KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Martynas Adomavičius

„Medaus puodynės“ metodo panaudojimas sudarant perspėjamąją sistemą

Magistro darbas

Darbo vadovas
doc. dr. A. Venčkauskas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Martynas Adomavičius

„Medaus puodynės“ metodo panaudojimas sudarant perspėjamąją sistemą

Magistro darbas

Recenzentas
dr. D. Rimkus
2010-05-26

Vadovas
doc. dr. A. Venčkauskas
2010-05-26

Atliko
IFN-8/3gr. stud.
Martynas Adomavičius
2010-05-26

Kaunas, 2010

Darbo turinys

Using "Honey pot" method in developing a warning system

Summary

As computer applications develope, it is important to ensure computer and their network security in everyday life. This is done by using various instruments: from firewalls, intrusion detection and prevention systems to antivirus programs.

With the development of security technologies hackers evolve too. They are adjusting to new security technologies - examining their operating principles, learning how to pass by them. To be able to strengthen the network or server security, we need to know the latest methods used by hackers. Often, the method used by hacker is sorted out only when losses in server or network are experienced. Hackers should be monitored to collect information about their activities, then administrators should structure and process that information to develope new methods in protecting networks and (creation of new rules for firewalls or intrusion detection systems).

Firewall, intrusion detection and prevention protection system is the typical solution, but their opportunities are limited, these systems are often limited to certain acts excluding a new attack method that is not recognized. Partly to resolve these problems, „Honey pot“ approach is used. „Honey pot“ principle in short: attract and catch. Thus, using of „honey pot“ method in conjunction with firewall and intrusion detection and prevention system one can create a warning system, in which administrators analyzes the information to develope new methods of protection.

Santrauka

Plečiantis kompiuterių pritaikymo galimybėms ir integracijai kasdieniniame gyvenime, svarbu užtikrinti kompiuterių ir jų tinklų saugumą. Tam naudojamos įvairios priemonės pradedant ugniasienėmis, įsilaužimo aptikimo ir prevencijos sistemomis baigiant antivirusinėmis programomis. Besivystant saugumo technologijoms, tobulėja ir įsilaužėliai. Jie prisitaiko prie naujų saugumo technologijų – analizuodami jų veikimo principus, išmoksta jas apeiti. Kad galėtume sustiprinti tinklo ar serverių saugumą, turime žinoti naujausius įsilaužėlių naudojamus metodus. Dažnai įsilaužėlio naudojamas metodas išsiaiškinamas

įvykus įsilaužimui į tinklą ar serverį - patyrus nuostolių. Įsilaužėlį derėtų stebėti, rinkti informaciją apie jo veiksmus, ją susisteminti ir apdoroti, o vėliau panaudoti kuriant naujus metodus tinklų ar serverių apsaugai didinti (naujų taisyklių ugniasienėms ar įsilaužimų aptikimo sistemoms taikymas). Saugant tinklą dažniausiai naudojamos ugniasienės, įsilaužimo aptikimo ir prevencijos sistemos, tačiau jų galimybės ribotos. Dažnai šios sistemos apsiriboja tam tikrų veiksmų blokavimu, esant naujam atakos būdui šis neatpažįstamas. Iš dalies šias problemas gali išspręsti „medaus puodynės“ metodo panaudojimas. „Medaus puodynės“ metodo esmė - įviliok ir pagauk. Taigi panaudojus „medaus puodynės“ metodą kartu su ugniasiene bei įsilaužimo aptikimo ir prevencijos sistema galima sukurti perspėjamąją sistemą, kurioje administratorius analizuoja informaciją bei kuria naujus apsaugos metodus.

1. ĮVADAS

Informacijos rinkimas, apdorojimas, platinimas – glaudžiai susijęs su sparčiu kompiuterinės technikos tobulėjimu. Atsiradus pirmosioms ne mechaniniu pagrindu veikiančioms skaičiavimo mašinoms, tokioms kaip Hermano Holerito gyventojų surašymui 1890 m. sukurtoji skaičiavimo mašina, atsirado poreikis informaciją perduoti kitiems šaltiniams, Holerito atveju – perfokortos buvo naudojamos JAV gyventojų apskaitos vykdymui. Informacijos mainai paspartėjo atsiradus asmeniniams kompiuteriams sujungtiems tarpusavyje. Pradžioje tai buvo vietiniai tinklai sukurti informacijos mainams, vėliau peraugę į galingą sistemą, kuri buvo pavadinta Internetu. Plečiantis kompiuterių tinklams fizinė informacijos sauga tapo ne tiek aktuali, kaip informacijos sauga virtualioje terpėje.

Kiekvieną dieną kompiuteriniais tinklais perduodamas ypatingai didelis kiekis informacijos, kurio vertė kartais nenusakoma pinigais. Perduodamai informacijai turi būti užtikrintas slaptumas, vientisumas, autentiškumas, prieinamumas - tai pagrindiniai kompiuterinių tinklų saugumo uždaviniai. Siekiant užtikrinti kompiuterinių tinklų bei pačių kompiuterių saugumą naudojama didelė aibė saugumo priemonių: virtualūs privatus tinklai (VPN angl. Virtual Private Network), duomenys perduodant šifruojami, naudojama tiek programinė, tiek aparatinė kontrolė, tinkle esantys kompiuteriai naudoja antivirusines programas, užkardas ir kitas kartais sudėtines priemones. Besivystant saugumo technologijoms vietoje nestovi ir kita pusė – įsilaužėliai. Jie prisitaiko prie naujų saugumo technologijų ir analizuodami išmoksta jas apeiti. Kad galėtume sustiprinti tinklo ar serverių saugumą, turime žinoti naujausius įsilaužėlių naudojamus metodus. Dažnai įsilaužėlio naudojamas metodas išsiaiškinamas po fakto – įvykus įsilaužimui į tinklą ar serverį (patyrus nuostolių). Įsilaužėlį derėtų stebėti, rinkti apie jo veiksmus informaciją, ją susisteminti ir apdoroti, o vėliau panaudoti kuriant naujus metodus tinklų ar serverių apsaugai didinti (naujų taisyklių užkardoms ar įsilaužimų aptikimo sistemoms taikymas).

Įsilaužimams aptikti naudojamos įvairios įsilaužimų aptikimo sistemos (angl. Intrusion detection system - IDS) arba integruotų sistemos komponentų įvykių istorijos įrašų analizė. Įsilaužimų aptikimo sistemos pagal įvairius žinomus požymius, dar vadinamus parašais, bando aptikti bet kokią įtartiną veiklą kompiuteryje arba vietiniame tinkle ir informuoja apie tai sistemos administratorių. Įsilaužimo aptikimo sistemos turi parašų bazę, kuri turi būti nuolat atnaujinama, nes vis atsiranda naujų įsilaužimo metodų. Dažnai įsilaužimo aptikimo sistemos gali praleisti kenkėjiškus veiksmus.

Taigi šioje vietoje susiduriama su keliomis pagrindinėmis problemomis: kaip išsiaiškinti naujus įsilaužėlių metodus bei kaip sumažinti nuostolius, kuriuos gali padaryti įsilaužėliai.

Dar viena priemonė, tai įsilaužimo į kompiuterinį tinklą detektorius - „Medaus puodynė“ (angl. Honey pot) Vienareikšmiškai apibrėžti „medaus puodynės“ technologiją yra gana sudėtinga. Šios technologijos realizacijų gali būti kelių tipų saugumo problemoms spręsti ir jos neskirtos vienos konkrečios saugumo problemos sprendimui. „Medaus puodynė“ - tai informacinės sistemos resursai, kurių tikslas atkreipti ir pritraukti įsilaužėlio dėmesį, aptikti bet kokią nelegalią ar neautorizuotą jo veiklą. [24]

Taigi integruojant saugumo priemones kaip ugniasienė, įsilaužimo aptikimo sistema, įvykių scenarijus su keliomis virtualiomis mašinomis, galima sukurti įsilaužimo perspėjimo sistemą, kurios veikimas pagrįstas „medaus puodynės“ metodu.

2. ĮSILAUŽIMO APTIKIMO SISTEMŲ ANALIZĖ

2.1 Įsilaužimo aptikimui naudojamos duomenų apdorojimo technologijos

Analizuojant įsilaužimo aptikimo sistemas svarbūs yra jų naudojami analizavimo metodai. Yra išskiriama nemažai metodų, kuriais apdorojamas duomenų srautas, dažnai naudojami keli metodai kartu: [13, 22]

- Ekspertinės sistemos;
- Parašų analizė;
- Spalvotieji Petri tinklai;
- Būsenos pakitimo analizė;
- Statistinė analizė;
- Neuroniniai tinklai;
- Vartotojo ketinimų identifikavimas;
- Kompiuterio imunologija;
- Mašinos mokymasis;
- Anomalijų aptikimas;
- Euristinė analizė;
- Kiti metodai.

Ekspertinės sistemos

Šios sistemos naudoja prieš tai apibrėžtas taisykles, kurios nusako atakas. Visi su saugumu susiję įvykiai audito metu paverčiami jei-tuomet-kitu atveju (angl. if-then-else) taisyklėmis.

Parašų analizė

Panašus į ekspertines sistemas, tačiau šis metodas naudojasi žinių baze. Metodas perverčia semantinį atakos apibrėžimą į auditui tinkamą formatą. Atakų parašai gali būti randami istorijos įrašuose arba įeinančiame sraute. Šis metodas naudoja abstraktų audito tyrimo duomenų ekvivalentą. Aptikimas realizuotas naudojant pagrindinį teksto sutapatinimo mechanizmą. Šis metodas efektyvus, tačiau ribotų galimybių – sudėtingesnius veiksmus blogai interpretuoja.

Spalvotieji Petri tinklai

Šis metodas naudojamas paverčiant žinių bazės duomenis į grafišką atitikmenį. Su šiomis sistemomis lengva administratoriui įdėti naujus atakos parašus. Tačiau ieškant sudėtingesnės atakos atitikmens dažnai sistema užtrunka ilgiau nei tikėtasi. Ši technika nenaudojama komercijoje.

Būsenos pakitimo analizė

Ataka apibrėžiama kaip būsenų rinkinys, kuris atsiranda įsilaužėliui sukompromitavus sistemą. Pakitimai demonstruojami būsenos pakitimo diagramomis, lyginant su normalaus veikimo sistemos būsena.

Statistinė analizė

Tai dažniausiai naudojamas metodas. Vartotojo ar sistemos elgesys (požymių komplektas) yra išmatuotas naudojant daug kintamųjų tam tikrais laiko periodais. Tokių kintamųjų pavyzdžiui : vartotojo prisijungimo ir atsijungimo vardai, pasiektų failų skaičius per tam tikrą laiką, disko panaudojimas, atminties bei procesoriaus naudojimas. Kintamųjų atnaujinimas galimas skirtingais laiko periodais: pradedant keliomis minutėmis baigiant mėnesiu. Sistemos saugykloje tikrinama ar buvusios reikšmės neperžengia apribojimų veiksmams. Šis modelis patobulintas lyginant vartotojo trumpojo laikotarpio profilį (elgseną) ir ilgojo laikotarpio profilį, taip aptinkant įsilaužimą. Šie elgsenos profiliai reguliariai atnaujinami.

Neuroniniai tinklai

Šis metodas naudoja mokymosi algoritmus, kad išmoktų apie santykius tarp įvesties ir išvesties vektorių ir juos apibendrintų taip, kad ateityje atskirtų naujus santykius tarp įvesties

ir išvesties vektorių. Sistemų, kurios naudoja neuroninių tinklų metodus, pagrindinis tikslas mokytis apie vartotojus esančius sistemoje. Statistiniai metodai iš dalies atitinka neuroninių tinklų metodą. Tačiau neuroninių tinklų metodas pranašesnis tuo, kad gali geriau išreikšti netiesinius santykius tarp dviejų kintamųjų ir automatiškai mokintis. Šis metodas yra tebetobulinamas ir nelabai paplitęs įsilaužimo aptikimo sistemose.

Vartotojo ketinimų identifikavimas

Ši technika sudaro vartotojo elgsenos modelį, kuriame apibrėžiami aukšto lygio veiksmų komplektai sistemoje. Šie duomenys lyginami su audito metu gautaisiais ir pavojaus atveju generuojamas pavojaus signalas.

Kompiuterio imunologija

Metodas kuria normalios sistemos elgsenos modelį, o ne atskiro vartotojo. Šis modelis susideda iš sistemos kreipinių, kuriuos sukuria tam tikras procesas. Atakos, išnaudodamos sistemos pažeidžiamumus, generuoja neįprastus veiksmus (veiksmų atlikimo kelius). Audito metu surenkama informacija apie tinkamą – normalų sistemos veikimą. Tuomet į žinių bazę įdedami visi žinomi gerai veikiančios sistemos kreipiniai ir procesai. Tuomet lyginamas modelis su naujuoju, pavojaus atveju signalizuojamas pranešimas. Šio modelio veikimas pasižymi aukštu produktyvumu. Trūkumas - nesugeba aptikti tinklo paslaugų konfigūracijos saugumo skylių.

Mašinos mokymasis

Tai dirbtinio intelekto technologija, kuri kaupia visą vartotojo įvedamą informaciją ir numato ją kaip „legalią“ veiklą. Tuomet grupuoja šią informaciją vartotojų komandų bibliotekoje ir sudaro komandos tipinę struktūrą, kuri lyginama su įvedamais duomenimis.

Anomalijų aptikimas

Šio metodo esmė – normalios vartotojo ar tinklo elgsenos nustatymas, duomenų apdorojimo metu lyginant su esamuoju elgsenos profiliu aptinkama anomalija ir suformuojamas signalinis pranešimas. Dažnai šis metodas turi ir dalį statistinio metodo, tad puikiai aptinka naujas nežinomas grėsmes. Tačiau šis metodas pasižymi ir dideliu klaidingų perspėjimų skaičiumi.

Euristinė analizė

Uždavinių sprendimo metodu pagrįsta analizė, kai sprendžiant naudojamasi apytiksliais rezultatais, artėjančiais prie priimtino galutinio atsakymo.

2.2 Įsilaužimų aptikimo sistemos

Įsilaužimų aptikimo sistema (toliau ĮAS) stebi tinklo veiklą bei esant reikalui praneša apie tai sistemos ar tinklo administratoriui. Kai kuriais atvejais ĮAS gali atsakyti į tinkle vykstančią anomaliją ar piktybinius veiksmus blokuojant vartotojo IP (angl. Internet Protocol) adresą, nutraukiant teikiamą paslaugą ar atliekant kitus veiksmus. Įsilaužimų aptikimo sistemų yra didelė įvairovė, tad ir įtartino srauto aptikimas galimas skirtingais būdais.

Įsilaužimų aptikimo sistemas galima skirstyti pagal veikimo principą:

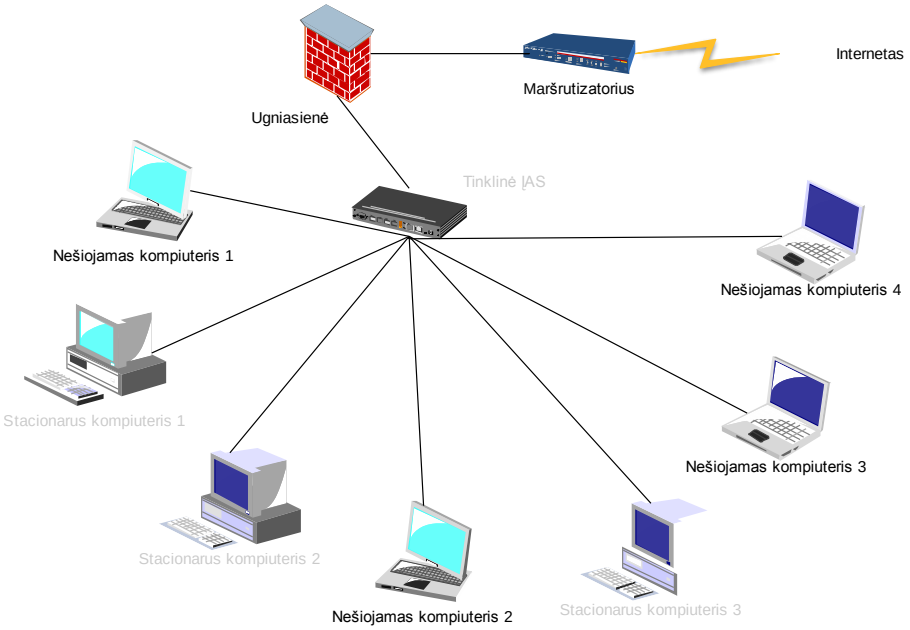
- Atpažįstančios parašus (angl. Signature based intrusion detection system - SIDS);
- Statinė protokolo analizė;
- Aptinkančios anomalijas (angl. Anomaly - based intrusion detection system);
- Mišrios (prieš tai išvardintų metodų derinys).

Kiekvienas įsilaužimas turi savo pėdsaką – t.y. tam tikras savybes kuriomis jis pasižymi ir pagal kurias galima teigti, kad tai būtent tokio tipo įsilaužimas. Šio tipo įsilaužimo aptikimo sistemose saugoma didelė bazė parašų, kuriuos papildo administratorius. Sekant srautą stebima ar tinkle nevyksta tam tikri veiksmai, kurie sutampa su parašų bazėje aprašytaisiais, jeigu vyksta atitinkamai reaguojama į juos. Anomalijas aptinkančios įsilaužimo aptikimo sistemos stebi srautą ir jį rūšiuoja į normalų bei anomalų. Rūšiavimas atliekamas euristiniais ir matematiniais modeliais. [22]

Pagal saugomą sistemą:

- Tinklinės (angl. Network based intrusion detection system NIDS);
- Lokalios (angl. Host based intrusion detection system HIDS);
- Mišrios.

Tinklinės įsilaužimų aptikimo sistemos statomos specialiose tinklo vietose, kad galėtų sekti visą tinklo duomenų srautą. Idealiausias atvejis tai viso srauto sekimas, tačiau tai gali sulėtinti tinklo veikimą, tad geriau sekti tik įeinančius srautus. Tai pademonstruota 1 paveikslėlyje - Tinklinės ĮAS naudojimo pavyzdys. [16]



1 pav. Tinklinės ĮAS naudojimo pavyzdys [16]

Lokalios įsilaužimų aptikimo sistemos diegiamos vartotojo kompiuteryje ir stebi ar nevyksta keisti veiksmai sistemoje: failų sistemos modifikacijos, programų veikimo nukrypimai, darbinės atminties naudojimo pažeidimai ir kita. Šio tipo įsilaužimo aptikimo sistemos dažniausiai naudojamos su tinklo įsilaužimo aptikimo sistemomis, kad padengtų viena kitos trūkumus.[13,16]

Taip pat ĮAS skirstomos pagal protokolą:

- Programos protokolu pagrįstos (Application protocol-based intrusion detection system - APIDS);
- Protokolu pagrįstos (Protocol-based intrusion detection system - PIDS).

Programos protokolu pagrįstos įsilaužimo aptikimo sistemos stebi protokolų naudojimą kompiuterinėse sistemose. Tipinė vieta, kurioje dislokuojama tokia sistema yra tarp žiniatinklio serverio ir duomenų bazės. Dislokuota sistema kaupia informaciją (pėdsakus) taip, kad įvykus įsilaužimui ji informuotų , jog pėdsakas neatitinka gerai veikiančios sistemos pėdsako.

Protokolu pagrįstos įsilaužimo aptikimo sistemos dažniausiai instaliuojamos žiniatinklio serveryje, kad stebėtų ir analizuotų protokolų naudojimą sistemoje (stebi protokolų naudojimą su kitais sujungtais įrenginiais).

Pagal ĮAS pobūdį:

- Aktyvios;
- Pasyvios.

Pasyvios įsilaužimo aptikimo sistemos nesiima jokių veiksmų su aptiktu įsilaužimu, tik informuoja administratorių apie tai. Reaguojančios įsilaužimo aptikimo sistemos aptiktą įsilaužimą užblokuoja (vartotoją ar IP adresą) ar atjungia teikiamą paslaugą ir apie tai informuoja administratorių.

Pagal struktūrą:

- Centralizuotos;
- Paskirstytos (agentai).

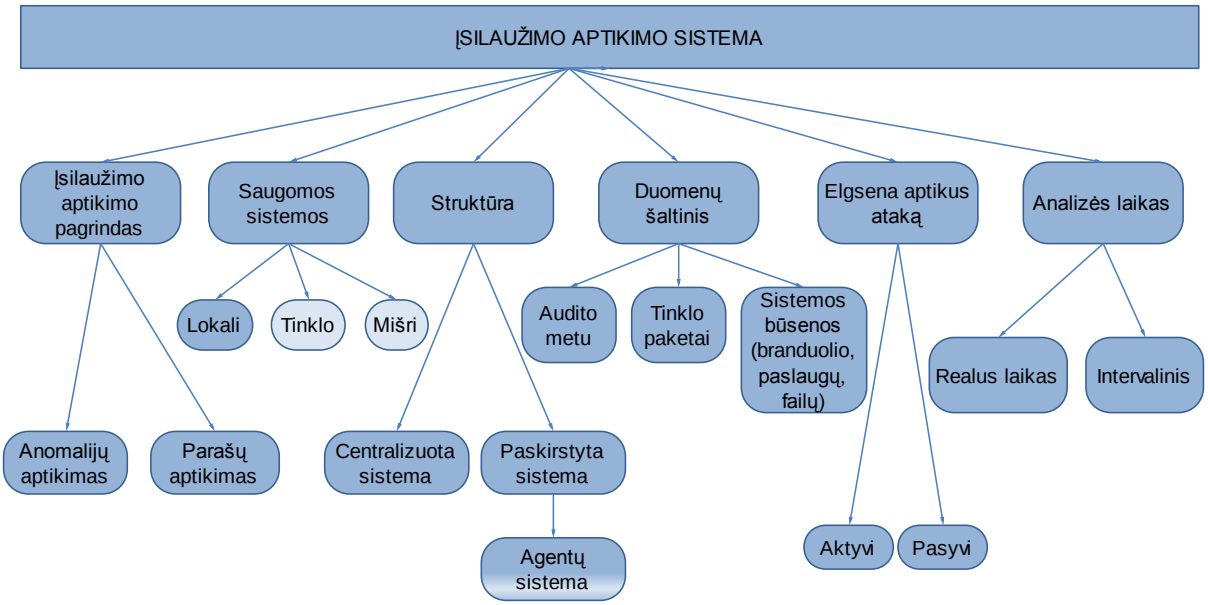
Pagal duomenų šaltinį:

- Gauti atlikus saugumo auditą;
- Tinklo paketai;
- Gauti iš sistemos būsenos analizės (branduolio, paslaugų, failų stebėjimas).

Pagal analizavimo intervalą:

- Realiu laiku;
- Intervaliniai.

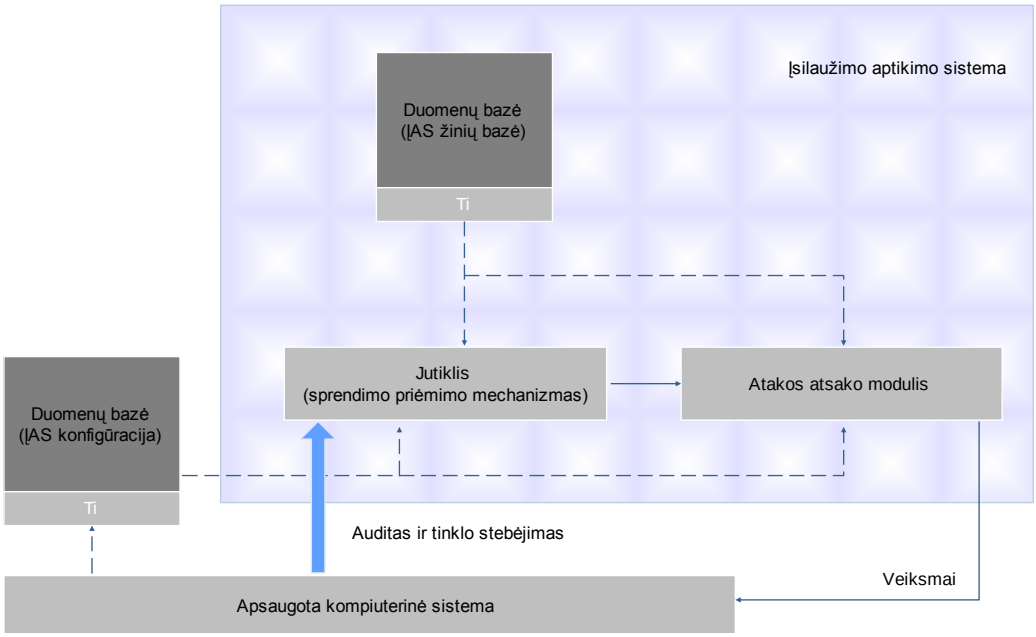
Įsilaužimo aptikimo sistemas galima klasifikuoti įvairiai. Vienas iš klasifikavimo pavyzdžių pateiktas 2 paveikslėlyje.



2 pav. Įsilaužimo aptikimo sistemų klasifikacija [13]

Struktūra ir veikimo principas

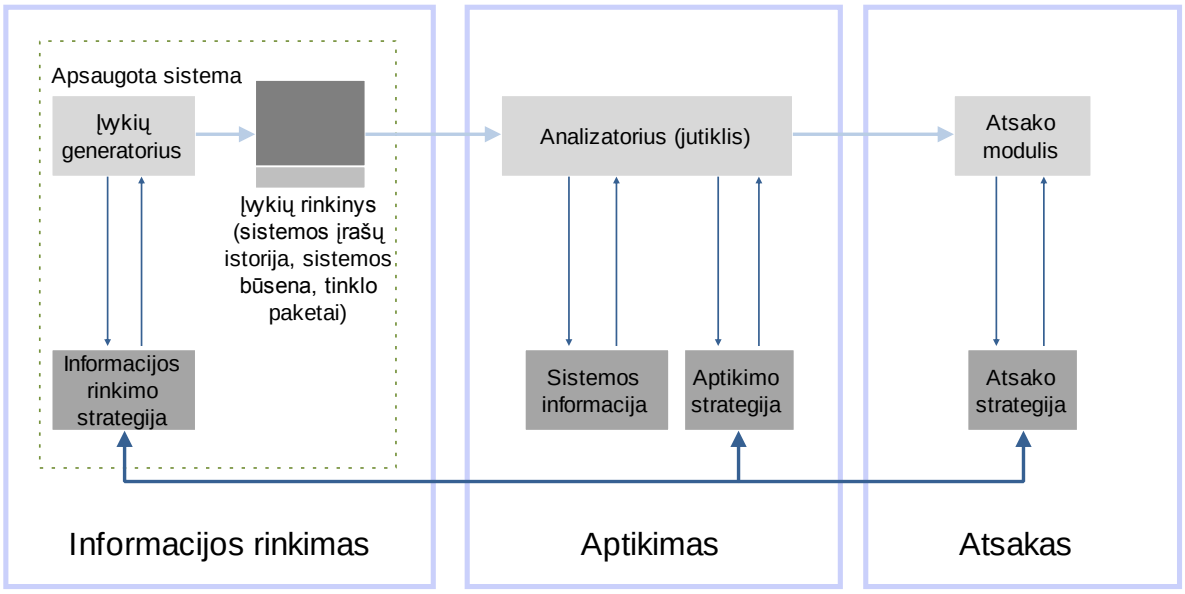
Įsilaužimų aptikimo sistemos struktūra skiriasi priklausomai nuo tipo. Dažnai šios sistemos būna mišrios. Pagrindiniai komponentai visoms įsilaužimo aptikimo sistemoms - tai jutiklis, kuris yra pats sistemos branduolys ir atsakingas už įsilaužimo aptikimą. Šis komponentas priima sprendimą dėl įsilaužimo. Jutiklis gauna neribotą informacijos kiekį iš trijų pagrindinių šaltinių: nuosavos žinių duomenų bazės, sistemos įrašų bei audito duomenų. Sistemos įrašai turi tokius duomenis kaip failų sistemos konfigūracija, vartotojo autorizacijos ir kita. Ši informacija sudaro pirminį sprendimo priėmimo procesą. Bendrai informacijos srautai parodyti 3 paveikslėlyje, kur linijos storis proporcingas informacijos kiekiui. [12]



3 pav. Informacijos srautai įsilaužimo aptikimo sistemoje[12]

Jutiklis integruotas su komponentu, kuris yra atsakingas už duomenų rinkimą. Duomenų rinkimo būdas nustatytas veiksmų generatoriaus politikoje, kuri apibrėžia filtravimo tipą. Įvykių generatorius (operacinė sistema, tinklas, programa) daro šiai strategijai neprieštaraujančią įvykių aibę, kurie gali būti kaupiami įrašuose ar audituojami kaip sistemos veiksmai ar tinklo paketai. Ši informacija gali būti saugoma ir išorinėje saugioje sistemoje. Jutiklio darbas - atmesti teisingus sistemos veikimo įrašus taip aptinkant įtartiną veiklą. Analizatorius naudoja įsilaužimo parašų duomenų bazę, kur palyginęs gautus sistemos įrašus su įsilaužimo įrašais, priima sprendimą. Taip pat duomenų bazėje laikomi ir įsilaužimo aptikimo sistemos konfigūracijos parametrai, kurie apima ir komunikaciją su atsako moduliu. Jutiklis taip pat turi savo duomenų bazę skirtą sudėtingesniems įsilaužimams aptikti.

Pagrindiniai įsilaužimo aptikimo sistemos komponentai ir ryšiai tarpusavyje pateikti 4 paveikslėlyje. [12]



4 pav. Įsilaužimo aptikimo sistemos komponentai [12]

2.3 Esamų įsilaužimo aptikimo sistemų sprendimų lyginamoji analizė

Egzistuoja labai daug įvairių įsilaužimo aptikimo sistemų, tad kiekvienai organizacijai galima pasirinkti įsilaužimo aptikimo sistemą pagal savo specifinius reikalavimus ir galimybes. Kad būtų galima patogiau palyginti įsilaužimo aptikimo sistemas, aktualu informaciją pateikti susistemintą 1 lentelėje. Kai kurios ĮAS, kainos, diegimo, naudojimo įverčiai paimti iš elektroninio žurnalo windowsecurity [12,13] bei papildyta savaisiais.

Dažniausias įsilaužimo aptikimo sistemos pasirinkimo kriterijus yra kaina, tačiau ne visuomet reikia šiuo kriterijumi vadovautis, reikia atkreipti dėmesį ir į funkcionalumą, valdymo patogumą, išplečiamumą, parašų bazės atnaujinimo dažnumą, įsilaužimo aptikimo sistemos tipą, pritaikymą individualiam vartojimui ir kita.
Lyginamos bus atviro kodo įsilaužimo aptikimo sistemos: OSSEC, Snort, Prelude, BRO, Suricata bei mokamos įsilaužimo aptikimo sistemos: Sax2, ELM, GFI LANguard, Cisco Secure IDS, Dragon Enterasys (komercinių įsilaužimų aptikimo sistemų gausybė, tad pasirinktos žinomesnės). Komercinės ĮAS detaliau nebus nagrinėjamos dėl sunkiai pasiekiamos informacijos ir poreikių neatitikimo (pinigine išraiška).

1 lentelė Įsilaužimo aptikimo sistemų palyginimas[1]

	OSSEC	**Snort**	**Prelude IDS**	**BRO**	**Suricata**	**Sax2**	**ELM**	**GFI LAN-Guard**	**Cisco secure**	**Dragon Enterasys**
Tipas	HIDS	NIDS	mišri	NIDS	NIDS	NIDS	HIDS	HIDS	NIDS	NIDS
Kaina (1 m.)	a.k.	a.k.	a.k.	a.k.	a.k.	329$ 1 lic.	~12000 $	4 -32 € 1 IP	~8000 $	~8000 $
Apar./prog.	Prog.	Prog.	Prog.	Prog.	Prog.	Prog.	Apar.	Prog.	Apar.	Apar.
Diegimas	*	**	*	*	**	****	****	***	****	****
Naudojimas	**	***	*	**	**	***	****	***	****	***
Integravimas	***	****	**	**	***	*	*	*	*	*
Atnaujinimai	**	****	**	**	***	****	****	****	****	****
Windows	**	***	**	---	---	***	***	***	***	***
Unix	***	***	***	***	***	---	***	**	***	***
Taisyklių rinkinys	>400	>10 tūkst.	0 Snort	0 Snort	0 Snort	---	---	---	---	---
Taisyklių formatas	XML OSSEC	Snort	Snort	Bro, Snort	Snort	---	---	---	---	---
Teisingi perspėjimai	***	****	***	**	***	****	****	****	****	****

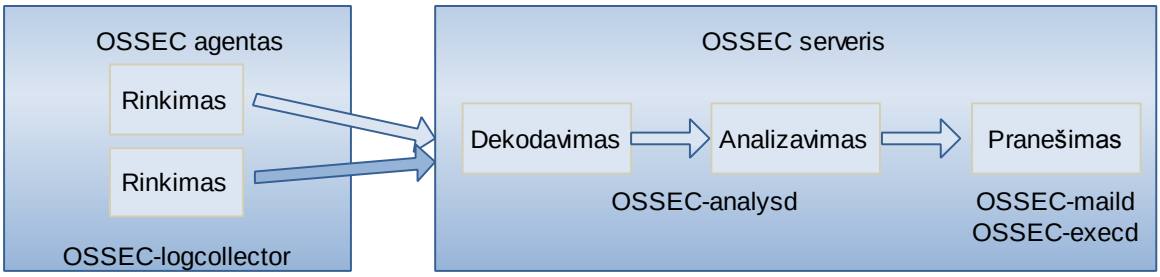
Atviro kodo įsilaužimo aptikimo sistemų ypatybės

OSSEC - pilna platforma stebėti ir kontroliuoti sistemą. Tai mišinys, susidedantis iš visų lokalios įsilaužimo aptikimo sistemos komponentų: įrašų stebėjimo, saugumo informacijos valdymo, saugumo informacijos ir įvykių valdymo modulio. Realaus laiko sistema turi konfigūruojamus perspėjimus bei centralizuotą valdymą. Susideda iš pagrindinės programos, agento ir interneto svetainės. Taigi programa sukurta remiantis kliento-serverio architektūra. Kliento programos renka informaciją, o serverio programa šią informaciją apdoroja perduodama ją 1514 prievadu UDP (angl. User Datagram Protocol) protokolu. OSSEC-logcollector modulis, esantis agento programose, renka informaciją, kurią siunčia į dekodavimo ir analizavimo modulį OSSEC-analysd, esantį serverio programoje, tuomet

[1] Žvaigždutė (*) yra kokybinis matas, didesnis žvaigždučių skaičius reiškia geresnį vertinimą. Maksimalus įvertinimas 5 žvaigždutės. Trumpinių reikšmės: a.k – atviro kodo; Prog. – programinė; Apar. – aparatinė.

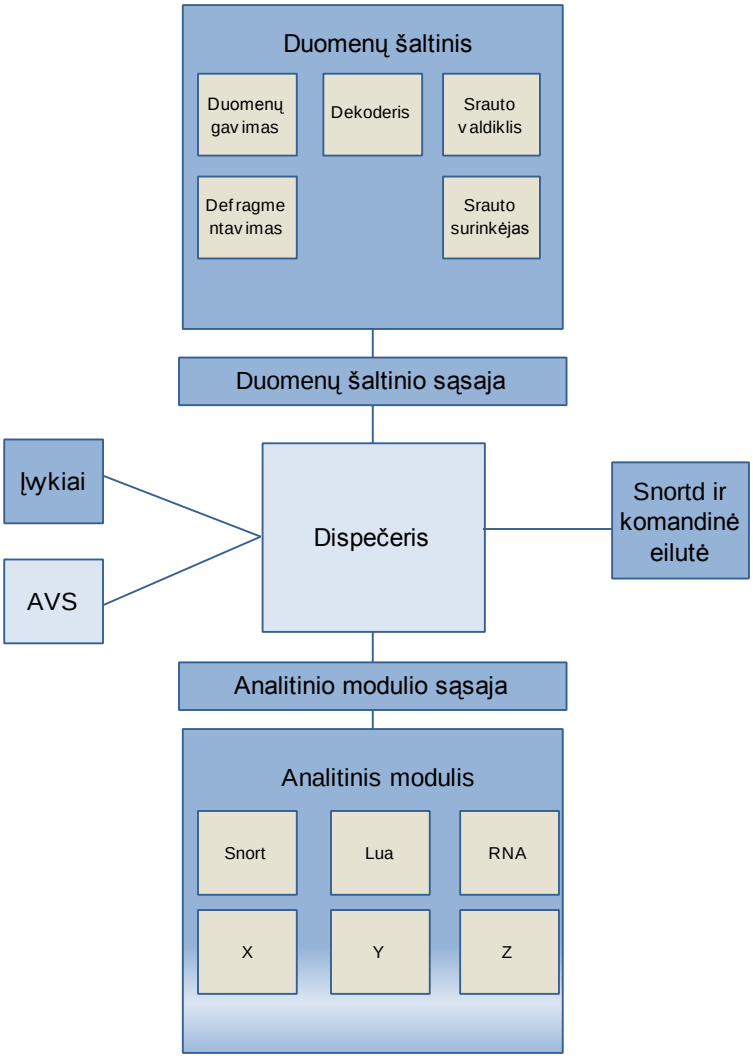
formuojamas pranešimas (OSSEC-maild modulis) ir atsakas (OSSEC-execd modulis). Įrašų istorijos kelias pateiktas 5 paveikslėlyje.

OSSEC turi galingą įrašų analizavimo priemonę, kuri palaiko daugelio įrašų formatus. Taisyklės galimos dviejų tipų: atomic bei composite. Pirmuoju variantu taisyklė orientuota į vieną įvykį, antruoju į daugelį. Taisyklės rašomos XML (angl. Extensible Markup Language) formate. [18]



5 pav. Įrašų istorijos kelias OSSEC sistemoje [18]

Snort - tai atviro kodo atakų aptikimo ir šalinimo sistema skirta kompanijoms bei paprastiems vartotojams, padedanti apsaugoti tinklus nuo įvairaus pobūdžio atakų. Snort - tai realaus laiko paketų peržiūros ir analizavimo įrankis skirtas IP tinklams. Jis gali analizuoti, vykdyti paiešką ir gali būti panaudotas daugeliui atakų aptikti, buferio perpildymui, nematomiems prievadų skenavimams, SMB(angl. Server Message Block) užklausoms, OS (operacinės sistemos) piršto antspaudų užklausimams. Sistema taip pat gali būti panaudota kaip atsakas atakoms jas blokuojant ar pranešant apie tam tikrus įvykius. ĮAS Snort gali dirbti su papildoma programine įranga, skirta grafiškai atvaizduoti atakų grafikus pagal laiką ar kitus kriterijus, kaip SnortSnarf, Sguil, OSSIM ir BASE (angl. BASE Basic Analysis and Security Engine). Snort gali būti integruojamas su automatiniu atakų blokavimu naudojant įvairias ugniasienes: ipfw, ipchains, iptables. Dėl savo lankstumo ji tapusi viena populiariausių pasaulyje įsilaužimo aptikimo sistemų. Ši ĮAS susideda iš 3 pagrindinių dalių: duomenų šaltinio, dispečerio ir analitinio modulio. Sąveikai naudojamos atskiros sąsajos. Yra įvykių sistema, kuri atlieka tam tikrus veiksmus, AVS – atributų valdymo sistema bei ryšys su komandine eilute ir snortd procesu. Snort, Lua, RNA, X,Y,Z - tai srauto analizės moduliai. Bendra Snort architektūra pateikta 6 paveiksliuke. Snort kūrėja – kompanija Sourcefire. [21,23]



6 pav. Snort architektūra [21]

PreludeIDS – renka, normalizuoja, rūšiuoja bei koreliuoja visą informaciją susijusią su saugumu, taip padarydama vientisą formatą dar vadinamą įsilaužimo aptikimo žinučių apsikeitimo formatą. Ši įsilaužimo aptikimo sistema turi daug jutiklių, kurie stebi daugelį sistemos veiksmų, taip užtikrindami detalų informacijos rinkimą. Saugumo informacijos valdymo sistema gali dirbti su AuditD, Nepenthes, NuFW, OSSEC, Pam, Samhain, Sancp bei Snort programinėmis įrangomis. PreludeIDS darbo schema panaši į visų tinklo ĮAS, kur egzistuoja agentai ir serverio programa, kuri koordinuoja visą darbą. Pagrindiniai PreludeIDS komponentai: pagrindinis valdytojas (angl. manager), Libprelude, PreludeDB, Prelude –LML, Prelude- Correlator, Prewika sąsaja, Prelude-PFLoger. Valdytojas - pagrindinis sistemos komponentas, renkantis informaciją iš jutiklių. Libprelude yra biblioteka, kuri garantuoja saugius susijungimus tarp valdytojo ir jutiklių. PreludeDB – saugumo žinučių saugojimui duomenų bazėje skirta sąsaja. Prelude –LML - tai įrašų istorijos analizatorius. Prelude-Correlator - srauto palyginimo su taisyklėmis modulis. Prewika - tai grafinė sąsaja sistemos

vartotojui, kur pateikiama visa informacija ir dar turi kelias papildomas funkcijas (traceroute komanda ir pan.) Prelude-PFLoger modulis siunčia perspėjimus į valdymo modulį.[19]

BRO – Unix šeimos operacinėms sistemoms skirta įsilaužimo aptikimo sistema pasyviai stebinti tinklą ir ieškanti įtartinų veiksmų. Bro aptinka ataką tinklo srauto analizavimo metu, kada išskleidžia veiksmus iki taikomojo lygio semantikos, taip palygindama su kenkėjiškų veiksmų modeliais. Jei aptinkamas įsilaužimas ši sistema realiu laiku generuoja operacinės sistemos komandą, kuri nutraukia ar blokuoja piktybinius veiksmus. Bro tikslas – greitai aptikti atakas dideliame sraute, dėl to naudojamas dideliems srautams sekti. Bro naudoja specializuotą taisyklių kalbą, kuri suteikia galimybę aptikti naujus metodus. Jei sraute aptinkama įdomi elgsena, ji įrašoma į istoriją, kur administratorius gali ją peržiūrėti. Žinoma, kad aptiktų atakas pagrinde Bro remiasi parašų baze bei žinomų įsilaužimų duomenų baze ir standartinio tinklo veikimo modeliu. Kadangi jis naudojamas dideliems srautams sekti, reikalingas pakankamai galingas kompiuteris. Pvz. 5000 paketų per sekundę stebėjimui reikalingas 1 GHz procesorius, 2 GB darbinės atminties, 50 GB kietasis diskas bei trys tinklo sąsajos. Papildomų nustatymų dėka Bro gali naudoti Snort taisykles, tačiau jos bus konvertuojamos į Bro formatą. [5]

Suricata - nors daugelio gijų sąvoka nėra nauja, tačiau įsilaužimo aptikimo sistemose ji yra nauja. Suricata turi ypatybę, kurios veikimas pagrįstas gijų naudojimu. Taip pat turi automatinį protokolo aptikimą, Gzip išarchyvavimą bei greitą adresų atitikimą. Suricata - naujos kartos įsilaužimo aptikimo sistema, nepakankamai ištestuota, nes pasirodė 2009 metų gale. Planuojamas susiejimas su aparatūrine įranga. Tinklo paketams paimti naudojamas LibPcap. Ši ĮAS turi integruotą HTTP (angl. Hyper - Text Transfer Protocol) užklausų įrašų konvertavimą į apache įrašų formatą. Kūrėjai - Open Information Security Foundation (OISF). Kadangi ši ĮAS yra visiškai nauja, pateiktos informacijos apie ją sunku rasti. [25]

2.4 „Medaus puodynės“ metodas

Tradiciškai kovojant su informacijos vagystėmis naudojamas vartotojo autentiškumo nustatymas. Vartotojo teisės prie tam tikro resurso gali būti nustatomos įvairiais būdais, pradedant vartotojo vardo ir slaptažodžio įvedimu baigiant asmens biometrinių duomenų panaudojimu. Įsilaužimo aptikimo sistemos pagal tam tikrus vartotojo veiksmų požymius (dar vadinamus parašus) bando aptikti įtartiną veiklą kompiuteryje ar vietiniame tinkle bei praneša apie tai sistemos administratoriui. Parašu gali būti tiek besikartojantis slaptažodžio įvedimas, tiek atvirų prievadų žvalgymas ir kita tam tikromis taisyklėmis apibrėžta veikla. Dauguma

įsilaužėlių naudoja panašius būdus įsilaužimui į tinklus ar kompiuterius, tai ir jų paliktos „pėdos“ bus tokios pačios ir gerai pažįstamos ugniasienėms bei antivirusinėms sistemoms. Kad šios sistemos gerai veiktų, jos turi turėti visą įsilaužimo parašų duomenų bazę bei nuolatos ją atnaujinti. Tradiciniai saugos metodai padeda apsaugoti informaciją, tačiau jie neefektyvūs kai siekiama išsiaiškinti įsilaužėlio veiksmų eiliškumą, motyvus ir mąstyseną. Šiuos metodus galima vadinti pasyviais ar gynybiniais. Visgi įveikti piktavalius išvardintais būdais nėra taip lengva, norint apsaugoti informaciją, būtina imtis papildomų priemonių. Dar vienas metodas kovai su informacijos vagystėmis - tai informacinių sistemų apsaugos priemonės, naudojančios spąstus įsilaužėliams ar brukalo siuntėjams, kitaip dar vadinamos „medaus puodynėmis“ (angl. honey pot). Sąvoka pirmą kartą panaudota Cliff Stoll ir Bill Cheswick knygose.[4,9,24]

Vienareikšmiškai apibrėžti „medaus puodynės“ technologiją yra gana sudėtinga. Šios technologijos realizacijų gali būti kelių tipų saugumo problemoms spręsti ir jos neskirtos vienos konkrečios saugumo problemos sprendimui. „Medaus puodynė“, tai informacinės sistemos resursai, kurių tikslas atkreipti dėmesį ir pritraukti įsilaužėlį, aptikti bet kokią nelegalią ar neautorizuotą veiklą, pavyzdžiui, prie vietinio tinklo prijungtas kompiuteris, kuriame patalpinta tam tikra duomenų bazė ir programa, prie kurios jungiamasi norint skaityti, keisti, sukurti naują įrašą duomenų bazėje. Šiame kompiuteryje palikta šiek tiek daugiau saugumo spragų, tačiau ne per daug, kad nesuprastų įsilaužėlis, jog tai spąstai. Taip pat įdiegta visa programinė įranga, kuri turėtų būti tokiame duomenų bazę ir žiniatinklio svetainę palaikančiame serveryje, taip pat papildomai įdiegiama nematoma sekimo programa. Bet koks bandymas prisijungti prie šios tarnybinės stoties bus bandymas įsilaužti.[2,24]

Taikant „medaus puodynę“ nebūtina pastatyti tikro kompiuterio ir prijungti jį prie interneto, galimi įvairūs variantai: elektroninio laiško formato „medaus puodynės“, programa – „medaus puodynė“ arba tiesiog prievadų monitorius ar kitokia realizacija. Taip pat galima sukonfigūruoti virtualų kompiuterį, kuris imituotų realaus kompiuterio darbą. [9,24]

Pavyzdžiui, elektroninio laiško siuntimas žinomam adresatui, kuriame pateikiama informacija apie svarbius firmos duomenis, patalpintus tam tikrame puslapyje, kuris yra tik „medaus puodynė“. Laišką turi skaityti tik siuntėjas ir gavėjas, norint įsitikinti, kas dar skaito atsiųstą laišką, reikia po laiško išsiuntimo stebėti puslapio prisijungimų įrašus, nes kiekvienas egzistuojantis prisijungimas rodo nesankcionuotą laiško skaitymą. [9,24]

„Medaus puodynių“ pagalba efektingai kovojama su brukalo siuntėjais. Projektas „Medaus puodynė“, sumanytas Čikagos teisininko Matthew Prince, yra vienas iš sėkmingų bandymų apsaugoti žmonių privatumą elektroninėje erdvėje. Jo sugalvotu metodu sučiuptas brukalo siuntėjas, kuris per diena išsiųsdavo apie 10 milijonų nepageidaujamų laiškų. Šio metodo esmė paprasta: brukalo siuntėjai naudojasi specialia programine įranga, kuri šniukštinėja po įvairiausius puslapius ir renka elektroninio pašto adresus, kuriais vėliau siunčiami nepageidaujami laiškai. Puslapio savininkas, įdiegęs specialią „medaus puodynės“ programinę įrangą, gali fiksuoti tokių adresų ieškotojus ir jiems pateikti ne tikrus puslapyje esančius elektroninius adresus, o daugybę specialiai tam skirtų elektroninių adresų, bei užfiksuoti adresų ieškotojo IP adresą. Gavus nepageidaujamą laišką į nurodytus paštus, jau turima įrodymų ir galima kelti bylą brukalo siuntėjui.[20,24]

Norint surinkti daugiau informacijos galime realizuoti net ir kompiuterinio tinklo mastu, sudarant galimybę įsilaužėliui bandyti patekti į dirbtinai padarytą kompiuterių tinklą, kuris yra didelė „medaus puodynė“. Derėtų žinoti, jog kai kurios „medaus puodynės“ modifikacijos yra ne saugumo sprendimas, o jo radimas. „Medaus puodynė“ susideda ne tik iš įdomios informacijos, bet ir iš neteisėto įsilaužimo registravimo. [9,24]

2.4.1 „Medaus puodynės“ technologijų įvairovė

„Medaus puodynės“ technologijas galima skirstyti pagal keletą būdų. Vienas iš jų tai klasifikavimas pagal paskirtį:

1. Produkcinė „medaus puodynė” (angl. Production honey pot);
2. Tyrimų „medaus puodynė“ (angl. Research honey pot).

Produkcinė „medaus puodynė” yra lengvai naudojama ir skirta įmonių informacijos apsaugai, tačiau su ja surinkta informacija yra ribota. Ši „medaus puodynė“ įdiegiama į kompiuterį, kuris pastatomas tinkle kartu su kitais serveriais, taip gerinamas saugumą. „Medaus puodynė“ bando tam tikrame laike ir erdvėje kuo realiau imituoti serverių programinę įrangą ir jos veikimą. Produkcinė „medaus puodynė“ dažniausiai surenka duomenis ir siunčia sistemos administratoriui, o šis juos apdoroja ir imasi atitinkamų veiksmų. Ši informacija kur kas tikslesnė ir mažesnių apimčių, nei informacija kurią pateikia įsibrovimo susekimo programos ar ugniasienės. Ugniasienės ir įsibrovimo aptikimo sistemos pateikia labai didelius duomenų kiekius, kurių analizei reiktų daug laiko ir darbo jėgos išteklių, tad „medaus puodynės“ metodas efektyvus užtikrinant sistemų ar kompiuterio saugą . Paprastai produkcinė „medaus puodynė“ yra žemos sąveikos. Ši „medaus puodynė“ pateikia

mažiau duomenų apie atakas ar įsilaužimus nei mokslinė tyrimų „medaus puodynė“. Puodynės tikslas sumažinti atakų riziką tikriems serveriams. Tai yra kaip papildoma saugumo priemonė prie ugniasienių ir įsilaužimo aptikimo sistemų. Buvo toks atvejis, kai kompanija buvo užpulta ir buvo sugadintas jos vienas serveris, tačiau, kaip vėliau paaiškėjo, tai buvo serveris – „medaus puodynė“. Žala padaryta „medaus puodynės“ serveriui, tikrasis serveris ir toliau funkcionuoja, naudojamas metodas atliko savo darbą.[4,9,24]

Tyrimų „medaus puodynė“ skirta surinkti kuo didesnį ir įvairesnį informacijos kiekį, kad vėliau surinktą informaciją būtų galima panaudoti saugumo spragoms taisyti ir užtikrinti didesnį sistemos ar atskiro komponento saugumą. Tyrimų „medaus puodynė“ naudojama moksliniuose, kariniuose ir kituose valstybinės reikšmės projektuose. Ši „medaus puodynė“ neskirta saugumui padidinti realiu laiku, o tiesiog skirta jį padidinti ateityje - nuspėjant galimus įsilaužimus ir jiems užkertant kelią. Tyrimų „medaus puodynė“ gamybinę lenkia keliais žingsniais. Valstybinės ir nevalstybinės saugumu besirūpinančios organizacijos siekia, kad būtų sukurta iš anksto perspėjanti „medaus puodynė“. Tokios stambios kompanijos kaip Microsoft savo vizijoje mato „medaus puodynės“ naudojimą visur, suteikiant jai didesnį prioritetą nei palaikymui ir konfigūracijoms. Tiksliai imituojant tikruosius serverius ateityje „medaus puodynės“ metodo naudojimas atneš didelę naudą.[24]

Matome, jog „medaus puodynes“ dar galima skirstyti į žemos ir aukštos sąveikos. Didžiausias skirtumas tarp jų tai sąveika su operacine sistema, kuri leidžiama puolėjui. Mėgdžiodama operacinę sistemą žemos sąveikos „medaus puodynė“ suteikia nedaug kontrolės galimybių. Svarbiausias tokių „medaus puodynių“ pranašumas tai paprastumas, kuris leidžia lengvą išdėstymą ir palaikymą. Taip pat žemas rizikos lygis, nes nedirbama su pagrindine tikrąja sistema. Pavyzdžiui, pamėgdžiojama FTP (angl. File Transfer Protocol) paslauga, tuomet klausomasi 21 prievado. Galima pamėgdžioti tik prisijungimo vardą arba palaikyti ir kitas papildomas FTP komandas. Prie žemos sąveikos „medaus puodynių“ priskiriamos Specter, Honeyd, KFSensor, BackOfficer friendly, namų „medaus puodynės“ (angl. homemade honey pot) programos ir kitos (LaBrea, PatriotBox, Jackpot SMTP Tarpit). [24]

Aukštos sąveikos „medaus puodynėje“ realizuojamos tikros operacinės sistemos bei programos, niekas neimituojama. Tarkime, reikalinga FTP paslauga, veikianti Linux operacinėje sistemoje. Tokiu atveju statomas tikras Linux serveris su šia paslauga. Galima sukombinuoti iš daugiau dalių, suteikiant tikrumo jausmą puolėjui. Puolėjui suteikiama tikra

operacinė sistema, tad surenkamos informacijos kiekis žymiai padidėja ir galima pamatyti naujų įsilaužimo būdų. Puikus pavyzdys kaip naudojantis šiuo „medaus puodynės“ tipu buvo aptiktos užkoduotos slaptos komandos nestandartiniame IP protokole. Esminis trūkumas, lyginant su žemos sąveikos „medaus puodyne“, yra didesnis rizikos laipsnis, nes puolėjas gali perimti tikros operacinės sistemos kontrolę ir padaryti daugiau žalos. Reikia imtis papildomų technologinių sprendimų užkirsti kelią šiai blogybei. Aukštos sąveikos „medaus puodynės“ pavyzdžiais gali būti Symantec Decoy Server, Mantrap ir „medaus puodynės“ tinklai (angl. honey nets). ir daugelis kitų. Esminiai skirtumai tarp žemos ir aukštos sąveikos „medaus puodynių“ pateikti 2 lentelėje. [24]

2 lentelė Žemos ir aukštos sąveikos medaus puodynių savybės [24]

Žemos sąveikos sprendimas imituoja sistemas ir paslaugas	Aukštos sąveikos sprendimas pateikia realias sistemas ir paslaugas
• Lengva įdiegti ir išdėstyti, dažniausiai reikalauja lengvo įdiegimo ir programų konfigūracijos kompiuteryje. • Minimali rizika, nes galima kontroliuoti ką įsilaužėlis gali ir ko ne. • Surenka ribotą informacijos kiekį.	• Gali būti sudėtingas įdiegimas ir išdėstymas (komercinės versijos paprastesnės). • Didelė rizika, nes puolėjui suteikiama tikra operacinė sistema. • Galima surinkti daugiau informacijos, įskaitant naujus įsilaužimo būdus.

Apžvelgsime keletą žemos ir aukštos sąveikos „medaus puodynės“ realizacijų ir metodų.

BackOfficer friendly (toliau BOF) – tai Marcus Ranum ir komandos NFR kūrinys. Tai yra puikus pavyzdys žemos sąveikos „medaus puodynės“. Ši „medaus puodynė“ pasižymi paprastu valdymu ir savybėmis. Ja gali naudotis nepatyręs specialistas. BOF- tai Windows šeimai skirta programa, nors yra ir Unix platformai skirta versija. Ši realizacija yra visiškai nemokama. Gali pamėgdžioti 7 paslaugas, iš kurių populiariausios HTTP, FTP , Telnet (angl. TErminaL NETwork), paštą. Tai yra pastangų jungtis prie šių paslaugų prievadų registracija. Taip pat ši programa kuria dirbtinius atsakus į bandymą prisijungti. Tačiau imitacijų atsakas yra menkas, dėl to surenkama pakankamai mažai informacijos. Galima registruoti bandymą jungtis prie HTTP ar brutalios jėgos naudojimą siekiant atspėti slaptažodį (angl. brute force).

Tai panašu į įsilaužimų aptikimą ir signalizaciją, skiriant dėmesį pačioms populiariausioms paslaugoms. Programa parodo laiką, kas buvo daryta, su kokiu IP adresu buvo tai padaryta, kartais parodo net įsilaužimo tipą. BOF veikimas pagrįstas imituojamų paslaugų prievadų stebėjimu. Jei sistema palaiko komandą netstat –a, įdiegus BOF galima stebėti šiuos prievadus. Tačiau jei antivirusinė programa nuolat kontroliuoja POP3 (angl. Post Office Protocol Version 3) srautą (tikrina ar neužkrėstas paštas) BOF neatvaizduos šio prievado, nes jis tuo metu yra naudojamas. Tas pats ir su kitomis paslaugomis. [24]

BOF programa užima tik 92 Kb. ir paleidžiama akimirksniu. Valdymas lengvas, nes yra tik vienas valdymo langas. Tereikia pasirinkti kokias paslaugas imituoti. [24]

Galimos paslaugos:

- Back Orifice Trojos arklio veikimo principu pagrįsta programa, stebi 31337 prievadą;
- FTP – protokolas skirtas duomenų persiuntimui. Klausomasi 21 prievado. Populiariausia tarp programišių taikinių. Tikėtina, kad bus didelė veikla šiame prievade;
- Telnet – skirta tolimų sistemų valdymui. Stebimas prievadas 23;
- SMTP (angl. Simple Mail Transfer Protocol) – protokolas, skirtas pašto siuntimui ir gavimui. Prievadas – 25;
- HTTP – protokolas, skirtas žiniatinkliui. Prievadas – 80. Nėra galimybės klausytis SSL (angl. Secure Socket Layer), skirto šifravimui. Prievadas – 443;
- POP3 (angl. Post-Office Protocol)- pašto atkūrimo protokolas. Stebimas 110 prievadas;
- IMAP (angl. Internet Message Access Protocol) – dar vienas pašto protokolas, stebimas 143 prievadas.

Taip pat galima nustatyti apgaulingų atsakymų rinkinius tam tikroms paslaugoms. Nenustačius šių atsakymų programišiai supras, kad neteikiamos jokios paslaugos ir pasitrauks. Šie paslaugų mėgdžiojimai yra ribotų galimybių, pavyzdžiui, Telnet paslauga gali suteikti piktavaliui tik slaptažodžio ir vartotojo vardo įvedimą, nėra specifinių antraščių. Ši „medaus puodynė“ netinka didelėms organizacijoms dėl savo paprastumo ir įrankių stokos, taip pat neturi perspėjimo paštu sistemos, nes informacija pateikiama tik tekstiniu formatu tam tikrame faile. Taip pat BOF yra lengvai aptinkamas dėl savo paprastų atsakymų ir dėl to, kad lengvai prieinamas studijavimui, nes yra atviro kodo programa. [24]

Specter turi panašumų su anksčiau aptartu projektu, tačiau jis gali pamėgdžioti daugiau paslaugų bei operacinių sistemų. Lengvai įgyvendinamas su ganėtinai maža rizika. Ši programa dirba Windows operacinėje sistemoje. Ji gali greitai nustatyti kas ir ko ieško. Tačiau ši informacija yra daugiau statistinė nei informatyvi. Galimų operacinių sistemų imitavimo sąrašas gan platus : Windows 98, Windows NT, Windows 2000, Windows XP, Linux, Solaris, Tru64, NeXTStep, Irix, Unisys Unix, AIX, MacOS, MacOS X, FreeBSD. Turint tokią aibę operacinių sistemų, galime puikiai prisitaikyti prie įmonės poreikių. Taip pat galima suteikti tam tikras žymes imituojamai operacinei sistemai, kuri tampa kur kas realesnė. Tačiau reikia nepamiršti tokių paprastų dalykų, kaip pavyzdžiui, imituojant Solaris (Unix) operacinę sistemą, reikia imituoti šios operacinės sistemos paslaugas, o ne pavyzdžiui IIS (angl. Internet Information Services) serverį, kuris galimas tik Windows operacinėse sistemose. Kiekviena operacinė sistema turi unikalias savybes, kurios gali būti panaudotos, kad teigiamai identifikuotų sistemą. Egzistuoja keletas tokių programų kaip Nmap, kurios, siųsdamos įvairias užklausas į nutolusią sistemą, gali identifikuoti jos operacinę sistemą. Nustatymuose galima pasirinkti kokią norite turėti imituojamą operacinę sistemą: nesėkmingą, saugią, atidarytą, agresyvią ir keistą. Priklausomai nuo pasirinkimo, atitinkamai elgsis imituojama operacinė sistema. Priešingai nei BOF turi perspėjimo mechanizmą – siunčia pavojaus žinutes į paštą. Atsakai į piktavalio užklausas yra kur kas platesni nei BOF. Generuojami atsakymai kartais būna klaidinantys, kad atbaidytų piktavalį. Be to, yra pažeidžiamumo savybė, kuri piktavaliui imituoja, jog įvyko tam tikras sistemos pažeidimas, nors tai iš tikrųjų neįvyko. Dar vienas skirtumas – konfigūruojami spąstai, pavyzdžiui, norime žiūrėti MySQL prievadą- pažymime 3360 prievadą. Užpuolimo atveju, kai užpuolikas ne žmogus, ši programinė įranga mažai kuo padės administratoriui. Viena programos teigiamų savybių – galimybė nustatyti, kad informacija būtų renkama automatiškai. Dalis informacijos renkama pasyviai. Taip pat šią puodynę galima kontroliuoti nuotoliniu būdu tai palengvina administratoriaus darbą. [4,24]

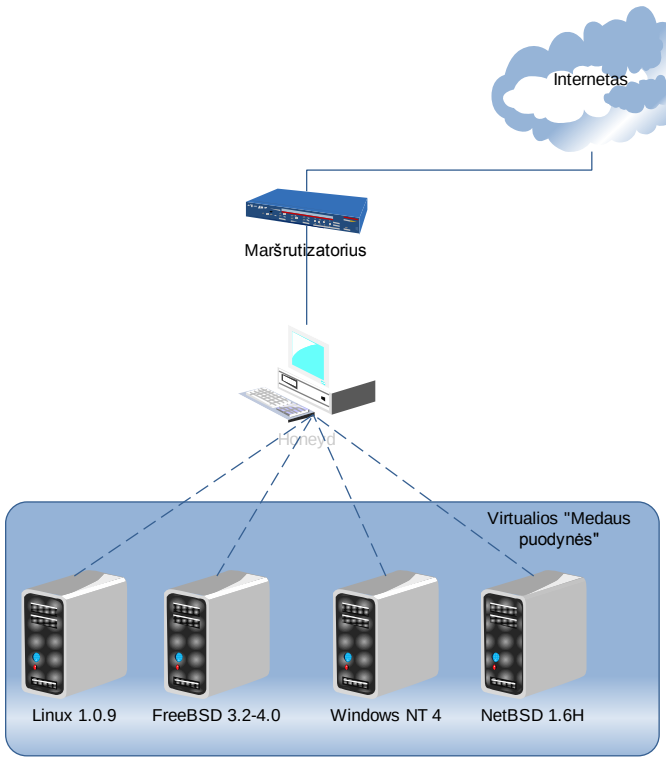
Dažnai nenorima, kad įsilaužėlis žinotų apie jo sekimą, tad reiktų aktyvius informacijos rinkimo būdus naudoti rečiau. Įspėjus piktavalį apie jo nesankcionuotus veiksmus jis dažniausiai pasitraukia. Specter kaip ir BOF negali klausytis tų prievadų, kurie yra naudojami tam tikros paslaugos ar programos. [4,24]

Ši puodynė palaiko panašias paslaugas kaip ir BOF, papildomai turi šias:

- FINGER – skirta gauti informacijai apie vartotojus iš nutolusios sistemos. Nustatomas 79 prievado sekimas;
- NETBUS, SUB-7, BO2K – Windows trojos arkliai. Stebimi 12345 bei 27374, 54320 prievadai;
- DNS (angl. Domain Name Service) – naudojamas atskleisti domenų pavadinimus ar perkelti zonų duomenis. Klausomasi 53 TCP prievado;
- SUN-RPC- prievadų ženklinimas ar nutolusios procedūros užklausimas. Stebimas 111 TCP prievadas;
- SSH (angl. Secure Shell) – užšifruojamas protokolas, kuris naudojamas saugiems nuotoliniams sistemos administravimams ar failų persiuntimams. Klausomas 22 TCP prievadas.

Nors ši programa naudinga, tačiau ji taip pat gali būti pavojinga - operacinėje sistemoje, kurioje įdiegta ši „medaus puodynė“ turi būti apsaugota aukščiausiu laipsniu, nes, piktavaliui užvaldžius tikrąją operacinę sistemą, gali būti neprognozuojamų pasekmių.[24]

Honeyd medaus puodynė yra atviro kodo ir pakankamai galinga. Sukurta Unix sistemoms. Ši programa gali imituoti apie 400 operacinių sistemų ir tūkstančius skirtingų kompiuterių vienu metu. Ji ne tik mėgdžioja operacines sistemas taikomajame lygmenyje, bet ir IP lygmenyje. Jokia kita „medaus puodynė“ šios savybės neturi. Taip pat Honeyd gali pamėgdžioti tūkstančius kompiuterių su skirtingais IP adresais. Kadangi tai yra atviro kodo programa, ji nuolatos tobulėja, nes saugumo bendruomenės darbuotojai ją nuolat tobulina. Pirmiausiai Honeyd naudojama atakų aptikimui. Kai tik puolėjas bando įsilaužti į neegzistuojančią sistemą, Honeyd nustato aukos IP adresą ir tiesiogiai bendrauja su įsilaužėliu pamėgdžiodama jam įvairias paslaugas. Pavyzdžiui, Telnet imitacija susijungus su Cisco ryšio paskirstytoju. Scenarijai gali būti parašyti bet kokia kalba. Su šia „medaus puodyne“ galima ne tik bendrauti su programišiumi, bet ir aptikti kitas veiklas stebint prievadus. Taigi sprendimas turi tokių galimybių, kurių neturi kiti sprendimai juos kartu sudėjus. Honeyd imituoja TCP (angl. Transmission Control Protocol) ir UDP paslaugas ir operacines sistemas. Schema pateikta 7 paveiksliuke.



7 pav. Honeyd „medaus puodynės“ naudojimo pavyzdys[24]

Keletas šios realizacijos ypatybių:

- Pilna atviro kodo programa;
- Lengva konfigūracija;
- Galima stebėti ne tik norimus prievadus, bet ir visą tinklą;
- Kadangi tai yra žemos sąveikos „medaus puodynė“, tai negalima aprūpinti realiais operaciniais sprendimais prieš priešininką. [4,24]

KFSensor - tai komercinė medaus puodynės versija, skirta padidinti saugumą organizacijoje. Ji užtikrina didesnį saugumo lygmenį nei ugniasienės. Tačiau kaina yra pakankamai didelė (990$), tad ją galėtų sau leisti įsigyti tik didesnės organizacijos. Ji veikia kaip masalo serveris. Šis sprendimas skirtas Windows operacinei sistemai ir turi novatoriškų realizacijų, tokių kaip nuotolinis valdymas (galima valdyti daug nutolusių KFSensor aplinkų), parašo variklis ir Windows protokolų pamėgdžiojimas. Daugiau galimybių nei prieš tai aptartuose produktuose: gali palaikyti ne keliasdešimt prievadų, o tūkstančius. Turi grafinėje vartotojo aplinkoje esantį valdymą, daug dokumentacijos. [24]

KFSensor susideda iš dviejų dalių:

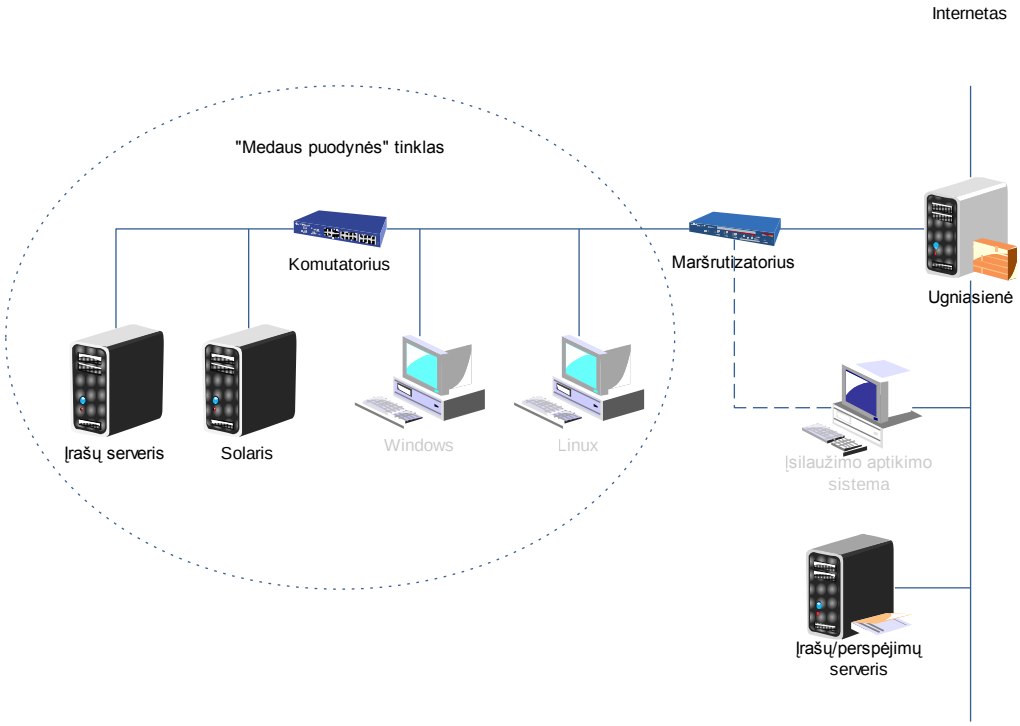
- KFSensor Server – skirtas prievadų klausymui, bendravimui su užpuoliku, paslaugos imitavimui;
- KFSensor Monitor – tai vartotojo sąsaja skirta stebėjimui ir analizei;

Papildomai prie ankstesniuose produktuose išvardintų imitacijų galima pridėti Microsoft SQL Server, Terminal Server, VNC (angl. Virtual Network Computing) ir kitas paslaugų imitacijas.[24]

Jackpot SMTP Tarpkit Brukalo (angl. spam) siuntėjai kiekvienam įstrigę į atmintį. Tad nekeista, kad egzistuoja „medaus puodynė“, kuri padeda surašyti šiuos siuntėjus į nepageidaujamųjų sąrašus. Brukalo siuntimo sustabdymui reikia įsitikinti, kad niekas neturi atidarytos prieigos prie SMTP serverio ir tuomet įdiegti antibrukalo įrankius. Kadangi sukurtasis pašto serveris yra netikras, tai kiekvienas atkeliavęs laiškas yra brukalas. Ši programa reaguoja iškart, kai gauna pirmąsias bandomąsias žinutes iš brukalo siuntėjo. Brukalai renkami ir grupuojami, vėliau panaudojami prieš brukalų siuntėją kaip įrodymai. Taip pat patikrinama ar šis siuntėjas yra įtrauktas į brukalų siuntėjų sąrašus. Ši „medaus puodynė“ pateikia atsakymus siuntėjui taip sulėtindama jo darbą.[24]

Mantrap yra komercinė „medaus puodynė“. Užuot imitavusi tam tikras paslaugas, ši programa kuria keturias posistemes, dažnai vadinamas „kalėjimais“. Šie „kalėjimai“ logiškai atskirti nuo pagrindinės operacinės sistemos. Administratoriai gali pakeisti „kalėjimų“ nustatymus taip, kad juose, kaip įprastame serveryje, būtų Oracle duomenų bazė, Apache serveris. Tai „medaus puodynę“ daro lankstesne ir tikresne. Taip pat galima IRC pokalbių imitacija ir kitos smulkmenos. Šis sprendimas skirtas tik Solaris operacinei sistemai.[24]

„Medaus puodynės “tinklai priskiriami prie tyrinėjimo tipo „medaus puodynių“. Tai produkcinių sistemų tinklas. Skirtingai nei kitose „medaus puodynėse“ čia nieko nėra imituojamo. Konkretus pavyzdys pateikiamas 8 paveikslėlyje. Tokia sistema duoda pilną sistemų, programų funkcionalumą. Dėl to galima analizuoti jų organizaciją, komunikavimo metodus, įsilaužimo motyvus, tačiau proporcingai didėja ir rizikos laipsnis. Šie tinklai skirti duomenų analizei ir apdorojimui bei sprendimų priėmimui.[4,9,24]



8 pav. „Medaus puodynės“ tinklas [24]

Symantec Decoy Server. Kadangi tai yra aukštos sąveikos puodynė, surenkamos informacijos kiekis ženkliai didesnis. Ši sistema leidžia stebėti kiekvieną piktavalio klavišo paspaudimą. Kadangi tai komercinis produktas, tai jo veikimo schema yra nedetalizuojama. Keletas pagrindinių šio metodo (produkto) savybių:

- Turi incidento valdymo savybę - praneša ir registruoja veiksmą, tai yra vienas iš prioritetų;
- Atsako į programišiaus veiklą pagal dažnumą ir apriboja galimybes piktavaliui, taip kontroliuodamas jo veiksmus;
- Aprūpina gyvu kontroliavimu, gyva analize;
- Aptinka ir įsilaužimą į tinklą, ir įsilaužimo autorių;
- Nežinomų žygdarbių ir atakų nulinės dienos pripažinimas. [24]

Kitos „Medaus puodynės“. „Medaus puodynių“ paskirtis dažniausiai būna specifinė: kirminų aptikimas ar sistemos stebėjimas. Jos gali būti panaudotos kaip produkcinės ar tyrimų „medaus puodynės“, priklausomai nuo jų tikslų. Šias „medaus puodynes“ vartotojai susikuria pagal savo poreikius, todėl galimų variantų yra begalė.[24]

Vienas bendras pavyzdys - kuriama paslauga, kuri klausosi 80 prievado (HTTP). Tai paprastai daroma, kad sugauti kirminus:

netcat -l -p 80 > c:\honeypot\worm

Kirminas galėjo jungtis prie netcat, klausančio 80 prievado. Jei kirminas padarytų sėkmingą TCP susijungimą, ši informacija būtų išsaugota „medaus puodynėje“ ir išanalizuota administratoriaus. „Medaus puodynėse “ galima realizuoti ir daugiau , tačiau tada galima nenaudingai padidinti saugumo riziką. [24]

Dar keletas „medaus puodynių“:

- Johannes B. Ullrich Perl kalba parašytas prievadų sekėjas kirmino W32/leaves aptikimui;
- Intd (super serverio) imitacija skirta Windows NT ir Windows 2000 operacinėms sistemoms;
- Pašto siuntimo „medaus puodynės“, skirtos šlamšto siuntėjams nustatyti;
- LaBrea Tarpit – unikalus įrankis ne tik aptikti, bet ir sulėtinti kirmino veikimą. Veikimas pagrįstas nenaudojamų IP adresų panaudojimu sukuriant virtualius serverius, kuriuose atsakinėjama į programišių ir kirminų užklausas, taip sulėtinant jų veiklą.

Taigi matome, jog egzistuoja tiek prievadų stebėjimo sistemos, tiek „kalėjimo“ tipo „medaus puodynės“.[24]

2.4.2 „Medaus puodynių“ palyginamas

Egzistuoja nemažai „medaus puodynių“ realizacijų bei metodų, tad naudinga būtų pagrindines savybes pateikti lentelėje. „Medaus puodynės“ palyginti yra ganėtinai sunku dėl jų didelio pasirinkimo spektro ir galimybių, tad palyginamoji informacija labiau informacinio pobūdžio nei funkcijų aprašymai. Lyginant „medaus puodynes“ svarbi jos sąveika, operacinė sistema (toliau OS), kuriai skirta, imituojamų paslaugų ir OS skaičius, kaina, paskirtis. Palyginimas pateiktas 3 lentelėje. Iš lentelės matome, jog daugiausiai paslaugų galinčios imituoti „medaus puodynės“ yra mokamos. Geriausias sprendimas tai yra „medaus puodynės“ tinklai, nes imituojamų paslaugų skaičius neribojamas, tačiau viską reikia pasidaryti pačiam administratoriui.

3 lentelė "Medaus puodynių" palyginimas

	Sąveika	OS tipas	Imituojamų paslaugų/ OS skaičius	Kaina	Tinkamumas
BackOfficer friendly	žema	Unix	7 paslaugos	Nemoka-ma	Asmeniniam naudojimui
Specter	žema	Windows	14 paslaugų/ 14 OS	900$	Didelėms įmonėms
Honeyd	žema	Unix	11 paslaugų/13 OS	Nemoka-ma	Asmeniniam naudojimui
KFSensor	žema	Windows	15 paslaugų	Nuo 600$	Didelėms įmonėms
Jackpot SMTP tapkit	žema	Unix	1 paslauga	Nemoka-ma	Asmeniniam naudojimui
Mantrap	aukšta	Solaris	1 OS	Nemoka-ma	Asmeniniam naudojimui
„Medaus puodynės“ tinklai	aukšta	Visos	daugelis	Nemoka-ma	Visiems
Symantec Decoy server	aukšta	Visos	daugelis	Neskelbiama	Didelėms įmonėms

2.4.3 „Medaus puodynės“ legalumas

„Medaus puodynės“ gali padėti išspręsti įvairias saugumo problemas organizacijos viduje, bet ar šios puodynės teisėtai panaudotos ? Į šį klausimą labai sunku atsakyti, nes nėra vientiso dokumento, apibrėžiančio tai. Kiekviena valstybė apibrėžia tai vis kitaip. Pavyzdžiui. Vokietijoje priimtas įstatymas, kuriuo numatomos bausmės už be slaptažodžio įkurtą bevielį ryšį (bevielio ryšio stotelę). Taigi bevielio ryšio stotelės („medaus puodynės“) šioje šalyje draudžiamos. Taip pat kiekvienos organizacijos viduje yra tam tikra saugumo politika. Priklauso nuo to, kokia informacija renkama ir kokiam tikslui. Disponavimas šia informacija apima tą patį lygmenį kaip ir, tarkim, paketų apdorojimas saugumui užtikrinti, bet ne kitiems tikslams. „Medaus puodynė“ - tai tam tikra provokacija programišiui , o tai nelegalu. Viena iš

„medaus puodynės“ paskirčių – stebėti programišiaus veiksmus, o tai gali būti traktuojama kaip pasiklausymas. Teisinės problemos konkrečiai neapibrėžiamos virtualiame pasaulyje. Be to, prieš įdiegiant tokią sistemą, derėtų pasikonsultuoti su teisininku ir nuolat sekti Lietuvos Respublikos įstatymus, kurie susiję su virtualiu pasauliu. [9,24]

2.5 Analizės išvados

Analizės metu pastebėta, jog dažniausiai įsilaužimo aptikimo sistemose naudojami parašų analizės, anomalijų aptikimo ir statistiniai ar mišrūs metodai. Rečiau naudojami neuroninių tinklų, kompiuterių mokymosi metodai. Taip pat įsilaužimo aptikimo sistemos skirstomos pagal dislokacijos vietą, aktyvumą, protokolą pobūdį ir kitas ypatybes.

Šioje dalyje atlikta lyginamoji esamų įsilaužimo aptikimo sistemų analizė, kur sprendimai palyginami funkcionalumo, patogumo, pritaikomumo aspektais. Lyginami buvo tiek komerciniai produktai, tiek atviro kodo įsilaužimo aptikimo sistemos. Nustatyta, jog Snort yra viena iš geresnių šio tipo įsilaužimo aptikimo sistemų pagal anksčiau minėtus kriterijus. Analizuojant įsilaužimo aptikimo sistemas pastebėta ir keletas jų trūkumų:

- Dažnai reaguoja į įvairius triukšmus (programų klaidų sugeneruoti paketai, sugadinti DNS duomenys, vietiniai nutekėję paketai)
- Tikrosios atakos dažnai praleidžiamos (pasenusi parašų bazė gali kelti keblumų)
- Reagavimas į ataką dažnai apsiriboja jos užblokavimu

Viena didžiausių problemų yra tai, kad šie apsaugos metodai remiasi turimomis žiniomis apie įsilaužėlį, tad naujus įsilaužėlio naudojamus metodus sunku identifikuoti.

Dalinai šias problemas gali išspręsti „medaus puodynės“ metodo panaudojimas, kurio esmė - suviliok ir pagauk. „Medaus puodynės “ pagal sąveiką gali būti skirstomos į du tipus: aukštos ir žemos sąveikos. Aukšta sąveika reiškia, kad įsilaužėliui pateikiama realiai veikianti paslauga ar operacinė sistema, o žemos sąveikos „medaus puodynėse“ tik imituojama paslauga ar operacinė sistema. Apžvelgtos įvairių tipų „medaus puodynės“ stengiantis suprasti jų galimybes bei pritaikymus. Pastebėta, kad daugiau galimybių duoda aukštos sąveikos sistemos, tačiau jos valdymo ir saugumo požiūriu yra sudėtingesnės. Žinant tikslus ir poreikius svarbu nepamiršti išanalizuoti „medaus puodynės“ teisinę pusę įmonės vidaus politikoje bei valstybiniuose įstatymuose.

3. PERSPĖJAMOSIOS SISTEMOS SU „MEDAUS PUODYNE“ MODELIS

3.1 Tikslas

Rinkti informaciją apie naujausius įsilaužimo metodus bei atvaizduoti ją įrašų analizės posistemėje, kur administratorius ją apdoroja bei sudaro naujus apsaugos metodus.

3.2 Reikalavimų specifikavimas

Projektuojamai perspėjimo sistemai, kurios pagrindas „medaus puodynės“ metodo panaudojimas, keliami atitinkami reikalavimai. Jie skirstomi į dvi grupes: funkciniai, nefunkciniai. Šiuo atveju minimi tik funkciniai, nes nefunkciniai tiesiogiai su funkcijomis nesusiję reikalavimai. Sistemos funkciniai reikalavimai - tai reikalavimai, kurie nusako sistemos funkcionalumą, apibrėžia ką, sistema turėtų daryti ir kokiais duomenimis manipuliuoti, kad atliktų pagrindines savo funkcijas.

Įsilaužimo aptikimo ir peradresavimo posistemė:

- nukreipia vartotoją į „medaus puodynę“ pagal:
  - sudarytas įsilaužimo aptikimo taisykles;
  - juoduosius IP sąrašus.
- turi galimybę nustatymuose nurodyti:
  - laiko tarpą, kuriam nukreipiamas vartotojas į „medaus puodynę“;
  - maksimalų galimų prisijungimų skaičių.
- generuoja perspėjimus apie įsilaužimą bei perduoda juos į „medaus puodynės“ serverį.

„Medaus puodynės“ serveris:

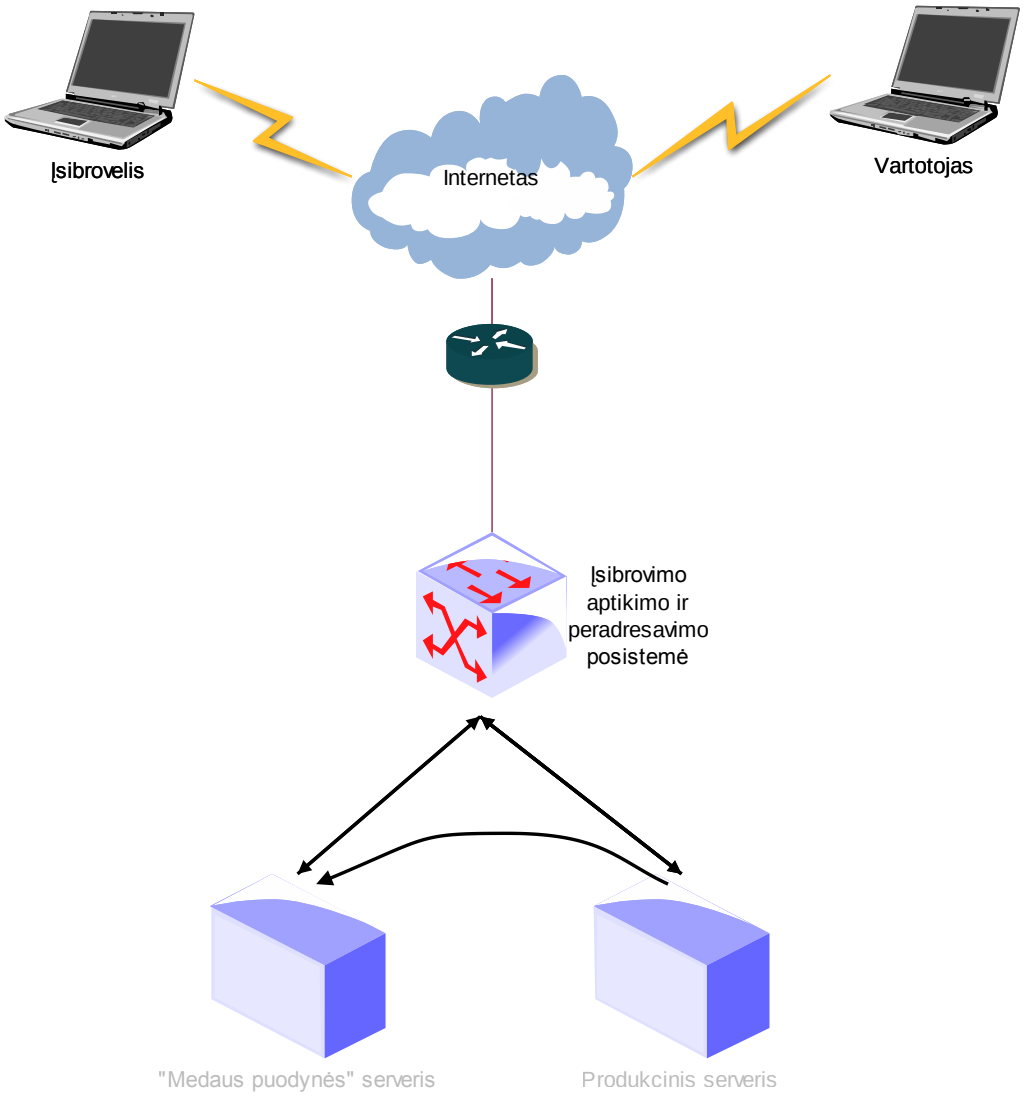
- turi „medaus puodynės“ svetainę su modifikuota prisijungimo duomenų baze
- turi įrašų analizės posistemę, kurioje:
  - naudojama autentifikacija;
  - pateikiama OS , ĮAS bei žiniatinklio serverio įrašų istorija;
  - yra galimybė ieškoti duomenų, rikiuoti.

Produkciniam serveriui:

- įdiegta apsauga nuo slaptažodžio atakų bei juodųjų IP sąrašų perdavimas į įsilaužimo aptikimo ir peradresavimo posistemę.

3.3 Perspėjimo sistemos architektūra

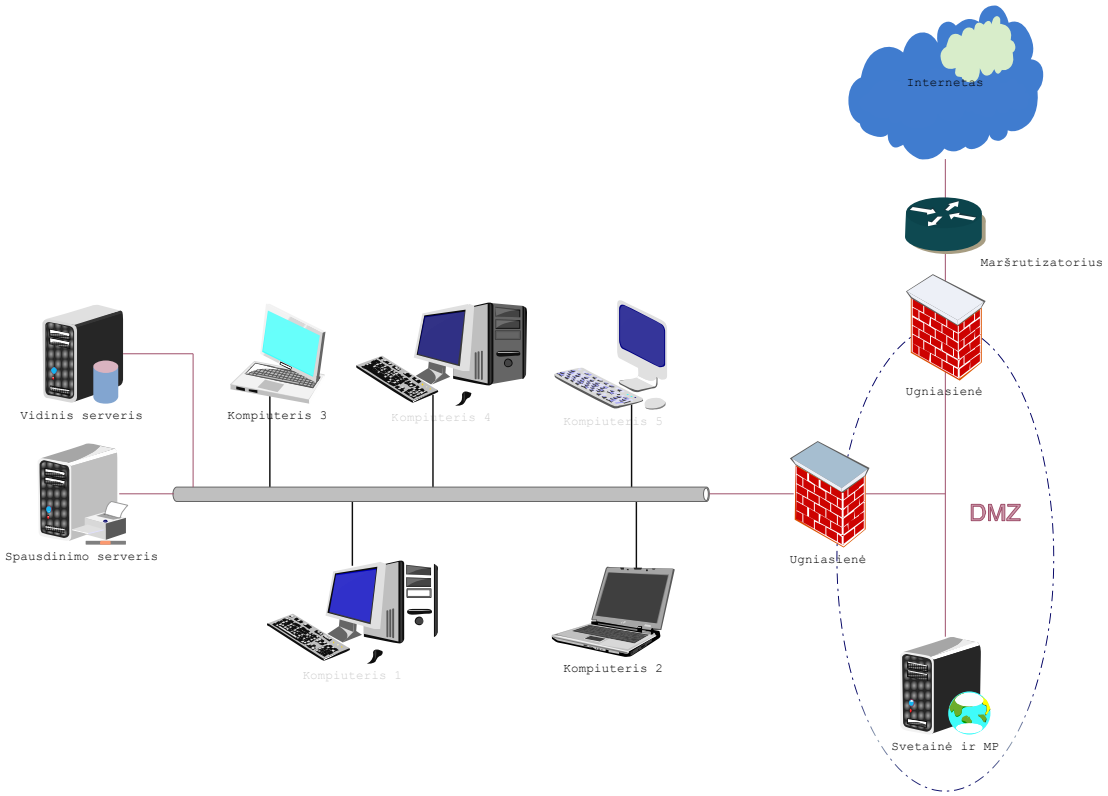
Dažnai administratorius susiduria su programišių veiksmais, nukreiptais prieš serverį, šie veiksmai gali būti tiek automatizuoti, tiek valdomi rankiniu būdu. Esant sėkmingoms atakoms, gali nukentėti serveryje saugomi duomenys, žiniatinklio puslapis ar paslauga gali tapti nepasiekiami, pablogėti jų darbas ar įvykti kitas nepageidaujamas scenarijus. Kad visa tai nenutiktų, administratorius turi sekti įsilaužimo aptikimo ar prevencinės sistemos įspėjimus, nuolatos privalo peržiūrėti kreipimosi į serverį įrašus, bei kiekvienu atveju imtis atitinkamų veiksmų, saugumui didinti. Tačiau dažnai atsitinka taip, kad įsilaužimas pastebimas tik patyrus nuostolių - įsilaužus. Taip pat atakos praleidžiamos dėl pasenusios parašų bazės. Panaudojus atviro kodo įsilaužimo aptikimo sistemą, keletą jos įskiepių, ugniasienę, kartu su „medaus puodynės“ metodo veikimo principu, galima sudaryti perspėjimo sistemos su „medaus puodyne“ modelį. Modelio schema pateikta 9 paveiksliuke. Įsilaužėliui bandant įsilaužti ar pakenkti produkciniam serveriui, kuriame patalpinta žiniatinklio svetainė, jis nukreipiamas į „medaus puodynės“ serverį, kur gali toliau atlikti savo kenkėjiškus veiksmus, tačiau čia jis gali pakenkti tik „medaus puodynės“ (toliau MP) svetainei, tuo tarpu tikroji svetainė lieka saugi. Vykdant ataką, nukreiptą prieš slaptažodį, pagal tam tikrus požymius vartotojas nukreipiamas į „medaus puodynę“. Abiem atvejais informacija apie piktavalį kaupiama įrašų analizės posistemėje, kur apdorodamas informaciją administratorius gali sukurti naujų priemonių produkcinio serverio saugumui padidinti.



9 pav. Perspėjimo sistemos panaudojant „medaus puodynę“ modelio schema

3.4 Galima perspėjimo sistemos dislokacijos vieta

Medaus puodynės serveris bei produkcinis serveris patalpinti į DMZ (Demilitarizuotą zoną). Demilitarizuota zona - tai sritis, atskirta ugniasienėmis, esanti tarp išorinio ir vidinio tinklo. Tinklo schema pateikta 10 paveiksliuke. Ugniasienė – tai įranga (aparatinė arba programinė), sukurianti apsauginę sieną tarp kompiuterio ir interneto. Ji gali apsaugoti kompiuterį bei tinklą nuo daugelio įsilaužėlių bei kompiuterinių virusų ir kirminų.



10 pav. MP ir produkcinio serverio vieta tinkle

3.5 Modelio struktūrinės dalys

Modelį susidaro 3 pagrindinės dalys:

- Įsibrovimo aptikimo ir peradresavimo posistemė;
- „Medaus puodynės“ serveris;
- Produkcinis serveris.

3.5.1 Įsibrovimo aptikimo ir peradresavimo posistemė

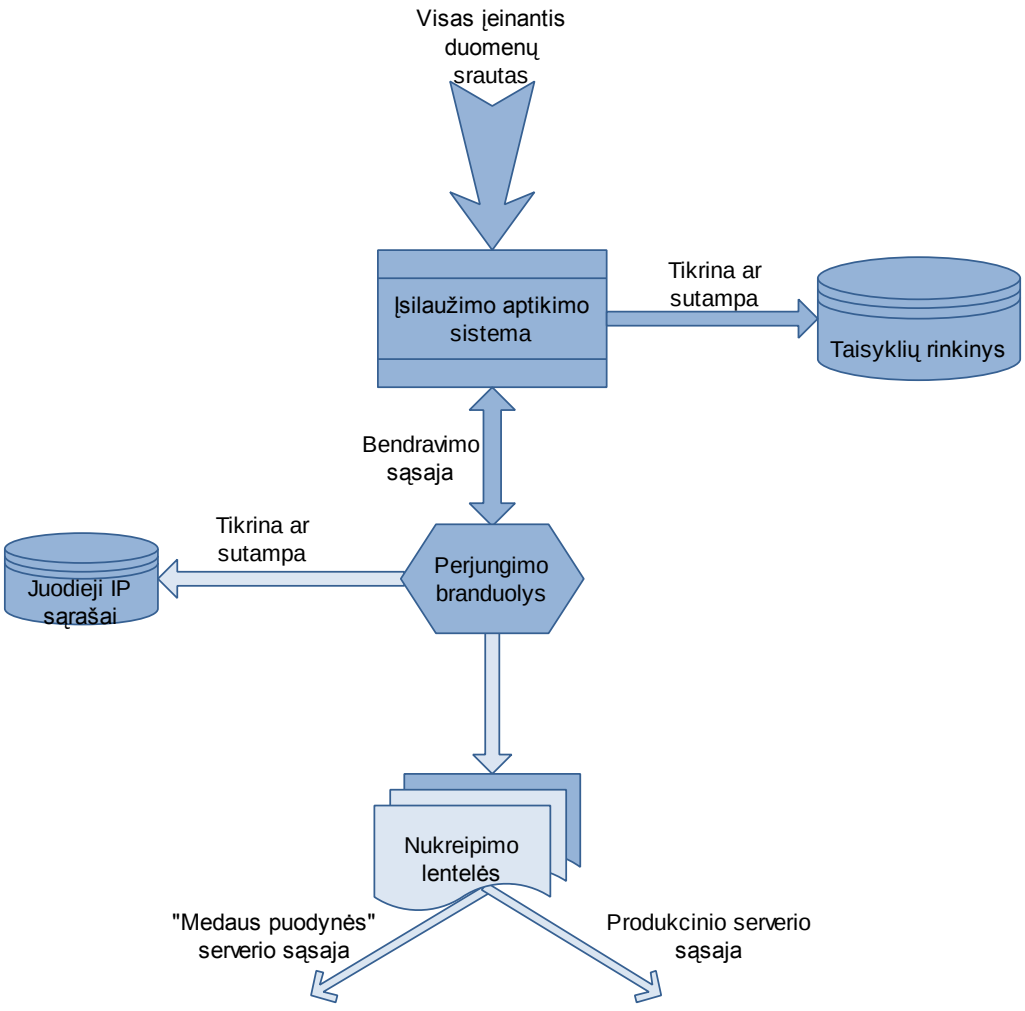
Įsibrovimo aptikimo ir peradresavimo posistemė yra „medaus puodynės“ maršrutizatorius, kuris ateinantį „kenkėjišką“ srautą nukreipia į „medaus puodynės“ serverį. Šis „medaus puodynės“ maršrutizatorius turi 3 sąsajas:

0 sąsaja skirta visam įeinančiam srautui (interneto tiekėjo skiriamas adresas);
1 sąsaja skirta srautui nukreipti į „medaus puodynės“ serverį (potinklis A);
2 sąsaja skirta srautui nukreipti į produkcinį serverį (potinklis B).

Įsibrovimo aptikimo ir peradresavimo posistemę sudaro šios dalys:

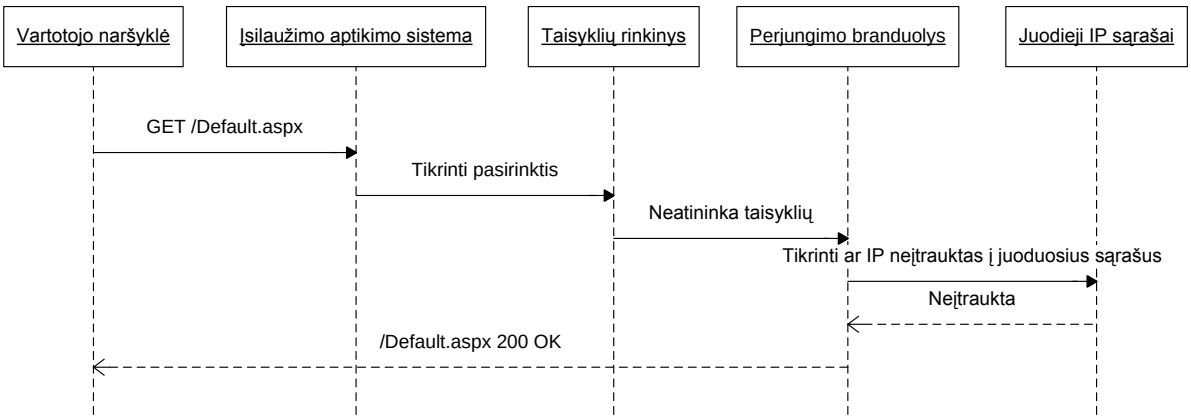
- Perjungimo branduolys;
- Taisyklių rinkinys;
- Juodieji IP sąrašai;
- Bendravimo sąsaja tarp perjungimo branduolio bei įsilaužimo aptikimo sistemos;
- Įsilaužimo aptikimo sistema;
- Nukreipimo lentelės.

Perjungimo branduolys bendrauja specialia sąsaja su įsilaužimo aptikimo sistema, kuri savo ruožtu tikrina taisyklių rinkinius. Įsilaužimo aptikimo sistema grėsmės atveju generuoja perspėjimą, kuris perjungimo branduolyje aktyvuoja statinių nukreipimo lentelių sudarymą, kurios nukreipia srautą į „medaus puodynės“ serverį. Posistemės schema pateikta 11 paveiksliuke.



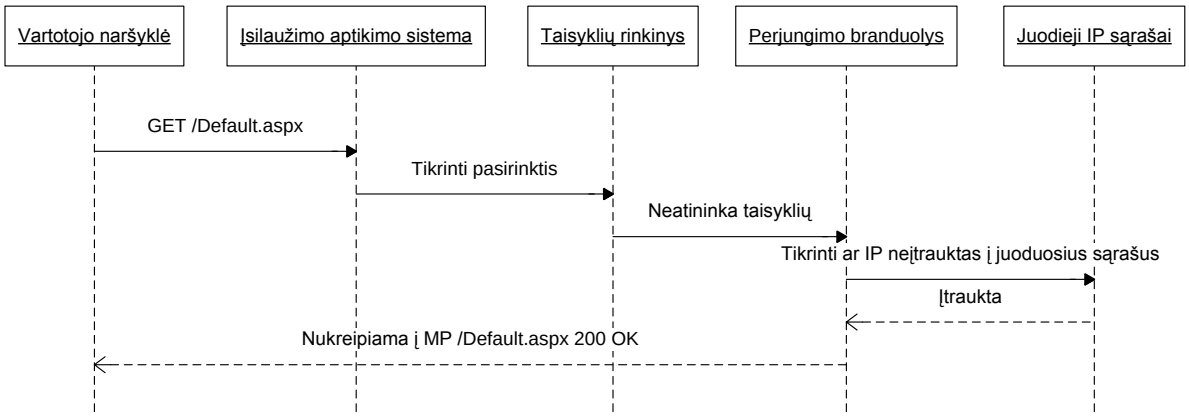
11 pav. Įsibrovimo aptikimo ir peradresavimo posistemė

Jei vartotojui atlikus užklausą ši neatitinka ĮAS taisyklių ir šio vartotojo IP adresas neįtrauktas į juoduosius sąrašus, jam leidžiama pasiekti produkcinę svetainę. Sekų diagrama pateikta 12 paveiksliuke.



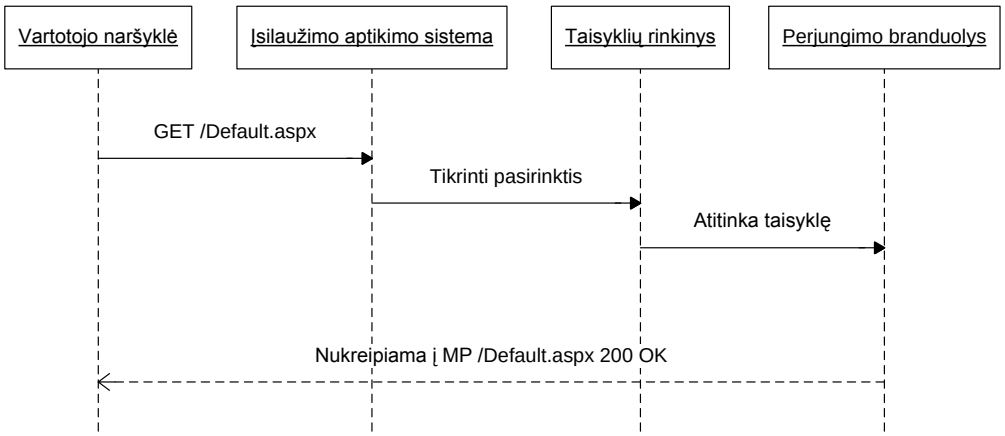
12 pav. Veiksmų sekų diagrama pateikiant tikrąją svetainę

Įsilaužimo aptikimo ir perjungimo branduolys tikrina juoduosius IP sąrašus, radus sutapimą, srautas perjungiamas į „medaus puodynės“ serverį. Sekų diagrama pateikiama 13 paveiksliuke.



13 pav. Veiksmų sekų diagrama aptikus IP adresą juoduosiuose sąrašuose

Jei vartojo atliekami veiksmai atitinka sudarytą ĮAS taisyklę, šis nukreipiamas į MP svetainę. Šio scenarijaus sekų diagrama pateikiama 14 paveiksliuke.



14 pav. Veiksmų sekų diagrama sutapus taisyklei su pasirinktimi

ĮAS taisyklių sudarymas

Daugumos įsilaužimo aptikimo sistemų taisyklių struktūra yra panaši. Kiekviena taisyklė turi antraštę bei pasirinkimus (opcijas). Antraštė turi šaltinio IP adresą, prievadą, protokolą, tikslo IP adresą bei prievadą. Pasirinkimų dalis turi perspėjamąją žinutę bei dalį paketo, koks jis turi būti, kad būtų imtasi reakcijos – aktyvuoti perspėjamąją žinutę.

Taisyklės pavyzdys:

alert tcp any any -> 192.168.1.0/24!111: (content:|.000186a5.|; msg .mountd access.)

Ši taisyklė generuoja perspėjamąją žinutę, jei iš TCP protokolo iš bet kokio IP adreso ir bet kokio prievado kreipiamasi į adresus nuo 192.168.1.0 iki 192.168.1.255, ne didesniu nei 111 prievadu, ir paketo turinys turi šešioliktainį kodą, parodantį, jog kažkas bando įkelti direktoriją per paslaugos prievadą, kuris normaliomis sąlygomis nenaudojamas. [14]

Taisyklės veiksmas galimas trejopas:

- Perspėjamoji žinutė - perspėja ir kaupia įrašų istoriją
- Įrašyti – kaupia įrašų istoriją
- Praleisti – ignoruoja paketą

Svarbi taisyklės dalis – protokolas. Populiariausi TCP, UDP, ICMP (angl Internet Control Message Protocol), galima įkelti ir kitų.

Taip pat galima nurodyti konkretų šaltinio ar tikslo IP adresą, šauktukas (!) naudojamas norint išreikšti negatyvų operatorių. Nurodant tiek prievado numerį, tiek IP adresą galima naudoti žodelį any, taip apibrėžiant visus prievadus ar IP adresus. Apibrėžiant prievadus galima

naudoti operatorių dvitaškį (:), kur prieš dvitaškį nurodomo intervalo pradžia - po intervalo pabaiga. Operatorius -> taisyklėje naudojamas apibrėžti duomenų srautų kryptį. Priklausomai nuo įsilaužimo aptikimo sistemos, sudarinėjant sudėtingesnes taisykles, galima naudoti kintamuosius.[14]

Norint sukurti taisykles piktybinei veiklai aptikti reikia žinoti naujausius ir dažniausiai pasitaikančius įsilaužimo būdus.

3.5.2 „Medaus puodynės“ serveris

„Medaus puodynė” turi 2 pagrindinius modulius:

- MP Svetainę;
- Įrašų analizės posistemę.

MP Svetainė

Pagal medaus puodynės apibrėžimą: „Medaus puodynė“ - tai informacinės sistemos resursai, kurių tikslas atkreipti dėmesį ir pritraukti, aptikti bet kokią nelegalią ar neautorizuotą veiklą. Kaip žinia, dažnai įsilaužėliai savo tikslams pasiekti naudoja viešai prieinamą informaciją (pasiekiamą internete). Taigi pirmieji žingsniai kuriant „medaus puodynę“, padaryti šią „medaus puodynę“ kuo realesnę bei patrauklesnę įsilaužėliui. Konkrečiu atveju naudojama tikrosios svetainės kopija su modifikuota autentifikacijos duomenų bazė.

MP svetainėje autentifikuojant vartotoją reikia naudoti modifikuotą duomenų bazę - t.y. be tikrų prisijungimo vardų, naudojant nedidelį kiekį brutalios jėgos sugeneruotų slaptažodžių bei iš žodyno failo.

Priklausomai nuo paslaugos, kuri teikiama serveryje, reikia projektuoti ir „medaus puodynę“. Tai gali būti tiekiamas vidinis firmos FTP su svarbia informacija. Dar kitu atveju, tai gali būti firmos paštas ar vidinė duomenų bazė su informacija, kuri gali sudominti įsilaužėlį.

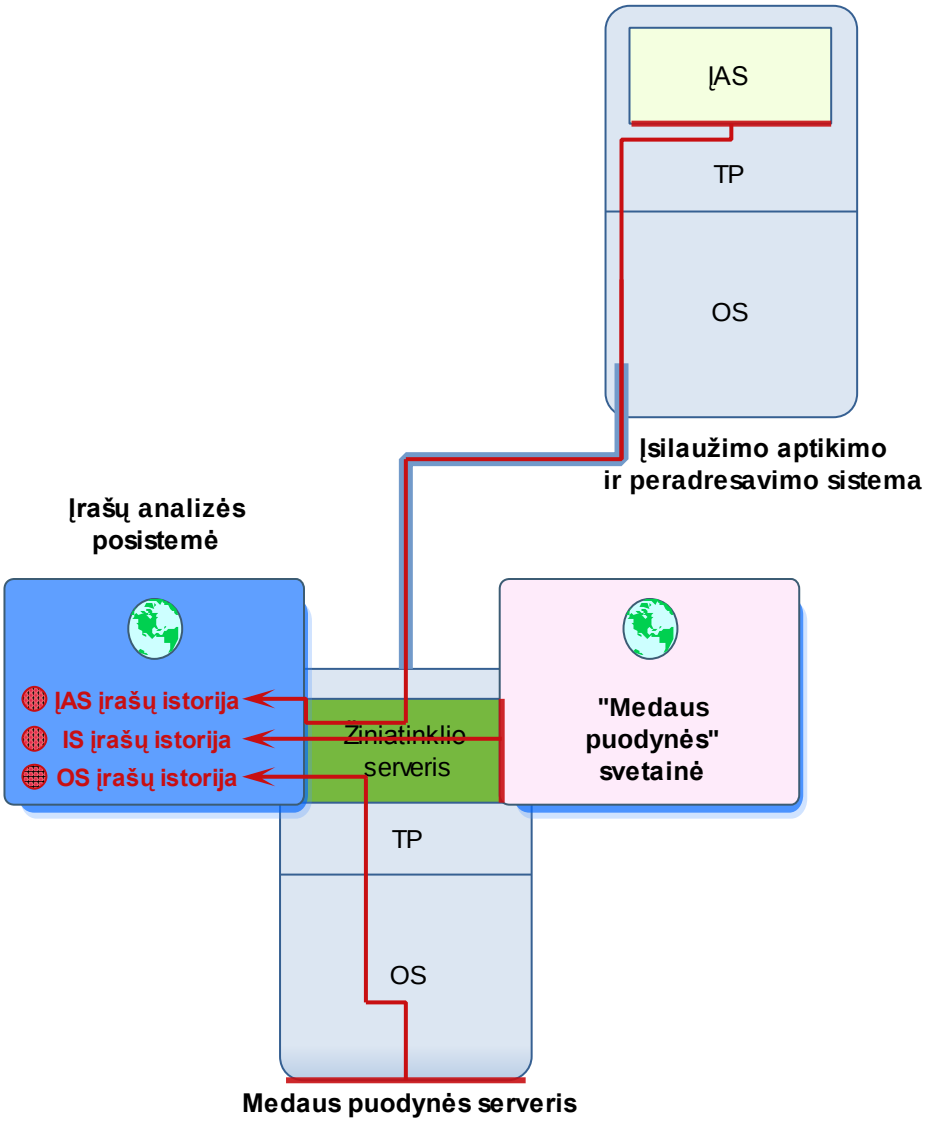
Įrašų analizės posistemė

Posistemė pateiks trijų tipų įrašus:

- Įsilaužimo aptikimo įrašų istoriją – perspėjimus;
- OS įrašų istoriją;

- Žiniatinklio serverio įrašų istoriją.

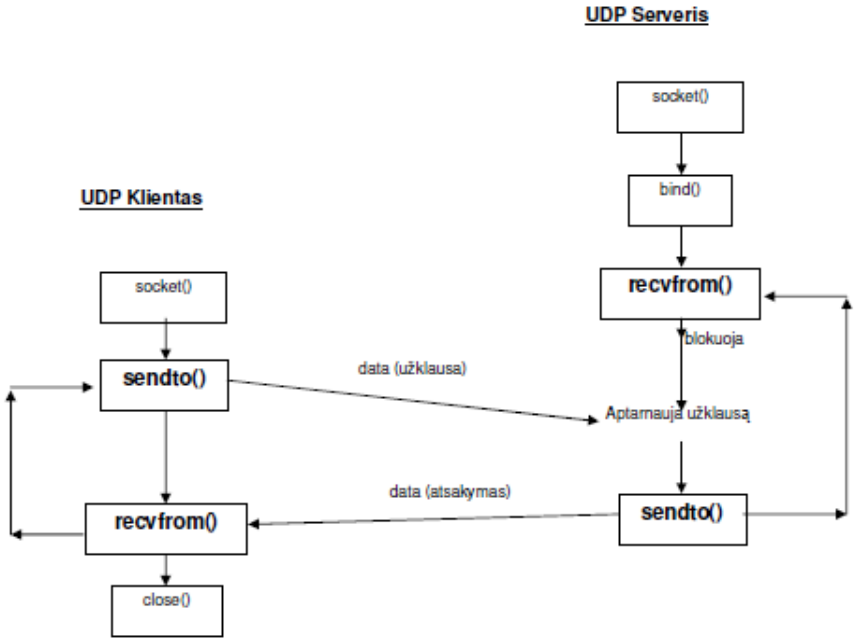
Visi šie įrašai pateikiami vienoje interneto svetainėje, kad administratorius patogiai, koncentruotai galėtų juos peržiūrėti ir analizuoti. Perspėjimo sistemos įrašų surinkimo vietos pateiktos 15 paveiksliuke. TP – taikomosios programos, IS – imituojama svetainė.



15 pav. Įrašų analizės posistemės įrašų surinkimo vietos

Įsilaužimo aptikimo įrašų istorija – perspėjimai yra generuojami įsilaužimo aptikimo ir peradresavimo posistemėje. Kadangi įrašų analizės posistemė yra dislokuota „medaus puodynės“ serveryje, įsilaužimo aptikimo įrašų istoriją – perspėjimus reikia turėti šiame serveryje. Tai galima atlikti perduodant duomenis tiesiai iš įsilaužimo aptikimo sistemos perspėjimo modulio panaudojant UNIX soketus, kur kitas soketas klausosi jų ir įrašęs į DB pateikia į įrašų analizės posistemę. Dažniausiai perspėjimams perduoti naudojamas soketas,

kuris naudoja UDP protokolą. Apibendrintas UDP soketo darbo algoritmas pateiktas 16 paveiksliuke.



16 pav. Kliento - serverio darbo algoritmas (UDP atveju) [3]

Dažniausiai įsilaužimo aptikimo sistemų perspėjimų įrašas (ir kitų sričių įrašai) susideda iš:

- Įvykio datos;
- Įvykio laiko;
- Įvykio pavadinimo;
- Užuominų – informacijos šaltinio;
- Šaltinio IP adreso bei prievado;
- Taikinio IP adreso bei prievado;
- Kitos informacijos.

Sudėtis priklauso nuo naudojamos įsilaužimo aptikimo sistemos. Pavyzdžiui plačiai paplitusios Snort įsilaužimo aptikimo sistemos perspėjimų įrašas atrodo taip:

Jul 23 22:34:15 dissent snort[8094]: BACKDOOR ATTEMPT-Back Orifice 2000: 192.168.10.10:22652 -> 192.168.1.1:8787 [7]

Operacinės sistemos įrašų istorija kaupiama „medaus puodynės“ serveryje, tad papildomų priemonių pasiekti juos, tokių kaip soketai, nereikia. OS kaupia didelius kiekius įrašų istorijos, todėl yra svarbu kuriuos įrašus pasirinkti atvaizduoti.

Linux OS visos įrašų istorijos saugomos /var/log/ kataloge:

- */var/log/message*
- */var/log/auth.log*
- */var/log/kern.log*
- */var/log/cron.log*
- */var/log/maillog*
- */var/log/qmail/*
- */var/log/httpd/*
- */var/log/lighttpd*
- */var/log/boot.log*
- */var/log/mysqld.log*
- */var/log/secure*
- */var/log/utmp arba /var/log/wtmp*
- */var/log/yum.log*

Atvaizdavimui derėtų pasirinkti tokius įrašus, kurie būtų naudingi ir perspėjimo sistema nebūtų perkrauta informacija. Tai galėtų būti auth.log – autentifikacijos įrašų istorija, httpd – Apache prieiga ir klaidos, Lighttpd prieiga ir klaidos, */var/log/utmp ar /var/log/wtmp* – prisijungimų įrašų istorija. [8]

Pavyzdžiui faile */var/log/auth.log* dažnai galima išvysti eilutę, kurioje matome mėginimą prisijungti prie SSH:

Dec 18 12:20:18 blah sshd[30294]: Failed password for invalid user postmaster from ::ffff:218.206.92.226 port 60532 ssh2

Windows OS pagrindinės įrašų istorijos yra renkamos ir klasifikuojamos į:

- Svetainės įrašų istorijas;
- Saugumo įrašų istorijas;
- Sistemos įvykių įrašų istorijas;
- Domeno kontrolerių įvykių įrašų istorijas;
- DNS įvykių įrašų istorijas;
- Failų mutavimo įrašų istorijas.

Įrašų analizės posistemėje reikalinga atvaizduoti saugumo įrašų istoriją ar sistemos įvykių įrašų istoriją, kurios saugo pavykusius ir nepavykusius prisijungimus, veiksmus su failais ir fiksuoja sistemos veiksmus. [15]

Visos Windows įrašų istorijos saugomos *C:\WINDOWS\system32\config* kataloge (pagal nutylėjimą) failuose : *SecEvent.evt, SysEvent.evt, SecEvent.evt* ir kiti. Šiuos failus Windows aplinkoje peržiūrėti galima Event Viewer programa, taip pat galima juos konvertuoti į txt ar log bylų formatą su visiems prieinama nemokama programėle PsTool.

Pavyzdžiui SysEvent.evt bylos informacinis įspėjimas apie laiko sinchronizaciją:

System log on \\Video:

[9721] W32Time

Type: INFORMATION

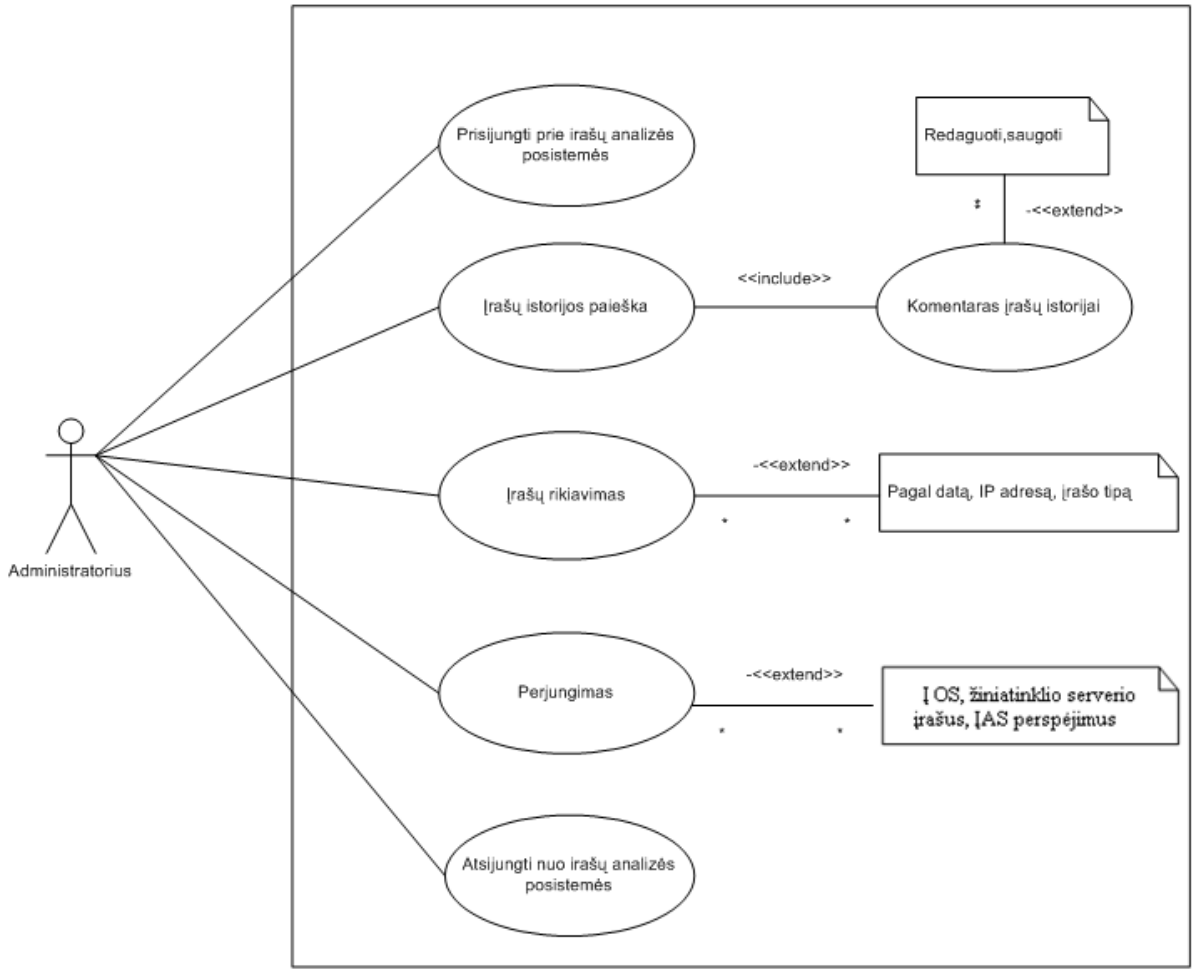
Computer: VIDEO

Time: 2010.04.20 10:24:56 ID: 35

The time service is now synchronizing the system time with the time source time.windows.com (ntp.m|0x1|192.168.0.1:123->207.46.197.32:123).

Įvykių tipai Windows sistemose skirstomi į: informacinius, įspėjimus, klaidas, pavykusius bei nepavykusius auditus.

Detaliau apie žiniatinklio serverio įrašų istoriją bei jos pateikimą detalizuojama panaudos atvejų diagramomis, veiklos diagramomis, vartotojo sąsajos langais. Administratoriaus veiklos su serveriu ir įrašų analizės posisteme pateiktos 17 paveiksliuke.

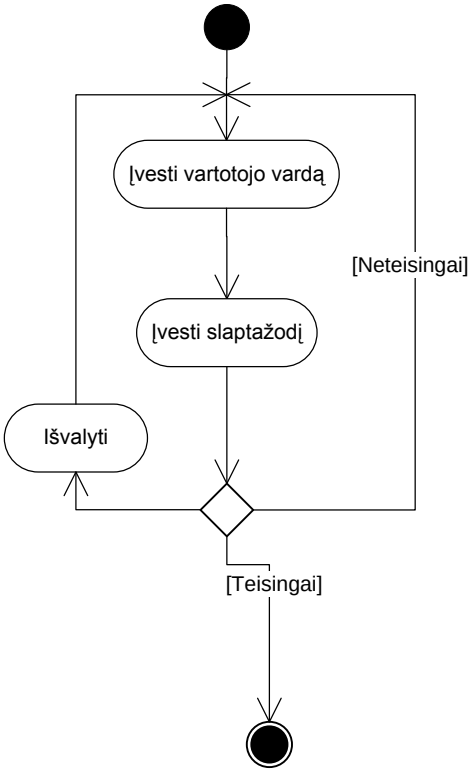


17 pav. Administratoriaus įrašų analizės posistemės panaudojimų atvejų diagrama

Administratoriaus veiklos diagramos

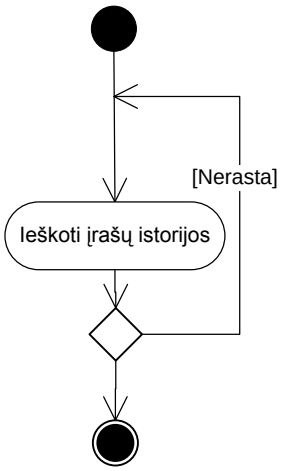
Šio tipo diagramos taip sistemos panaudojimo galimybes, viskas yra labiau detalizuota, parodomi veiksmų eiliškumo principai ir išsišakojimai atlikus konkrečius veiksmus. Pažvelgus į šias diagramas, geriau suprantamas veiksmų eiliškumas ir papildomos galimybės. Diagramos yra skirtos administratoriaus veiklai žiniatinklio puslapyje, kuriame pateikiama įrašų istorija (įrašų analizės posistemėje).

Administratorius jungiasi prie interneto svetainės, kur jam reikia įvesti vartotojo vardą bei slaptažodį. Sėkmingai įvedęs prašomus duomenis administratorius naudojasi sistema. Nesėkmingo įvedimo atveju administratoriaus paprašoma pakartotinai suvesti duomenis. Matydamas, kad įvedimo metu padarė klaidą, administratorius gali išvalyti laukus ir pradėti iš naujo. Prisijungimo veiklos diagrama pateikta 18 paveiksliuke.



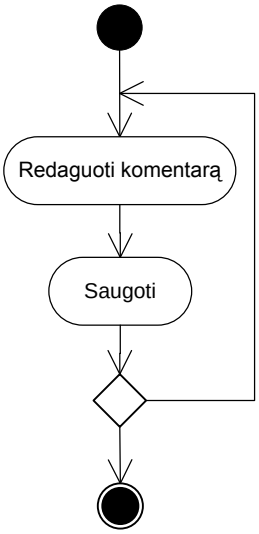
18 pav. Prisijungimo veiklos diagrama

Norint peržvelgti įrašus reikia juos susirasti, suradus galima vykdyti kitus veiksmus, neradus paieška prasideda iš naujo - administratorius įveda raktinius žodžius. Įvykių istorijos paieškos veiklos diagrama pateikta 19 paveiksliuke.



19 pav. Įvykių istorijos paieškos veiklos diagrama

Suradęs atitinkamą įvykių istoriją, administratorius gali pridėti prie jos komentarą bei jį išsaugoti. Išsaugojęs gali iš naujo pakeisti komentaro turinį. Komentaro rašymo veiklos diagrama pateikta 20 paveiksliuke.

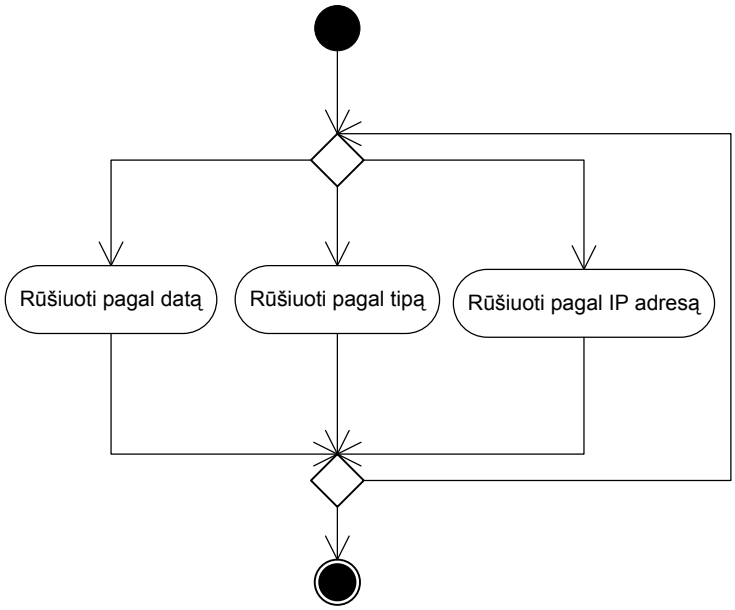


20 pav. Komentaro rašymo veiklos diagrama

Suradęs reikiamą įvykių istoriją administratorius gali ją rūšiuoti pagal 3 kriterijus.

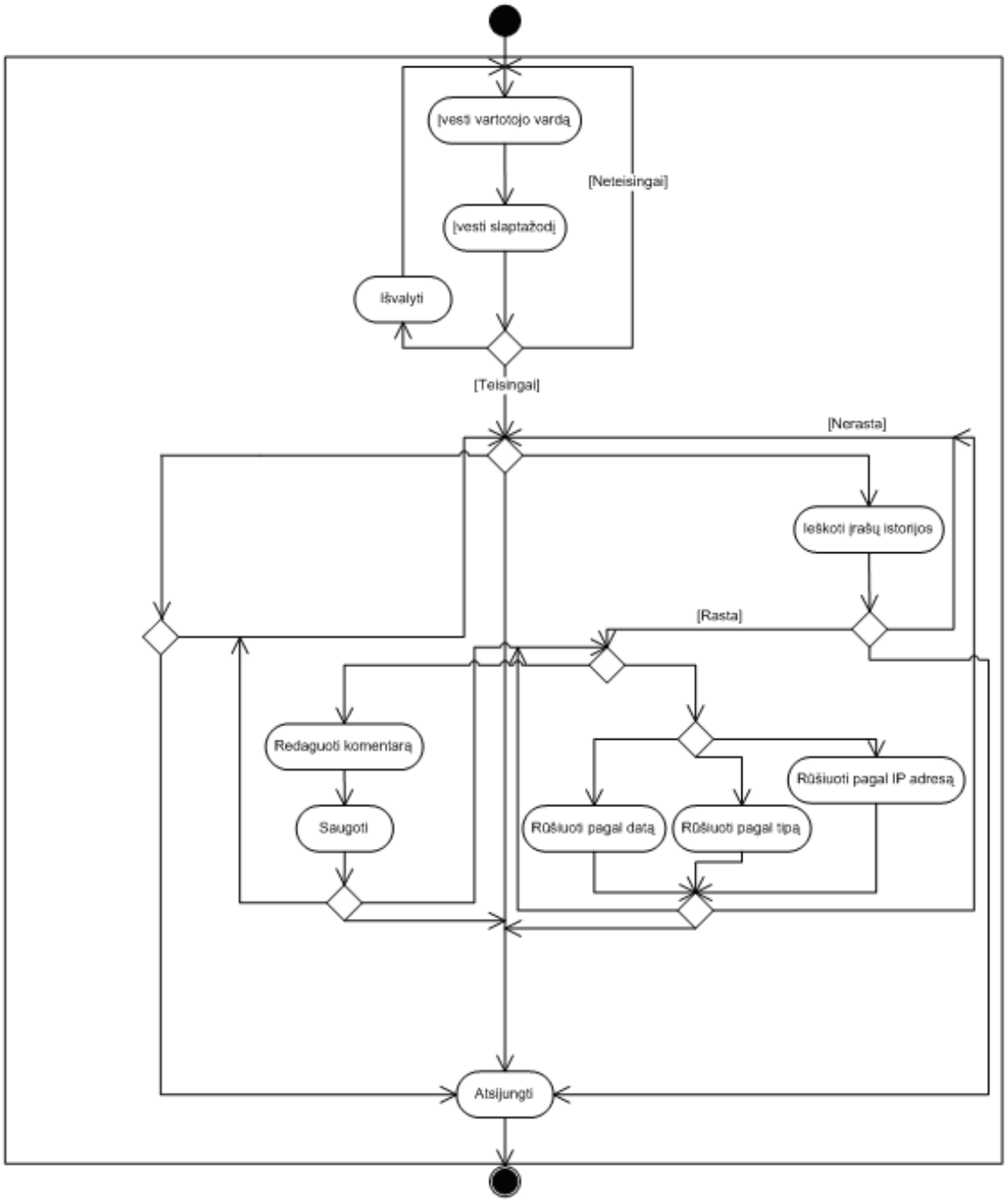
1. Pagal kreipimosi į puslapį datą
2. Pagal įvykio HTTP užklausos tipą
3. Pagal besikreipiančio IP adresą

Pasirinkus vieną iš 3 galima pakeisti rūšiavimo kriterijų. Rūšiavimo pasirinkimo veiklos diagrama pateikta 21 paveiksliuke.



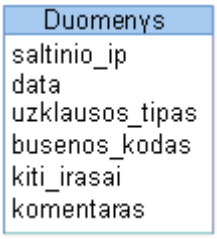
21 pav. Rūšiavimo pasirinkimo veiklos diagrama

Sudėję šias visas veiklas į vieną diagramą gauname administratoriaus veiklos diagramą. Diagrama pateikta 22 paveiksliuke.



22 pav. Administratoriaus veiklos diagrama

Įrašų analizės posistemė turi savo duomenų bazę, kuri turi 3 lenteles (pateikiamas žiniatinklio serverio įrašams kaupti skirtos lentelės pavyzdys), struktūra pateikta 23 paveiksliuke ir lentelės duomenų tipai bei ilgiai pateikti 5 lentelėje.



23 pav. Įrašų analizės posistemės duomenų bazės lentelė (žiniatinklio serverio įrašų)

Lentelės laukų paskirtis:

saltinio_ip – saugomas vartotojo IP adresas;

data – saugoma įrašo atsiradimo data;

uzklausos_tipas – saugomas užklausos tipas (GET, PUT ir pan.);

busenos_kodas – saugomas HTTP būsenos kodas;

kiti_irasai – visi likę įrašų failo duomenys;

komentaras – administratoriaus komentaras prie tam tikro įrašo.

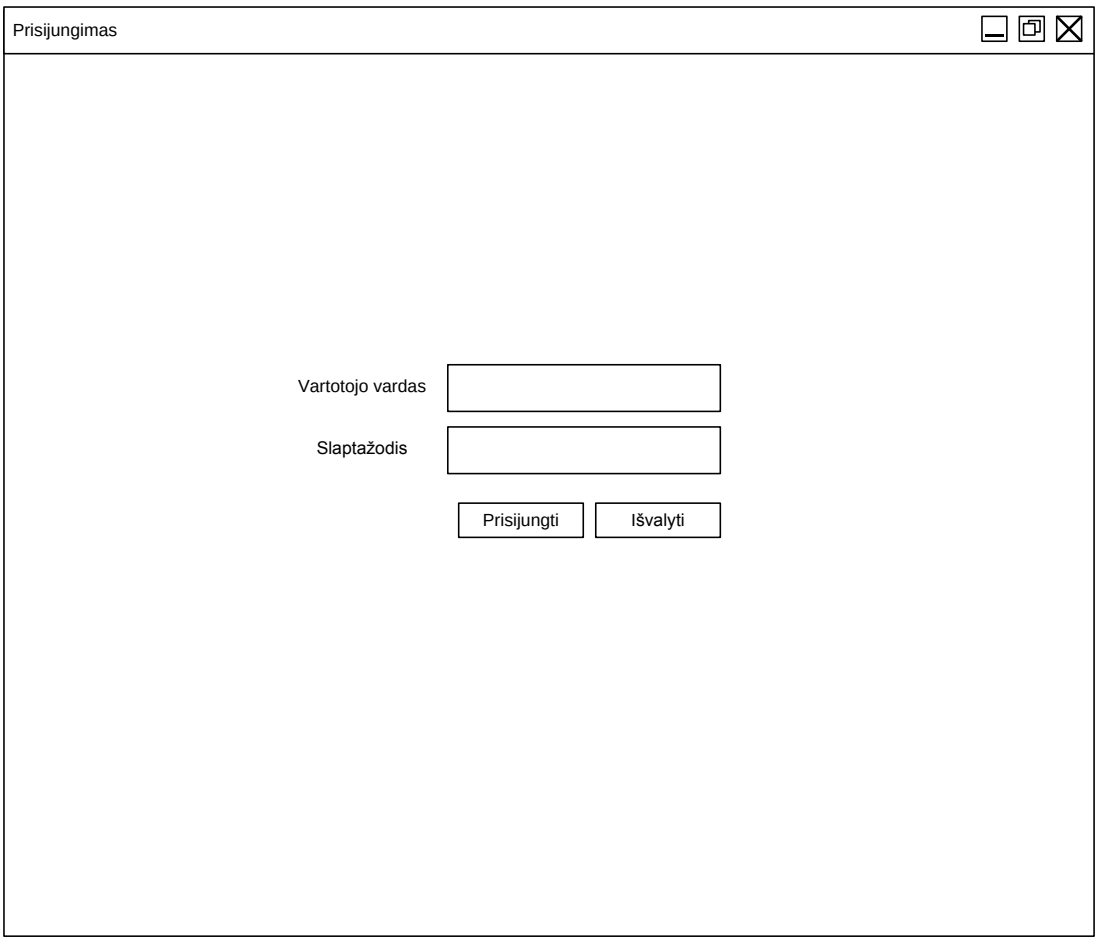
Duomenų laukai su tipu ir pavyzdžiu pateikiami 5 lentelėje.

4 lentelė Duomenų tipai, ilgiai, bei lauko pavyzdys

Duomenų laukas	Duomenų tipas (ilgis)	Pavyzdys
saltinio_ip	nvarchar (15)	*192.168.114.201*
data	datetime	*03/20/01, 7:55:20*
uzklausos_tipas	nvarchar (8)	*GET*
busenos_kodas	integer	*200*
kiti_irasai	nvarchar (255)	*-,W3SVC2, SERVER, 172.21.13.45 4502, 163, 3223, 0, /DeptLogo.gif, -,*
komentaras	nvarchar (255)	*Keista elgsena*

Įrašų analizės posistemės langų projektai

1. Prisijungimo lango projektas pateiktas 24 paveiksliuke.



24 pav. Įrašų analizės posistemės prisijungimo langas

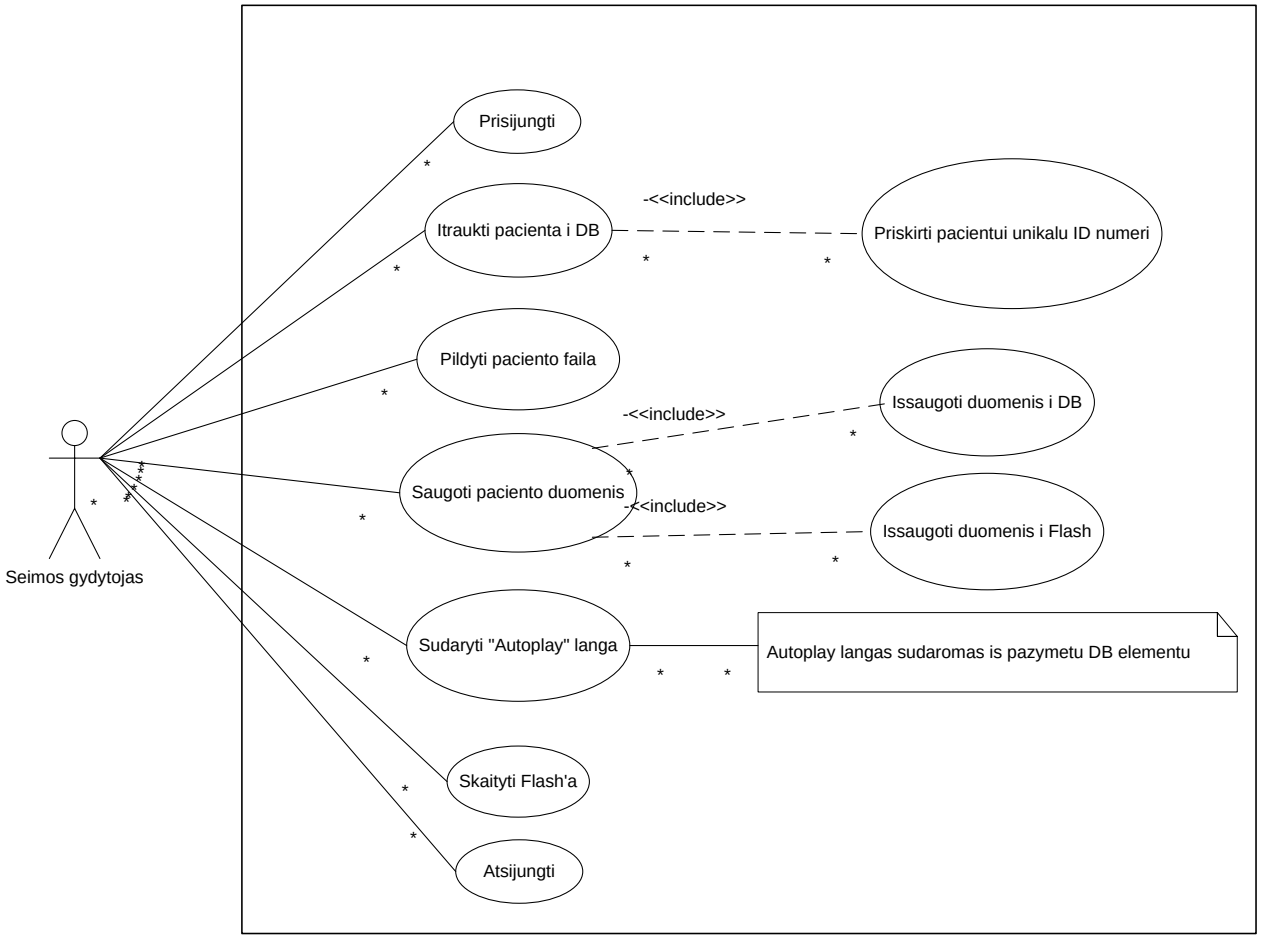
2. Identifikuotam vartotojui pateikiamas pagrindinis sistemos langas, kuriame jis gali pasirinkti tam tikrus veiksmus (iliustracija pateikta 25 paveiksliuke):

    - Ieškoti įrašų
    - Rikiuoti įrašus pagal datą, tipą, IP adresą
    - Išsaugoti komentarą ar jį redaguoti
    - Atsijungti

25 pav. Pagrindinis įrašų analizės posistemės langas

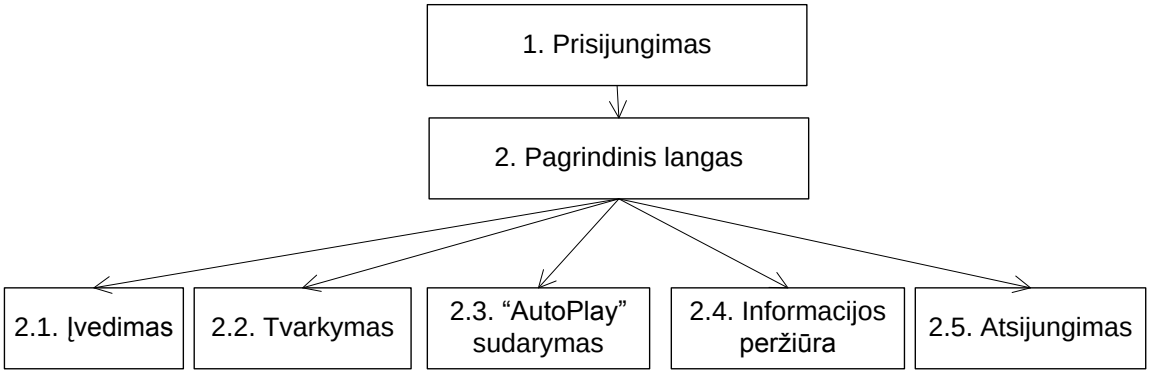
3.5.3 Produkcinis serveris

Produkciniame serveryje turėtų būti kopijų šaltinis klonuojant į „medaus puodynės“ serverį bei vėliau jį modifikuojant. Produkciniame serveryje diegiama saugoma svetainė. Konkrečiu atveju tai gali būti sveikatos duomenų saugojimo sistema, kuri vien sistemos pavadinimas įsilaužėliui turėtų kelti „apetitą“. Panaudojimų atvejai gydytojui pateikti 26 paveiksliuke.



26 pav. Sistemos panaudojimo atvejai gydytojui [1]

Įrašų analizės posistemės langų planas, kuris nurodo langų hierarchiją bei kiekybę, pateikiamas 27 paveiksliuke.



27 pav. Sveikatos duomenų saugojimo sistemos langų planas [1]

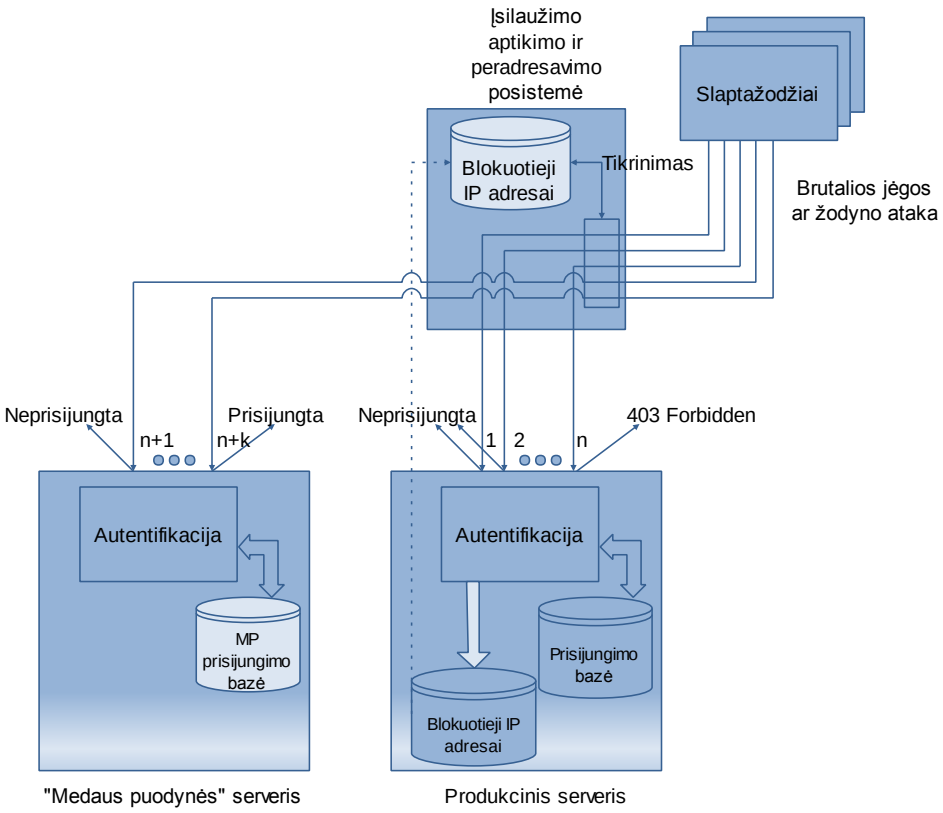
Paleidus sistemą matomas prisijungimo langas. Sėkmingai prisijungus matome pagrindinį sistemos langą, kuriame galime pasirinkti veiksmą. Priklausomai nuo pasirinkimo, pateikiamas paciento įvedimo arba paciento istorijos langas.

Prisijungimo langas (1): įvedamas vartotojo vardas ir slaptažodis. Pagrindinis sistemos langas (2). Jame galima pasirinkti naujo paciento įvedimą, jo duomenų tvarkymą, galima sudaryti “AutoPlay” langą, peržiūrėti paciento duomenis ir atsijungti nuo sistemos. Naujo paciento įvedimo langas (2.1): čia įvedama informacija apie pacientą. Paciento istorijos pildymo langas (2.2): kiekvieną kartą pacientui apsilankius pas gydytoją informacija yra papildoma, sudaromas “AutoPlay” langas (2.3). Taip pat naudojamas informacijos peržiūrėjimo langas (2.4) bei atsijungimas (2.5). [1]

Žiniatinklio serveryje patariama įdiegti žiniatinklio ugniasienę arba modifikuotą autentifikacijos sistemą, taip pat nuo slaptažodžio atakų apsaugančią posistemę, šios posistemės pavyzdys bus nagrinėjamas detaliau.

Dažnai prieš slaptažodį gali būti nukreipta brutalios jėgos ataka, ataka paremta algoritmu, kuris nuosekliai vieną po kito tikrina kiekvieną sekos simbolį tol, kol jis sutampa su pirmuoju fragmento simboliu, toliau lygina fragmento ir sekos antruosius simbolius, po to trečiuosius, ir t.t. Jei bent vienoje vietoje simboliai nesutampa, reikia pereiti prie sekančio sekos simbolio ir tikrinimą pradėti iš naujo.

Dabar paplitusiose programose šis algoritmas yra pakankamai optimizuotas ir veikia ganėtinai greitai. Tad apie bandančiuosius naudoti šiuos įrankius reikia kaupti informaciją (IP adresus) bei panaudoti juos nukreipiant per įsilaužimo aptikimo ir peradresavimo posistemę į „medaus puodynę“. Nukreipimo schema pateikta 28 paveiksliuke.



28 pav. Įsilaužėlių nukreipimas į MP esant slaptažodžio atakoms

Į blokuotų IP adresų sąrašą patenka tie adresai, kurie viršina šias normas:

- Maksimalų prisijungimų skaičių vienu metu iš vieno IP adreso;
- Maksimalų užklausų skaičių per atitinkamą laiką

Viršinus šias normas galima uždėti blokavimą IP adresui nustatytam laikui ir parodyti įspėjamąjį pranešimą (Pavyzdžiui 403 Forbidden).

3.6 Išvados

Apibrėžtas modelis, kuriame detalizuota kaip įsilaužimo aptikimo ir peradresavimo posistemė nukreipia piktavališkus veiksmus į „medaus puodynės“ serverį, kur veikia „medaus puodynės“ svetainė (tikrosios svetainės kopija) . Lygegrečiai kaupiama informacija apie potencialų įsilaužėlį bei pateikiama įrašų analizės posistemėje. Sprendimas dėl nukreipimo į „medaus puodynę“ priimamas tikrinant du kriterijus: juoduosius IP sąrašus bei įsilaužimo aptikimo taisykles.„Medaus puodynės“ svetainės prisijungimo duomenų bazė yra modifikuota, tam, kad „įsileistų“ įsilaužėlį – išgautų iš jo kuo daugiau informacijos. Produkcinis serveris apsaugotas programų lygmens ugniasiene, pavyzdžiui, apsaugotas nuo

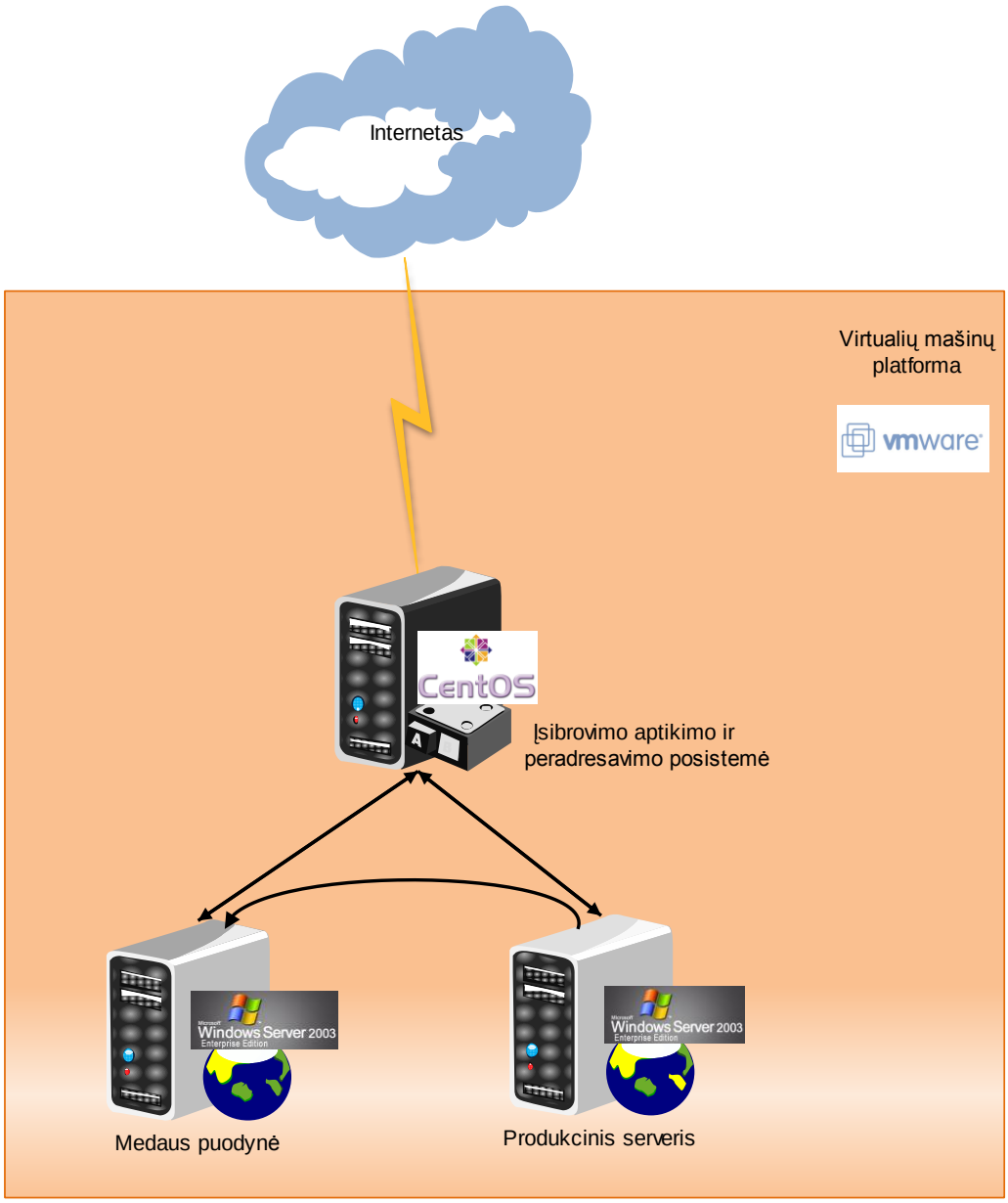
prieš slaptažodį nukreiptų atakų, kurios pagal detalizuotą schemą nukreipiamos į „medaus puodynės“ serverį. Svarbi modelio dalis yra įrašų analizės posistemė, skirta administratoriams, kur jie gali analizuoti koncentruotą įrašų istoriją iš 3 skirtingų vietų: įsilaužimo aptikimo sistemos, serverio OS bei žiniatinklio serverio. Detalizuotai aprašyta žiniatinklio serverio įrašų analizės posistemės dalis.

Apibrėžtos sistemos modelis iš dalies padeda apsaugoti tikrąjį serverį (paslaugą) bei suteikia galimybę rinkti ir analizuoti koncentruotą informaciją apie įsilaužėlio veiksmus, tuomet galima sukurti naujus apsaugos metodus.

4. SISTEMOS PROTOTIPO REALIZAVIMAS IR TYRIMAS

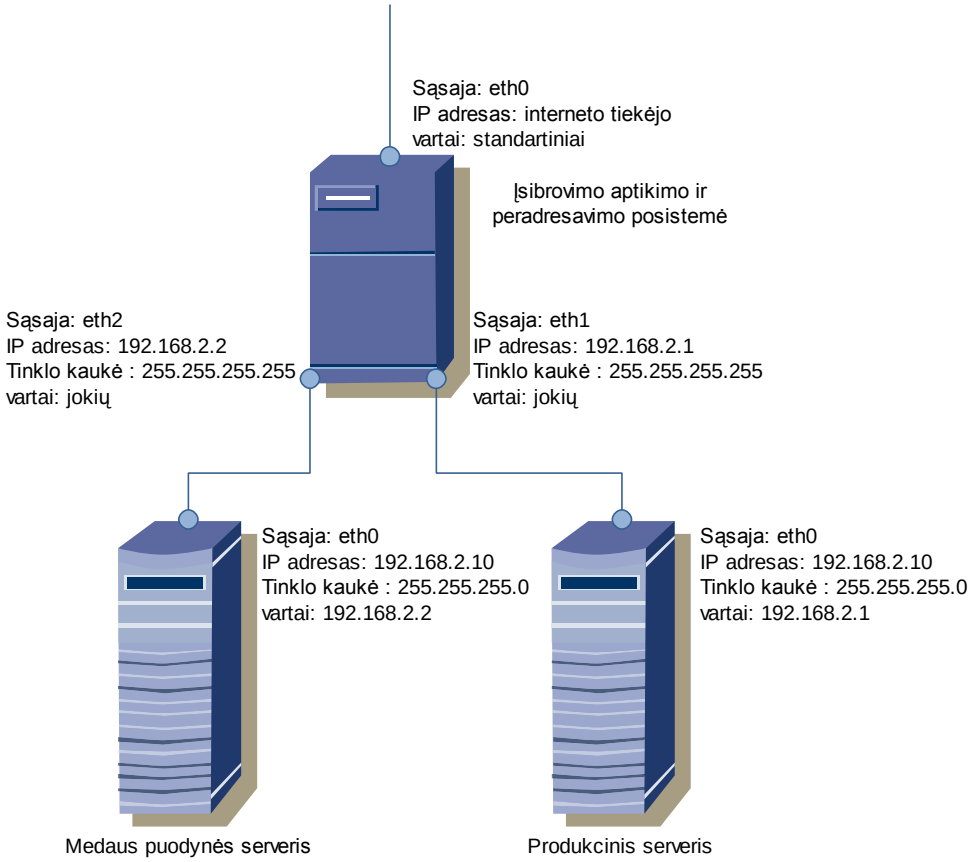
4.1 Sistemos prototipo architektūra

Siekiant imituoti produkcinį serverį, „medaus puodynės“ serverį bei peradresavimo posistemę, pasirinkta VMware Workstation programinė įranga (skirta virtualizavimui), kur sukurtos 3 virtualios mašinos: 2 su Windows Server 2003 operacine sistema bei viena su CentOS operacine sistema. Virtuali mašina, kurioje instaliuota CentOS, turi tris tinklo sąsajas, pirmoji skirta jungčiai su internetu, antroji – „medaus puodynei“, trečioji – produkciniam serveriui. Naudojamų programinių įrangų schema pateikta 29 paveiksliuke.



29 pav. Prototipe naudojamų virtualių mašinų operacinės sistemos

Visa ši sistema, siekiant tikroviškumo turėtų būti dislokuota demilitarizuotoje zonoje. Eksperimento metu kompiuteris su virtualiomis mašinomis prijungtas prie paprasto interneto tiekėjo tinklo ne demilitarizuotoje zonoje, kitomis sąsajomis su „medaus puodyne“ ir produkciniu serveriu. IP adresacijos konfigūracija reikalinga tokia, kokia pateikta 30 paveiksliuke, nes tik tokią tinklo adresaciją palaiko viena įsibrovimo aptikimo ir peradresavimo posistemės dalių - Bait and Switch. Ši tinklo struktūra suformuojama atlikus reikiamas konfigūracijas.



30 pav. Įrankio Bait and Switch IP adresacija [26]

Darbe virtualizavimui naudojamos įrangos VMware Workstation versija 6.5.1 build – 126130. Ši virtualių mašinų platforma instaliuota į nešiojamąjį kompiuterį, kurio pagrindinės charakteristikos pateiktos 6 lentelėje.

5 lentelė Eksperimente naudojamo kompiuterio charakteristikos

Modelis	Acer Aspire 5040
Procesorius	AMD Turion 64 Mobile 1,8 GHz
Darbinė atmintis	1 GB
Vaizdo akseleratorius	ATI Radeon Xpress 200M 128 MB (integruota)
Kietasis diskas	80 GB
Tinklo plokštė	Realtek RTL8169/8110
Bevielio tinklo plokštė	Atheros AR5005G
Operacinė sistema	Microsoft Windows XP SP3 (5.1.2600)

Virtualių mašinų konfigūracijos:

Microsoft Windows Server 2003 virtualios mašinos nustatymuose 128 MB skirti operacinės sistemos veiklai. Sukurtas virtualus diskas, kurio talpa 8 GB. Kadangi „medaus puodynės“ serverio virtuali mašina yra produkcinio serverio virtualios mašinos kopija, tad konfigūracija tokia pati.

Įsilaužimo aptikimo ir peradresavimo posistemė realizuota CentOS operacinėje sistemoje, kuriai skirta 512 MB darbinės atminties (kadangi didžioji dalis darbo vykdoma joje). Virtualiam kietajam diskui skirta 8 GB, sukurtos 3 virtualios tinklo sąsajos.

Virtualių mašinų ypatybės, pagal kurias galima atskirti, jog tai „medaus puodynė“ (virtuali mašina): [11]

- MAC (angl. Media Access Control address) adresas (VMware turi tam tikrą MAC adreso pradžią);
- Windows sistemose Add/Remove ir tam tikrose direktorijose matoma VMware programinė įranga;
- Įtartina informacija: kompiuteris turintis 1,8 GHz Procesoriaus turi tik 128 MB. darbinės atminties;
- Virtualios mašinos prietaisų pavadinimai dažnai turi VMware ar "virtual" žodžius.

Virtualios mašinos turi savo MAC adresą, kuris yra automatiškai priskiriamas kiekvienam tinklo įrenginiui, taip pat ir tinklo sąsajoms. Adresas priskiriamas kaip unikalus, tačiau virtualią perkėlus į kitą vietą ar keičiant pagrindines konfigūracijos opcijas jis priskiriamas iš naujo.

IEEE standartuose VMware išskirta MAC adresų grupės :

00-05-69-xx-xx-xx *00-0C-29-xx-xx-xx* *00-50-56-xx-xx-xx* [10]

Koks MAC adresas priskirtas Windows virtualioje mašinoje galima pažiūrėti

start - > run -> cmd įvedus komandą *ipconfig /all arba getmac*

Pakeisti šį adresą galima keičiant VMware konfigūracijos failus: Windows Server 2003 Enterprise Edition.vmx, Windows Server 2003 Enterprise Edition II.vmx bei Centos.vmx.

Reikia ištrinti eilutes :

ethernet0.addressType = "generated"
ethernet0.generatedAddress = "00:0c:29:19:87:f1"
ethernet0.generatedAddressOffset = "0"

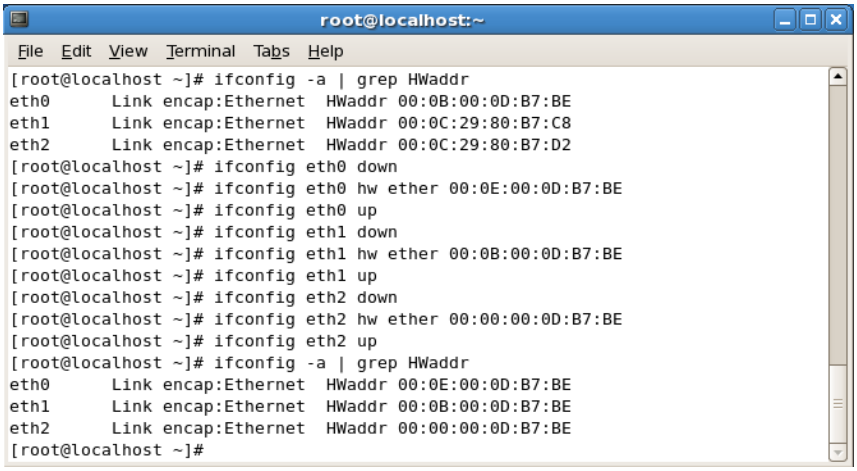
Bei įterpti naują eilutę (Centos.vmx atveju dar ir 1 ir 2 sąsajai):

ethernet[0].address = Naujas MAC adresas (XX.XX.XX.XX.XX.XX)

Esant nesėkmingam bandymui pakeisti konfigūraciją, gali neveikti virtuali mašina, tad pakeisti MAC adresą virtualioje mašinoje (Windows) galima ir grafinėje sąsajoje:

Start->Settings->Network Connections-> Local Area Connection->Properties->Configure->Advanced-> Network Address -> Value įvedama norima MAC adreso reikšmė.

Linux aplinkoje komandinėje eilutėje būtų atitinkamai kaip pateikta 31 paveiksliuke. Žinoma, keičiant derėtų pasidomėti apie MAC adresų grupes, kurios priklauso tam tikriems gamintojams, kad nekeltų įtarimo dėl virtualios mašinos naudojimo.



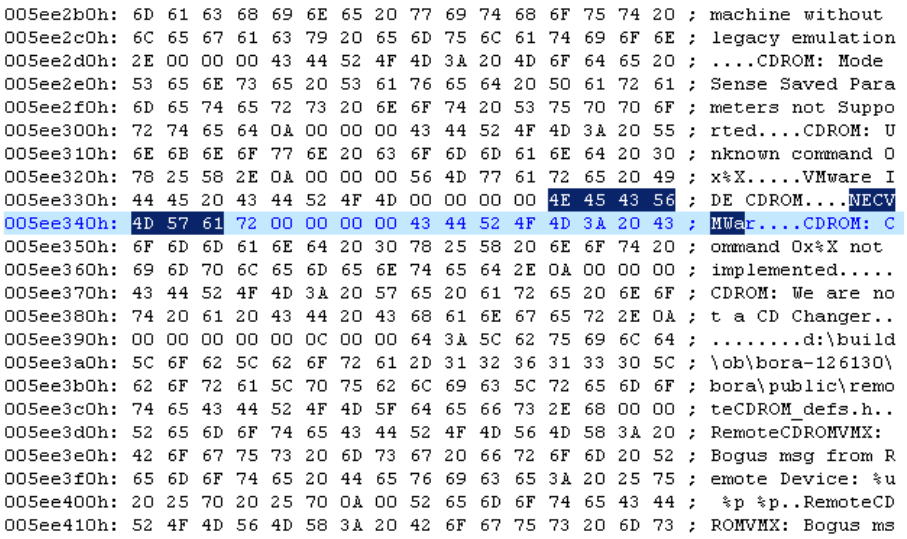
31 pav. MAC adresų keitimas CentOS komandinėje eilutėje

Dažnai virtualioje mašinoje būna įdiegta specifinė VMware programinė įranga ar įsilaužėlio sekimo įranga. Ją derėtų užmaskuoti. Kadangi „medaus puodynė“ realizuota Windows aplinkoje, derėtų pasirūpinti Add or remove programs rodomu vaizdu (įdiegta programine įranga). Tai galima atlikti koreguojant registrus. Veiksmų seka: *Start->Run->regedit->OK->HKEY_CLASSES_ROOT->Installer->Products* tuomet bus matomi du ilgų skaičių sekų katalogai, kuriuose, radus reikiamos programinės įrangos pavadinimą, reikia ištrinti visą tą katalogą, kuriame radome norimos programinės įrangos pavadinimą (product name). Eksperimente tai katalogas pavadinimu *2BC5FF3A53BF85647815E9EDD1563BAA*.

Vienas iš pagrindinių įtartinų dalykų įsilaužėliui būna keisti kompiuterio parametrai. Eksperimento metu - tai kompiuteris su 1,8 GHz procesoriumi ir 128 MB darbinės atminties bei 8 GB kietasis diskas. Taigi, kuriant virtualią mašiną derėtų skirti bent 512 MB darbinės atminties bei 20 GB kietajam diskui. Darbinės atminties kiekį galima pakeisti prieš paleidžiant virtualią mašiną darbui, o kietojo disko parametrus kuriant virtualią mašiną. VMware įrenginiams virtualioje mašinoje suteikia pavadinimus *VMware virtual IDE Hard Drive, NECVMWare VMware IDE CDR10, VMware Accelerated AMD PCNET Adapter* ir

kitus. Juos pakeisti galima naudojantis *Ultraedit* ar kita šešioliktainį kodą redaguojančia programėle. Reikia pakeisti dvejetainio kodo failą *vmware-vmx.exe*.

Atsidarius šį failą matomas turinys pateiktas 32 paveiksliuke.



32 pav. VMware-vmx.exe failo turinio keitimas

Pakeičiame užrašą *NECVMWare* į *NEC 2000* bei kietojo disko pavadinimą.

CentOS saugumo didinimas

Saugumo didinimas CentOS reikalingas siekiant apsaugoti peradresavimo posistemę nuo įsilaužėlių. Geriausias saugumo skylių užtaisymo būdas yra atnaujinimų parsiuntimas bei įdiegimas. Šioje CentOS versijoje yra programų (tuo tarpu ir OS) atnaujinimo modulis randamas *Aplication->System Tool-> Software update*. Eksperimento metu modulis rado nemažai atnaujinimų ir papildinių (apie 70).

Taip pat derėtų išjungti įvairias nenaudojamas paslaugas (pateiktos 7 lentelėje) iki kol jų prireiks. Tai galima padaryti su komanda :

[root@localhost]# /sbin/chkconfig paslaugos pavadinimas off

Žvaigždutėmis (*) paženklintos tinklinės paslaugos, kurias išjungti reikia pirmiausiai.

6 lentelė Dalis CentOS paslaugų [17]

anacron	haldaemon	messagebus
apmd	hidd	microcode_ctl
autofs`	hplip*	pcscd
avahi-daemon*	isdn	readahead_early
bluetooth	kdump	readahead_later
cups*	kudzu	rhnsd*
firstboot	mcstrans	setroubleshoot
gpm	mdmonitor	xfs

CentOS saugumui padidinti puikiai tinka Bastille programinė įranga.

Kadangi eksperimentas yra vykdomas namuose, kur interneto tiekėjas skiria vieną adresą, jo tikslas yra slėpti produkcinį bei medaus puodynės serverį nuo išorės. Tam panaudota CentOS esanti iptable ugniasienė tam tikroms taisyklėms bei adresų transliacijai (angl. network address translation - NAT) sudaryti, kurios užtikrintų reikiamą Bait n Switch su Snort programų veikimą. Naudojamos 3 paprastos taisyklės:

- Srautas iš įsilaužimo aptikimo ir peradresavimo mechanizmo įėjimo sąsajos eth0 ir „medaus puodynės“ sąsajos galimas bet kur ir bet kuris protokolas.
- Srautas iš bet kurio šaltinio į „medaus puodynės“ serverį bet kuriuo protokolu leidžiamas.
- Srautas iš bet kurio šaltinio į bet kurį šaltinį bet kuriuo protokolu leidžiamas.

Adresų transliacija:

- Srautas iš „medaus puodynės“ serverio išeinantis į bet kurį adresą bei turintis bet kokią paslaugą transliuojamas naudojant eth0 sąsają.
- Srautas iš bet kurio šaltinio į eth0 sąsają transliuojamas į „medaus puodynę“.

Taisyklės pateiktos **Priede 1 iptable taisyklės**.

Bait n Switch, jo sąsajos ir Snort konfigūravimas

Turėdami snort 2.8.0 versiją bei Bait n Switch 2.2. juos išarchyvavę – įdiegiame.

Pradžioje atliekama Bait n Switch konfigūracija, papildomas Snort diegimas (Papildomai įdiegta gcc biblioteka).

CentOS terminalo lango vaizdas pateiktas **Priede 2 Terminalo lango turinys**.

Snort senos versijos instaliavimas buvo komplikuotas dėl senų bibliotekų nesutapimo. Teko pasirinkti naujesnę versiją. Jos įdiegimas:

[root@localhost snort]#./configure

[root@localhost snort]# make

[root@localhost snort]# make install

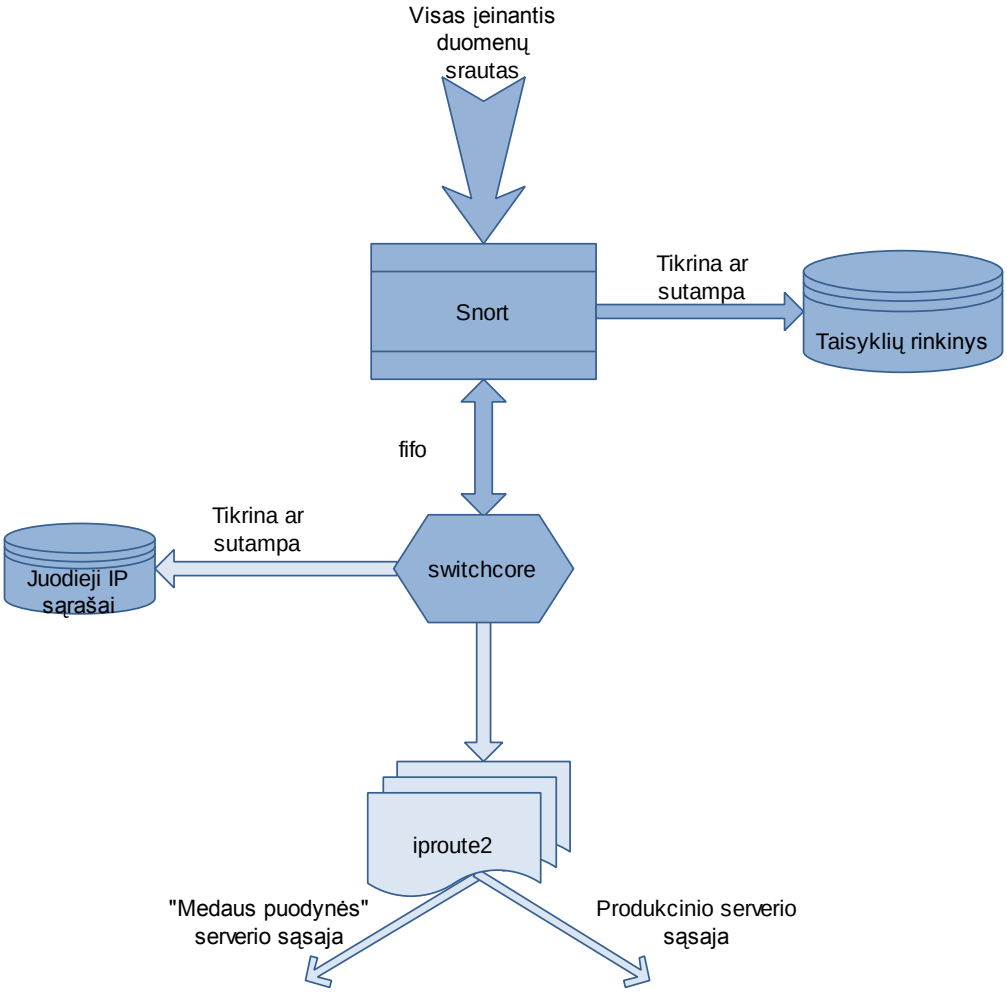
Įsilaužimo aptikimo ir peradresavimo posistemės paleidimas vykdomas taip:

[root@localhost bns]#./switchcore

[root@localhost snort]# snort –c /direktorija/ rules.script

rules.script - tai taisyklės pagal kurias peradresuojama į medaus puodynę.

Taigi atlikus visus konfigūracinius veiksmus gautas modelis su konkrečiomis priemonėmis. Visos dalys, išskyrus įsilaužimo aptikimo sistemą – Snort ir jo taisyklių rinkinį galima vadinti Bait n Switch komponentais. Naudojamos priemonės įsilaužimo aptikimo ir peradresavimo posistemei realizuoti pateiktos 33 paveiksliuke.



33 pav. Įsilaužimo aptikimo ir peradresavimo schema su konkrečiais įrankiais

Juodųjų IP sąrašų sudarymas yra ne vienkartinis procesas,o ciklišKas. Juos galima rinkti pačiam administratoriui naudojantis DenyHost 2.6 programine įranga, kuri įdiegiama į Linux

operacinę sistemą, eksperimente - tai CentOS. Ši programa renka (įrašo į failą) ir blokuoja IP adresus, kurie įvairiais būdais bando prisijungti prie serverio per SSH protokolą (22 prievadas). DenyHost 2.6 diegimas:

Iš pradžių parsiunčiama ir įdiegiama Python 2.5 versija, tai atliekama su komandomis:

[root@localhost Python]# ./configure

[root@localhost Python]# make

[root@localhost Python]# make install

Tuomet instaliuojamas DenyHost 2.6:

[root@localhost DenyHost 2.6]# python setup.py install

[root@localhost DenyHost 2.6]# cp denyhost.cfg-dist denyhost.cfg

[root@localhost DenyHost 2.6]# cp deamon-control-dist deamon-control

[root@localhost DenyHost 2.6]# vi deamon-control

Pakeičiame failų kelius į tikruosius (priklauso nuo OS):

DENYHOSTS_BIN = "/usr/bin/denyhosts.py"
DENYHOSTS_LOCK = "/var/lock/subsys/denyhosts"
DENYHOSTS_CFG = "/usr/share/denyhosts/denyhosts.cfg"

[root@localhost DenyHost 2.6]# ./ deamon-control start /var/log/denyhost

Surinktieji juodieji IP sąrašai saugomi faile esančiame etc/hosts.deny

Taip pat galima pasinaudoti susistemintais duomenimis, kurie pateikti internete. Duomenų pavyzdys iš stats.denyhosts.net pateiktas 8 lentelėje.

7 lentelė Juodieji IP sąrašai

IP adresas	vardas	šalis	pirmą kartą pastebėtas	blokuotas kartų
212.18.195.102	102-195-018-212.ip-addr.teresto.net	Vokietija	Dec 24, 2009 08:54 AM	5304
85.17.200.81	d85.colo.logol.ru	Olandija	Jan 29, 2010 05:44 PM	3318
92.46.123.11	92.46.123.11	Kazachstanas	Feb 04, 2010 10:35 PM	3213
85.21.139.69	zeon2.tdzr.ru	Rusija	Jan 02, 2010 11:04 AM	3123
218.56.61.114	218.56.61.114	Kinija	Oct 13, 2008 12:40 PM	2572
217.160.16.156	217.160.16.156	Vokietija	Oct 02, 2009 11:45 AM	2535
85.17.90.10	ice.edwh.net	Olandija	Nov 06, 2009 09:38 PM	2368
59.108.230.130	59.108.230.130	Kinija	Dec 23, 2009 05:12 PM	2034
220.90.134.2	220.90.134.2	Korėja	Jan 13, 2010 03:52 AM	1866
61.129.60.23	61.129.60.23	Kinija	Jan 07, 2009 07:52 PM	1861

Kelios panašios programėlės, kurios turi savų privalumų ir trūkumų tai Fail2Ban, BlockHosts, Blacklist.

Taisyklių sudarymas

Taisyklės sudaromos Snort įsilaužimo aptikimo sistemai. Tai nebus galutinis administratoriui tinkantis taisyklių rinkinys, kuris padės užtikrinti originalios sistemos saugumą. Tai tik specializuotų „medaus puodynei“ taisyklių sudarymo principų paaiškinimas. Kurių laikantis galima sukurti pilną taisyklių rinkinį, kuris sumažins įsilaužimo į tikrąjį serverį tikimybę. Prieš pradėdamas vykdyti ar bandyti atakas įsilaužėlis turi atlikti serverio saugumo spragų auditą, tam naudojamos automatizuotos skanavimo priemonės: Nikto, Nessus ir kt. Taigi norint, kad įvykiai būtų nukreipti į „medaus puodynę“ derėtų nurodyti taisykles, kurios aptinka vykdomą skanavimą bei sugeneruoja pavojaus pranešimą. Daugelį taisyklių galima parsisiųsti iš oficialaus Snort puslapio.

Nikto susideda iš daug įvairių testavimo įrankių, tad kiekvienas įrankio veiksmas aprašomas skirtinga taisykle, keletas taisyklių pavyzdžių:

Taisyklė aptinka TCP protokolu serveryje vykstantį vieningojo resurso identifikatoriaus doc ieškojimą. Galima įvesti ir kitų žodžių: password, pass, admin ir kt. [23]

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC /doc/ access"; flow:to_server,established; uricontent:"/doc/"; nocase; metadata:service http; reference:bugtraq,318; reference:cve,1999-0678; classtype:web-application-activity; sid:1560; rev:8;)

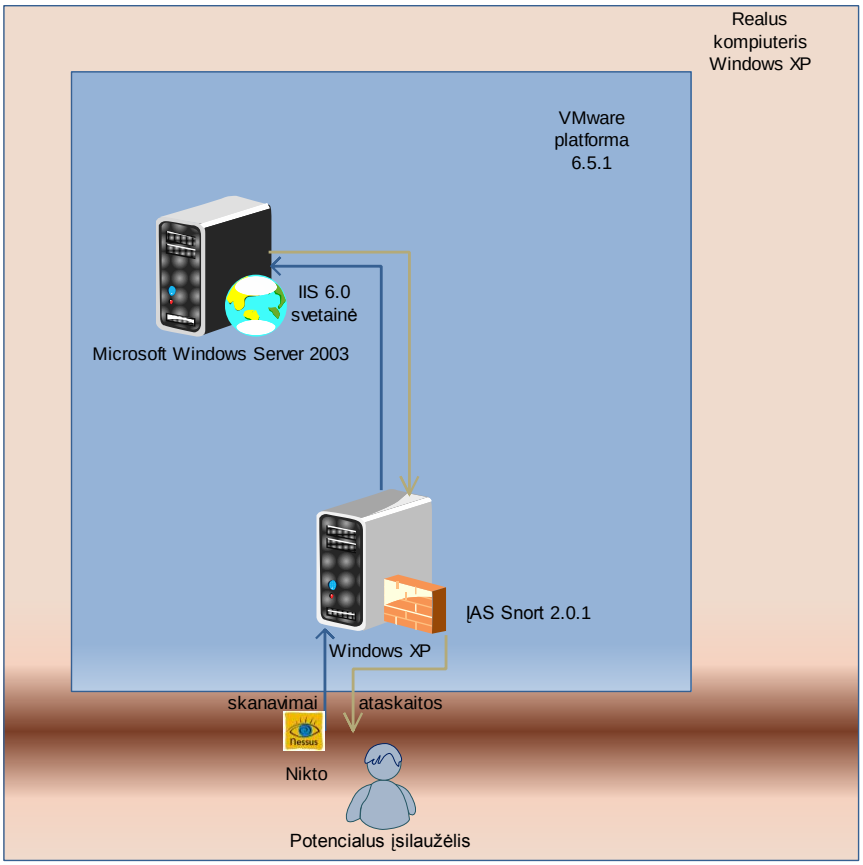
Panašaus tipo taisyklės, ieško robots.txt bei IIS .cnf (konfigūracinių) failų. [23]

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-MISC robots.txt access"; flow:to_server,established; uricontent:"/robots.txt"; nocase; metadata:service http; reference:nessus,10302; classtype:web-application-activity; sid:1852; rev:4;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS .cnf access"; flow:to_server,established; uricontent:".cnf"; nocase; metadata:service http; reference:bugtraq,4078; reference:cve,2002-1717; reference:nessus,10575; classtype:web-application-activity; sid:977; rev:15;)

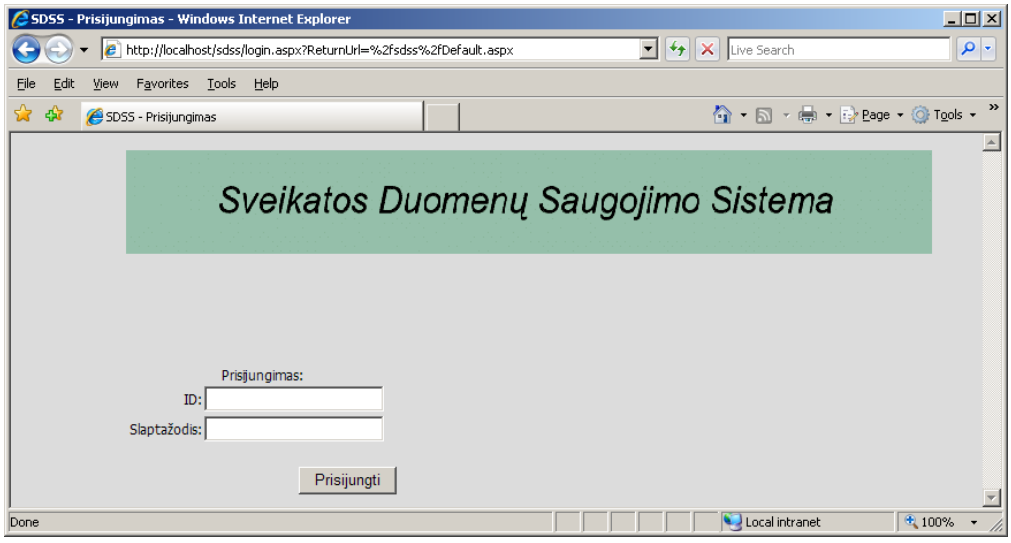
4.2 Sistemos prototipo tyrimas

Kadangi modelis pilnai nerealizuotas, tad daroma prielaida, jog aptikus nepageidaujamą įvykį, bus sugeneruotas perspėjimas bei šio įvykio šaltinio IP adresas nukreipiamas į „medaus puodynės“ serverį bei pateikiami perspėjimai, bei įrašų istorija puslapyje. Tyrimui naudojama supaprastinta modelio schema: serveris, kuriame paleista svetainė; serveris, kuriame instaliuota įsilaužimo aptikimo sistema Snort, bei atakuotojo kompiuteris. Atakuotojo kompiuteris – tikrasis kompiuteris, o abu serveriai virtualizuoti panaudojant VMware programinę įrangą. Potencialus įsilaužėlis iš realaus kompiuterio siunčia užklausas į virtualią mašiną žiniatinklio serveriui, šis atsako, visa tai stebi įsilaužimo aptikimo sistema. Darbo schema pateikta 34 paveiksliuke.

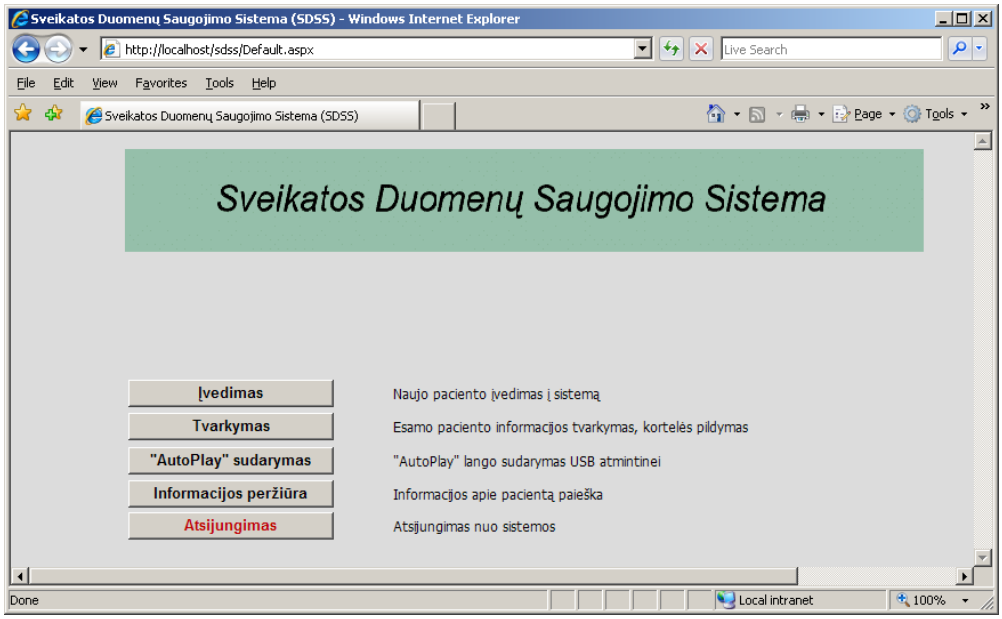


34 pav. Tyrimui naudojamų priemonių schema

Windows Server 2003 mašinoje įdiegta sveikatos duomenų saugojimo sistema, sistemos langų pavyzdžiai pateikti 35 ir 36 paveikslėliuose. 35 Paveikslėlyje pateiktas sistemos prisijungimo langas, 36 paveikslėlyje pateiktas pagrindinis sistemos darbo langas.



35 pav. Sveikatos duomenų saugojimo sistemos prisijungimo langas[1]



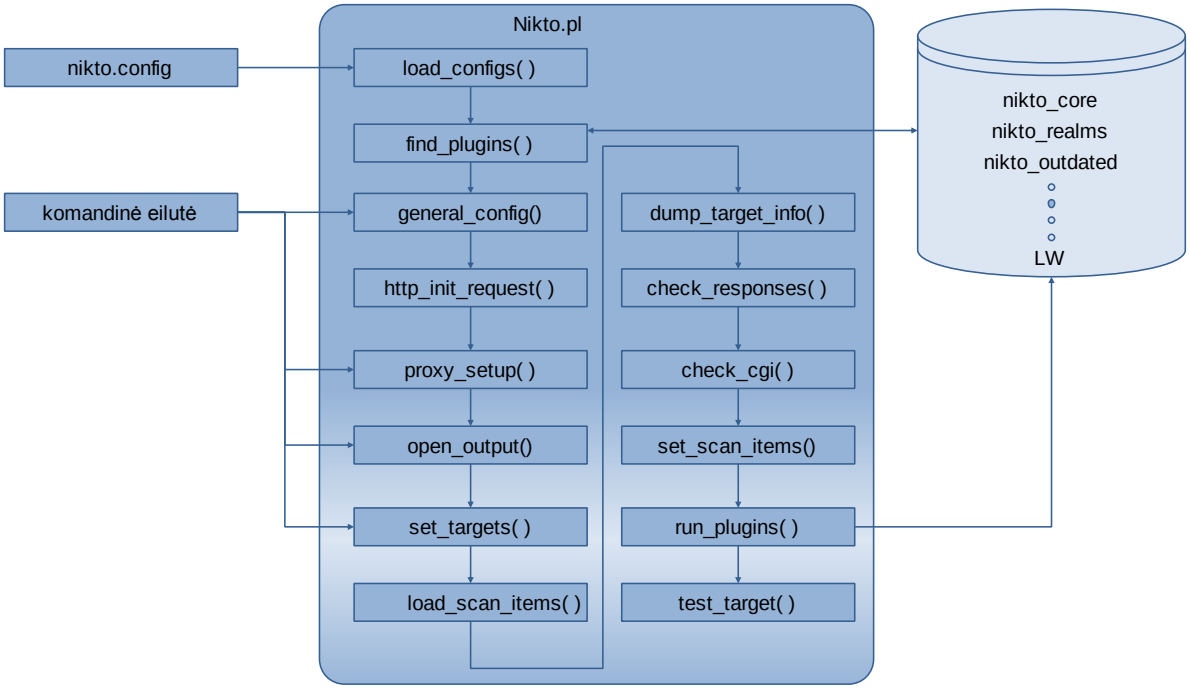
36 pav. Sveikatos duomenų saugojimo sistemos pagrindinis langas [1]

Dažnai potencialų įsilaužėlį galima identifikuoti iš vykdomų tinklo ar serverio skanavimų. Šiems skanavimams naudojamos įvairios automatizuotos priemonės: Nikto, Nessus, Paros ir kt. Bus atlikti keli skanavimai Nikto bei Nessus priemonėmis.

Nikto - tai atviro kodo žiniatinklio serverio skaneris, pasižymintis išsamių testų rinkiniais, tikrina programinės įrangos atnaujintumą, serverio konfigūraciją, HTTP opcijas, nesaugius failus ir programas bei daugelį kitų silpnybių.

Paleidus programą vykdomas metodas *load_configs()*, kuris užkrauna pradinius nustatymus iš nikto.config failo. Tuomet surandami įskiepiai bei įvedama iš komandinės eilutės pagrindiniai nustatymai, proxy nustatymai bei išvedimo formatas. Su *set_targets()*

metodu nustatomi taikiniai su specifiniais prievadais. Tuomet funkcija *dump_target_info()* parodo taikinio informaciją ir tikrina gerų ir blogų HTTP užklausų atsakus (*check_responses()*). Vėliau tikrinamos pagrindinės CGI (angl. Common Gateway Interface) direktorijos. Tuomet leidžiami įskiepiai ir demonstruojamas prieš tai rastas rezultatas. Įskiepiuose aprašyti pažeidžiamumų tikrinimo metodai. Pavyzdžiui, *nikto_outdated* įskiepis tikrino žiniatinklio serverio versiją; *nikto_realms* mėgina autentifikacijoje standartinius vardus. [6] Veikimo schema pateikta Pav. 37.



37 pav. Nikto veikimo schema

Testuojamas serveris kuo mažesniame laiko intervale, tačiau įrašų istorijoje akivaizdžiai matomas, tad puikiai tinka ĮAS perspėjimams tikrinti. Nikto skanavimo priemonės diegimas Windows XP aplinkoje (realiame kompiuteryje) yra ganėtinai sudėtingas, kadangi tai yra Unix tipo operacinėms sistemoms skirta priemonė. Be Nikto papildomai reikalingos priemonės:

- Active Perl;
- Microsoft Visual C++ 2008 Redistributable Package;
- Nmake;
- MiniGW;
- OpenSSL;
- Perl SSL module Net_SSLey.

Įdiegus šias priemones į atitinkamas direktorijas būtina nurodyti failų kelius. Tai padaroma My Computer-> System Properties -> Advanced->Environment Variables tuomet skiltyse User ir system variables PATH įkeliami suinstaliuotų priemonių failų keliai.Tuomet naudojantis Command Prompt nebūtina eiti į direktoriją, kurioje yra įdiegta priemonė.

Tuomet instaliuojama Perl SSL:

C:\Temp\Net_SSLeay.pm-1.25> perl Makefile.PL -windows C:/OpenSSL
C:\Temp\Net_SSLeay.pm-1.25> nmake
C:\Temp\Net_SSLeay.pm-1.25> -nmake install

Atliekame atnaujinimą bei paleidžiame parametrais papildytą skanavimą, kuris išsamiau testuoja bei pateikia rezultatus ekrane bei faile

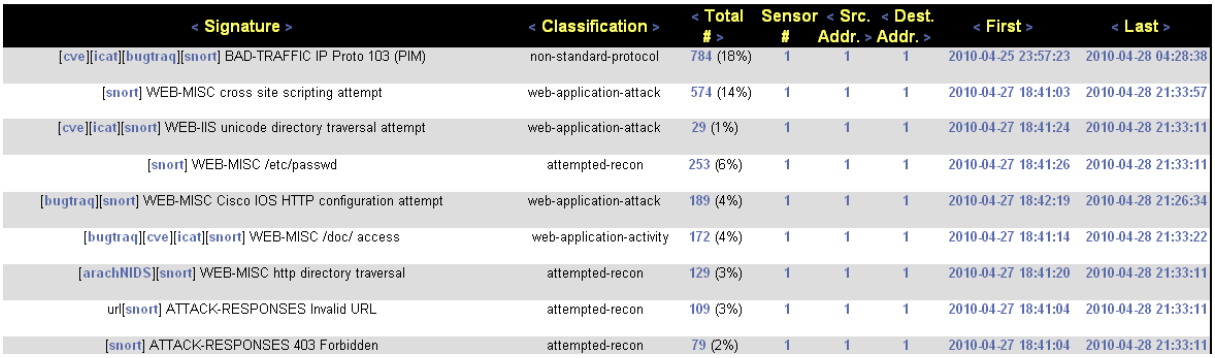
C:\Nikto2> perl nikto.pl update
C:\Nikto2> perl nikto.pl -h 192.168.1.13 –Display 1234 –evasion 123456789AB output

Dalis Nikto ataskaitos pateikta **Priede 3 Nikto ataskaita**. Iš ataskaitos matome, jog dalis pranešimų yra informacinio pobūdžio, kiti tikslesni. Nurodomas šaltinio šeimininko vardas ADMIN-3A05325DC, naudojamas prievadas 80, naudojama ASP.NET technologija, kurios versija 4.0.30319. Taip pat nurodomi kokios HTTP užklausos galimos : OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH. Dalis šių užklausų saugumo požiūriu turėtų būti uždraustos: PUT metodas gali įgalinti vartotoją išsaugoti failus serveryje, DELETE – ištrinti, MOVE – pakeisti failo lokacijos vietą (prie šių aprašymų nurodomos nuorodos su ID, kuriose detaliau aptariamos problemos).

Pavyzdžiui: OSVDB -*425* – Microsoft IIS palaiko SEARCH HTTP metodą WebDAV (angl. Web Distributed Authoring and Versioning), kuris savyje turi trūkumą – gali privesti prie nesankcionuoto informacijos atskleidimo. Kai indeksacijos serveris įgalintas, nutolęs atakuotojas turi galimybę pamatyti direktorijų vaizdą , kuriose gali būti saugoma jautri informacija (žinynai ir rinkmenos, kuriuose gali būti slaptažodžių). [27]

Taip pat pateikiama, kad IIS versija yra 6.0 yra pasenusi, nes egzistuoja naujesnės versijos. Iš ekrane išvestų vaizdų matoma, jog buvo bandoma atlikti žodyno ataką , naudojant įvairių katalogų pavadinimus, tačiau nesėkmingai.

Su Nikto priemone galima atlikti visas pagrindines atakų imitacijas, pažeidžiamumų ir konfigūracijos klaidų paiešką. Tai įrodo Snort perspėjimų gausa (pavyzdys pateiktas 38 paveiksliuke, panaudojant ACID žiniatinklio sąsają).



< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
[cve][icat][bugtraq][snort] BAD-TRAFFIC IP Proto 103 (PIM)	non-standard-protocol	784 (18%)	1	1	1	2010-04-25 23:57:23	2010-04-28 04:28:38
[snort] WEB-MISC cross site scripting attempt	web-application-attack	574 (14%)	1	1	1	2010-04-27 18:41:03	2010-04-28 21:33:57
[cve][icat][snort] WEB-IIS unicode directory traversal attempt	web-application-attack	29 (1%)	1	1	1	2010-04-27 18:41:24	2010-04-28 21:33:11
[snort] WEB-MISC /etc/passwd	attempted-recon	253 (6%)	1	1	1	2010-04-27 18:41:26	2010-04-28 21:33:11
[bugtraq][snort] WEB-MISC Cisco IOS HTTP configuration attempt	web-application-attack	189 (4%)	1	1	1	2010-04-27 18:42:19	2010-04-28 21:26:34
[bugtraq][cve][icat][snort] WEB-MISC /doc/ access	web-application-activity	172 (4%)	1	1	1	2010-04-27 18:41:14	2010-04-28 21:33:22
[arachNIDS][snort] WEB-MISC http directory traversal	attempted-recon	129 (3%)	1	1	1	2010-04-27 18:41:20	2010-04-28 21:33:11
url[snort] ATTACK-RESPONSES Invalid URL	attempted-recon	109 (3%)	1	1	1	2010-04-27 18:41:04	2010-04-28 21:33:11
[snort] ATTACK-RESPONSES 403 Forbidden	attempted-recon	79 (2%)	1	1	1	2010-04-27 18:41:04	2010-04-28 21:33:11

38 pav. ĮAS Snort užfiksuotų perspėjimų pavyzdys (grafinėje formoje)

IIS įrašų istorijos pavyzdys:

2010-04-28 18:20:04 W3SVC1 ADMIN-3A05325DC 192.168.1.13 HEAD / - 80 - 192.168.1.3 HTTP/1.1 Mozilla/4.75+(Nikto/2.1.1)+(Evasions:12345678AB)+(Test:Port+Check) - - ADMIN-3A05325DC 302 0 0 260 214 51031
2010-04-28 18:21:14 W3SVC1 ADMIN-3A05325DC 192.168.1.13 GET / - 80 - 192.168.1.3 HTTP/1.1 Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+en-US;+rv:1.9.1.9)+Gecko/20100315+Firefox/3.5.9+(.NET+CLR+3.5.30729) ASP.NET_SessionId=d1e0fu5ms1gba1d0jf4aaqrd - 192.168.1.13 302 0 0 402 439 171
2010-04-28 18:21:40 W3SVC1 ADMIN-3A05325DC 192.168.1.13 GET /login.aspx ReturnUrl=%2f 80 - 192.168.1.3 HTTP/1.1 Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+en-US;+rv:1.9.1.9)+Gecko/20100315+Firefox/3.5.9+(.NET+CLR+3.5.30729) ASP.NET_SessionId=d1e0fu5ms1gba1d0jf4aaqrd - 192.168.1.13 200 0 0 1825 463 2109

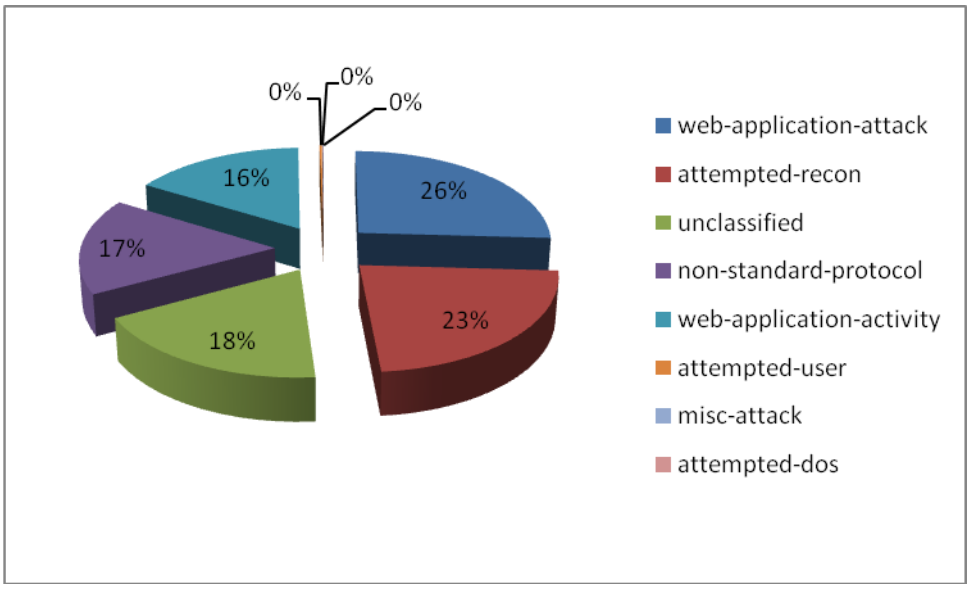
Operacinės sistemos programų įvykio informacinio pranešimo dalis, rodanti, kad nepavyko prisijungti:

Event Type: Information
Event Source: ASP.NET 4.0.30319.0
Event Category: Web Event
Event ID: 1315
Date: 4/28/2010
Time: 7:58:55 PM
User: N/A
Computer: ADMIN-3A05325DC
Description:
Event code: 4005
Event message: Forms authentication failed for the request.
Event time: 4/28/2010 7:58:55 PM
Event time (UTC): 4/28/2010 4:58:55 PM
Event ID: d14ba989170a4f8aabb8631bec8afbe0
Event sequence: 67
Event occurrence: 4
Event detail code: 0

Nikto gali sugeneruoti apie 4000 skirtingų tikrinimų. Eksperimento metu sugeneruota 3823 atakų bei pažeidžiamumų tikrinimai.

Tuo tarpu ĮAS Snort 2.0.1 sugeneravo 473 unikalius perspėjimus. Perspėjimai suskirstyti į 8 Snort kategorijas. Kaip pasiskirstę šios kategorijos procentine išraiška pateikta 39 paveiksliuke (pateikiama orginalia Snort kalba). Daugiausia perspėjimų sugeneruota su web-application-atack kategorija, kadangi skenuota buvo su žiniatinklio atakas imituojančia programa Nikto.

23 % perspėjimų sudaro pažeidžiamumų žvalgyba. 18% užima neklasifikuoti perspėjimai.



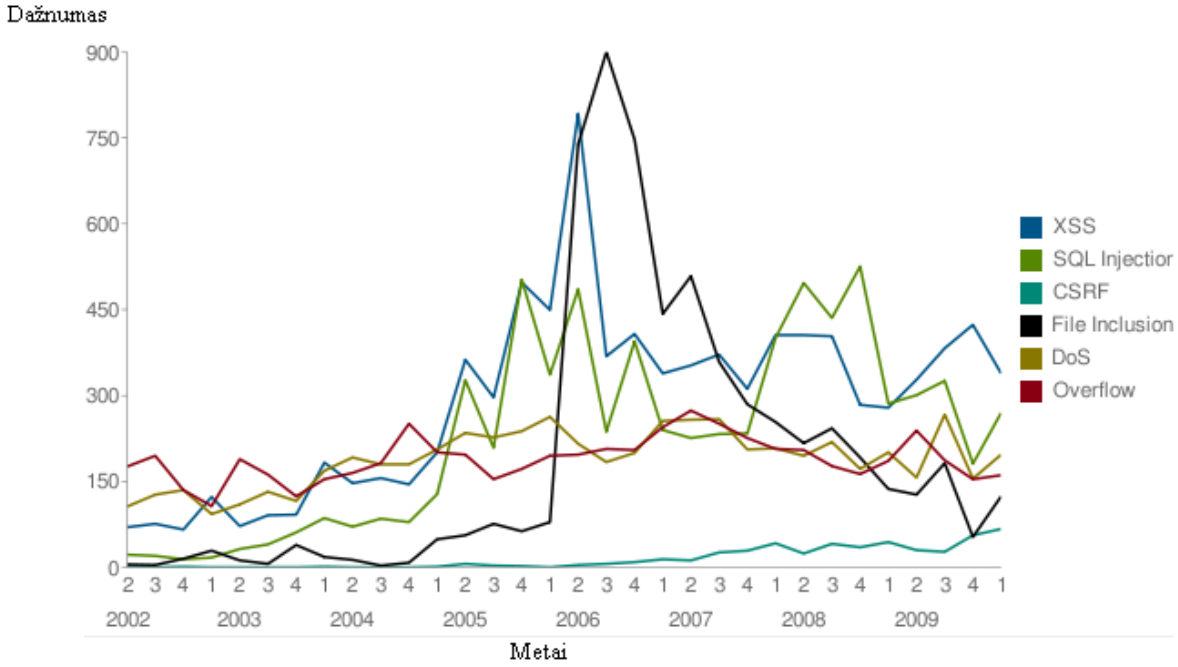
39 pav. Perspėjimų susiskirstymas į Snort kategorijas

Žiniatinklio atakų kategorijoje iš viso sugeneruota apie 50 atakų perspėjimų, kur dažniausiai pasitaikančių 10 perspėjimų pateikta 9 lentelėje (pateikiama orginalia Snort kalba), kur matome, jog daugiausiai perspėjimų gauta, kai Nikto programa tikrino ar įmanomi Cross Site Scripting tipo pažeidžiamumai. Antroje vietoje esantis perspėjimas tai Nessus skanerio atliktas veiksmas, kuris tikrina Cisco konfigūracijų pažeidžiamumus (versijos) ir yra galimybė perimti nuotolinį įrenginio valdymą. Toliau matome, jog IIS serveryje buvo ieškoma standartinių pavyzdžių, kurie gali būti panaudoti piktiems tikslams. 5 vietoje esantis perspėjimas rodos, jog buvo mėginama pasiekti konfidencialius failus ar direktorijas, naudojant užkoduotą „/“ išreiškimą formuojant URL (angl. Uniform Resource Locator) užklausas, pavykus atakai, įsilaužėlis gali vykdyti komandas serveryje.

8 lentelė Dažniausiai pasitaikantys perspėjimai (žiniatinklio atakų kategorijoje)

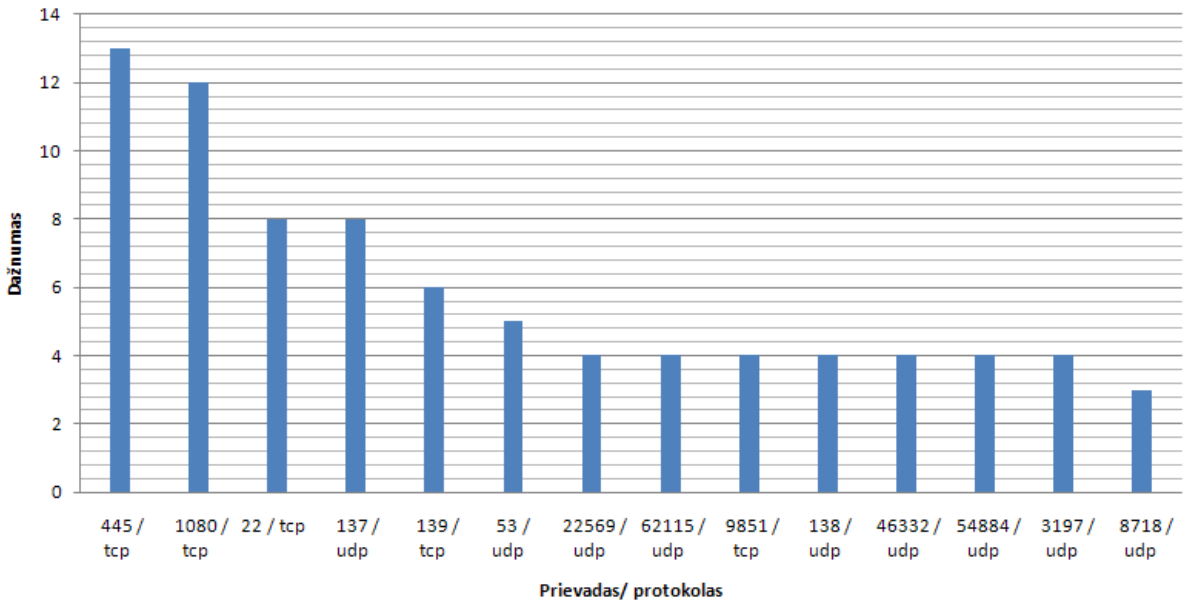
Nr.	Perspėjimo pavadinimas	Dažnumas
1	WEB-MISC cross site scripting attempt	936
2	WEB-MISC Cisco IOS HTTP configuration attempt	284
3	WEB-IIS iissamples access	69
5	WEB-IIS unicode directory traversal attempt	67
4	WEB-IIS /scripts/samples/ access	44
7	WEB-FRONTPAGE fourdots request	23
8	WEB-IIS SAM Attempt	17
9	WEB-MISC Tomcat servlet mapping cross site scripting attempt	16
10	WEB-IIS iisadmin access	15

Gauta lentelė patvirtina, kad naudojama skanavimo priemonė yra pakankamai nauja, nes generuoja tokias atakas kaip Cross Site Scripting, kuri yra šiuo metu viena iš populiaresnių, tuo galima įsitikinti atviro kodo duomenų bazės pateiktu grafiku 40 paveiksliuke.



40 pav. Dažniausiai pasitaikantys žiniatinklio pažeidžiamumai[27]

Dažniausiai pasitaikantis taikinio prievadas žinoma yra žiniatinklio prievadas – 80. Likusiųjų dažniausiai pasitaikančių taikinio prievadų pasiskirstymas pateiktas 41 paveiklsiuke.



41 pav. Dažniausiai pasitaikantys taikinio prievadai atlikus skanavimus

Dažniausiai pasitaikantis po 80 prievado tai yra 445 prievadas. Nessus skaneris rado 4 kritinius pažeidžiamumus susijusius su 445 prievadu. 445 prevadas tai SMB (angl. Server Message Block) naudojamas failų dalinimuisi Windows XP, 2000, 2003, ME operacinėse sistemose bei kitose su SAMBA susijusiuose sujungimuose. Šiuo prievadu dažnai naudojasi virusai bei kirminai. Šie pažeidžiamumai pateikti 42 paveiksliuke.

42 pav. Nessus skanerio aptikti kritiniai pažeidžiamumai

Antroje vietoje esantis prievadas 1080 skirtas soketams, išvykstančio (HTTP, FTP) srauto palaikymui. Pažeidžiamumai panašūs į FTP.

22 prievadas naudojamas SSH prisijungimams atlikti. Dažnai naudojamas prieš slaptažodį nukreiptoms atakoms vykdyti. 137-139 prevadai naudojami NetBIOS (angl. Network Basic Input/Output System). 53 prievadas naudojamas DNS sprendimams. Dažniausiai pasitaikančių šaltinio prievadų pasiskirstymas buvo 3394 bei 80 prievadams. Besikreipiančiųjų į eksperimento serverį šaltinio adresai (kai kuriais atvejais juodieji IP adresai) bei kilmės šalis pateikti 10 lentelėje. Matome, jog dominuoja Rusijos bei Kinijos IP adresai.

9 lentelė Besikreipiančiųjų į eksperimento serverį duomenys

IP adresas	Šeimininko vardas	Šalis
84.16.88.15	imu138.infomaniak.ch	Šveicarija
114.176.85.186	p14186-ipngn100201niho.hiroshima.ocn.ne.jp	Japonija
59.51.24.244	negalima nustatyti	Kinija
61.19.71.83	negalima nustatyti	Tailandas
87.245.141.164	negalima nustatyti	Rusija
193.106.92.144	negalima nustatyti	Rusija
193.206.39.129	negalima nustatyti	Italija
193.206.47.46	dp1.fasf.uniba.it	Italija
195.24.90.13	PDHASKOVO	Bulgarija
202.102.120.202	negalima nustatyti	Kinija
205.209.161.180	negalima nustatyti	JAV
222.45.112.221	QC-1EA3BE3CEDE7	Kinija
203.117.65.119	negalima nustatyti	Singapūras
58.56.123.212	negalima nustatyti	Kinija
208.68.93.216	negalima nustatyti	Kanada
187.117.141.174	ip-187-117-141-174.user.vivozap.com.br	Brazilija
122.225.56.45	negalima nustatyti	Kinija

4.3 Išvados

Prototipo realizacijoje buvo įgyvendinta dalis modelio. „Medaus puodynės“ serveryje įdiegta sveikatos duomenų saugojimo sistema su modifikuota prisijungimo duomenų baze, pakeistos šio serverio kai kurios ypatybės, leidžiančios nustatyti, jog tai yra „medaus puodynė“ (pagrindinės virtualių mašinų atpažinimo savybės).

Įsilaužimo aptikimo ir peradresavimo posistemė realizuota dalinai, peradresavimas pavyko tik pagal juoduosius IP sąrašus, nes pasirinktos programinės įrangos versijos buvo nesuderinamos su operacinės sistemos versija. Buvo susidurta su keblumais, kurie atsirado dėl darbinės atminties kiekio trūkumo kompiuteryje, kuriame buvo atliekamas eksperimentas (3 virtualios mašinos bei lokalios mašinos programos veikė nepriimtinai lėtai, nors ir buvo išjungtos labiausiai resursams imlios OS dalys). Nepaisant keblumų, su kuriais buvo susidurta realizuojant modelį, prototipo tyrimas buvo atliekamas su prielaida, jog sugeneravus ĮAS perspėjimą, srautas bus nukreipiamas į „medaus puodynę“.

Imituojant atakas bei ieškant pažeidžiamumų buvo pasirinktos priemonės Nikto bei Nessus. Nustatyta, kad iš sugeneruotų 4000 pažeidžiamumų ir veiksmingų atakų paieškų ĮAS Snort sugeneravo 473 įvykių perspėjimus. Taip pat iš skanerių ataskaitų pastebėta, kad dauguma pažeidžiamumų, jų tarpe ir kritiniai, atsiranda tiek dėl programinės įrangos neatnaujinimo, tiek dėl blogo žiniatinklio serverio konfigūravimo, tiek dėl programavimo klaidų. Iš dažniausiai pasitaikančių taikinio prievadų matome, jog didelį dėmesį reikia sutelkti stebint tokius pažeidžiamus prievadus kaip 445,1080, 22, 53, 137, 139 ir kt. Taip pat

užfiksuoti apie 20 IP adresų, kurie kreipėsi į eksperimentinį serverį – pasikartojančius daugiau nei 5-10 kartų iškart galima dėti į juoduosius sąrašus taip padidinant piktavalių nukreipimą į „medaus puodynės“ svetainę.

5. MD BENDROS IŠVADOS

Darbe išanalizuoti įsilaužimo aptikimo sistemų informacijos apdorojimo metodai, veikimo principai, atlikta lyginamoji šių sistemų analizė. Analizėje, dėl teigiamų savybių daugiau dėmesio skirta atviro kodo ĮAS ir pastebėta, kad jų tarpe geriausia yra Snort, bet visoms šioms sistemoms yra keletas bendrų trūkumų: veiksmų trūkumas aptikus įsilaužimą, pasenusi parašų bazė neaptinka naujų įsilaužimo būdų. Iš dalies šias problemas gali išspręsti „medaus puodynės“ metodo panaudojimas. Apžvelgtos taip pat ir „medaus puodynių“ technologijų galimybės ir įvairovė. Šios nuo įsilaužimo aptikimo sistemų skiriasi tuo, kad galima aptikti naujus įsilaužėlių naudojamus metodus. Metodo esmė: įvilioti potencialius įsilaužėlius į „medaus puodynės“ bei rinkti apie juos informaciją, kurią apdorojus galima kurti naujus saugos metodus. Išanalizavus ši metodą modeliuojama perspėjimo sistema.

Panaudojus esamus sprendimus: įsilaužimo aptikimo sistemą, ugniasienę, kelis scenarijus ir aibę taisyklių bei juoduosius IP sąrašus galima sukurti įsilaužimo aptikimo ir peradresavimo posistemę. Ši posistemė pagal taisykles ir IP sąrašus piktybinį srautą gali nukreipti į „medaus puodynės“ svetainę, kur įrašų analizės posistemėje administratorius analizuoja koncentruotą įrašų istoriją bei kuria naujus saugos metodus. Įrašų analizės posistemėje įrašai pateikiami iš ĮAS, OS bei žiniatinklio serverio.

Realizacijoje iš dalies realizuota įsilaužimo aptikimo ir peradresavimo posistemė, pašalintos „medaus puodynės“ atpažinimo savybės (virtualizavimo pasėkmės). Modifikuoti autentifikacijos duomenų bazės įrašai. Tyrime panaudotos kelios populiarios įsilaužimus imituojančios ir pažeidžiamumų ieškančios priemonės. Pastebėta, kad iš sugeneruoto „piktybinio“ srauto įsilaužimo aptikimo sistema pateikia apie 12 % perspėjimų, vadinasi 12 % šio srauto bus nukreipiama į „medaus puodynės“ svetainę, taip dalinai apsaugant produkcinį serverį. Taip pat pastebėti ir užregistruoti dažniausiai skanuojami prievadai bei šaltinio IP adresai, į kuriuos derėtų atkreipti dėmesį. Visa ši sistema turėtų būti sujungta į bendrą informacijos kaupimo ir analizavimo tinklą, taip didinant naujų įsilaužimo metodų aptikimo tikimybę. Taip pat nereikia pamiršti, kad tokios sistemos naudojimas yra ne saugumo sprendimas, o saugumo didinimo priemonė, tad nereiktų atsisakyti standartinių tinklo saugos metodų.

6. LITERATŪROS SĄRAŠAS

[1] ADOMAVIČIUS M.; MATONIS M.; SINKEVIČIUS V. Sveikatos duomenų saugojimo sistema: bakalauro darbas. KTU, Informatikos fakultetas. [Kaunas], 2008.

[2] ČENYS A. et al. Medaus puodynės (Honeypot) technologijos taikymas ankstyvam kenkėjiškų programų aptikimui : PFI mokslinės konferencijos medžiaga. Vilnius, 2005, Gruodis. p. 125 -128 ISBN:9986-9284-7-8.

[3] LAGZDINYTĖ I. Paskaitos medžiaga. Kliento-serverio sąveika. 61 p. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.ila.lt/moduliai/P175B103/src/kliento_serverio_saveika_paskaitav2.pdf>

[4] AMIT D. L. Techniques using Honeypots : master‘s thesis. University of London, Information Security Group Royal Holloway. [London], 2003, p. 2-21.

[5] Bro documentation, support, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.bro-ids.org>

[6] Clarke J.; Dhanjani N. Network Security Tools. 2005, April. p. 75-89. ISBN:0-596-00794-9

[7] FLYNN H. Snort Installation and Basic Usage Part Two. Symantec [interaktyvus]. 2000, July. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.symantec.com/connect/articles>

[8] GITE V. Linux log files location and how do I view logs files? Cyberciti [interaktyvus]. 2007. [žiūrėta 2010.05.15]. Prieiga per internetą: < http://www.cyberciti.biz>

[9] GRIMES R.A. Honeypots for Windows. 2005, February. ISBN:1590593359.

[10] IEEE Standart Associasion. VMware MACs. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://standards.ieee.org/regauth/oui>

[11] INESS S.; VALLI C. Honeypots: How do you know when you are inside one? : 4th Australian Digital Forensics Conference. Perth, 2006, December. ISBN 0-7298-0624-3.

[12] KAZIENKO P.; DAROSZ P. Intrusion Detection Systems (IDS) part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture). Windowsecurity [interaktyvus]. 2004, June [žiūrėta 2010.03.07]. Prieiga per internetą:<http://www.windowsecurity.com>

[13] KAZIENKO P.; DAROSZ P. Intrusion Detection Systems (IDS) part II - Classification; methods; techniques. Windowsecurity [interaktyvus]. 2004, July [žiūrėta 2010.03.07]. Prieiga per internetą:<http://www.windowsecurity.com>

[14] KEYS B. Creating Snort Rules with EnGarde. Linuxsecurity [interaktyvus]. 2007. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.linuxsecurity.com>

[15] MAGALHAES R. M. Understanding Windows loging. Windowsecurity [interaktyvus]. 2003, May. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.windowsecurity.com>

[16] MAGALHAES R. M. Host-Based IDS vs Network-Based IDS (Part 1) Windowsecurity [interaktyvus]. 2003, July [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.windowsecurity.com>

[17] National Security Agency Central Security Service. Hardening Tips For Default Installation of Red Hat Enterprise Linux 5. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.nsa.gov>

[18] OSSEC documentation, support, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.ossec.net>

[19] PreludeIDS documentation, support, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą: <https://dev.prelude-technologies.com>

[20] PRESTON E. A New Tool in the Spam War. Securityfocus [interaktyvus]. 2005, January [žiūrėta 2010.05.15]. Prieiga per internetą:<http://www.securityfocus.com>

[21] ROESCH M. Snort 3.0 Architecture Series Part 1: Overview. Securitysauce.blogspot [interaktyvus]. 2007, Nowember [žiūrėta 2010.05.15]. Prieiga per internetą: <http://securitysauce.blogspot.com>

[22] SCARFONE K.; MELL P. Guide to intrusion detection and prevention systems (IDPS). National Institute of Standards and Technology Gaithersburg U.S. [interaktyvus]. 2007, February [žiūrėta 2010.03.07] Prieiga per internetą: <http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf>

[23] Snort documentation, support, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.snort.org>

[24] SPIZNER L. Honeypots: tracking hackers. Boston, 2003, September. ISBN:0-321-10895-7.

[25] Suricata documentation, support, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://www.openinfosecfoundation.org>

[26] The Bait and Switch documentation, downloads. [žiūrėta 2010.05.15]. Prieiga per internetą:<http://baitnswitch.sourceforge.net>

[27] The Open Source Vulnerability Database. [žiūrėta 2010.05.15]. Prieiga per internetą: <http://osvdb.org>

7. PRIEDAI

Priedas 1 Iptable taisyklės

Chain INPUT (policy DROP)

target prot opt source destination

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

RULE_0 all -- 192.168.1.6 anywhere state NEW

RULE_0 all -- 192.168.2.1 anywhere state NEW

RULE_0 all -- 192.168.2.10 anywhere state NEW

RULE_2 all -- anywhere anywhere

Chain FORWARD (policy DROP)

target prot opt source destination

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

RULE_0 all -- 192.168.2.10 anywhere state NEW

RULE_1 all -- anywhere 192.168.2.10 state NEW

RULE_2 all -- anywhere anywhere

Chain OUTPUT (policy DROP)

target prot opt source destination

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

RULE_0 all -- anywhere anywhere state NEW

RULE_1 all -- anywhere 192.168.2.10 state NEW

RULE_2 all -- anywhere anywhere

Chain RULE_0 (5 references)

target prot opt source destination

LOG all -- anywhere anywhere LOG level info prefix `RULE 0 -- ACCEPT '

ACCEPT all -- anywhere anywhere

Chain RULE_1 (2 references)

target prot opt source destination

LOG all -- anywhere anywhere LOG level info prefix `RULE 1 -- ACCEPT '

ACCEPT all -- anywhere anywhere

Chain RULE_2 (3 references)

target prot opt source destination

LOG all -- anywhere anywhere LOG level info prefix `RULE 2 -- DENY '

DROP all -- anywhere anywhere

Priedas 2 Terminalo langu turinys

root@localhost Desktop]# cd bns
[root@localhost bns]# ls
bns-HOWTO.html bns-HOWTO.pdf config README routing snort switching
[root@localhost bns]# cd config
[root@localhost config]# ls
bns_conf.bash
[root@localhost config]# ./bns_conf.bash

Menu for Bait N Switch Configuration (GO IN ORDER)

*1) Set Up Routing Tables. (**RUN ONCE PER MACHINE**)*
2) Configuration
3) Patch Snort (ONLY AFTER OPTION 2)
4) Exit

Your Choice: 1

Menu for Bait N Switch Configuration (GO IN ORDER)

*1) Set Up Routing Tables. (**RUN ONCE PER MACHINE**)*
2) Configuration
3) Patch Snort (ONLY AFTER OPTION 2)
4) Exit

Your Choice: 2

##
 # Welcome to the Bait N Switch Configuration Generator.
 # This will generate bns.conf for you, enjoy.
#
 # - Team Violating
##

External Interface: eth0
External IP Address: 192.168.1.6

Production Interface: eth1
Production Gateway IP Address: 192.168.2.1
Honeypot's Interface: eth2
Honeypot's Gateway: 192.168.2.2
IP of <both> Honeypot and Production (Same IP, Diff GW): 192.168.2.10
Length of Time (in minutes) that the mark time should be incremented: 5
Length of Time DoS Protection: Max Alerts (how many alerts is 'too many' within dos_time): 100
Length of Time DoS Protection: Period ((in seconds) to look for too many marks from a single IP): 30
IP DoS Protection: How many Ip's per a certain amount of time is too many? 50
IP DoS Protection: Within what length of time should a certain number of IP's be too many?100
Fifo File Location (ie: /fifo/bnsfifo): /root/Desktop/bns/switching
Log Location (ie: /var/log/switchcore.log): /var/log/switchcore.log
Blacklist Location (ie: /etc/bns_blist): /etc/bns_list
Path to snort 1.9.0 directory (ie: /root/snort-1.9.0): /root/Desktop/snort

Done with variables, writing configuration(s) file
Done with Configuration, creating routes

Menu for Bait N Switch Configuration (GO IN ORDER)

*1) Set Up Routing Tables. (**RUN ONCE PER MACHINE**)*
2) Configuration
3) Patch Snort (ONLY AFTER OPTION 2)
4) Exit

Your Choice: 3
Path to bns.diff (ie: /root/bns/snort/bns.diff) /root/Desktop/bns/snort/bns.diff
patching file src/Makefile.in
Hunk #1 FAILED at 170.
1 out of 1 hunk FAILED -- saving rejects to file src/Makefile.in.rej
patching file src/output-plugins/Makefile.am
Hunk #1 FAILED at 9.
1 out of 1 hunk FAILED -- saving rejects to file src/output-plugins/Makefile.am.rej
patching file src/output-plugins/Makefile.in
Hunk #1 FAILED at 90.
Hunk #2 FAILED at 106.
2 out of 2 hunks FAILED -- saving rejects to file src/output-plugins/Makefile.in.rej
patching file src/output-plugins/spo_alert_bns.c
patching file src/output-plugins/spo_alert_bns.h
patching file src/plugbase.c
Hunk #1 FAILED at 103.
Hunk #2 FAILED at 146.
2 out of 2 hunks FAILED -- saving rejects to file src/plugbase.c.rej
done patching...

exit or menu [e/m]: e
[root@localhost config]# ls
bns.conf bns_conf.bash

[root@localhost config]# cd..
bash: cd..: command not found
[root@localhost config]# cd ..
[root@localhost bns]# ls
bns-HOWTO.html bns-HOWTO.pdf config README routing snort switching
[root@localhost bns]# cd routing
[root@localhost routing]# ls
bnsroutes.bash bnstables.bash
[root@localhost routing]# ./bnsroutes.bash
bash: ./bnsroutes.bash: Permission denied
[root@localhost routing]# chmod 777 bnsroutes.bash
[root@localhost routing]# ./bnsroutes.bash
RTNETLINK answers: Network is unreachable
RTNETLINK answers: Network is unreachable
RTNETLINK answers: File exists
[root@localhost routing]# ./bnsroutes.bash
RTNETLINK answers: Network is unreachable
RTNETLINK answers: Network is unreachable
RTNETLINK answers: File exists
[root@localhost routing]# cd ..
[root@localhost bns]# ls
bns-HOWTO.html bns-HOWTO.pdf config README routing snort switching
[root@localhost bns]# cd switching
[root@localhost switching]# ls
fifoc.c switchcore.c switch.vars
[root@localhost switching]# gcc -lpthread switchcore.c -o switchcore
bash: gcc: command not found
[root@localhost switching]# yum install gcc
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
** addons: ftp.nb.lug.ro*
** base: ftp.nb.lug.ro*
** extras: ftp.nb.lug.ro*
** updates: ftp.nb.lug.ro*
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package gcc.i386 0:4.1.2-46.el5_4.2 set to be updated
--> Processing Dependency: cpp = 4.1.2-46.el5_4.2 for package: gcc
--> Processing Dependency: libgomp >= 4.1.2-46.el5_4.2 for package: gcc
--> Processing Dependency: libgcc >= 4.1.2-46.el5_4.2 for package: gcc
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc
--> Running transaction check
---> Package cpp.i386 0:4.1.2-46.el5_4.2 set to be updated
---> Package glibc-devel.i386 0:2.5-42.el5_4.3 set to be updated
--> Processing Dependency: glibc = 2.5-42.el5_4.3 for package: glibc-devel
--> Processing Dependency: glibc-headers = 2.5-42.el5_4.3 for package: glibc-devel
--> Processing Dependency: glibc-headers for package: glibc-devel
---> Package libgcc.i386 0:4.1.2-46.el5_4.2 set to be updated
---> Package libgomp.i386 0:4.4.0-6.el5 set to be updated
--> Running transaction check

--> Processing Dependency: glibc = 2.5-42 for package: nscd
---> Package glibc.i686 0:2.5-42.el5_4.3 set to be updated
--> Processing Dependency: glibc-common = 2.5-42.el5_4.3 for package: glibc
---> Package glibc-headers.i386 0:2.5-42.el5_4.3 set to be updated
--> Processing Dependency: kernel-headers >= 2.2.1 for package: glibc-headers
--> Processing Dependency: kernel-headers for package: glibc-headers
--> Running transaction check
---> Package glibc-common.i386 0:2.5-42.el5_4.3 set to be updated
---> Package kernel-headers.i386 0:2.6.18-164.15.1.el5 set to be updated
---> Package nscd.i386 0:2.5-42.el5_4.3 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

```
================================================================================
 Package            Arch       Version                 Repository     Size
================================================================================
Installing:
 gcc                i386       4.1.2-46.el5_4.2        updates       5.2 M
Installing for dependencies:
 glibc-devel        i386       2.5-42.el5_4.3          updates       2.0 M
 glibc-headers      i386       2.5-42.el5_4.3          updates       601 k
 kernel-headers     i386       2.6.18-164.15.1.el5     updates       1.0 M
 libgomp            i386       4.4.0-6.el5             base          70 k
Updating for dependencies:
 cpp                i386       4.1.2-46.el5_4.2        updates       2.6 M
 glibc              i686       2.5-42.el5_4.3          updates       5.2 M
 glibc-common       i386       2.5-42.el5_4.3          updates       16 M
 libgcc             i386       4.1.2-46.el5_4.2        updates       95 k
 nscd               i386       2.5-42.el5_4.3          updates       163 k

Transaction Summary
================================================================================
Install      5 Package(s)
Update       5 Package(s)
Remove       0 Package(s)

Total download size: 33 M
Is this ok [y/N]: y
Downloading Packages:
(1/10): libgomp-4.4.0-6.el5.i386.rpm                |  70 kB     00:00
(2/10): libgcc-4.1.2-46.el5_4.2.i386.rpm            |  95 kB     00:00
(3/10): nscd-2.5-42.el5_4.3.i386.rpm                | 163 kB     00:00
(4/10): glibc-headers-2.5-42.el5_4.3.i386.rpm       | 601 kB     00:02
(5/10): kernel-headers-2.6.18-164.15.1.el5.i386.rpm | 1.0 MB     00:05
(6/10): glibc-devel-2.5-42.el5_4.3.i386.rpm         | 2.0 MB     00:10
(7/10): cpp-4.1.2-46.el5_4.2.i386.rpm               | 2.6 MB     00:22
(8/10): gcc-4.1.2-46.el5_4.2.i386.rpm               | 5.2 MB     00:40
```

```
(9/10): glibc-2.5-42.el5_4.3.i686.rpm                | 5.2 MB     00:44
(10/10): glibc-common-2.5-42.el5_4.3.i386.rpm        | 16 MB     01:24
--------------------------------------------------------------------------------
Total                                   160 kB/s |  33 MB     03:34
warning: rpmts_HdrFromFdno: Header V3 DSA signature: NOKEY, key ID e8562897
updates/gpgkey                                       | 1.5 kB     00:00
Importing GPG key 0xE8562897 "CentOS-5 Key (CentOS 5 Official Signing Key) <centos-5-
key@centos.org>" from /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
Is this ok [y/N]: y
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating     : glibc-common                           1/15
  Installing   : kernel-headers                         2/15
  Updating     : libgcc                                 3/15
  Updating     : glibc                                  4/15
  Updating     : cpp                                    5/15
  Installing   : libgomp                                6/15
  Updating     : nscd                                   7/15
  Installing   : glibc-headers                          8/15
  Installing   : glibc-devel                            9/15
  Installing   : gcc                                   10/15
  Cleanup      : nscd                                  11/15
  Cleanup      : cpp                                   12/15
  Cleanup      : glibc-common                          13/15
  Cleanup      : libgcc                                14/15
  Cleanup      : glibc                                 15/15

Installed:
  gcc.i386 0:4.1.2-46.el5_4.2

Dependency Installed:
  glibc-devel.i386 0:2.5-42.el5_4.3        glibc-headers.i386 0:2.5-42.el5_4.3
  kernel-headers.i386 0:2.6.18-164.15.1.el5 libgomp.i386 0:4.4.0-6.el5

Dependency Updated:
  cpp.i386 0:4.1.2-46.el5_4.2              glibc.i686 0:2.5-42.el5_4.3
  glibc-common.i386 0:2.5-42.el5_4.3       libgcc.i386 0:4.1.2-46.el5_4.2
  nscd.i386 0:2.5-42.el5_4.3

Complete!
You have new mail in /var/spool/mail/root
[root@localhost switching]# gcc -lpthread switchcore.c -o switchcore
[root@localhost switching]#
```

Priedas 3 Nikto ataskaita

- Nikto v2.1.1/2.1.1

+ Target Host: ADMIN-3A05325DC
+ Target Port: 80
+ GET /: Retrieved X-Powered-By header: ASP.NET
+ GET /: Retrieved x-aspnet-version header: 4.0.30319
+ GET /: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
+ OSVDB-5646: GET /: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ OSVDB-397: GET /: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5647: GET /: HTTP method ('Allow' Header): 'MOVE' may allow clients to change file locations on the web server.
+ GET /: HTTP method ('Allow' Header): 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get directory listings if indexing is allow but a default page exists.
+ GET /: HTTP method ('Allow' Header): 'PROPPATCH' indicates WebDAV is installed.
+ OSVDB-425: GET /: HTTP method ('Allow' Header): 'SEARCH' indicates DAV/WebDAV is installed, and may be used to get directory listings if Index Server is running.
+ GET /: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
+ OSVDB-5646: GET /: HTTP method ('Public' Header): 'DELETE' may allow clients to remove files on the web server.
+ OSVDB-397: GET /: HTTP method ('Public' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5647: GET /: HTTP method ('Public' Header): 'MOVE' may allow clients to change file locations on the web server.
+ GET /: HTTP method ('Public' Header): 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get directory listings if indexing is allow but a default page exists.
+ GET /: HTTP method ('Public' Header): 'PROPPATCH' indicates WebDAV is installed.
+ OSVDB-425: GET /: HTTP method ('Public' Header): 'SEARCH' indicates DAV/WebDAV is installed, and may be used to get directory listings if Index Server is running.
+ OSVDB-13431: GET /: PROPFIND HTTP verb may show the server's internal IP address: http://192.168.1.13/
+ HEAD /: Microsoft-IIS/6.0 appears to be outdated (4.0 for NT 4, 5.0 for Win2k, current is at least 7.0)