuteriui, nuolatos skanuoja bylas esančius kompiuteryje ir taip leidžia efektyviai apsisaugoti nuo virusų.

Užkardų kartu su antivirusinėmis programomis naudojimas, leidžia ženkliai sumažinti tikimybę būti pažeistam.

Saityno taikomųjų sistemų užkardos. Šios užkardos leidžia apsaugoti saityno taikomąją sistemą, nuo grėsmių, kurias gali aptikti ir atremti viena ar kita pasirinkta užkarda. Šios užkardos įdiegiamos kiekvienoje saityno taikomojoje sistemoje, taip leidžiant jas apsaugoti nuo galimų, plačiai paplitusių saityno grėsmių. Kaip pavyzdį, galime paminėti Roko Paškevičiaus magistro darbą tema „Žiniatinklio programų ugniasinės sudarymas ir tyrimas“ [22]. Darbe aprašomas metodas, kurį įdiegus į saityno taikomąją sistemą, apsaugoma sistema nuo pagrindinių saityno sistemų grėsmių.

Duomenų bazių apsauga. Gali būti taikomi vietiniai saugos sprendimai arba naudojami populiarūs saugos sprendimai nuo duomenų pakeitimo ar išnaikinimo. Tarp tinkamų sprendimų gali būti GreenSQL sprendimas [27], kuris apsaugo nuo aktualiausios šių dienų saityno taikomųjų sistemų atakos, kuri paveikia duomenų bazėse saugomą informaciją.

1.8 Įsilaužimų į taikomąsias sistemas aptikimo priemonių palyginimas

4 lentelė. Įsilaužimų į taikomąsias sistemas aptikimo priemonių palyginimas

Kriterijai / Priemonė	Integravimas	Greitaveika	Paskirties apsaugojimo lygis
Užkardos	*Vidutinis*	*Didelė*	*Vidutinis-Didelis*
Saityno užkardos	*Lengvas*	*Vid.-Didelė*	*Didelis*
Antivirusinės	*Lengvas*	*Maža*	*Vidutinis*
DXJL	*Lengvas*	*Vidutinė*	*Didelis*
XGD	*Vidutinis*	*Didelė*	*Didelis*
RP	*Lengvas*	*Vidutinė*	*Didelis*
GSQL	*Vidutinis*	*Vidutinė*	*Didelis*

Sudarėme 4 lentelę, norėdami pavaizduoti ir palyginti mūsų aprašytų priemonių pasirinktas savybes, tai: integravimo galimybės, greitaveika bei apsaugojimo lygis, priklausant kokios paskirties yra atitinkama priemonė. Pagal nusakytas savybes matome, kad priemonių galimybės nėra vienodos, nėra pilnai apsaugančios sistemas, bet pavyzdžiui naudojant užkardą ir kartu antivirusinę, apsauga taikomosioms sistemoms sustiprėja. Saityno užkardos leidžia apsaugoti saityno taikomąsias sistemas nuo pagrindinių pavojų, o duomenų bazių valdymo sistemų apžvelgti apsaugos metodai leidžia užtikrinti saugumą dėl savo sudėtingos architektūros ir dėl to kenčia greitaveika.

5 lentelė. Įsilaužimų į taikomąsias sistemas aptikimo priemonių tipai

Įsilaužimų į taikomąsias sistemas aptikimo priemonės	Įterptinės
Užkardos	
Saityno užkardos	✓
Antivirusinės	
DXJL	✓
XGD	✓
RP	✓
GSQL	✓

5 lentelėje išskyrėme mūsų aprašytas taikomųjų sistemų įsilaužimų aptikimo priemones, pažymėdami kurias priemones laikome įterptinėmis apsaugos priemonėmis.

Įterptinė sistema.

Įterptine sistema laikome programinio kodo dalį, kurį galima patalpinti į jau veikiančią taikomąją sistemą, taip pat atliekami nedideli pakeitimai reikalingi jos diegimui. Pati įterptinė sistema suprojektuota ir sukurta įgyvendinti tam tikrą funkcionalumą. [29, 30]

Įterptinių sistemų savybes pateikiame lentelėje nr. 6.

6 lentelė. Įterptinių sistemų savybės

Įterptinių sistemų savybės
• Talpinimas į veikiančią taikomąją sistemą; • Reikalingi nedideli pakeitimai taikomojoje sistemoje, įterptinės sistemos diegimo metu; • Įterptinė sistema atlieka konkrečia paskirtį.

1.9 Analizės išvados

- 2010 metais kompiuterinių atakų skaičius, palyginus su 2009 m., padidėjo 36%;
- 55% visų kompiuterinių atakų įvykdoma prieš saityno taikomąsias sistemas;
- Dažniaisiai pasitaikanti šių dienų ataka, panaudojama tiek lokaliose, tiek saityno taikomosiose sistemose – *SQL* injekcija. Tai nustatėme išskyrę taikomąsias sistemas į lokalias ir saityno, išanalizavę pagrindines lokalių taikomųjų sistemų grėsmes ir priemones joms įgyvendinti, taip pat ir populiariausias saityno taikomųjų sistemų atakų tipus.
- Dažniaisiai pasitaikančios saityno taikomųjų sistemų atakos: *SQL* injekcijos ir *XSS* atakos. Šiomis atakomis išnaudojamos taikomųjų sistemų apsaugos spragos, kurios leidžia atakuoti duomenų bazes;
- aprašėme pagrindines priemones, taikomas apsaugai nuo įvairių grėsmių taikomosioms sistemoms, tai: ugniasienės, antivirusinės, saityno taikomųjų sistemų užkardos, bei duomenų bazių saugos priemonės.
- Atlikus kelių metodų taikomų duomenų bazių apsaugai nuo *SQL* atakų analizę, nustatyta, kad siekiant pritaikyti apsaugą realizuotų taikomųjų sistemų duomenims, patogiausia ir efektyviausia taikyti įterptines sistemas, kadangi jos turi konkrečią paskirtį, pvz.: apsauga, tam tikras funkcionalumas ir kitos.

  Aprašytų metodų savybės:

- DXJL – įterptinė sistema, kurioje analizuojamas tinklu keliaujantis srautas, kas reikalauja didelių resursų;
- XGD – įterptinė sistema, pasižyminti sudėtinga architektūra, dirbanti kaip saugumo vartai tarp duomenų bazės ir saityno serverio (angl. *web server*), kuri reikalauja didelių resursų, tačiau apsaugo duomenis;
- RP – įterptinė sistema, pritaikyta konkrečioms saityno taikomosioms sistemoms;
- GSQL – įterptinė sistema, veikianti kaip įgaliotasis serveris (angl. *proxy*) filtruojanti įeinančias užklausas.

2. ĮTERPTINĖS ĮSILAUŽIMŲ Į TAIKOMĄSIAS SISTEMAS APTIKIMO PRIEMONĖS PROJEKTAS

2.1 Darbo tikslas ir keliami reikalavimai

Pirmoje darbo dalyje atliktoje lokalių ir saityno taikomųjų programų analizėje, išsiaiškinome svarbiausias šių dienų grėsmes. Vienas iš didžiausių kiekvieno saugumo specialisto uždavinys – apsaugoti kliento informaciją, talpinamą duomenų bazėje, todėl projektuosime taikomosios programos įterptinę sistemą, siekiant apsaugoti brangiausią turtą – informaciją. Tikslas – suprojektuoti įterptinę įsilaužimų į taikomąsias sistemas aptikimo sistemą, siekiant apsaugoti duomenų bazes nepriklausomai nuo to, kokia yra taikomoji sistema: lokali ar saityno.

2.2 Funkciniai ir nefunkciniai reikalavimai

Funkciniai reikalavimai, keliami įterptinei sistemai į taikomąsias sistemas:

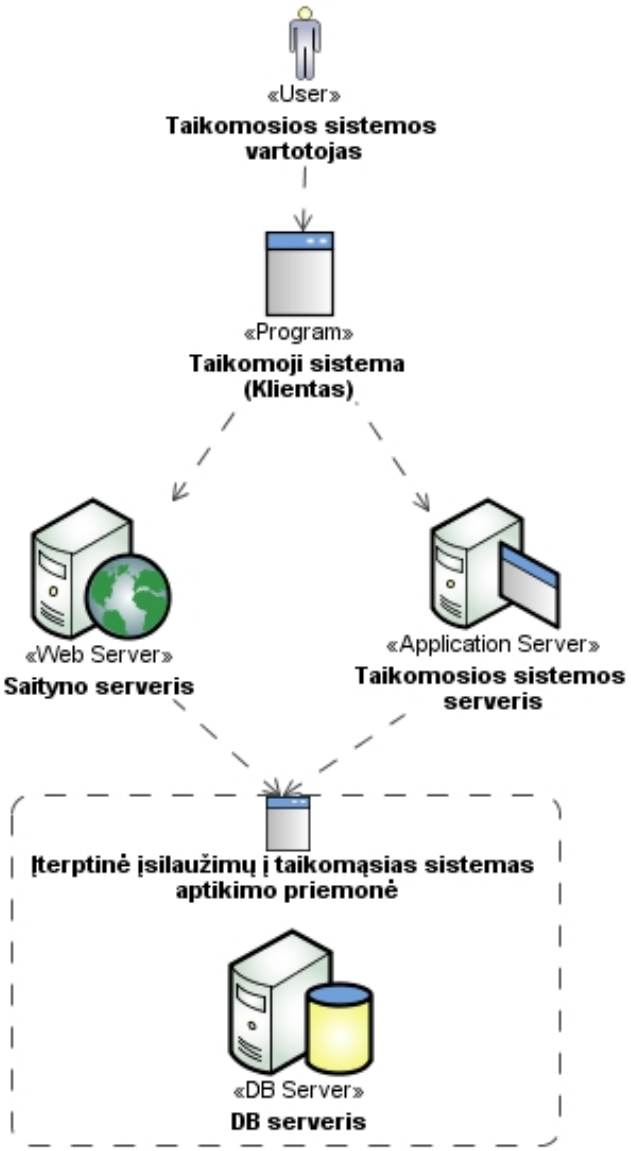
- įterptinė sistema turi apsaugoti nuo galimų kenksmingų vartotojo įvestų duomenų;
- apsauga nuo naujų galimų grėsmių, jeigu nepakanka, pildoma taisyklių bazė arba tobulinama priemonė;
- įterptinė sistema naudos taisyklių bazę, kuri:
  - bus pildoma taisyklėmis;
  - talpins galimų grėsmių informaciją apie įeinantį turinį ir apie įeinančių užklausų logiškumą;
  - bus naudojama kaip pagrindinis galimų atakų šaltinis;
- fiksuojami kenksmingi veiksmai juodųjų sąrašų bazėje;
- automatiškas blokavimas kenksmingus veiksmus atlikusio vartotojo;
- užblokuoto vartotojo pašalinimas iš juodųjų sąrašų bazės.

Nefunkciniai reikalavimai, keliami įterptinei sistemai:

- neturi būti prarandama, nutekinama ar modifikuojama informacija, patalpinta duomenų bazėje;
- sistemos greitis, neturi trukdyti vartotojo darbo;
- įterptinė sistema nepriklauso nuo taikomosios sistemos programavimo kalbos.

2.3 Įterptinės sistemos architektūra

Mūsų siūlomos įterptinės sistemos architektūros principas – įterptinės sistemos talpinimas ne taikomosios programos lygmenyje, o žemesniame – duomenų bazės lygyje. Tokiu būdu, visas įeinantis srautas bus patikrinamas ir bus išvengiama kenksmingų atakų padarinių (pav. 9). Duomenų bazei aptarnaujant daugiau negu vieną taikomąją sistemą ir panaudojus įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę, apsauga būtų užtikrinta visoms aptarnaujamoms taikomosioms sistemoms.



9 pav. Įterptinės sistemos architektūra

Paveiksle nr. 9 pavaizduotos įterptinės įsilaužimų aptikimo sistemos architektūros aprašymas:

- sistemos(ų) vartotojai besinaudojantys taikomąja sistema;
- duomenų bazė gali aptarnauti kelias taikomąsias sistemas;
- duomenų bazės lygmenyje patalpinama įterptinė sistema;
- įterptinė sistema naudodamasi taisyklių bei juodąja bazėmis aptinka ir atmeta galimai žalingai modifikuotas užklausas;
- įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė registruoja atitikimus taisyklių ir juodųjų sąrašų bazėse įvykių bazėje.

Įterptinės sistemos architektūra atvaizduoja, kad taikomoji sistema gali būti tiek lokalaus, tiek saityno tipo ir programavimo kalbos pasirinkimas įtakos neturi įterptinės sistemos veiklai, kadangi ji patalpinama duomenų bazės lygmenyje ir yra iškviečiama iš taikomosios sistemos.

Mūsų įterptinės sistemos architektūrą sudarančios dalys:

- Taisyklių bazė – pildoma taisyklių bazė, kurią sistema naudodama, ieškos galimų kenksmingų veiksmų įeinančiame sraute;
- Juodųjų sąrašų bazė – bazė, talpinanti informaciją apie kenksmingus veiksmus atlikusio vartotojo *IP* adresą, siekiant identifikuoti vartotoją;
- Įvykių bazė – bazė, kurioje registruojama pažeidžiamumų informacija;
- Įterptinė sistema – programinis kodas, kuriame atliekamas įeinančio srauto tikrinimas pagal taisyklių ir juodąsias bazes.

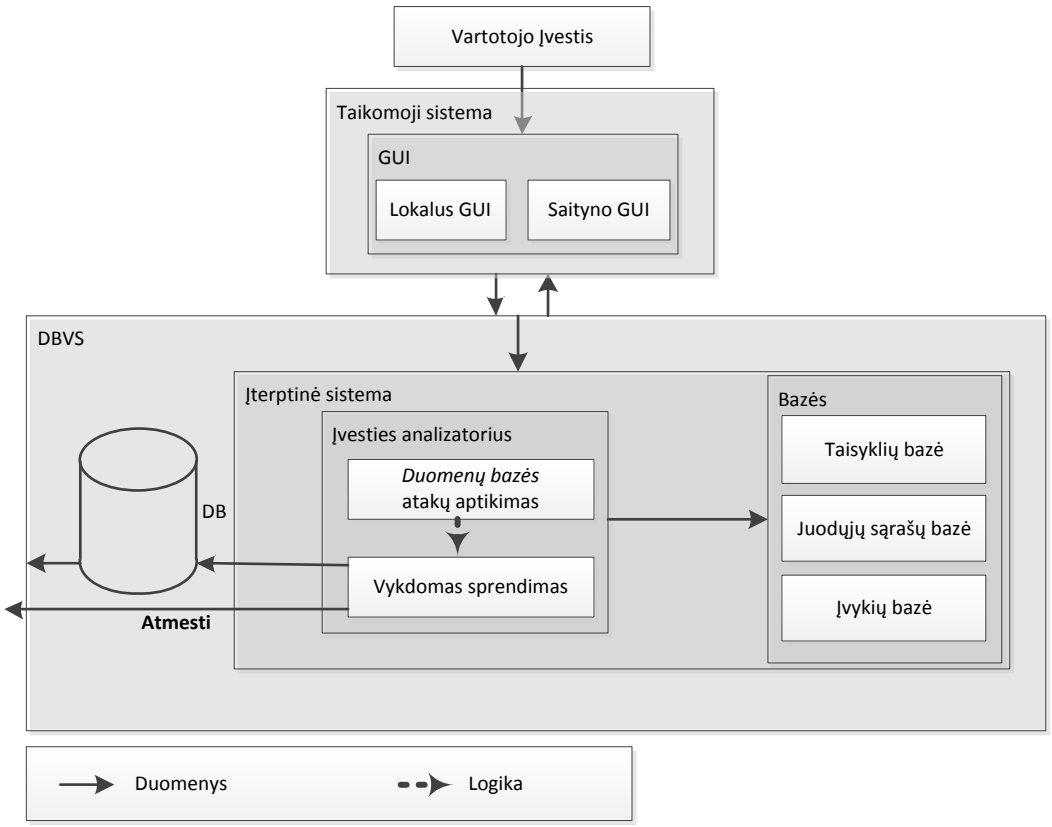
Tokiu principu, visas įeinantis srautas bus patikrinamas duomenų bazės lygmenyje, kas užtikrina patikimą saugumą nuo kenksmingų vartotojo modifikuotų užklausų.

2.4 Įterptinės sistemos veikimo principas

Aprašytų atakų apsauga.

Panaudojant mūsų siūlomą įterptinę sistemą, įmanoma apsaugoti taikomąsias sistemas nuo šių dienų populiariausių pirmame skyriuje aprašytų atakų, kurios galimos saityno ir lokaliose taikomosiose programose.

Įterptinės įsilaužimų aptikimo sistemos veikimas.



10 pav. Įterptinės sistemos schema

10 pav. pateikiame įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos schemą. Vartotojo įvesti duomenys gali būti tiek per saityno, tiek per lokalių sistemų vartotojo sąsajas. Duomenų bazių valdymo sistemoje patalpinta įterptinė sistema filtruoja perduotus įvestus vartotojų duomenis pagal taisyklių ir juodąsias bazes (jeigu nurodyta naudoti juodąją bazę). Atsižvelgiant į gautus rezultatus, priimamas sprendimas užklausas vykdyti ar įvykus atakai fiksuoti vartotoją juodųjų sąrašų bazėje bei įvykių registravimo bazėje bei užklausą atmesti.

2.4.1 Taisyklių bazė

Įterptinė sistema remiasi taisyklių baze, kuri bus nuolatos papildoma, atsižvelgiant į galimas grėsmes. Įterptinė sistema naudodamasi taisyklių baze, patikrina kiekvieną įeinantį duomenų srautą ir esant taisyklių atitikimams – jį atmeta.

Taisyklės sudaromos pagal galimas *SQL* užklausų modifikavimo sąlygas panaudojant reguliariąsias išraiškas, kuriomis bus patikrinamas vartotojo perduotas srautas. Tai specialūs taisyklių rinkiniai, aprašantys tekstinį šabloną, pagal kurį randamas reikalingas tekstas ar teksto fragmentas. Tikslui apsaugoti duomenų bazes nuo įvestų galimai kenksmingų vartotojo duomenų, būtina sudaryti taisykles, kurios aptiktų *SQL* užklausas modifikuojantį tekstą. Stipresnei apsaugai užtikrinti, taisyklėse aprašysime ir šešioliktaines reikšmes.

Taisyklių bazės veikimo algoritmo užrašymas pseudokodu:

Kintamieji:

r ← taisyklių sąlygų sąrašas [ ]

i ← vartotojo įvestis

s ← sprendimas

e ← rezultatai

```
while ( r )
        if ( r sąlyga atitinka i )
                s ← fiksuojamas sprendimas
        else
                r ← naudojama sekanti taisyklė pagal didžiausią prioritetą
        end
end
        if ( s = užfiksuota ataka )
                e ← tuščias rezultatas, užklausa nevykdoma
        else
                e ← užklausa vykdoma, grąžinami jos rezultatai
        end
return e
```

Taisyklės kuriamos siekiant išvengti įeinančiame sraute esančių galimai *SQL* užklausas modifikuojančių sąlygų, tokių kaip:

- *SELECT;*
- *UPDATE;*
- *DELETE;*
- *ALTER;*
- *DROP;*
- *CREATE;*
- *UNION.*

Pateikiame kelis lengvai suprantamus taisyklių sudarymo pavyzdžius.

Vartotojo įvestuose duomenyse ieškome kabutės arba kabliataškio, kurie leistų uždaryti esamą užklausą ir vykdyti naują su duomenų bazės užklausų kūrimo žodžiais *select*, *update*, *delete*, *alter*, *drop*, *create*:

- '*.*((\x27)|(')|(\x3B)|(;)).*select*' ;
- '*.*((\x27)|(')|(\x3B)|(;)).*update*' ;
- '*.*((\x27)|(')|(\x3B)|(;)).*delete*' .

Reguliariose išraiškose galima tikrinti ir pagal šešioliktainę skaičiavimo sistemą perduotus duomenis. Tokiu sąlygų sudarymo būdu galima aprašyti įvairias taisykles, siekiant apsaugoti duomenų bazę. Pateiksime pavyzdį reguliariosios išraiškos, kuri aptinka kenksmingai modifikuotoje užklausoje populiarius *SQL* užklausos loginius jungiklius *OR* bei *AND*:

- '*.*((\x27)|(')|(\s*\w*))\s*((\x6F)|o|(\x4F))\s*((\x72)|r|(\x52))' ;
- '\s*\w*\s*((\x27)|(')|(\s*\w*))\s*((\x61)|a|(\x41))\s*((\x6E)|n|(\x4E))\s*((\x64)|d|(\x44))' .

Pateikiame reguliariosios išraiškos pavyzdį, kuriame siekiama aptikti įvestuose duomenyse išliekančių *XSS* ar *CSRF* atakų požymius:

- Siekiama nepraleisti duomenų su galimai nuotraukų, nuorodų, *JavaScript* kalbos žymėmis, kurias panaudojus patalpinamos šios atakos duomenų bazėje. Pavyzdys:

  '*.*((\\x3C)|(\\x3E)|(<)|(>))' .

Taisyklių bazės įrašai pildomi naujais perduodant duomenis į numatytas procedūras. Taisyklės įvertinamos prioriteto įverčiais, pagal kurias pirmiausiai bus tikrinami duomenys.

2.4.2 Juodųjų sąrašų bazė

Įterptinei sistemai atmetus įeinantį duomenų srautą ar komandas dėl neatitikimo taisyklių bazėje esantiems įrašams, informacija apie vartotojo *IP* adresą fiksuojama juodųjų sąrašų bazėje. Taip vyksta ir grįžtamasis ryšys – sistema gavusi signalą iš atitinkamo *IP* adreso, kuris patalpintas juodųjų sąrašų bazėje, automatiškai atmeta jo įeinantį srautą. Vartotojo *IP* adreso registravima nėra garantuota apsauga nuo atakuotojo, kadangi jis jį gali pasikeisti arba įvykdyti iš vidinio tinklo iš kurio tėra vienas išorinis *IP* adresas. Todėl *IP* adresai fiksuojami kaip įtartini ir tik po nustatyto (5 kartai) kiekio kartų adresas blokuojamas. Įvykus nesklandumams ar pagal taisykles užblokuotam *IP* adresui kurį būtina atstatyti, reikia iš naujo fiksuoti juodųjų sąrašų bazėje šį adresą kaip įtartiną.

Juodųjų sąrašų bazės naudojimas yra konfigūruojamas:

- kiekvienai užklausai galima nurodyti – jai naudoti juodųjų sąrašų bazės fukncionalumą ar ne, perduodant *IP* adresą;
- perdavus *IP* adresą, tolimesni įterptinės sistemos veiksmai:
    - filtruoja esamus *IP* adresus patalpintus juodųjų sąrašų bazėje – radus atitikimą įvykis fiksuojamas įvykių žurnale ir grąžinamas tuščias rezultatas;
    - tikrinami vartotojo įvesti duomenys – radus atitikimą pagal taisyklių bazėje aprašytas reguliariąsias išraiškas, įvykis registruojamas įvykių žurnale, *IP* adresas patalpinamas juodųjų sąrašų bazėje ir taikomąjai sistemai grąžinamas tuščias rezultatas;
    - *IP* adresai fiksuojami kaip įtartini bent 5 kartus, po kurių adresas yra blokuojamas ir reikia papildomos įterptinės sistemos administratoriaus priežiūros norint leisti naudotis taikomąja sistema šių adresų vartotojams.

Juodųjų sąrašų bazė gali būti nenaudojama, norint užtikrinti didesnę užklausos vykdymo spartą, tačiau taip prarandant naudingą statistiką apie atakuotojus ir jų atakų kiekį bei atakavimo laikus.

2.4.3 Įvykių bazė

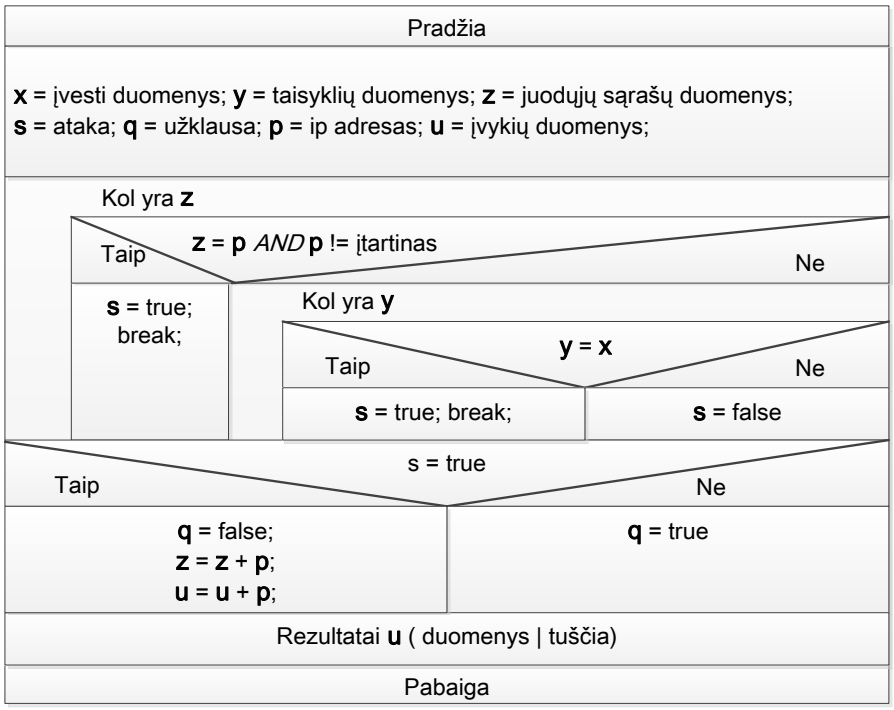
Įvykių bazėje registruojama informacija apie įeinančio srauto atitikimus taisyklių ir juodųjų sąrašų bazėse. Šiuo būdu surenkama reikalinga informacija pažeidžiamumų statistikos peržiūrai atlikti. Registruojama informacija įvykių bazėje:

- Laikas;
- Taisyklių bazėje atitikusios taisyklės identifikavimo numeris;
- Jeigu įjungtas juodųjų sąrašų bazės funkcionalumas, registruojamas *IP* adresas.

2.4.4 Įterptinės sistemos logika

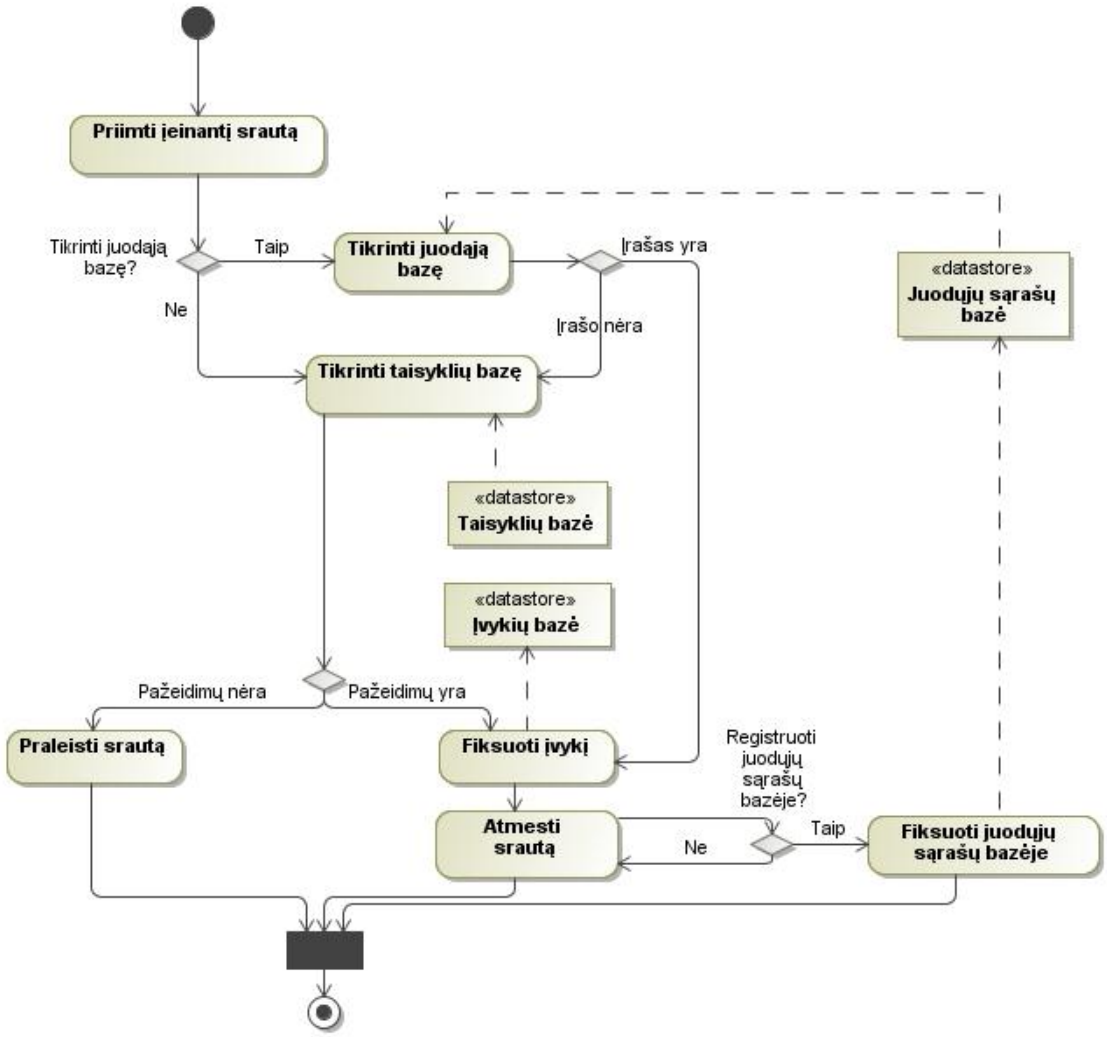
Pagrindinė galimų pažeidžiamumų logika patalpinta įterptinėje sistemoje, kuri naudojasi tiek taisyklių, tiek juodąja bazėmis. Ji apsprendžia kaip elgtis su vartotojo perduotais duomenimis, kokius sprendimus priimti atitikus tam tikras sąlygas.

11 paveiksle pateikiame įterptinės sistemos algoritmo struktūrogramą.



11 pav. Įterptinės sistemos algoritmo stuktūrograma

12 pav. pateikiame įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės veikimo schemą.



12 pav. Įterptinės sistemos algoritmas

Visų pirmiausia, įterptinė sistema priims įeinantį srautą, kuriuo manipuliuos atlikdama tikrinimus. Kitas žingsnis – priklausomai nuo konfigūracijos, vartotojas identifikuojamas juodųjų sąrašų bazėje arba ne. Identifikavus vartotoją pagal *IP* adresą, vartotojo pateiktas srautas atmetamas ir įvykis registruojamas, jeigu ne – tikrinama taisyklių bazė. Esant atitikimams su aprašytomis reguliariomis išraiškomis taisyklių bazėje – srautas taipogi yra atmetamas ir informacija fiksuojama juodųjų sąrašų bazėje bei įvykių žurnale. Kitu atveju patikrintas, saugus srautas praleidžiamas tolimesniems sistemos veiksmams ir vartotojui grąžinama informacija atitinkanti originalias užklausas.

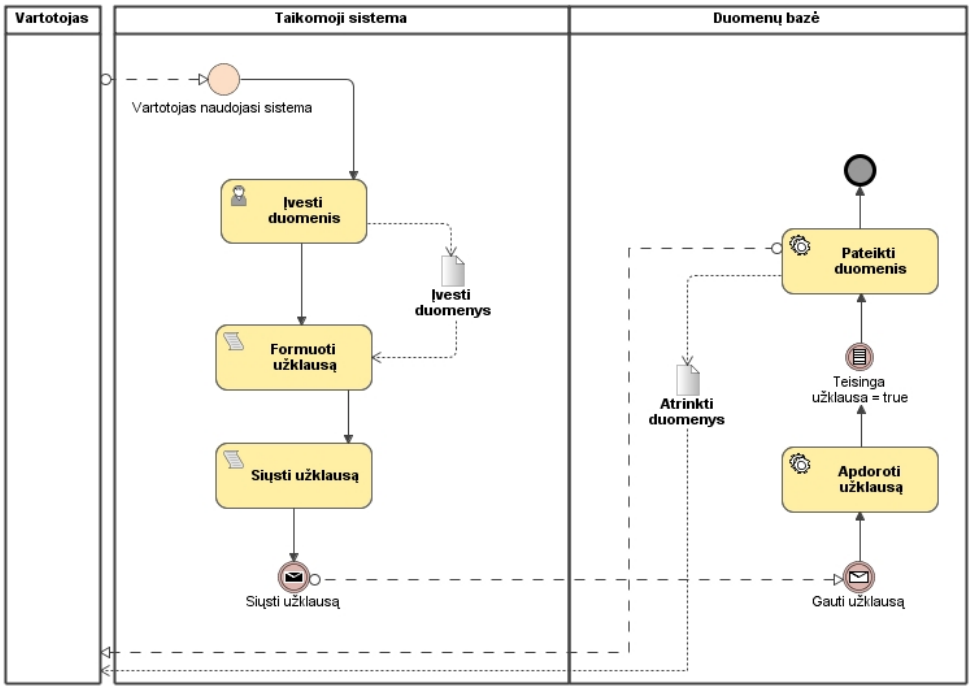
7 lentelė. Naudojamų bazių struktūros

Bazė	**Struktūra**			
Taisyklių	*Identifikacinis nr.*	*Taisyklė*	*Komentaras*	
Juodųjų sąrašų	*Identifikacinis nr.*	*IP*	*Kiekis*	
Įvykių	*Identifikacinis nr.*	*Data/laikas*	*Taisyklės Identifikacinis nr.*	*IP*

7 lentelėje aprašytos įterptinėje įsilaužimų į taikomąsias sistemas aptikimo priemonėje naudojamų bazių struktūros.

2.5 Esamas taikomųjų sistemų veiklos procesas

13 paveiksle pavaizduosime situaciją iš esamų realizacijų – taikomųjų sistemų bendravimas su duomenų bazėmis. Pateikiamama situacija „esama (angl. *as-is*)“, kurioje nėra naudojama įeinamo srauto validacija nei taikomojoje sistemoje, nei duomenų bazėje.



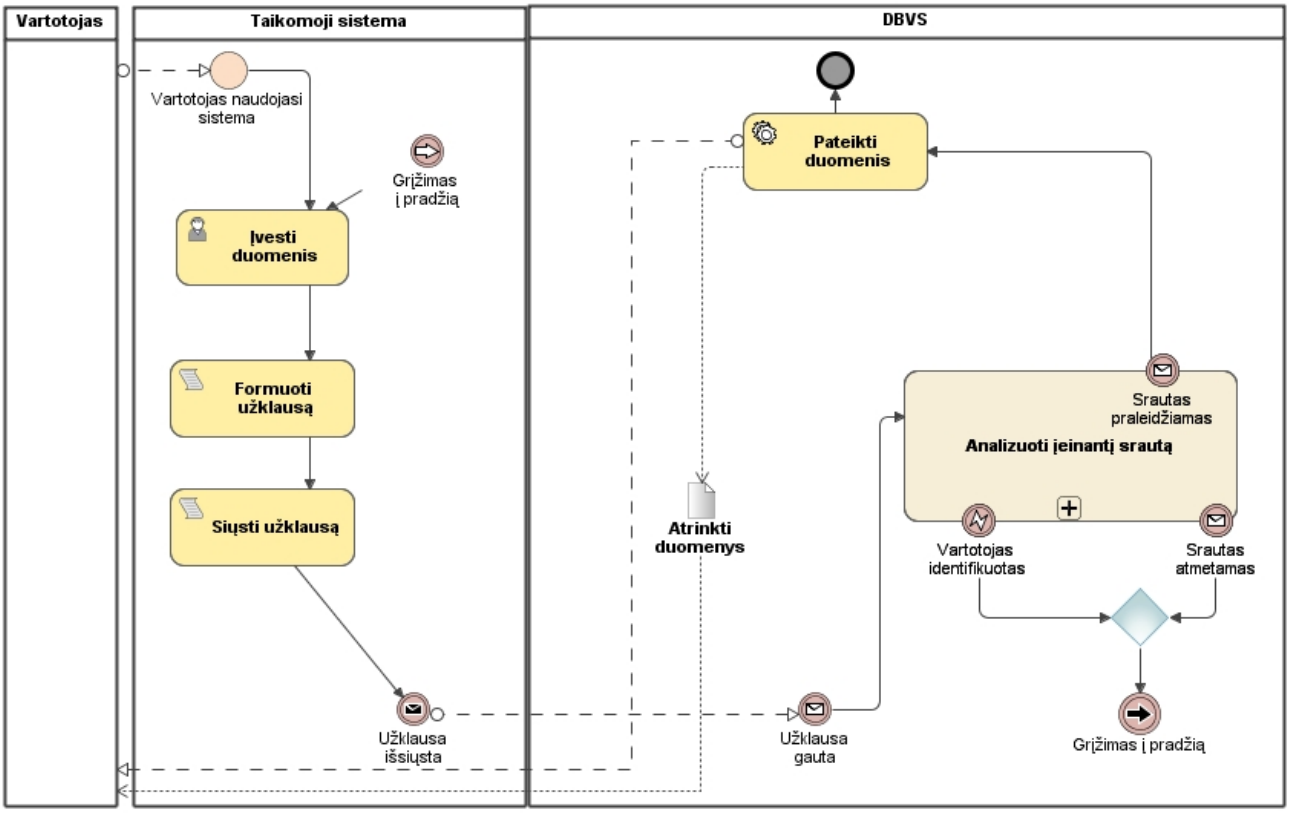
13 pav. Esamas (angl. *as-is*) veiklos procesas „Taikomosios sistemos bendravimas su duomenų baze“ (BPMN notacija)

13 pav. BPMN „esama (angl. *as-is*)“ diagramoje perteikta situacija puikiai parodo sistemos pažeidžiamumus *SQL* užklausų modifikavimams. Pats svarbiausias dalykas – tikrinti kiekvieną vartotojo įvedimą. Taikomųjų sistemų kūrėjai, dažniausiai pamiršta, praleidžia įvedamų duomenų patikrinimą, kas gali sukelti nepageidaujamų pasekmių. Todėl mes ir

projektuojame įterptinę sistemą, kuri duomenų bazės lygyje patikrintų visus vartotojo įvestus duomenis, bei apsaugotų sistemas nuo informacijos nutekėjimo, modifikavimo ar praradimo.

2.6 Siekiamas taikomųjų sistemų veiklos procesas

„Saugus taikomosios sistemos bendravimas su duomenų baze“ procesas pateiktas BPMN „siekiamoje (angl. *to-be*)“ diagramoje 14 paveiksle. Paveiksle matome, jog nėra jokios duomenų validacijos taikomosios programos lygmenyje, nors saugumo sumetimais pravartu validavimą naudoti ir šiame lygmenyje.

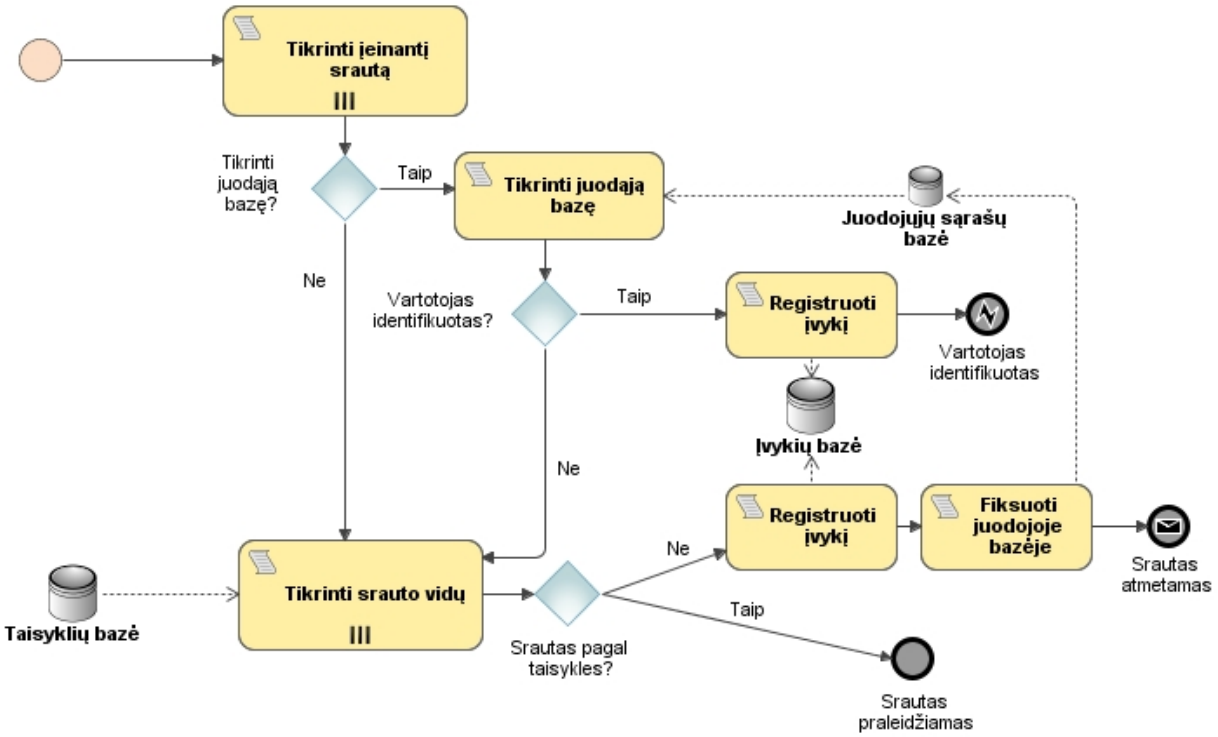


14 pav. Siekiamas (angl. *to-be*) veiklos procesas „Saugus taikomosios sistemos bendravimas su duomenų baze“ (BPMN notacija)

Įterptinės sistemos kodas bus įterpiamas į naudojamą duomenų bazę – tuomet bus filtruojamas ir apsaugotas visas įeinantis srautas – įvykus kenksmingiems veiksmų aptikimui aprašytose taisyklių bazės taisyklėse, įeinantis srautas bus atmetamas ir ši informacija bus

fiksuojama juodųjų sąrašų bazėje. Duomenų validavimas pačioje taikomojoje sistemoje nebus privalomas, tačiau patartinas didesniam saugumo užtikrinimui.

Kitoje diagramoje (15 pav.) pavaizduosime išplėstąjį „Analizuoti įeinantį srautą“ procesą, kuriame detaliai atvaizduota proceso logika.

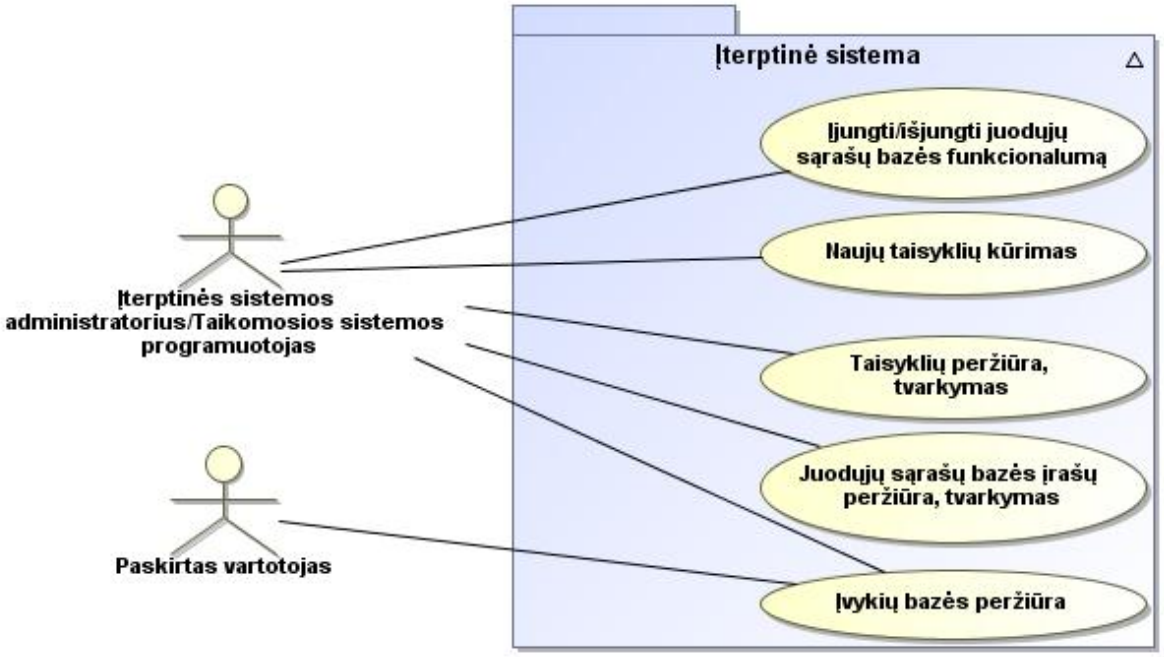


15 pav. Veiklos procesų diagrama „Analizuoti įeinantį srautą“ (BPMN notacija)

Detalus „Analizuoti įeinantį srautą“ proceso atvaizdavimas parodo, kokie procesai jame yra atliekami ir kokios sąlygos vykdomos – srauto tikrinimas, fiksavimas juodųjų sąrašų bazėje pagal aprašytas sąlygas. Pavaizduoti salygų rezultatai bei dvi naudojamos bazės: taisyklių ir juodųjų sąrašų.

Įeinančio srauto patikrinimas užtikrins duomenų bazės saugumą žemesniame lygmenyje (t.y. ne taikomojoje sistemoje), o tai reiškia didesnį informacijos saugumą nuo galimų nesankcionuotų atakų.

2.7 Įterptinės sistemos konfigūracija



16 pav. Įterptinės sistemos konfigūracijos panaudojimo atvejų diagrama (UML notacija)

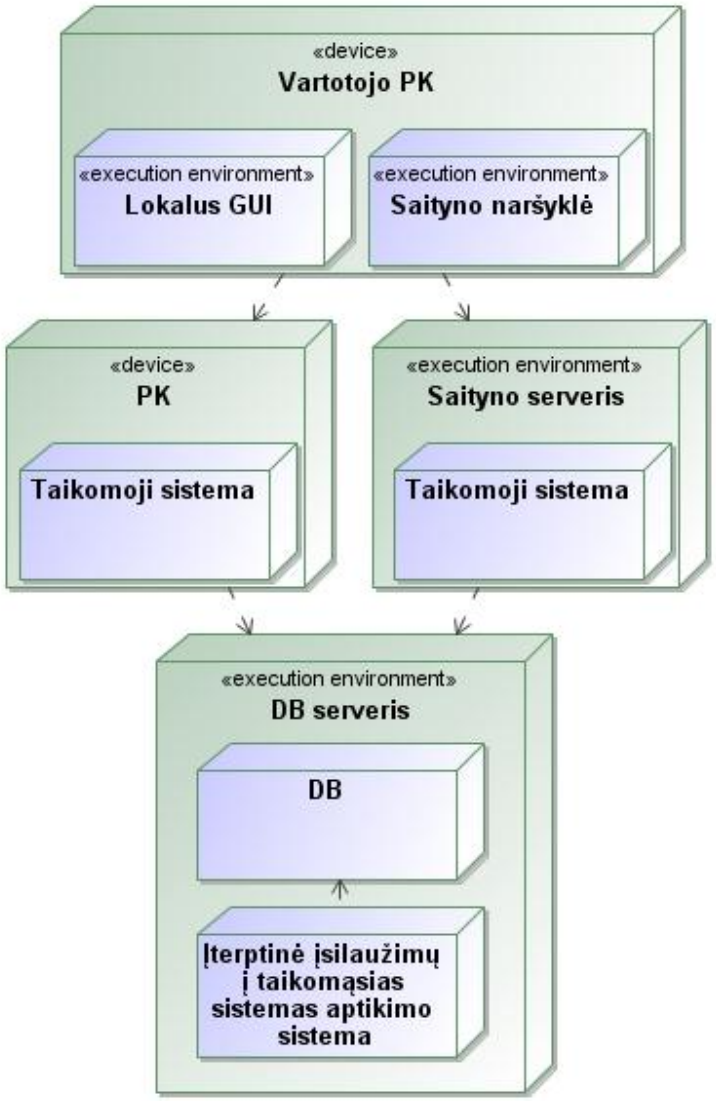
Pagal 16 paveikslo panaudojimo atvejų diagramą matome išskirtus „įterptinės sistemos administratoriaus/taikomosios sistemos programuotojo“ ir „paskirto vartotojo“ veikėjų galimus atlikti veiksmus įterptinėje įsilaužimų į taikomąsias sistemas aptikimo sistemoje. Pirmojo veikėjo galimi veiksmai:

- Juodųjų sąrašų bazės funkcionalumo įjungimas/išjungimas;
- Naujų taisyklių kūrimas;
- Taisyklių peržiūra bei tvarkymas;
- Juodųjų sąrašų bazės įrašų peržiūra bei tvarkymas;
- Įvykių bazės peržiūra – įvykių statistika.

„Paskirto vartotojo“ veikėjas gali peržiūrėti įvykių bazėje sukauptus įrašus apie užregistruotus įvykius, vesti statistiką.

2.8 Diegimo diagrama

Diegimo diagramoje (žr. 17 pav.) pavaizduosime įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos realizaciją.



17 pav. Diegimo diagrama (UML notacija)

Mūsų projektuojamos sistemos realizacijos atvejo diegimo diagrama pavaizduoja, kurioje sistemos vietoje savo vietą užims įterptinė sistema. Iš diagramos aiškiai matyti, jog į naudojamą duomenų bazę bus patalpintas papildomas programinis kodas, kuris atliks jame aprašytus veiksmus ankstesniuose skyreliuose.

2.9 Išvados

- Projektavimo etape apibrėžtas kuriamos įterptinės sistemos tikslas – sukurti įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės projektą, leidžiantį universaliai panaudoti šią sistemą įvairioms taikomosioms sistemoms, siekiant užtikrinti apsaugą nuo duomenų bazės atakų;
- Projektuojamos sistemos architektūra išskiriama į šias dalis: taisyklių bazė, juodųjų sąrašų bazė, įvykių bazė ir įterptinės sistemos logika;
- Taisyklių bazėje laikoma taisyklių informacija, aprašyta reguliariųjų išraiškų metodu, kurias pasitelkus aptinkamas įeinančio srauto kenksmingas turinys;
- Vartotojo identifikavimui juodųjų sąrašų bazėje talpinama vartotojo *IP* informacija. Jeigu *IP* adresas jau yra juodųjų sąrašų bazėje, jo perduotas srautas yra iškart atmetamas, jeigu įrašo nėra juodųjų sąrašų bazėje ir aptinkamas kenksminga įvestis – tuomet įvykis registruojamas ir *IP* adresas talpinamas juodųjų sąrašų bazėje. *IP* adresas fiksuojamas kaip įtartinas 5 kartus, vėliau jis automatiškai užblokuojamas – leisti šio adreso vartotojui toliau naudotis taikomąja sistema, reikia įterptinės priemonės administratoriaus įsiterpimo;
- Priklausomai nuo taikomosios sitemos poreikių, juodųjų sąrašų bazė gali būti įjunta/išjungta;
- Pažeidžiamumų statistikos kaupimas vykdomas registruojant įvykių bazėje informaciją apie įterptinės sistemos aptiktus atitikimus tiek taisyklių, tiek juodųjų sąrašų bazėje;
- Duomenų bazių apsauga nuo galimų atakų įterptinėje sistemoje užtikrinama apjungiant taisyklių, juodąją ir įvykių bazes. Besinaudodama taisyklių ir juodaja bazėmis, įterptinė sistema patikrina vartotojo įvestą srautą ir jį filtruoja pagal aprašytas taisykles.
- Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės vieta bendroje architektūrinėje informacinėje sistemoje nurodyta diegimo diagramoje.

3. ĮTERPTINĖS ĮSILAUŽIMŲ Į TAIKOMĄSIAS SISTEMAS APTIKIMO PRIEMONĖS REALIZACIJA

Šios dalies tikslas – sukurti eksperimentinę realizaciją, kuri atitiktų mūsų projektinėje dalyje aprašytą modelį, siekiant į realizuotą taikomąją sistemą įterpti įsilaužimų aptikimo sistemą, gebančią aptikti kenkėjiškas *SQL* atakas. Šioje darbo dalyje aprašysime kuriamos įterptinės sistemos realizacijos dalį. Pateiksime savo eksperimentinę realizaciją nuo šių kenksmingų *SQL* atakų:

- *SQL* injekcijos;
- *XSS* atakos (išliekančios duomenų bazėje);
- *CSRF* atakos (išliekančios duomenų bazėje).

Situacija nenaudojant įterptinės įsilaužimų aptikimo sistemos.

Taikomoji sistema (tiek lokali, tiek saityno) suprogramuota neatsižvelgiant į duomenų saugumą, į galėjimą juos modifikuoti, pasisavinti ar net šalinti. Tokiu būdu kiekvienas įvedimo laukelis suteikia sistemos vartotojui galimybę pakeisti užklausą sau norima linkme. Tokių sistemų savininkai rizikuoja savo klientų duomenimis, kurie gali būti itin aktualūs ar slapti, kas galėtų sukelti atgarsį viešumoje ir padaryti neigiamą reklamą įmonei. Todėl kiekvienos sistemos analitikų/projektuotojų/programuotojų vienas iš svarbiausių tikslų – užtikrinti reikalaujamą duomenų saugumą.

Situacija naudojant įterptinę įsilaužimų aptikimo sistemą.

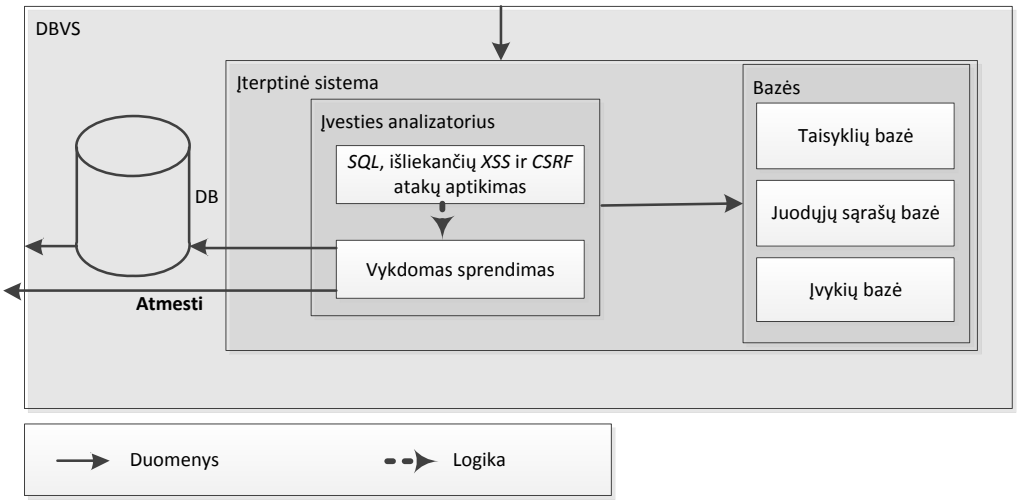
Taikomojoje sistemoje panaudojus mūsų siūlomą įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę yra užtikrinama tinkama apsauga nuo galimų *SQL* atakų. Tam tikslui pasiekti yra įgyvendintas užklausų duomenų filtravimo funcionalumas pagal taisykles, pagal fiksavimą juodųjų sąrašų bazėje, jeigu įjungtas šis funkcionalumas, bei įvykių registravimas, leidžiantis vesti statistiką apie patirtas atakas.

3.1 Realizacijos sprendimas

Eksperimentinei realizacijai pasirinktas *SQL* atakų aptikimo ir neutralizavimo sprendimas (pav. nr. 17).

Realizacijos dalys:

- Vartotojo sąsajos realizacijos dalis atlikta naudojant *ASP.NET* technologiją (*C#* programavimo kalba);
- Įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė realizuota panaudojant atviro kodo reliacinių duomenų bazių valdymo sistemą *MySQL*. Naudotas *MySQL WorkBench 5.2.36* įrankis.



18 pav. Įterptinės sistemos realizacijos sprendimas

Mūsų įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos realizacijos sprendimas (pav. 18) filtruoja vartotojo įvestus duomenis *MySQL* duomenų bazėje, kurie yra patikrinami taisyklių bei juodųjų sąrašų bazėje ir pagal atitinkamus rezultatus yra grąžinami rezultatai bei įvykiai registruojami duomenų bazėje.

3.2 Konfigūravimas

Taisyklių bazė.

Taisyklių bazėje laikomos aprašytos taisyklės reguliariųjų išraiškų pavidalu, siekiant aptikti galimas *SQL* užklausų pakeitimo ar kenksmingų įrašų patalpinimo sąlygas. Šiam kenksmingam aptikimui naudojamos reguliarios išraiškos, *MySQL* duomenų bazėje panaudojant specialų *REGEXP* žodį su aprašytomis taisyklėmis.

Projekto dalyje aprašėme pavyzdžius kaip atrodo reguliariosios išraiškos ir nuo kokio įvedamo srauto saugome duomenų bazę, tai – *SQL* kalbos užklausų formavimo žodžiai,

atitinkamo formavimo perduotas srautas, galintis pakeisti *SQL* užklausos logiką ir pan. Taip galima apsaugoti duomenų bazę nuo galimų *SQL* injekcijų, išliekančių *XSS* bei *CSRF* atakų.

Įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos taisyklių bazė gali būti papildoma įterptinės sistemos administratoriaus, panaudojant procedūrą:

- *call addSqlRule('.*((\\x27)|(\\%27)|(')).*delete','delete salyga');* .

Taisyklių bazės pavyzdys pateikiamas 19 paveiksle.

.*((/*)|(--)|(\x2F\x2A)|(\x2D\x2D))	komentaru salyga
.*((\x27)|(\%27)|(')).*alter	alter salyga
.*((\x27)|(\%27)|(')).*drop	drop salyga
.*((\x27)|(\%27)|(')).*create	create salyga
.*((\x27)|(\%27)|(')).*union	union salyga
.*((\x3C)|(\x3E)|(<)|(>))	XSS aptikimo salyga
.*((\x27)|(')|(\s*\w*))\s*((\x6F)|o|(\x4F))\s*((\x72)|r|(\x52))	or salyga
\s*\w*\s*((\x27)|(')|(\s*\w*))\s*((\x61)|a|(\x41))\s*((\x6E)|n|(\x4E))\s*((\x64)|d|(\x44))	and salyga
.*((\x27)|(\%27)|(')).*select	select salyga
.*((\x27)|(\%27)|(')).*update	update salyga
.*((\x27)|(\%27)|(')).*delete	delete salyga

19 pav. Taisyklių bazės pavyzdys

Juodųjų sąrašų bazė.

Juodųjų sąrašų bazėje yra saugomi potencialių atakuotojų *IP* adresai. Kiekvienos įvesties metu yra patikrinamas perduotas *IP* adresas. Jeigu jisai saugomas šioje bazėje, užklausa nėra tikrinama, įvykis registruojamas ir grąžinamas tuščias rezultatas.

Iš taikomosios programos perduodant *IP* adresą ir užfiksavus *SQL* injekcijos atakos pobūdį, įvykis yra registruojamas, bei perduotas *IP* adresas patalpinamas juodųjų sąrašų bazėje. Taip atmetami galimi būsimi šio vartotojo kenksmingi veiksmai.

Juodųjų sąrašų bazės pavyzdys pateikiamas 20 paveiksle.

id	ip	amount
1	127.168.1.64	5
2	127.168.1.65	1

20 pav. Juodųjų sąrašų bazės pavyzdys

Kaip matome 20 paveiksle, *IP* adresai užregistruojami juodųjų sąrašų bazėje ir kaskart jiems bandant vykdyti užklausas (jeigu įjungtas juodųjų sąrašų bazės funkcionalumas), bendras atitinkamo *IP* adreso perduodamų užklausų kiekis yra sumuojamas.

Įvykių bazė.

Įvykių bazėje talpinami įrašai apie bandymus įvykdyti kenksmingus veiksmus pagal reguliariųjų išraiškų sąlygas, aprašytas taisyklių bazėje.

Įvykus pažeidimui, fiksuojamas įvykio laikas, atitinkančios taisyklių bazės taisyklės identifikacinis numeris, bei *IP* adresas, jeigu yra įjungtas juodųjų sąrašų bazės funkcionalumas.

21 paveiksle pateikiamas įvykių bazės įrašas be įjungtos juodųjų sąrašų bazės funkcionalumo.

	id	createdate	ruleid	ip
	1	2012-02-25 12:46:24	83	NULL

21 pav. Įvykių bazės pavyzdys

3.3 Diegimas

Įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos diegimas duomenų bazėje atliekamas tokia seka:

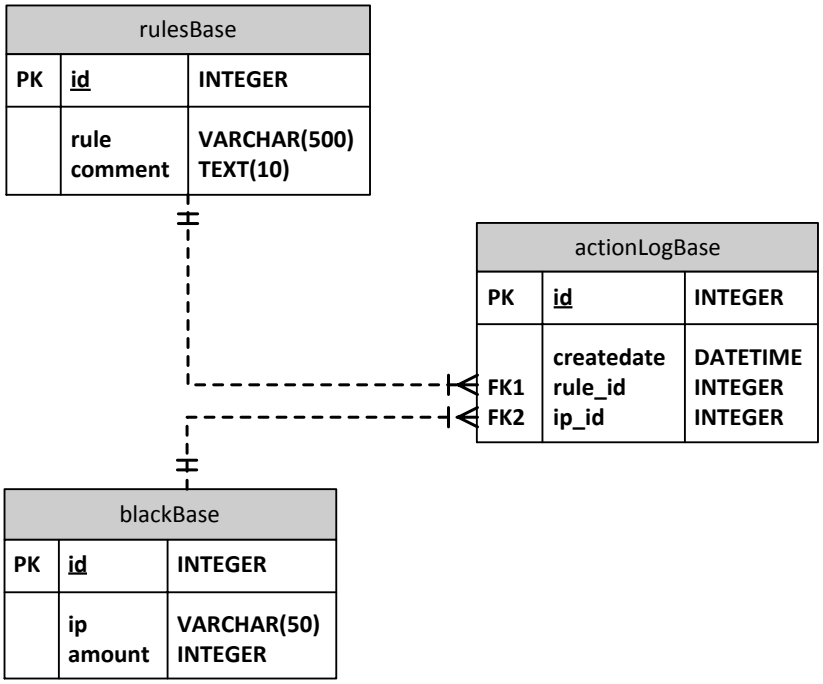
- įdiegiamas programinis įterptinės sistemos kodas;
- patalpinamos aprašytos pagrindinės taisyklės ir parametrai.

Iš taikomosios sistemos pusės diegimas atima daugiau laiko, pakeičiant esamas užklausas į perdavimą procedūrai. Pateikiamas pavyzdys iš eksperimentinės realizacijos:

```
  MySqlCommand cmd = new MySqlCommand(securityProcedure, sqlConn);
  cmd.CommandType = CommandType.StoredProcedure;
  cmd.Parameters.Add(new MySqlParameter("iQuery", "select * from rTableWebNet where id =
?;"));
  cmd.Parameters.Add(new MySqlParameter("iVar1", TextBox2.Text));
  cmd.Parameters.Add(new MySqlParameter("iIp", DBNull.Value));
```

3.4 Duomenų bazės schema

22 paveiksle pateikiame įterptinės įsilaužimų į taikomąsias sistemas aptikimo sistemos loginę duomenų bazės schemą.



22 pav. Loginė duomenų bazės schema

Pateikiame loginę duomenų bazės schemą, parodančią kaip buvo realizuota taisyklių, juodųjų sąrašų ir įvykių bazės duomenų bazėje ir kaip jos siejasi tarpusavyje.

3.5 Ypatumai

Privalumai:

- įterptinė sistema patalpinama naudojamoje duomenų bazių valdymo sistemoje, kuria gali naudotis tiek saityno, tiek lokalių taikomųjų sistemų kūrėjai, naudojantys jiems priimtiną programavimo kalbą;
- taisyklės gali būti tobulinamos ir papildomos (apsauga nuo išliekančių *XSS*, *CSRF* atakų);
- apsauga pagal *IP* adresus (jeigu įjungtas juodųjų sąrašų bazės funkcionalumas);

- lengvai pritaikoma taikomosiose sistemose, parašytose skirtingomis programavimo kalbomis;
- paprastas diegimas;

3.6 Išvados

- Tikslas apsaugoti taikomąsias sistemas nuo galimų *SQL* užklausų atakų duomenų bazės lygmenyje įgyvendintas panaudojant taisyklių, juodąją ir įvykių bazes, apjungtas įsilaužimų aptikimo logika;
- Juodųjų sąrašų bazės paskirtis kaupti informaciją apie potencialius kenkėjiškus *IP* vartotojus;
- Taisyklių bazės paskirtis laikyti pildomą taisyklių sąrašą, kuris naudojamas perduodamų užklausų duomenų filtravimui, siekiant atpažinti galimus kenksmingus veiksmus panaudojant reguliariąsias išraiškas;
- Įvykių bazės paskirtis kaupti informaciją apie aptiktus kenksmingus veiksmus: įvykio laikas, taisyklės identifikacinis numeris bei *IP* adresas (jeigu įjungtas juodųjų sąrašų bazės funkcionalumas);
- Įterptinė įsilaužimų į taikomąsias sistemas sistema veikia nepriklausomai nuo to ar tai lokali, ar saityno taikomoji sistema;
- Įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė lengvai pritaikoma taikomosiose sistemose, nesvarbu kokiomis programavimo kalbomis programuotos sistemos;
- Kelios taikomosios sistemos gali naudotis viena duomenų baze, kas leidžia turėti bendrą vieną patikimą apsaugą nuo kenksmingų *SQL* atakų;
- Klientinėje taikomųjų sistemų pusėje nebūtina tikrinti vartotojo įvestus duomenis, kadangi visa tai yra atliekama duomenų bazės lygmenyje;

4. ĮTERPTINĖS ĮSILAUŽIMŲ Į TAIKOMĄSIAS SISTEMAS APTIKIMO PRIEMONĖS TYRIMAS

4.1 Taikomųjų sistemų atakavimo eksperimentas

Po įsilaužimų ir atakų analizės pateiktos pirmajame skyriuje padarėme išvadą, jog labiausiai paplitusi šių dienų ataka yra *SQL* injekcija. Ji dažniausiai naudojama saityno taikomosiose sistemose, tačiau ją lengva panaudoti ir lokaliose taikomosiose sistemose.

Šiame taikomųjų sistemų atakavimo eksperimente išbandysime šios atakos galimybes populiariose šių dienų taikomosiose sistemose:

- **Microsoft Dynamics AX** – tai vienas iš populiarių Microsoft kompanijos sprendimų visame pasaulyje, skirtas įmonėms, turintis specialiai sukurtą pagrindą penkiose pramonės šakose bei teikiantis visapuses esmines finansų, žmogiškųjų išteklių ir operacijų valdymui skirtas ERP funkcijas [28];
- **Sandėlio valdymo sistema „X“**– universalus sandėlio valdymo sistemos sprendimas, skirtas gamintojams, didmeniniams paskirstymo centrams, viešiesiems logistikos centrams, sandėliavimo logistikos paslaugų tiekėjams. Sistema valdanti visus sandėlio procesus ir operacijas realiu laiku, kontroliuoja personalą bei naudojamą techniką.

Šiose sistemose atliksime paprasčiausią *SQL* injekcijos atakos bandymą, siekiant pažiūrėti išsiaiškinti jų atsparumą šioms atakoms.

Atakavimo metodas.

Atakuosime naudodami paprasčiausią *SQL* komandos pakeitimą. Pasirinksime atsitiktines formas ir bandysime pakeisti *SQL* užklausos logiką šiuo sakiniu: *tekstas‘ or 1=1;* -- .

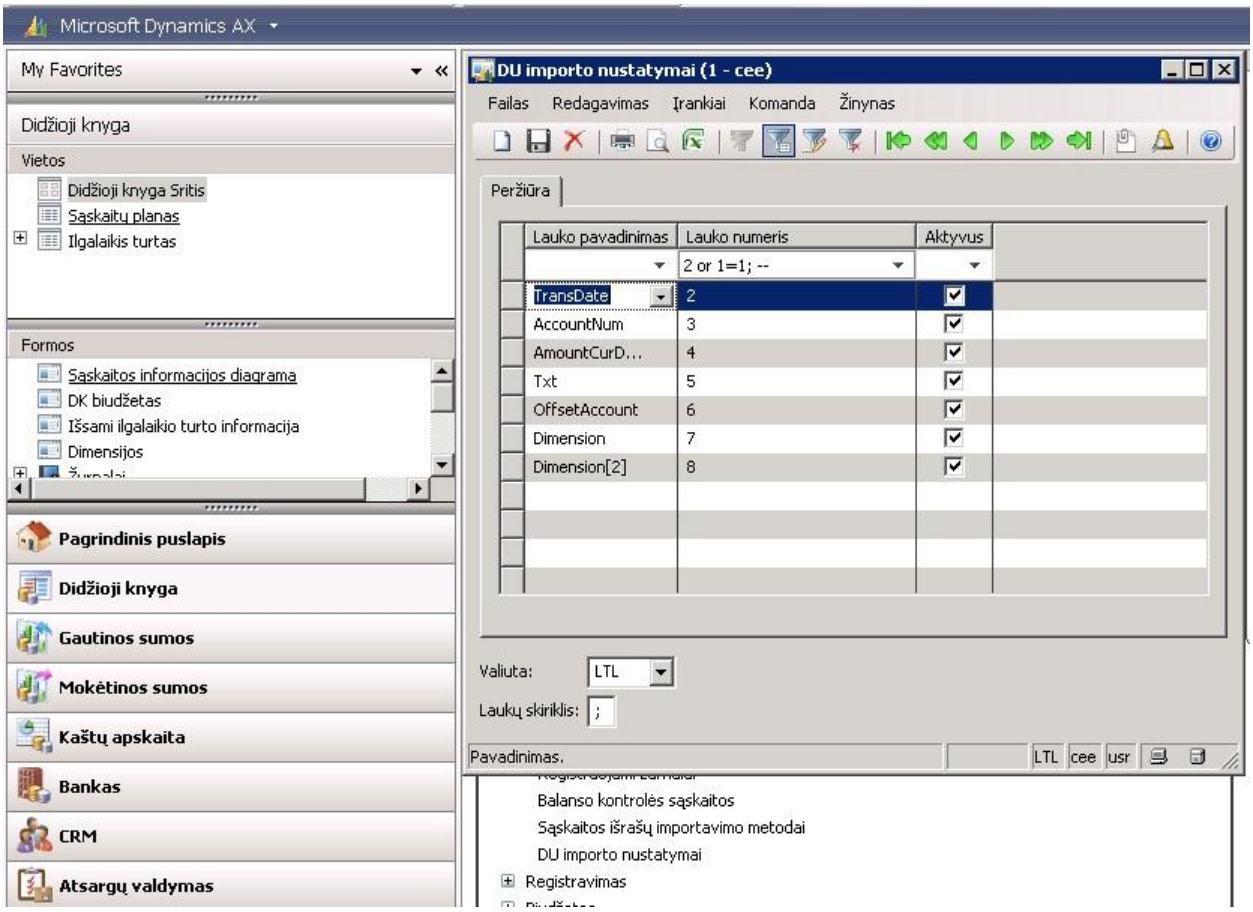
4.1.1 Microsoft Dynamics AX.

Standartinėje MS Dynamics AX versijoje naudojamose formose įterpti *SQL* injekcijos nepavyko. Tačiau tokia galimybė yra reali, kuomet įmonėse pagal kliento pageidavimus yra kuriami nauji sprendimai.

Šios problemos labiausiai priklauso nuo programuotojų žinių ir išsilavinimo. Tai parodo, kad standartinė versija yra suprogramuota laikantis saugių programavimo standartų, bet tai

neapsaugo nuo kitame pasaulio krašte programuojamų atskirų modulių, kurie gali būti pažeidžiami vienokių ar kitokių atakų.

Paveiksliuke nr. 23, matome pavykusią įvykdyti *SQL* injekciją. Tam panaudojome į filtro laukelį įvestą eilutę: *2 or 1=1; --* . Tokiu būdu atfiltruojami visi užklausos duomenys. Užklausa nėra kenksminga, tačiau ji parodo *SQL* injekcijos atakos pažeidžiamumą. Tokiu būdu galima net ir pašalinti duomenis laikomus duomenų bazėje, pašalinti lentelę, tačiau to gali būti išvengiama pasinaudojant vartotojui suteikiamomis duomenų bazės manipuliavimo teisėmis.



23 pav. MS Dynamics AX atakavimo eksperimentas

Iš paveikslo nr. 23 įsitikinome, įmanomos *SQL* injekcijų atakos ir gerai pasaulyje žinomose taikomosiose sistemose.

4.1.2 Sandėlio valdymo sistema „X“.

Panašią ataką pabandėme atlikti ir vienoje iš sandėlio valdymo sistemų, plačiai paplitusių Lietuvoje. Išsirenkame atsitiktinę formą ir pabandome įvykdyti ataką filtre, panaudojant tokią įvestį: *WHD') or 1=1;* -- . Rezultatas taipogi akivaizdus (žr. pav. 24).

24 pav. Sandėlio valdymo sistemos „X“ atakavimo eksperimentas

Šiomis atakomis sistemai nepakenksime, tačiau įvedus šalinimo ar modifikavimo sakinius, galima lengvai pakoreguoti svarbius duomenis.

4.1.3 Eksperimento rezultatai

Šis eksperimentas parodė, jog ir plačiai naudojamos taikomosios sistemos turi saugumo spragų. Jos gali būti tiek standartinėse versijose, tiek naujuose sistemų funkcionalumuose. Tyrimus atliksime su pačių realizuota eksperimentine saityno taikomąja sistema, siekiant ištirti mūsų siūlomos įterptinės apsaugos priemonės galimybes.

4.2 Tyrimo dalys

Realizuotos įsilaužimų į taikomąsias sistemas aptikimo sistemos tyrimą išskirsime į šias dalis:

- ✓ Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinės realizacijos atremiamumo tyrimas, panaudojant aprašytus atakų šablonus, bei pasirinktą saityno taikomųjų sistemų pažeidžiamumų analizavimo įrankį;
- ✓ Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinės realizacijos tyrimas apkrovos testu (angl *load test*);
- ✓ Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinės realizacijos tyrimas streso testu (angl. *stress test*).

4.3 Įterptinės sistemos atakų atremiamumo tyrimas

4.3.1 Naudojant žinomų atakų šabloną

Šiame tyrime naudosimės mūsų sudarytu atakų šablonu, kuris sudarytas remiantis OWASP [2] surinktų pažeidžiamumų informacija. Tiriamas atsparumas nuo šių atakų:

- ✓ *SQL* injekcijos;
- ✓ Išliekančios (angl. *stored*) *XSS* atakos;
- ✓ Išliekančios *CSRF* atakos.

Tyrimo eiga:

- ✓ Saityno eksperimentinės realizacijos taikomoji sistema patikrinama naudojant atakų šabloną su įterptine įsilaužimų į taikomąsias sistemas aptikimo priemone;
- ✓ Aprašomi gauti rezultatai abejais aprašytais atvejais.

Akivaizdu, jog nėra tikslinga atlikti šio tyrimo be įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės, kadangi visos atakos bus įvykdytos.

Tyrimas su įterptine įsilaužimų į taikomąsias sistemas aptikimo priemone.

Realizacijai pasirinktas elementarus saityno taikomosios programos GUI, kuriame yra įvedimo ir paieškos laukeliai (žr. 25 pav.).

25 pav. Eksperimentinės realizacijos GUI

Pasinaudojus šiais įvedimo laukeliais atakuojame taikomąją saityno sistemą sudarytu atakų šablonu. Aprašytos atakos: *SQL* injekcijos, *XSS* ir *CRSF*.

Rezultatai pateikiami 8 lentelėje.

Rezultatai.

8 lentelė. Testavimo rezultatai su atakų šablonu

Bendra statistika	Įvykių bazės įrašai be *IP*	Įvykių bazės įrašai su *IP*	
Duomenų bazės parametruose užregistruotas aptiktų pažeidžiamumų kiekis: **BAD_SQL_TOTAL: 100**	***id, createdate , ruleid, ip***	***id, createdate , ruleid, ip***	***id***
	5 ,'2012-04-26 11:04:30',77,NULL	16 ,'2012-04-26 11:09:21',77,2	16
	6 ,'2012-04-26 11:04:42',82,NULL	17 ,'2012-04-26 11:09:32',82, 2	
	7 ,'2012-04-26 11:05:18',84,NULL	18 ,'2012-04-26 11:09:44',84, 2	
	8 ,'2012-04-26 11:05:26',87,NULL	19 ,'2012-04-26 11:09:51',87, 2	
	9 ,'2012-04-26 11:05:33',83,NULL	20 ,'2012-04-26 11:09:59',83, 2	
	10 ,'2012-04-26 11:05:38',89,NULL	21 ,'2012-04-26 11:10:06',89, 2	
	11 ,'2012-04-26 11:05:39',88,NULL	22 ,'2012-04-26 11:10:17',88, 2	
	12 ,'2012-04-26 11:05:46',75,NULL	23 ,'2012-04-26 11:10:23',75, 2	
	13 ,'2012-04-26 11:05:48',71,NULL	24 ,'2012-04-26 11:10:29',71, 2	
	14 ,'2012-04-26 11:07:02',87,NULL	25 ,'2012-04-26 11:10:36',87, 2	
	15 ,'2012-04-26 11:07:10',86,NULL	26 ,'2012-04-26 11:10:45',86, 2	

8 lentelėje pateikti gauti rezultatai tiek su įjungtu juodųjų sąrašų bazės funkcionalumu, tiek su išjungtu. Visos atakos buvo aptiktos, pagal aprašytas taisyklių bazėje laikomas taisykles.

4.3.2 Naudojant saityno pažeidžiamumų analizavimo įrankį

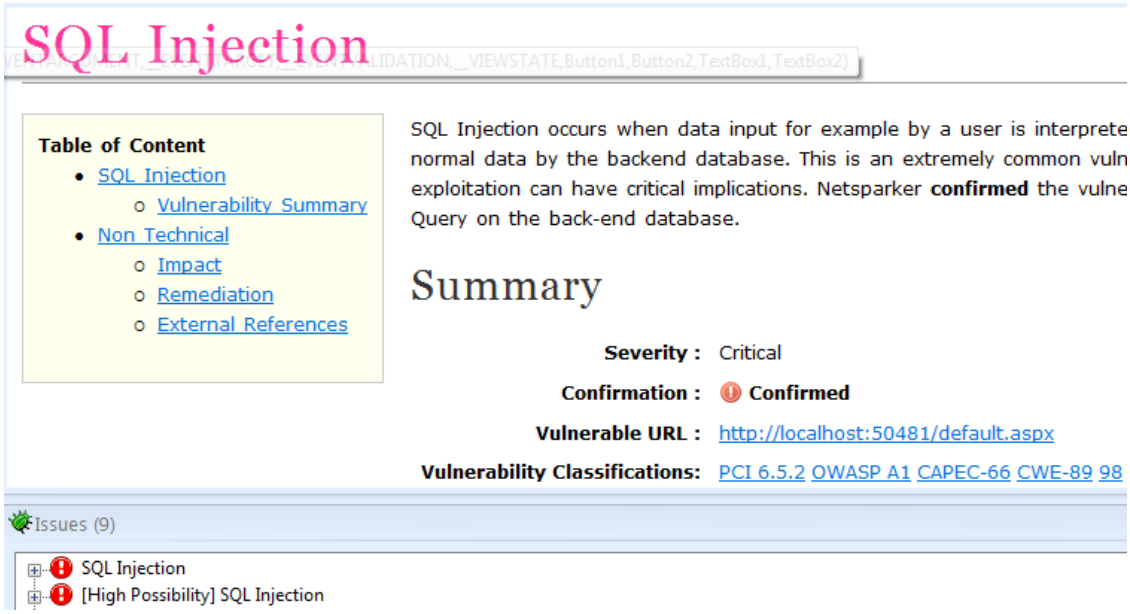
Pažeidžiamumams analizuoti buvo pasirinktas *NETSPARKER* – vienintelis saityno taikomųjų sistemų saugos analizavimo įrankis, kuriame naudojamas vidinis pažeidžiamumų išnaudojimo variklis teigiamai patvirtininantis apie pažeidžiamumus (angl. *false-positive free*) [31].

NETSPARKER privalumai:

- ✓ Paprastas naudoti;
- ✓ Plačios įvairių saityno taikomųjų sistemų pažeidžiamumų aptikimo galimybės;
- ✓ Pateikiama išsami informacija apie atrastus saityno taikomosios programos pažeidžiamumus;
- ✓ Kaip teigiama, *NETSPARKER SQL* injekcijų analizavimo įrankis yra vienas iš geriausių komercinių, nemokamų ir atviro kodo saityno pažeidžiamumų analizavimo įrankių, pagal šaltinį [32] testavimuose aptinkantis net 98.53% visų *SQL* injekcijų atakų.

Tyrimas be įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės.

Pasinaudodami *NETSPARKER* įrankiu, testuojam neapsaugotą saityno taikomąją sistemą.

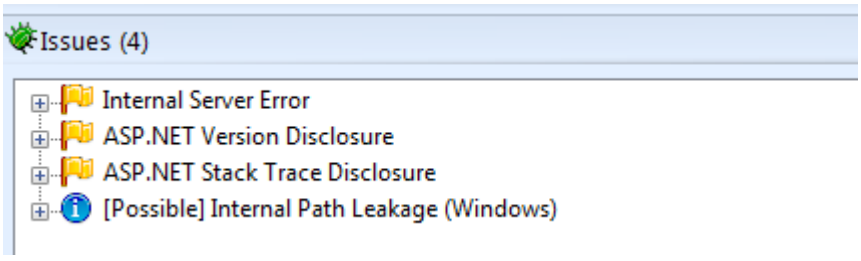


26 pav. Tyrimas be apsaugos, panaudojant pažeidžiamumų analizavimo įrankį

Iš 26 paveikslo matome, jog atlikus testą aptiktos kritinės *SQL* injekcijų atakų galimybės, atveriančios kelius į duomenų pasisavinimą, šalinimą ar modifikavimą.

Tyrimas su įterptine įsilaužimų į taikomąsias sistemas aptikimo priemone.

27 paveiksle pateikiami rezultatai, testuojant saityno taikomąją sistemą panaudojus įsilaužimų į taikomąsias sistemas aptikimo priemonę.



27 pav. Tyrimas su apsauga, panaudojant pažeidžiamumų analizavimo įrankį

Atlikus tyrimą su įterptine įsilaužimų į taikomąsias sistamas aptikimo priemone, panaudojant pažeidžiamumų analizavimo įrankį, nebuvo aptikta *SQL* injekcijų pažeidžiamumų.

Gauti įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės rezultatai pateikti 9 lentelėje.

Rezultatai.

9 lentelė. Testavimo rezultatai su pažeidžiamumų analizavimo įrankiu

Bendra statistika	Įvykių bazės įrašai be *IP*
Duomenų bazės parametruose užregistruotas aptiktų pažeidžiamumų kiekis: **BAD_SQL_TOTAL: 28** Atakų aptikimas: 100%	***id, createdate , ruleid, ip*** 86,"2012-04-27 02:30:26",75,NULL 87,"2012-04-27 02:30:26",75,NULL 88,"2012-04-27 02:30:26",84,NULL 89,"2012-04-27 02:30:26",81,NULL 90,"2012-04-27 02:30:26",82,NULL 91,"2012-04-27 02:30:26",82,NULL 92,"2012-04-27 02:30:26",75,NULL 93,"2012-04-27 02:30:26",84,NULL 94,"2012-04-27 02:30:26",82,NULL 95,"2012-04-27 02:30:26",81,NULL 96,"2012-04-27 02:30:26",81,NULL 97,"2012-04-27 02:32:06",84,NULL 98,"2012-04-27 02:32:06",81,NULL 99,"2012-04-27 02:32:06",82,NULL 100,"2012-04-27 02:32:06",82,NULL 101,"2012-04-27 02:32:06",84,NULL 102,"2012-04-27 02:32:06",82,NULL 103,"2012-04-27 02:32:06",81,NULL 104,"2012-04-27 02:32:06",82,NULL 105,"2012-04-27 02:32:06",81,NULL 106,"2012-04-27 02:32:06",82,NULL 107,"2012-04-27 02:33:46",75,NULL 108,"2012-04-27 02:33:46",84,NULL 109,"2012-04-27 02:33:46",82,NULL 110,"2012-04-27 02:33:46",81,NULL 111,"2012-04-27 02:33:46",82,NULL 112,"2012-04-27 02:33:46",81,NULL 113,"2012-04-27 02:33:46",82,NULL

9 lentelėje pateikti įvykių bazės įrašai, parodantys apie aptiktas atakas. Pažedžiamumų analizavimo įrankis, bei duomenų bazės įrašų patikrinimas leidžia daryti išvadą, jog mūsų

įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė sugebėjo apsaugą 100% nuo *SQL* injekcijų atakų.

10 lentelė. Atakų atremiamumo tyrimo rezultatai

Tyrimas	Atakos	Taikant/netaikant įterptinę priemonę	Atakos aptiktos/neaptiktos	Registravimas Įvykių Bazėje
Naudojant žinomų atakų šabloną	*SQL* injekcijos	Netaikant	Neaptiktos (visos)	Neįvyko
	XSS atakos *CSRF* atakos	Taikant	Aptiktos (visos)	Įvyko
Naudojant saityno pažeidžiamumų analizavimo įrankį	*SQL* injekcijos	Netaikant	Neaptiktos (visos)	Neįvyko
		Taikant	Aptiktos (visos)	Įvyko

10 lentelėje pateikiame atakų atremiamumo tyrimo rezultatus. Buvo atremtos visos *SQL* injekcijų atakos, o naudojant žinomų atakų šabloną, buvo aptiktos ir atremtos išliekančios *XSS* ir *CSRF* atakos.

4.4 Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinės realizacijos tyrimas apkrovos testu.

Šiam tyrimui panaudosime *NEOLOAD* įrankį [34]:

- Tai apkrovos testavimo programinis produktas, skirtas saityno taikomosioms sistemoms, vartotojo veiksmų simuliacijai bei serverių elgsenos analizavimui;
- Leidžia greitai, efektyviai ir dažnai testuoti saityno taikomąsias sistemas;
- Prieinamas pilnas sistemos funkcionalumas vienam mėnesiui.

Eksperimentinę realizaciją testuosime su skirtingais kiekiais virtualių vartotojų (5, 7 ir 10), nustatytai 2 minučių trukmei. Atlikami veiksmai apkrovos testavimo režime:

- Informacijos įvedimas;
- Informacijos atnaujinimas;
- Informacijos paieška.

Apkrovos tyrimas be įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės.

11 lentelėje pateikti gauti apkrovos rezultatai, nenaudojant įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės.

11 lentelė. Apkrovos testavimo tyrimo rezultatai be įterptinės sistemos

Virtualių vartotojų	Vidutinis užklausų atsakymo laikas (*s*)	Vidutinis puslapių atsakymo laikas (*s*)	Vidutinis paspaudimų kiekis (*paspaudimai/s*)
5	0,039	0,048	0,8
7	0,046	0,055	1,2
10	0,050	0,062	1,8

Apkrovos tyrimas su įterptine įsilaužimų į taikomąsias sistemas aptikimo priemone.

12 lentelėje pateikti gauti apkrovos rezultatai, naudojant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę.

12 lentelė. Apkrovos testavimo tyrimo rezultatai su įterptine sistema

Virtualių vartotojų	Vidutinis užklausų atsakymo laikas (*s*)	Vidutinis puslapių atsakymo laikas (*s*)	Vidutinis paspaudimų kiekis (*paspaudimai/s*)
5	0,041	0,052	0,7
7	0,049	0,058	1,1
10	0,052	0,065	1,6

Pagal gautus rezultatus matome, jog pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę, greitaveika sumažėjo, tačiau skirtumai nėra dideli.

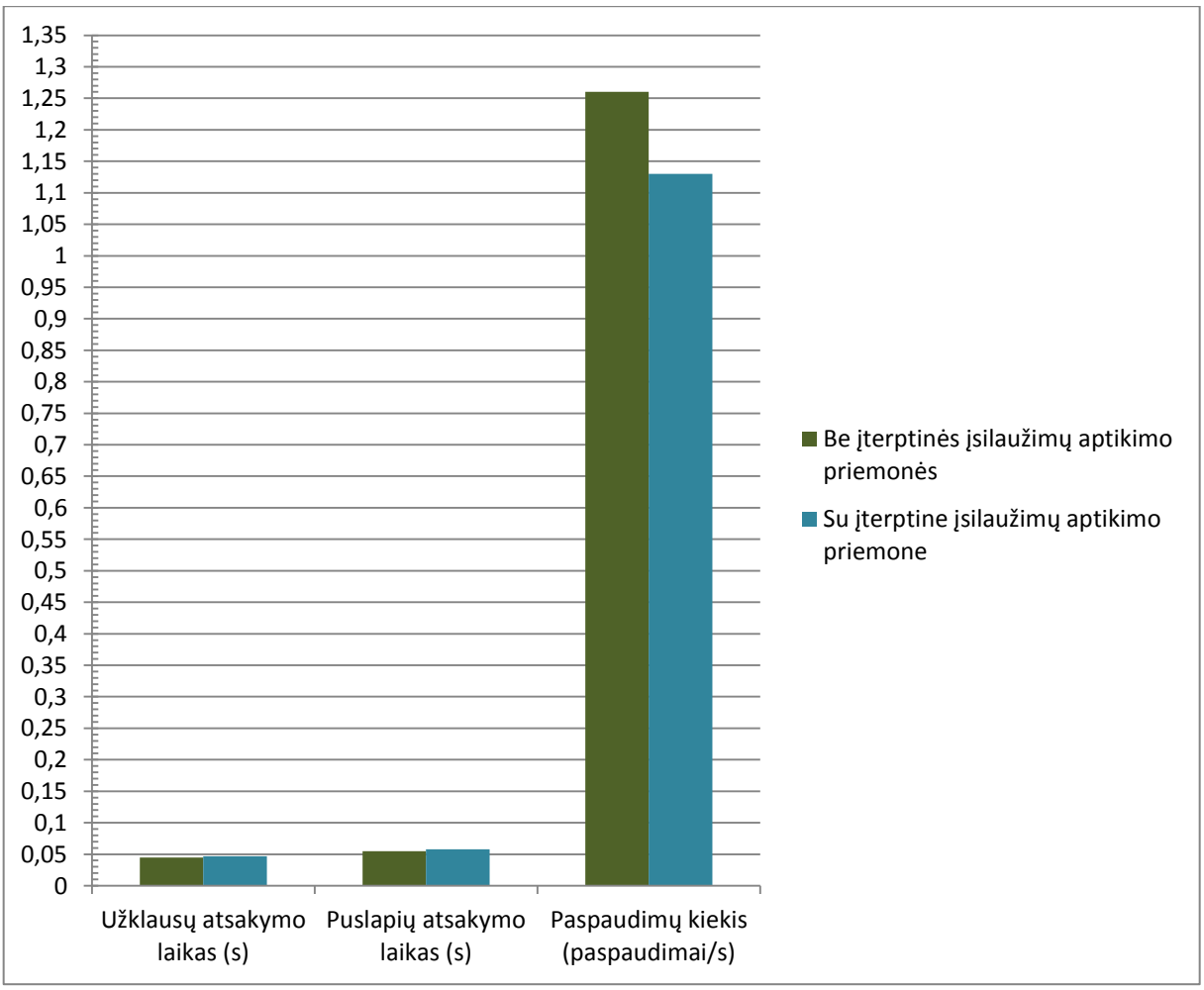
Apkrovos tyrimo rezultatai.

13 lentelėje pateikiame gautus apkrovos rezultatų pokyčius.

13 lentelė. Apkrovos testavimo procentiniai pokyčiai

Virtualių vartotojų skaičius	**5**	**7**	**10**
Užklausų atsakymo laiko pokytis (%) pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę	4,7%	6,1%	4,8%
Vidutinis paspaudimų kiekis skirtumas (%) pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę	12,5%	8,4%	11,2%

Vidutinis užklausų atsakymo laiko pokytis pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę – 5,2%. Vidutinis atliktų virtualių paspaudimų kiekio pokytis – 32,1%.



28 pav. Apkrovos testo vidurkių atvaizdavimas

28 paveiksle pavaizduoti rezultatų vidurkiai tiek su įterptine įsilaužimų į taikomąsias sistemas aptikimo priemone, tiek be: užklausųš atsakymo laikas, puslapių atsakymo laikas, paspaudimų kiekis.

4.5 Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės eksperimentinės realizacijos tyrimas streso testu.

Streso testą naudojame ištirti įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės ir tarnybinės stoties galimybių ribas, esant stresinėms sąlygoms. Streso testo programoje *WAPT* [35] nustatomi parametrai:

- Įvykdoma 100, 500 ir 1000 vartotojo sesijų iki testo pabaigos;

- Kas 30 sekundžių virtualių vartotojų skaičius didinamas po vieną iki maksimalaus galimo skaičiaus 20.

Rezultatai gauti nenaudojant įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės.

14 lentelė. Streso rezultatai nenaudojant įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės

Sesijų skaičius	**100**	**500**	**1000**
Testo trukmė	0:03:52	0:09:26	0:14:10
Užklausų kiekis	475	2310	4530
Užklausų kiekis (per *s*)	2,05	4,08	5,33

Iš 14-os lentėlės matome, jog įvykdytų užklausų kiekis atitinkamai didėdavo su nustatytu sesijų skaičiumi. Vidutinis užklausų kiekio per sekundę pokytis tarp trijų bandymų – 36,6%.

Rezultatai gauti naudojant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę.

15 lentelė. Streso rezultatai naudojant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę

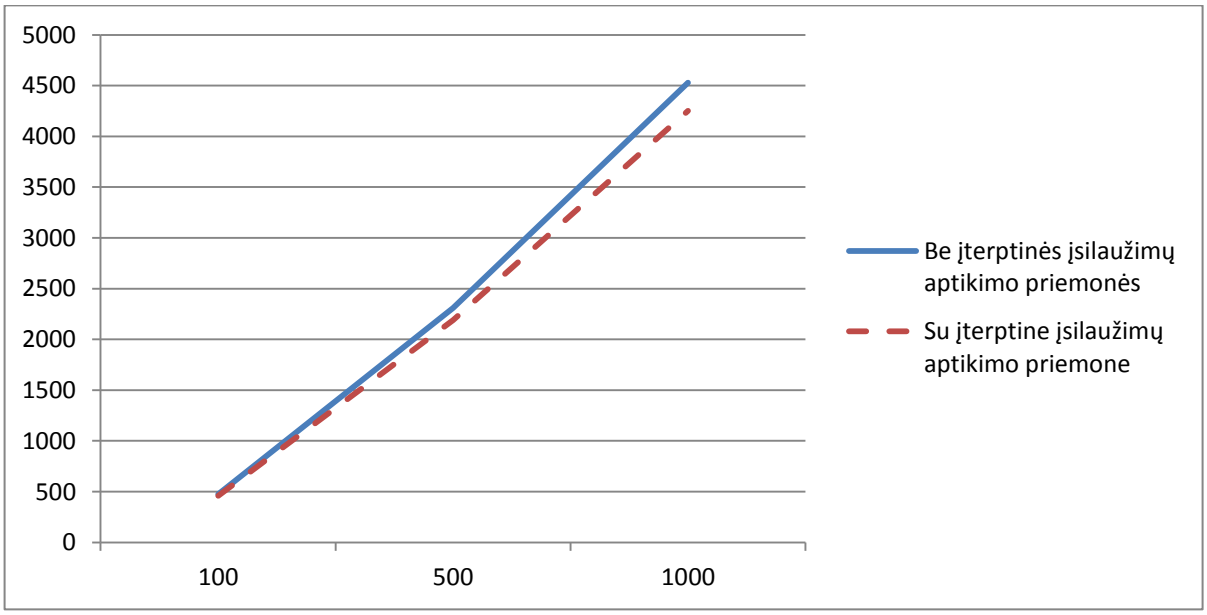
Sesijų skaičius	**100**	**500**	**1000**
Testo trukmė	0:04:10	0:09:32	0:14:08
Užklausų kiekis	460	2190	4250
Užklausų kiekis (per *s*)	1,84	3,83	5,01

Iš 15-os lentėlės matome, kad įvykdytų užklausų kiekis sumažėjo. Kaip ir buvo galima tikėtis, testo trukmė beveik nesiskyrė, o dėl taikomos apsaugos, įvykdytų užklausų kiekis per sekundę neženkliai sumažėjo.

Streso tyrimo rezultatų palyginimas.

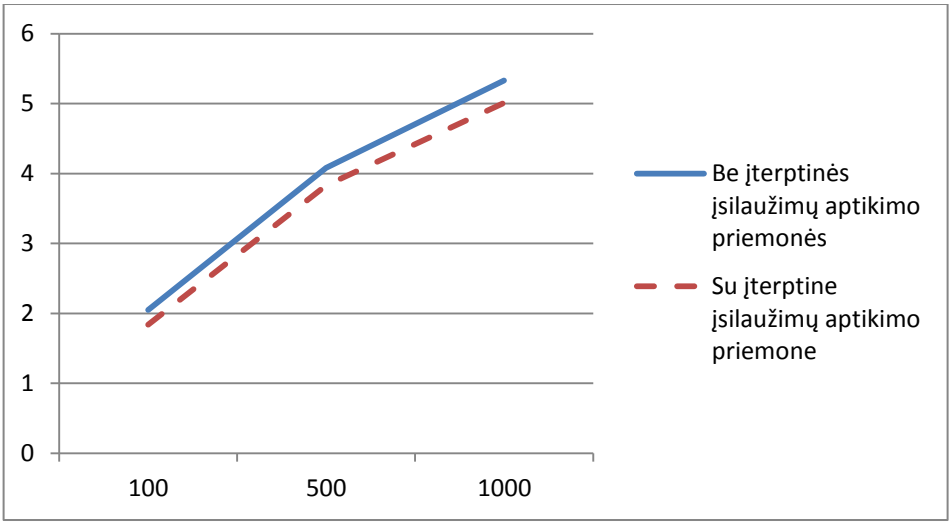
Pateikiame streso testo rezultatų palyginimus taikant ir netaikant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę.

29 paveiksle pateikiame kaip atitinkamai pagal vartotojo sesijų skaičių kito įvykdytų užklausų kiekis.



29 pav. Užklausų kiekio kitimas keičiantis vartotojo sesijų kiekiui

30 paveiksle pateikiame užklausų kiekio per sekundę lyginamąją diagramą, pagal naudotą vartotojo sesijų skaičių.



30 pav. Užklausų kiekio per sekundę kitimas keičiantis vartotojo sesijų kiekiui

Pagal gautus rezultatus 29 ir 30 paveiksluose matome, jog streso tyrime taikant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę akivaizdžiai sumažėjo įvykdytas užklausų kiekis. 16 lentelėje pateikiame procentinį palyginimą, kiek procentų sumažėjo įvykdytas užklausų kiekis ir užklausų kiekis per sekundę, taikant įterptinę įsilaužimų aptikimo priemonę.

16 lentelė. Procentinis užklausų pokytis, pritaikius įterptinę įsilaužimų aptikimo priemonę

Sesijų skaičius	**100**	**500**	**1000**
Užklausų kiekis	-3,2%	-5,2%	-6,2%
Užklausų kiekis (per *s*)	-10,2%	-6,1%	-6%

Pagal 16 lentelės duomenis nustatome, jog vidutinis įvykdytų užklausų kiekio procentinis pokytis - 4,9%. Vidutinis užklausų kiekio per sekundę procentinis pokytis – 7,4%.

Norint užtikrinti apsaugos funcionalumą taikomojoje sistemoje, reikia aukoti nežymų greitaveikos sumažėjimą, kuris yra apie 5%, tačiau taip 100% procentų užtikrinant sistemos saugumą nuo *SQL* injekcijų, išliekančių *XSS* ir *CSRF* atakų.

4.6 Išvados

- Paprastomis *SQL* injekcijų atakomis nustatėme, jog Microsoft Dynamics AX paketas, bei viena iš sandėlio valdymo sistemų „X“ yra pažeidžiamos šių atakų, nors pirmoje taikomojoje sistemoje šie pažeidžiamumai aptikti tik nestandartinėje versijoje, t.y. sistemoje suprogramuotame naujame funkcionalume;
- Atremiamumo tyrimui buvo panaudota tiek atakų šablonas, tiek pažeidžiamumų analizavimo įrankis *NETSPARKER*. Atakų šablonas buvo naudojamas siekiant patikrinti įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę savomis aprašytomis atakomis (remiantis šaltiniais), kad būtų aprėptos atakos nuo kurių skirta apsauga;
- Atremiamumo tyrimas parodė, jog mūsų pasiūlyta įterptinė įsilaužimų į taikomąsias sistemas aptikimo priemonė sugebėjo atremti vykdytus įsilaužimus 100% nuo *SQL* injekcijų, išliekančių *XSS* ir *CSRF* atakų.
- Apkrovos tyrimas parodė, jog pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę, užklausų atsakymo laikas vidutiniškai pailgėjo apie 5,2%. Šiam tyrimui buvo naudojami 5, 7 ir 10 virtualių vartotojų vienu metu.
- Streso tyrimas parodė, jog pritaikius įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę įvykdytų užklausų kiekis vidutiniškai sumažėjo apie 4,9% per 100, 500 ir 1000 vartotojo sesijų, didinant vartotojų skaičių po vieną iki 20 kas 30 sekundžių.

IŠVADOS

- Išanalizavus pagrindines šių dienų taikomųjų sistemų ir duomenų bazių grėsmes, išskyrėme taikomąsias sistemas pagal grėsmių pobūdį į saityno ir lokalias. Kadangi tiek saityno, tiek lokalios taikomosios sistemos dažniausiai kuriamos naudojant duomenų bazes, didžiausias dėmesys skirtas šių sistemų atakų analizei;
- Sukurtas įterptinis įsilaužimų į taikomąsias sistemas aptikimo metodas, apsaugantis duomenų bazes ir išbandytos jo galimybės apsaugant duomenų bazę nuo šių atakų: *SQL* injekcijos, išliekančios *XSS* ir *CSRF* atakos.
- Suprojektuota įsilaužimų į taikomąsias sistemas aptikimo priemonė, kuri patalpinama tarp taikomųjų sistemų ir duomenų bazės, taip apsaugant duomenis nuo kenksmingų veiksmų, bei nuo kitų kenksmingų atakų. Atakoms atpažinti naudojama taisyklių bazė, kurioje reguliariomis išraiškomis aprašomos taisyklės. Juodoji bazė skirta įtartino kenksmingus veiksmus atlikusio vartotojo registravimui ir blokavimui. Įvykių bazė registruoja įvykdytų kenksmingų veiksmų informaciją. Juodosios bazės funkcionalumas realizuotas su galimybe jį įjungti ar išjungti.
- Eksperimentinės realizacijos tyrimas parodė, jog realizuota įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonė:
    a) 100% aptiko vykdytas *SQL* injekcijų, išliekančias *XSS* ir *CSRF* atakas;
    b) Apkrovos tyrimo rezultatai parodė, kad užklausų atsakymo laikas vidutiniškai pailgėjo apie 5,2%. Šiam tyrimui buvo naudojami 5, 7 ir 10 virtualių vartotojų vienu metu;
    c) Streso tyrime buvo siekiama saityno taikomąją sistemą apkrauti kuo didesniais krūviais: buvo naudojamas vis didesnis virtualių vartotojų skaičius ir būtinas įvykdyti vis didesnis vartotojų sesijų kiekis. Streso metu taikant įterptinę įsilaužimų į taikomąsias sistemas aptikimo priemonę, įvykdomų užklausų kiekis sumažėjo apie 4,9% per 100, 500 ir 1000 vartotojo sesijų, didinant vartotojų skaičių po vieną iki 20 kas 30 sekundžių.

TERMINŲ IR SANTRUMPŲ SĄRAŠAS

Santrumpa, terminas		Paaiškinimas
SQL (angl. *Structured Query Language*)	-	struktūrizuota užklausų kalba
DB (angl. *Database*)	-	duomenų bazė
DBVS (angl. *Database Management System*)	-	duomenų bazių valdymo sistema
IS (angl. *Information System*)	-	informacijos sistema
IT (angl. *Information Technology*)	-	informacinės technologijos
IP (angl. *Internet Protocol*)	-	unikalus numeris, identifikuojantis kompiuterį tinkle ir internete
PK (angl. *Pesonal Computer*)	-	personalinis kompiuteris
GUI (angl. *Graphical User Interface*)	-	grafinė vartotojo sąsaja
CASE (angl. *Computer Aided Software Engineering*)	-	integruota IS kūrimo aplinka
Saitynas (angl. *Web*)	-	žiniatinklis, pasaulinis tinklas
Lokali programa (angl. *Stand-alone application*)	-	savarankiška taikomoji programa
Išliekanti (angl. *Stored*)	-	išliekanti duomenų bazėje pvz. *XSS* ar *CSRF* ataka

LITERATŪRA

[1] Street directory, Programming [interaktyvus]. [Žiūrėta 2010-12-29], Prieiga per internetą: <http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html>.

[2] OWASP – Open Web Application Security Project [interaktyvus]. [Žiūrėta 2010-12-29], Prieiga per internetą: < http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project >.

[3] Miscelaneous Security [interaktyvus]. [Žiūrėta 2010-12-29], Prieiga per internetą: <http://misc-security.com/2009/11/04/confidentiality-integrity-availability/>.

[4] Kazanavičius E., Venčkauskas A., Liutkevičius A., Vrubliauskas A. Informacijos Saugos Vadyba : mokomoji knyga. Kaunas: Vitae litera, 2008m. 170 p. ISBN 978-9955-686-72-9.

[5] Internet World Stats [interaktyvus]. [Žiūrėta 2010-12-29], Prieiga per internetą: <http://www.internetworldstats.com/stats.htm>.

[6] 2010 X-Force Mid-Year Trend and Risk Report *August 2010* [interaktyvus]. [Žiūrėta 2010-12-29], Prieiga per internetą: <ftp://public.dhe.ibm.com/common/ssi/ecm/en/wgl03003usen/WGL03003USEN.PDF>.

[7] Herbert H. Thompson, Scott G. Chase. The software vulnerability guide. - Hingham, Massachusetts, CHARLES RIVER MEDIA, INC., ISBN: 1-58450-358-0, 2005 m., 369 p.

[8] National Vulnerability database [interaktyvus]. [Žiūrėta 2011-01-04], Prieiga per internetą: < http://web.nvd.nist.gov/view/vuln/statistics-results?cid=1 >.

[9] Gary McGaw. Software Security: Building Security In. Addison Wesley Professional, January 23, 2006, ISBN-10: 0-321-35670-5, 448 p.

[10] Plėštys R., Rimkus D., Kavaliūnas R., Lagzdinytė I., Sarafinienė N. Kompiuterių tinklų sauga : mokomoji knyga, 2008 m. 187p. ISBN 9789955686705.

[11] Software attacks [interaktyvus]. [Žiūrėta 2011-01-04], Prieiga per internetą: < http://pmaungmaung.blogspot.com/2009/02/software-attacks.html >.

[12] ACUNETIX, Web Application Security [interaktyvus]. [Žiūrėta 2010-01-05], Prieiga per internetą: <http://www.acunetix.com/websitesecurity/xss.htm>.

[13] Testing Security [interaktyvus]. [Žiūrėta 2010-01-05], Prieiga per internetą: <http://www.testingsecurity.com/how-to-test/injection-vulnerabilities/ >.

[14] Information Assurance Technology Analysis Center (IATAC), Revision by Tzeyoung Max Wu. Information Assurance Tools Report – Intrusion Detection Systems. Sixth Edition, September 25, 2009, SPO700-98-D-4002, 298 (Rev. 8-98), p. 93.

[15] Robert Di Pietro, Luigi V. Mancini. Intrusion Detection Systems. 2008, p. 264, ISBN: 978-0-387-77265-3.

[16] OWASP Insecure Direct Object Reference [interaktyvus]. [Žiūrėta 2011-01-25], Prieiga per internetą:
< http://www.owasp.org/index.php/Top_10_2007-Insecure_Direct_Object_Reference >.

[17] OWASP Cross-Site Request Forgery (CSRF) [interaktyvus]. [Žiūrėta 2011-01-25], Prieiga per internetą: < http://www.owasp.org/index.php/CSRF >.

[18] Mark D., John M., Justin S. The Art of Software Security Assessment: Identyfying and Preventing Software Vulnerabilities. – Addison Wesley Professnional, November 10, 2006, ISBN-10: 0-321-44442-6, 1200 p.

[19] Nirali P. Emerging Trends in Information Technology. – Rachana Enterprises, March 2007, ISBN: 8185790884.

[20] D. Liwu, X. Ruzhi, J.Lizheng, L. Guangjuan. A database protection system aiming at SQL attack. 2009 Fifth International Conference on Information Assurance and Security, DOI 10.1109/IAS.2009.322.

[21] X. Ruzhi, G. Jian, D. Liwu. A database security gateway to the detection of SQL attacks. 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 978-1-4244-6542-2.

[22] Rokas Paškevičius. Žiniatinklio programų ugniasienės tyrimas ir sudarymas. Magistro darbas, - KTU, 2010m., 79 p.

[23] Coding Architecure, What is software architecture [interaktyvus]. [Žiūrėta 2011-06-14], Prieiga per internetą:
< http://www.codingthearchitecture.com/pages/book/what-is-software-architecture.html >.

[24] Ron Ben-Natan. Implementing database security and auditing: a guide for DBAs, information security administrators and auditors. 2005, p. 413, ISBN: 1-55558-334-2.

[25] Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. High Performance MySQL. 2008, p. 710, ISBN: 978-0-596-10171-8.

[26] OWASP [interaktyvus]. [Žiūrėta 2011-01-19], Prieiga per internetą: <https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet >.

[27] GreenSQL, Database Security solution [interaktyvus]. [Žiūrėta 2012-04-18], Prieiga per internetą: < http://www.greensql.net/ >.

[28] Microsoft Dynamics ERP Lietuva, Microsoft Dynamics AX [interaktyvus]. [Žiūrėta 2012-04-18], Prieiga per internetą < http://www.microsoft.com/lt-lt/dynamics/erp-ax-overview.aspx >.

[29] Webopedia, embedded system [interaktyvus]. [Žiūrėta 2012-04-19], Prieiga per internetą: < http://www.webopedia.com/TERM/E/embedded_system.html >.

[30] Introduction to Programming Embedded Systems, Department of Computer and Information Science, University of Pennsylvania [interaktyvus]. [Žiūrėta 2012-04-19], Prieiga per internetą: < http://www.cis.upenn.edu/~lee/06cse480/lec-into_to_prog_embedded_systems.pdf >.

[31] Netsparker, Mavituna Security (web application security experts) [interaktyvus]. [Žiūrėta 2012-04-26], Prieiga per internetą: < http://www.mavitunasecurity.com/netsparker/ >.

[32] Security tools benchmarking [interaktyvus]. [Žiūrėta 2012-04-26], Prieiga per internetą: < http://sectooladdict.blogspot.com/2011/08/commercial-web-application-scanner.html >

[33] E. Kazanavičius, R. Paškevičius, A. Venčkauskas. Securing Web Application by Embedded Firewall, Electronics and Electric Engineering, No.3(119), 2012, p. 65-68.

[34] Neotys, NeoLoad [interaktyvus]. [Žiūrėta 2012-04-30], Prieiga per internetą: < http://www.neotys.com/product/overview-neoload.html >.

[35] Web Application Testing, WAPT [interaktyvus]. [Žiūrėta 2012-05-03], Prieiga per internetą: < http://www.loadtestingtool.com/ >.

PRIEDAI

1. Priedas. Diegimo aktas

Diegimo akte nurodoma, kad šiame darbe sukurta programinė priemonė buvo įdiegta kaip saugos modulis plėtojamose interneto svetainės. Diegimo aktas įsegtas darbo gale.

DIEGIMO AKTAS

2012-05-18

Kaunas

Raimundo Stulpino magistrinio darbe „*Įterptinės įsilaužimų į taikomąsias sistemas aptikimo priemonės*“ sukurta programinė priemonė buvo įdiegta kaip duomenų bazės *MySQL* saugos modulis UAB "MEDIA INOVACIJOS" plėtojamose interneto svetainėse.

UAB "MEDIA INOVACIJOS" Direktorius Šarūnas Straševičius

(Parašas) (V. Pavardė)