



**Универзитет у Нишу
Електронски факултет
Катедра за рачунарство**



**Дигитална форензика
Системи за детекцију и превенцију напада
преко рачунарских мрежа**

**Студент:
Стефан Стојановић 1355**

**Ментор:
проф др Братислав Предић**

Ниш, август 2022.

Садржај

1 Увод	1
2 Основни концепти	2
2.1. Sniffing	2
2.2 Анализатор пакета	4
2.3 PCAP	5
2.4 Signature	6
3 Детекција и превенција упада	8
3.1 IDS - системи за детекцију упада у рачунарске мреже	8
3.2 IPS - системи за превенцију упада у рачунарске мреже	10
3.3 Позиционирање IDS и IPS система у мрежи	11
3.4 Примери упада и компромитовања безбедности система	12
3.5 Suricata	13
3.5.1 Suricata потписи	13
3.6 Додатни примери система за детекцију и превенцију упада у рачунарске мреже	18
3.7 Libtins	20
3.7.1 Инсталација библиотеке Libtins	22
4 Закључак	23
5 Литература	24

1 Увод

Packet analyzing технологија, која се често назива и Packet sniffing, омогућује пресретање и увид у саобраћај који се одвија између уређаја на мрежи. Технологија се може користити за различите сврхе, почев од детектовања сигурносних проблема на мрежи па до самог креирања сигурносних проблема. Администратори мреже често користе ову технологију за проверу сигурности система и проналазак рањивих делова.

Данас је доступно пуно разноврсног софтвера за анализирање пакета, детекцију и превенцију потенцијалних напада, као и готових библиотека које се могу користити за креирање сопствених решења. Сви ови софтвери су поткрепљени детаљном документацијом, па је само потребно добро је проучити како би се максимално искористио софтвер за одређену ситуацију.

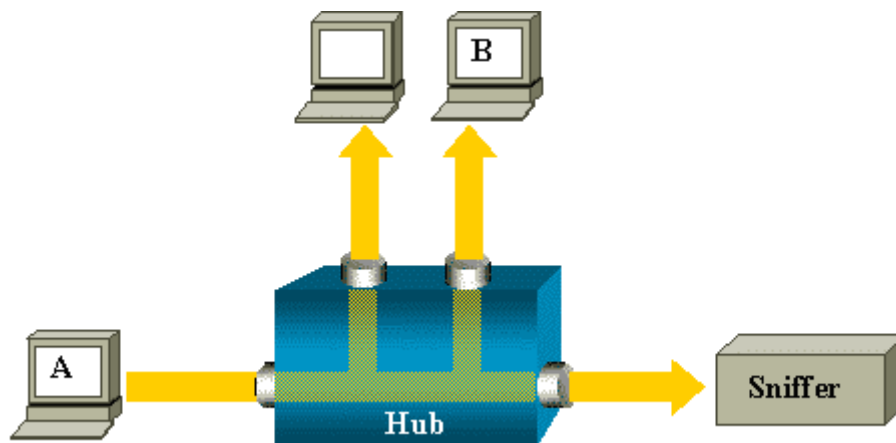
У овом раду биће описани битни концепти у овој области, функционалности готовог софтвера Suricata, и готова библиотека Libtins.

2 Основни концепти

2.1. Sniffing

У свакодневном раду за рачунаром када год рачунар или неки други уређај повежемо на локалну Ethernet мрежу, он врши слање и пријем TCP/UDP пакета тј. датаграма и комуницира са локалним рутером. Рутер пакете прослеђује даље, ка њиховом одредишту, према другим мрежним јединицама, рутерима, серверима на интернету односно где год је то потребно.

Sniffing представља процес пресретања, логовања и анализе свих ових пакета. Уколико се на мрежи користе различити уређаји а не само hub-ови, између нападача и потенцијалне жртве sniffing-а ће се појавити препреке. Препреке су свичеви који праве сегментацију на физичком и нивоу података OSI модела. За разлику од примене хабова код којих пакети стижу свима због broadcast-а, код мреже са свичевима пакети намењени уређају А стижу искључиво до уређаја А због unicast-а i multicast-а.



Слика 1. Sniffer на мрежи која има само хабове

Са позиције уређаја који жели да врши снифовање, пресретне пакете и логује их, оваква ситуација може да представља проблем. Потребно је да пакети стигну и до уређаја на ком желимо да вршимо снифовање а не само до тачне дестинације. Овај проблем се може превазићи на више начина:

- Мониторинг порт на свичу - ово је посебан порт који постоји на напреднијим свичевима а његова сврха је репликација свих пакета који пролазе кроз свич;
- Sniffing на самом рутеру - Снифовање на рутеру зависи од типа рутера. На пример на Микротик рутерима коришћење ове технике је омогућено;
- ARP spoofing - За коришћење ове технике потребно је да мрежна картица ради у тзв. promiscuous моду. Promiscuous означава да ће мрежни адаптер прихватати све пакете чак и оне који нису намењени директно њему. Најчешће је потребно да се овај мод подеси накнадно јер не спада у подразумевана подешавања. ARP spoofing се другачије назива и ARP poisoning. Ова техника користи слабост ARP протокола, јер

ARP не подржава никакав облик аутеникације. Уколико се неки уређај прикључи на мрежу он интерно повезује друге уређаје са одговарајућим MAC адресама, али заправо не постоји механизам којим би се потврдило да уређај А на адреси АА заиста јесте уређај А а не неки други уређај. Овај недостатак се употребљава за успешно коришћење ове технике.

Пример ARP spoofing-a

Претпоставимо да се уређај А прикључио на локалну мрежу. Потребно је да уређај А комуницира са уређајем Б на адреси 192.168.1.10, па ће на MAC адресу FF-FF-FF-FF-FF-FF послати упит:

„Ко зна на којој се MAC адреси налази уређај који има IP адресу 192.168.1.10?“

Сада се јавља уређај са IP адресом 192.168.1.10 и са своје MAC адресе, рецимо AB-CD-EF-12-34-56 шаље одговор:

„Ја имам адресу 192.168.1.10, а налазим се на MAC адреси AB-CD-EF-12-34-56“.

Могуће је послати и инверзни упит где се тражи MAC адреса на основу IP адресе. Овај процес функционише без додатних провера, при чему уређаји прихватају саобраћај и са адреса за које нису слали упите.

У ARP протоколу постоји и специјална врста одговора која се назива Gratuitous ARP. Ово је регуларан одговор, а разлика је само у томе да му није претходио упит ниједне мрежне јединице. Овај одговор може да објави сваки уређај у локалној мрежи, а сви остали уређаји ће без провере прихватити информацију као истиниту. Овај одговор се користи када се промени IP адреса мрежне картице уређаја, а уређај ову промену јавно објављује свим осталим уређајима у мрежи.

Претпоставимо да нападач жели да пресретне сав саобраћај који се одвија између циљаног уређаја који има IP адресу 192.168.1.10 (жртва у наставку текста) и рутера, чија је IP адреса 192.168.1.1. Оваквим пресретањем нападач добија могућност да снифује, практично сав саобраћај жртве ка интернету. Као први корак, нападач ће се прикључити на локалну мрежу и почети да преплављује рутер Gratuitous ARP одговорима (одговори без претходног упита) којима асоцира своју MAC адресу на IP адресу жртве:

„Ја имам IP адресу 192.168.1.10 (жртвина IP адреса), а налазим се на MAC адреси FF-FF-FF-BB-BB-BB (нападачева MAC адреса)“.

Услед недостатка механизма провере у оквиру ARP протокола, рутер ову објаву мора да прихвати и у свом ARP кешу ће асоцирати нападачеву MAC адресу са жртвином IP адресом.

Следећи корак је плавање жртве са Gratuitous ARP објавама, али сада објаве изгледају овако:

„Ја имам IP адресу 192.168.1.1 (IP адреса рутера), а налазим се на MAC адреси FF-FF-FF-BB-BB-BB (нападачева MAC адреса)“.

Због природе ARP протокола жртва ће ову информацију прихватити. Сада имамо ситуацију у којој се нападач жртви представио као рутер, а рутеру као жртва, при чему и рутер и жртва поуздано баратају са овим информацијама. Овиме је завршен процес ARP spoofing-а и остао је још само један елемент како би напад, који се назива и „Man in the middle (MiTM)“ напад, попримио свој коначан облик. Нападач сада треба да конфигурише свој оперативни систем тако да обезбеди IP forwarding. То значи да ће нападач пакете који му стижу прослеђивати према мрежним јединицама којима су и намењене. Ако жртва пошаље пакет рутеру, тај пакет ће реално стићи прво до нападача, а како би се остварила комуникација, нападач пакет мора да проследи рутеру.

Решење за блокирање ARP spoofing-а код мањих мрежа се огледа у коришћењу статичких уноса којима се на мрежним уређајима IP адресе везују за MAC адресе. За веће мреже, најбоље решење је коришћење свичева који имају port security опцију. Ова опција за сваки физички порт на свичу дозвољава само по једну MAC адресу, што би требало да спречи овај тип напада. Такође, препорука је коришћење неког софтвера за ARP мониторинг као што је на пример ARPwatch. Софтвер овог типа прати саобраћај на мрежи и пријављује сваки вид комуникације који је сумњив и све промене MAC адреса. Наравно, потребно је надлежно лице које ће анализирати детектоване догађаје и чинити даље кораке уколико је то потребно.

2.2 Аналитичар пакета

Анализатор пакета (Packet analyzer) или анализатор мреже је компјутерски програм или компјутерски хардвер који може да пресеће и логује саобраћај који пролази кроз рачунарску мрежу или део мреже. Аналитичари заправо врше процес Sniffing-а који је описан у претходном поглављу. Хватање пакета је процес пресретања и логовања саобраћаја. Како ток података пролази кроз мрежу, анализатор хвата сваки пакет из тока података и уколико је потребно декодира податке из пакета, па се вредности различитих поља из пакета могу видети и анализирати у складу са неком од доступних спецификација.



Слика 2. Wireshark (лево) и уређај за хватање пакета (десно)

Анализатор који се користи за пресретање саобраћаја на бежичним мрежама назива се бежични или WiFi анализатор. На жичаним мрежама са дељивим уређајима попут Ethernet, Token Ring и FDDI, у зависности од структуре тј. да ли се користи хаб или свич, може бити могуће да се ухвати цео саобраћај на једној машини. На модерним мрежама саобраћај се може ухватити коришћењем технике пресликавања порта која копира све пакете са једног порта на други. На бежичним мрежама саобраћај се може хватати по једном каналу у исто

време или више канала уз коришћење већег броја адаптера. Како би се на жичаним мрежама хватао саобраћај намењен другим машинама у мрежи, мрежни адаптер мора бити у promiscuous моду. За хватање саобраћаја намењеном другим уређајима на локалним бежичним мрежама promiscuous мод није довољан. Код ове мреже потребно је да мрежни адаптер буде у monitor моду.

Приликом хватања пакета може се памтити цео садржај пакета или само заглавља. Памћење само заглавља смањује заузету меморију и избегава проблеме са кршењем приватности. Детектован саобраћај се декодира у формат разумљив човеку. Неки анализатори имају могућност да и сами генеришу пакете. Они се могу користити као протокол тестери. Анализатори пакета могу поседовати прегршт различитих функционалности. Навешћемо само неке од њих:

- Анализа мрежних проблема
- Детекција покушаја упада
- Детекција злоупотребе мреже од локалних и спољашњих корисника
- Праћење коришћења мреже
- Праћење података у транзиту
- Праћење статуса безбедности WAN и крајњих уређаја на мрежи
- Прикупљање и пријава мрежне статистике
- Дебагирање клијент-сервер комуникације
- Праћење других локалних корисника мреже и прикупљање осетљивих информација

Неки примери анализатора пакета су: Wireshark, The Sniffer, Xplico, ngrep, Capsa итд.

2.3 PCAP

У домену администрације комјутерских мрежа, pcap представља формат за памћење мрежног саобраћаја. На системима базираним на линуксу овај формат се имплементира у оквиру libpcap библиотеке. На Виндовс системима до појаве Windows 7 коришћена је библиотека WinPcap која више није подржана, а од Windows 7 верзије користи се Npcap. Све ове библиотеке написане су у C језику.

Наведене библиотеке пружају подршку за детекцију и филтрирање пакета, као и чување детектованих пакета у фајлу. Сачувани фајлови се могу читати и анализирати коришћењем истог кода. Коришћењем libpcap, WinPcap, и Npcap могу се креирати сопствена решења.

Данас је доступно више софтвера који пружају подршку за овај формат а неки од њих су: tcpdump, Wireshark, Suricata, Snort, CA NetMaster, Microsoft Network Monitor итд.

На основу libpcap библиотеке написано је и много wrapper библиотека око ове библиотеке са додатним функционалностима, коришћењем различитих програмских језика. Неке од најпознатијих су:

- C++ - Libtins, Libcrafter, PcapPlusPlus;
- Python - python-libpcap, Pcapu, WinPcapu;
- Rust - pcap;

- Node.js - node_pcap;
- Java - jpcap, jNetPcap, Jpcap;
- .Net WinPcapNet, Pcap.Net, SharpPcap.

2.4 Signature

Први корак ка бољем разумевању безбедности рачунара је налажење тог рачунара у опасности. Рачунар се заправо стално налази у опасности од многобројних напада као што су cryptojacking, ransomware, вируси, и многи други. Како би се рачунар заштитио од напада, доступно је више различитих функционалности. Једна таква функционалност је детекција заснована на потпису.

У терминологији рачунарске безбедности, потпис је типичан отисак или образац повезан са злонамерним нападом на рачунарску мрежу или систем. Овај образац може бити низ бајтова у датотеци (секвенца бајтова) у мрежном саобраћају. Такође може бити у облику неовлашћеног извршавања софтвера, неовлашћеног приступа мрежи, неовлашћеног приступа директоријуму или аномалија у коришћењу мрежних привилегија.

Детекција заснована на потпису је једна од најчешћих техника које се користе за решавање софтверских претњи које се јављају на рачунару. Ове претње укључују вирусе, малвер, црве, тројанце и још много тога. Постизање заштите у великој мери може зависити од добро осмишљене, напредне детекције засноване на потписима. Овај тип детекције подразумева да антивирус има унапред дефинисан скуп статичких потписа (отисака прстију) који представљају познате мрежне претње. Ове претње се разликују једна од друге због свог јединственог кодирања.

Када антивирусни скенер крене у акцију, он креће да креира одговарајуће потписе за сваку датотеку и почиње да их упоређује са познатим потписима у свом складишту. Стално прати и тражи мрежни саобраћај у потрази за потписима. Ако се пронађе подударање, ова датотека се категорише као „претња“ и датотека је блокирана од предузимања било каквих даљих радњи.

Ова техника је веома популарна. Препознавање злонамерних претњи и додавање њихових потписа у складиште је примарна техника коју користе антивирусни производи. Детекција заснована на потпису је такође критични стуб безбедносних технологија као што су IDS, IPS, заштитни зид и друге. Детекција заснована на потпису се користи веома дуго, још од појаве првих антивирусних програма. Техника није превише сложена, брза је и једноставна за покретање и управљање.

Обзиром на то да свака претња има посебан потпис, овакви системи се боре са великим базама података. За сваки пакет који се анализира потребно је претражити целу базу података. Овај процес може узимати превише ресурса и успорити рад система, што систем чини рањивим за DoS нападе. Неки од злонамерних софтвера чији је циљ заобилажење IDS система користе ову слабост и бомбардују уређаје огромним количинама пакета тако да их IDS системи не могу довољно брзо обрадити. Као резултат тога IDS системи одбацују неке пакете и не врше целокупну анализу, па неки напад може проћи.

Иако је детекција заснована на потпису од велике важности за заштиту рачунара, скуп доступних претњи није статичан, већ се свакодневно повећава, а претње се усавршавају. Због тога постоји потреба за вишеслојним системима где се систем за детекцију упада

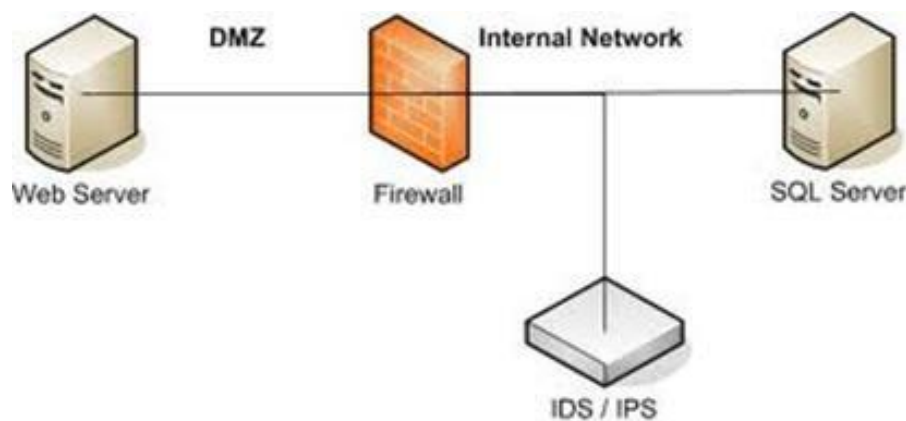
заснован на потпису користи у комбинацији са другим безбедносним техникама. Неке од тих техника су детекција заснована на понашању, откривање претњи помоћу вештачке интелигенције, напредно скенирање малвера и даљинско управљање безбедношћу система.

3 Детекција и превенција упада

3.1 IDS - системи за детекцију упада у рачунарске мреже

Детекција упада у рачунарске мреже представља начин за регистровање било какве неауторизоване активности која се јавља на некој рачунарској мрежи или било ком уређају или систему који је повезан на ту мрежу. Ову технику користимо да идентификујемо потенцијалног нападача пре него што нанесе штету мрежи или пословним активностима. IDS систем прикупља и анализира информације из различитог спектра области у оквиру рачунара или мреже како би идентификовао могуће безбедносне проблеме који укључују упаде од стране неауторизованих лица ван мреже или нападе од стране ауторизованих лица у оквиру мреже.

Детекција упада у рачунарске мреже може се поделити у три главне области у зависности од метода које користе: детекција заснована на потпису, детекција заснована на аномалијама и детекција заснована на спецификацији. Детекција заснована на потпису као што смо већ описали односи се на детекцију познатих рањивости и напада. Потребно је да постоји листа правила (потписа) познатих претњи како би ова детекција функционисала. Техника прати догађаје и покушава да детектује шаблоне. Уколико се шаблон поклопи са неким потписом издаје се упозорење, у супротном пакет се допушта на мрежи. Скуп потписа је потребно редовно ажурирати. За разлику од ове технике, детекција заснована на аномалијама детектује претње које нису раније регистроване. Филтрирање пакета се заснива на предефинисаном скупу правила и шаблона. Уколико пакет крши правила и не уклапа се у шаблоне издаје се упозорење и шаље SIEM систему. Овај тип детекције је комплекснији и обично укључује алгоритме машинског учења како би успешно вршио детекцију. Детекција заснована на спецификацији је заснива се на дефинисаној програмској спецификацији. Она је одговорна за праћење процеса и поклапање стварних података са програмским подацима и у случају било каквог неуобичајеног понашања биће издато упозорење. Систем се мора се одржавати и ажурирати сваки пут када је направљена промена у програмима за надзор како би био у стању да открије нове нападе. Код ових система број лажних позитивних резултата може бити мањи од система заснованом на аномалијама.



Слика 3. Позиција једног IDS система

Систем за детекцију упада у рачунарске мреже је најчешће тип софтвера који скенира мрежу тј. сав долазни и одлазни саобраћај и креира извештаје како би SIEM системи извршили додатне анализе на основу којих ће се предузети даље акције.

Системи за детекцију упада у рачунарске мреже могу се поделити у четири главне категорије. То су:

- Системи за детекцију упада у рачунарске мреже (NIDS) - као што само име говори, ови системи детектују упад скенирањем мрежног саобраћаја. NIDS системи добијају приступ мрежном саобраћају повезивањем на мрежне хабове, мрежне свичеве или тзв. network tap-ове. Код ових система сензори се постављају на одређеним тачкама мреже. Сензори хватају сав саобраћај и анализирају садржај појединачних пакета у потрази за злонамерним саобраћајем. Када се злонамерни саобраћај, напад или необично понашање детектује, упозорење може бити послато администратору. Snort је пример оваквог система. Коришћење овог система у пракси: овакав систем можемо инсталирати на некој подмрежи на којој постоје firewall-ови како бисмо детектовали да ли неко покушава да пробије неки од заштитних зидова;
- Системи за детекцију упада у хоста (HIDS) - овај тип система је специфичан за хоста. HIDS агент се инсталира на хосту и сав долазни и одлазни саобраћај се логује, а приступ хосту се надгледа. Систем такође анализира системске позиве, прати статус свих фајлова на хосту попут бинарних датотека, датотека лозинки, листи контрола приступа и обавештава администратора уколико је било који фајл обрисан или модификован. Систем може пратити само саобраћај на том хосту а не и целокупан саобраћај на мрежи. OSSEC је пример овог типа система. Коришћење овог система у пракси: Овакав систем се може користити на машинама које су неопходне за извршење неке операције или процеса, од којих се не очекује да мењају своје карактеристике;
- Системи за детекцију упада базирани на протоколу (PIDS) - овакви системи се обично инсталирају на серверу, пратећи и анализирајући коришћење неког протокола на серверу. На пример PIDS је инсталиран на web server-у и користи се за праћење и анализу HTTP и HTTPS долазног и одлазног саобраћаја.
- Хибридни системи за детекцију упада - хибридни системи комбинују неке од прве три категорије. Ови системи су обично ефикаснији од прва три типа система. Пример овог система је Prelude.

Овде ћемо такође поменути још два типа система: систем за детекцију упада на одређену површину (PIDS) и системи за детекцију упада базирани на надгледању виртуелних машина (VMIDS).

- PIDS - откривају и прецизирају покушаје упада кроз граничне тачке (ограде) критичне инфраструктуре. Коришћењем електронике или напреднијих оптичких влакана који су прикључени на граничне тачке (ограде) систем детектује сметње. Ако је упад откривен а систем га сматра покушајем упада, активира се аларм;
- VMIDS - детектује упаде надгледањем виртуелне машине. Ово је најновији тип система и још увек су у развоју.

Највећа мана ових система је велики број лажно позитивних “напада” тј. лажних аларма и количина напора коју треба уложити како би база потписа била исправна и стално ажурирана на најновију верзију. Зато је након инсталације оваквог софтвера потребно фино подешавање софтвера које подразумева исправно постављање параметара тако да софтвер добро разликује малициозан саобраћај од дозвољеног саобраћаја.

Поређење IDS са firewall-ом

И IDS и заштитни зид су повезани са безбедношћу мреже, али се IDS разликује од заштитног зида по томе што заштитни зид гледа споља на упаде како би спречио њихово појављивање. Заштитни зидови ограничавају приступ између мрежа како би спречили упад, а ако је напад унутар мреже, не шаље се упозорење. IDS анализира сумњиви упад када се догоди, а затим активира аларм.

3.2 IPS - системи за превенцију упада у рачунарске мреже

У домену безбедности рачунарских мрежа осим детекције упада у рачунарске мреже често се може чути и појам превенције упада у рачунарске мреже. Ови појмови су слични скоро у свему осим у једној карактеристици. IPS је активна технологија док је IDS пасивна технологија. IDS само детектује и алармира претње, док IPS покушава да блокира саобраћај и претње у тренутку када их детектује. Системи за спречавање упада се посматрају као проширење система за откривање упада (IDS) јер и IPS и IDS управљају мрежним саобраћајем и системским активностима за злонамерне активности.

IPS бележи информације у вези са уоченим догађајима, обавештава безбедносне администраторе о важним забележеним догађајима и прави извештаје. Многи IPS такође могу да одговоре на откривену претњу покушавајући да спрече њен успешан напад. Они користе различите технике одговора, које укључују заустављање самог напада, промену безбедносног окружења или промену садржаја напада.

IPS системи се могу поделити у 4 групе:

- системи за превенцију упада засновани на мрежи (NIPS) - прате целокупан мрежни саобраћај у потрази за сумњивим саобраћајем и анализирају активности различитих протокола;
- бежични системи за превенцију упада (WIPS) - прате саобраћај на бежичним мрежама у потрази за сумњивим саобраћајем и анализирају активности различитих бежичних протокола;
- анализа понашања мреже (NBA) - испитује мрежни саобраћај да би се идентификовале претње које стварају необичне токове саобраћаја, као што су дистрибуирани напади DoS, специфични облици малвера и кршења правила;
- системи за превенцију упада базирани на хосту (HIPS) - уграђени софтверски пакет који управља једним хостом у потрази за сумњиве активности скенирањем догађаја који се дешавају унутар тог хоста.

У наредној табели дато је поређење типова IPS система.

Тип IPS	Детектовани тип малициозне активности	Домет по сензору	Предности
NIPS	Активности мрежног, транспортног и апликационог нивоа TCP/IP стека	Више подмрежа и група хостова	Једини тип који може анализирати широк спектар протокола
WIPS	Бежичне активности, неауторизовано коришћење бежичних мрежа	Више бежичних мрежа и групе бежичних клијената	Једини тип који може да предвиди активности бежичних протокола
NBA	Активности мрежног, транспортног и апликационог нивоа TCP/IP стека	Више подмрежа и група хостова	Ефективнији у идентификацији извиђачког скенирања и DoS напада, као и спречавање великих напада злонамерног софтвера
HIPS	Активности хоста и оперативног система; Активности мрежног, транспортног и апликационог нивоа TCP/IP стека	Индивидуални хост	Може анализирати саобраћај који је креиран у криптованој комуникацији од тачке до тачке

Табела 1. Типови IPS система

Поређење IDS и IPS система

Главна разлика између система за превенцију упада и система за откривање упада је:

- Системи за спречавање упада су постављени на одбраменој линији и у стању су да активно спрече или блокирају упаде који се открију;
- IPS може да предузме радње као што је слање аларма, одбацивање откривених злонамерних пакета, ресетовање везе или блокирање саобраћаја са IP адресе која је идентификована као злонамерна;
- IPS такође може да исправи грешке у цикличној провери редунданције (CRC), дефрагментира токове пакета, ублажи проблеме са TCP секвенцирањем и очисти нежељене опције транспорта и мрежног слоја.

3.3 Позиционирање IDS и IPS система у мрежи

Правилно постављање система за откривање упада је критично и варира у зависности од мреже. Најчешћа позиција је иза заштитног зида, на ивици мреже. Ова пракса пружа IDS-у високу видљивост саобраћаја који улази у мрежу али неће детектовати никакав саобраћај између корисника на мрежи. Ивица мреже је тачка у којој се мрежа повезује са интернетом. Још једна пракса која се може постићи ако је доступно више ресурса је стратегија у којој ће техничар поставити свој први IDS на тачку највеће видљивости и у зависности од доступности ресурса поставити други на следећу највишу тачку, настављајући тај процес док се све тачке мреже не покрију.

Ако се IDS постави изван мрежног заштитног зида, његова главна сврха би била одбрана од шума са интернета, али, што је још важније, одбрана од уобичајених напада, као што су скенирање портова и мапирање мреже. IDS на овој позицији би надгледао слојеве од 4. до

7. нивоа ОСИ модела и био би заснован на потписима. Ово је веома корисна пракса, јер уместо приказивања стварних пробоја у мрежу који су прошли кроз заштитни зид, биће приказани покушаји упада што смањује количину лажних позитивних резултата. IDS на овој позицији такође помаже у смањењу времена потребног за откривање успешних напада на мрежу.

Понекад ће IDS са напреднијим функцијама бити интегрисан са заштитним зидом како би могао да пресретне софистициране нападе који улазе у мрежу. Друга опција за IDS постављање је унутар постојеће мреже. Ово ће открити нападе или сумњиве активности унутар мреже. Игнорисање безбедности унутар мреже може изазвати многе проблеме. Корисници могу да изазову безбедносне ризике или се омогућује нападачу који је већ провалио у мрежу да слободно лута. Сложена безбедност унутар мреже отежава чак и хакерима да злоупотребе мрежу.

3.4 Примери упада и компромитовања безбедности система

У овом поглављу таксативно ћемо навести неке примере који захтевају повећану пажњу корисника. То су упади у систем, упади у фајлове и упади у мреже.

Примери упада у систем:

- Активан приступ сајтовима на којима се дуго нисмо пријављивали
- Пријављивање у нерадно време или време спавања
- Системска грешка у идентификацији корисника
- Аутоматско креирање корисничког налога
- Измене у системским
- Брисање системских логова
- Смањење перформанси система
- Нестандардни приказ графичких елемената и нападних реклама
- Систем се изненада руши и покреће испочетка без учествовања корисника

Примери напада на фајлове:

- Непознате датотеке и програми на систему
- Измене величине датотеке и дозвола приступа датотеци
- Чудна идентификација датотека на локалном систему
- Обрисани фајлови

Примери мрежних упада:

- Поновљени покушаји пријаве са удаљених локација
- Нагло повећање саобраћаја по јединици времена
- Поновљене провере постојећих услуга и сервиса
- Насумични подаци логова у датотекама логовања

Како би се спречили и смањили наведени напади потребно је да мрежни администратори преузму одређене акције. Те акције могу бити:

- Често ажурирање базе података антивирусних потписа
- Конфигурисање заштитног зида да филтрира IP адресе уљеза
- Активирање звучног сигнала као индикација напада
- Присилно слање TCP, FIN или RST пакета како би се принудно прекинула веза
- Чување датотека детектованих пакета за будућу анализу
- Чување информација о свим досадашњим нападима (IP уљеза, IP жртве, временска ознака)
- Слање обавештења администратору о нападу

3.5 Suricata

Suricata је систем за детекцију и превенцију упада у рачунарске мреже и мрежну безбедност високих перформанси. Софтвер је бесплатан, отвореног је кода а систем развија заједница Open Information Security Foundation (OSIF). Систем користи скуп правила и језик потписа (Signature) како би детектовао и спречио претње, кршење правила и злонамерна понашања.

Као што је већ описано у претходним поглављима систем за детекцију упада детектује и алармира постојање претње. Систем за превенцију упада са друге стране такође преузима акције над догађајима и покушава да блокира нежељени саобраћај. Suricata пружа подршку за оба система а такође и врши дубоку и детаљну анализу пакета.

Suricata заузима мало меморије, јефтин је софтвер и може пружити одличан увид у дешавања на нашој мрежи са безбедносне стране. Осим Suricata постоје и други софтвери попут Snort i Bro. Оно што овај софтвер разликује од поменутих је то што Suricata има подршку за multithreading. Ово омогућава истовремено коришћење више језгара и боље распоређивање оптерећења саобраћаја. Самим тим, не морамо се обазирати на број креираних правила јер ће њихов број у мањој мери утицати на брзину обраде саобраћаја. Suricata се може користити и у спрези са тзв. SIEM системима. SIEM системи су системи за управљање информацијама и догађајима и они могу користити излаз из Suricata како би унапредили сопствена детекциона правила и процесе.

Suricata може логовати HTTP захтеве, логовати и чувати TLS сертификате, екстрактовати фајлове из токова и чувати их на диску, логовати све DNS захтеве и одговоре. Пружа потпуну подршку за креирање рсар фајлова, чиме је омогућена једноставна анализа. Све ово чини овај софтвер моћним за креирање једног система за управљање мрежом (NSM).

3.5.1 Suricata потписи

Код овог софтвера потписи могу на први поглед деловати комплексно, али након упознавања са њиховом структуром и принципима процесирања креирање сопствених правила постаје једноставно. Потпис се састоји из 3 дела. Ти делови су:

- Акција - која се предузима када се саобраћај поклопи са правилом;
- Заглавље - описује хоста, IP адресе, портове, протоколе и смер саобраћаја (долазни, одлазни)

- Опције - специфицирају идентификатор потписа (sid), логове, регуларне изразе који треба да се поклопе са заглављем пакета, класификациони тип и остале елементе који помажу да се лакше идентификује сумњив саобраћај.

Општа структура потписа је следећа: Акција Заглавље Опције. У наставку је дат пример потписа са идентификатором sid:2100498

```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0/28/root/29/"; classtype:bad-unknown; sid:2100498; rev:7; metadata:created_at
2010_09_23, updated_at 2010_09_23;)
```

Подвучени део је акција, подебљан део је заглавље а искошен део су опције правила.

Акција

У софтверу постоје 4 типа акција у зависности од мода у којем се користи Suricata (IDS или IPS мод). Типови акција су:

- Pass - софтвер престаје да скенира пакет и пропушта га без генерисања упозорења;
- Drop - уколико софтвер ради у IPS моду процесирање пакета се одмах зауставља, генерише се упозорење. Уколико је тип конекције TCP веза се одмах прекида;
- Reject - уколико софтвер ради у IPS моду, шаље се TCP reset пакет, а тренутни пакет се одбацује;
- Alert - софтвер генерише упозорење и логује у лог фајлу за будућу анализу.

Заглавље

Сваки потпис садржи заглавље које описује мрежни протокол, изворишну и одредишну IP адресу, портове и смер саобраћаја. Општа структура заглавља је следећа:

```
<PROTOCOL> <SOURCE IP> <SOURCE PORT> -> <DESTINATION IP> <DESTINATION PORT>
```

Протокол може бити TCP, UDP, ICMP, IP, неки апликациони протоколи. Изворишна и одредишна адреса могу бити конкретне IP адресе или опсези адреса, или може бити специјална реч ану која детектује све адресе и мреже. Стрелица -> указује на смер кретања саобраћаја. Потписи такође могу користити бидирекциони маркер <> који детектује саобраћај у оба смера. Уколико желимо да детектујемо малициозни саобраћај који се шаље са нашег уређаја као изворишну адресу стављамо нашу IP адресу, а као одредишну можемо ставити ану. Уколико желимо да детектујемо долазни малициозни саобраћај нашу адресу стављамо као одредишну, а као изворишну постављамо специјалну реч ану. У заглављу је могуће користити и негацију која је представљена ! знаком. У наставку је дат пример коришћења негације порта:

```
alert ssh any any -> 203.0.113.0/24 !22 (sid:1000000;) - овим потписом се прати долазни
саобраћај на датој адреси путем ssh протокола а чији одредишни порт није 22.
```

Овакав пример потписа није много користан јер не садржи никакве додатне информације о пакету, нити класификациони тип. Ове информације додају се у опцијама.

Опције

Параметри унутар заграда садрже различите опције и кључне речи које се могу користити за детектовање специфичних делова пакета, затим да класификују правило, или логују посебне поруке. Док се делови заглавља односе на адресу, портове или протоколе, опције се односе на податке унутар пакета. Опције се одвајају помоћу ; и користе кључ - вредност формат. Код неких опција се наводи само име у оквиру заграда.

Пример заглавља проширићемо додатном опцијом msg која представља поруку која објашњава о чему је упозорење. У нашем случају упозорење ће бити о томе да се ssh саобраћај одвија на другом протоколу. msg опција дизајнирана је да представља опис упозорења намењен човеку. Порука би требало да буде описна и да додаје контекст упозорењу како би неко ко анализира креирана упозорења разумео зашто је упозорење креирано. У овом поглављу детаљније ћемо описати кључне речи content и неке кључне речи које представљају метаподатке.

У нашем примеру *content:"uid=0/28/root/29/"* након кључне речи content следи стринг који ће се претраживати у подацима пакета. Помоћу ове кључне речи можемо тражити специфичне апликационе протоколе и онда проверити садржај пакета у потрази за специфичним бајтовима и стринговима или детекцији неког од регуларног израза.

Пример:

```
alert dns any any -> any any (msg:"DNS LOOKUP for your_domain.com"; dns.query; content:"your_domain.com"; sid:1000001;) - детектује све dns пакете са датим сајтом, али да би се детектовао и нпр. YOUR_DOMAIN.com потребно је у опције ставити и кључну реч nocase.
```

Сваки Suricata потпис мора да садржи јединствени идентификатор sid. Уколико два потписа имају исти идентификатор софтвер се неће покренути и генерисаће се грешка слична следећем примеру:

```
19/8/2022 -- 01:17:40 - <Error> - [ERRCODE: SC_ERR_DUPLICATE_SIG(176)] - Duplicate signature "drop ssh any any -> 127.0.0.0/8 !22 (msg:"blocked invalid ssh"; sid:10000000;)"
```

Уколико креирамо сопствене потписе можемо користити опсег 1000000-1999999, а уграђена правила користе опсег 2200000-2299999. Коришћење осталих опсега дато је на страници [15].

Потписе је могуће и ажурирати. На пример, почетни потпис можемо ажурирати тако да детектује само ssh саобраћај на порту 2022. У опцијама ћемо додати кључну реч rev:2.

```
alert ssh any any -> 203.0.113.0/24 2022 (msg:"SSH TRAFFIC on non-SSH port"; sid:1000000; rev:2;)
```

Уколико msg информација није довољна, могуће је дефинисати и reference опцију која омогућује да укључимо локацију на којој се може добити више информација о нападу или проблему које правило треба да детектује.

Suricata може класификовати саобраћај у складу са преконфигурисаним скупом категорија које су укључене приликом инсталације који се налази у оквиру `classification.config` фајла. Део садржаја тог фајла приказан је испод.

```
#
```

```
# config classification:shortname,short description,priority
```

```
#
```

```
config classification: not-suspicious,Not Suspicious Traffic,3
```

```
config classification: unknown,Unknown Traffic,3
```

```
config classification: bad-unknown,Potentially Bad Traffic, 2
```

Као што је напоменуто у заглављу фајла сваки тип класификације садржи три податка:

- Кратко име - читљиво од стране машине, `not-suspicious`, `unknown`, `bad-unknown`;
- Опис класификације који се користи уз упозорење, на пример `Not Suspicious Traffic` који означава да детектовани саобраћај није сумњив;
- Приоритетно поље - одређује редослед у коме ће се потписи процесирати од стране софтвера. Највећи приоритет је 1.

За класификацију се користи кључна реч `classtype` а у почетном примеру тип је `classtype:bad-unknown` који означава да је саобраћај потенцијално лош. Могуће је променити дефинисани приоритет класификације додавањем кључне речи `priority: n`, где је `n` нови приоритет од 1 до 255.

Могуће је користити и кључну реч `target`. Вредност може бити `src_ip` или `dest_ip` а сврха ове опције је правилно идентификовање изворних и одредишних хостова у логовима упозорења софтвера (`eve.json`) тј. одредити која страна у потпису је одредишна. Уколико у почетном примеру додамо опцију `target:dest_ip`; то ће резултовати додатним пољима у делу упозорења у `eve.json` фајлу.

```
...
"source": {
  "ip": "127.0.0.1",
  "port": 35272
},
"target": {
  "ip": "203.0.113.1",
  "port": 2022
}
...
```

Ово може бити корисно за даље прослеђивање логова SIEM софтверу како би се једноставније извршила претрага упозорења који потичу од одређеног хоста или напада који су усмерени ка одређеном одредишту у нашој мрежи.

Могуће је дефинисати и сопствене парове кључних речи и вредности коришћењем кључне речи metadata. Потребно је само испоштовати форму metadata: нова_кључна_реч вредност;. Осим описаних кључних речи постоји још много других речи које се односе на неки протокол, на пример: ttl везан за IP протокол, seq, ack, window везани за TCP протокол, http.uri, http.method, http.header, http.request_body итд. везани за HTTP протокол и још много других.

Ttl је кључна реч која се користи за проверу IP time-to-live вредности која се налази у заглављу пакета. Формат наредбе је ttl:10;

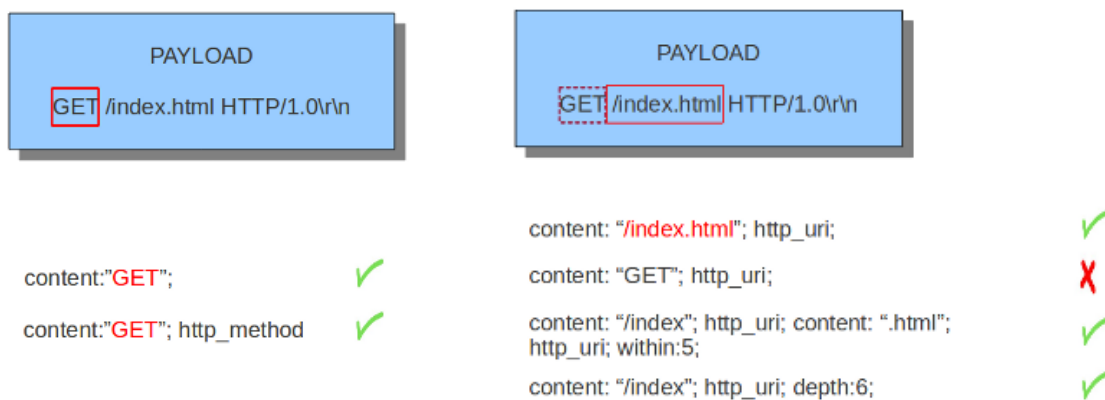
seq кључна реч се може користити за проверу специфично TCP броја секвенце. Пример: seq:0; Број секвенце представља број који се генерише насумично од стране обе крајње тачке TCP везе. И клијент и сервер креирају број секвенце, који се повећава за један са сваким бајтом који пошаљу. Дакле, овај редни број је различит за обе стране. Овај редни број морају да верификују обе стране везе. Преко бројева секвенце, TCP управља потврдом, редоследом и поновним преносом. Његов број се повећава са сваким бајтом података који пошљалац пошаље. Seq помаже у праћењу ком месту у току података припада бајт. Ако је SYN флег постављен на 1, тада је редни број првог бајта података овај број плус 1 (дакле, 2).

Кључна реч ack користи се за детекцију одређеног ack-а. ACK је потврда пријема свих претходних бајтова (података) послатих са друге стране TCP везе. У већини случајева сваки пакет TCP везе има ACK флег након првог SYN и број потврде који се повећава са пријемом сваког новог бајта података. Кључна реч ack се може користити у потпису за проверу одређеног броја TCP потврде. Пример: ack:1;

Кључна реч http.uri користи се за детекцију URI локације захтева.

Кључна реч http.method може се користити за детекцију тачно одређеног HTTP метода.

Обе кључне речи се заправо користи заједно са кључном речи content. На следећој слици су приказани примери коришћења ове две кључне речи.



Слика 4. Пример детекције HTTP заглавља са GET и POST методом

Постоји још мноштво кључних речи које се могу користити за креирање потписа у оквиру Suricata софтвера, а овде смо навели само неке од најбитнијих.

3.6 Додатни примери система за детекцију и превенцију упада у рачунарске мреже

Најбољи системи за детекцију и превенцију упада треба да се успешно боре са три задатка. То су: идентификација лажно позитивних напада, добро обучено особље и идентификација кључних ризика. Неки софтвери успешније савладавају ове захтеве од осталих, један пример смо детаљније описали у претходним поглављима, а неке мање познате ћемо навести у овом поглављу.

Snort

Snort је бесплатан софтвер за детекцију упада у рачунарске мреже отвореног кода. Креиран је 1998. године а данас је у власништву Cisco. Снорт има више функционалности почев од могућности да врши анализе саобраћаја у реалном времену и логовање пакета на IP мрежама, али не поседује могућност multithreading-a. Снорт изводи анализе протокола, претрагу садржаја и подударање садржаја. Може се користити за детекцију напада, укључујући покушаје детекције података о оперативном систему, семантичке URL нападе, прекорачење бафера, скенирање скривених портова итд. Снорт се може конфигурисати у 3 мода:

- Ослушкивач - програм чита пакете и приказује их на екрану;
- Логер пакета - у овом моду програм логира пакете на диск;
- детектор мрежних упада - у овом моду програм прати мрежни саобраћај и анализира у односу на скуп правила дефинисаних од стране корисника. Програм затим извршава одређене акције у зависности од тога шта је детектовано.

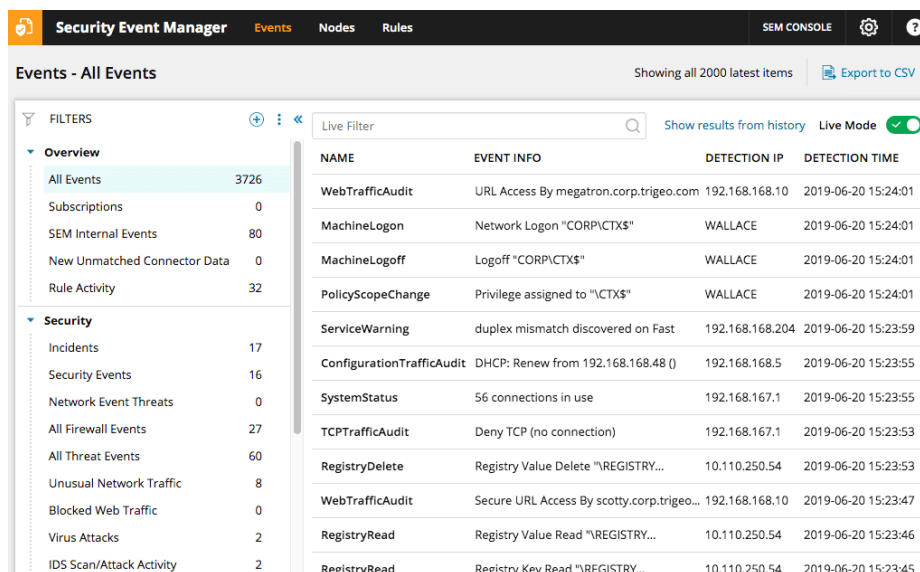
SolarWinds Security Event Manager (SEM)

Овај софтвер користи веома интелигентан приступ откривању претњи. Прикупљајући логове система за откривање упада у мрежу, SEM прикупља информације о врстама напада. Ове информације се затим интегришу са другим логовима инфраструктуре, стварајући огромну количину мрежних података који доприносе откривању претњи. Ови подаци константно оптимизују безбедносне системе и рад IDS-а. Помоћу SEM-а се могу идентификовати проблематични уређаји на мрежи, користити подаци за креирање извештаја о процени ризика за власнике уређаја и идентификовати високо напредне претње пре него што изазову огромну штету у систему.

SEM штеди време тако што прати и упозорава на све сумњиве догађаје или активности и аутоматски делује када се открију одређени догађаји. Он користи мрежне сензоре да помогне у откривању упада, спроводи анализу података, идентификује процесе који се користе. Аутоматизацијом процеса где год је то могуће, ове могућности смањују потребу да ручно откривамо претње и сумњиве активности.

Софтвер за откривање упада у мрежу је добар колико и његово окружење. Ако програм није прилагођен кориснику, онда није важно колико је софистициран или богат функцијама, јер просечан корисник неће моћи да комуницира са системом. SEM, упркос томе што нуди неке напредне функционалности, један је од најприкладнијих програма на овој листи. Његов интерфејс је једноставан, са догађајима, чворовима и правилима доступним на

главној траци са функцијама. Све картице су супер брзе за навигацију, а подаци су представљени на графички начин који је лако читљив. Контролна табла је шарена, чиста и динамична.



The screenshot shows the SolarWinds Security Event Manager (SEM) interface. The top navigation bar includes 'Security Event Manager', 'Events', 'Nodes', and 'Rules'. The main header shows 'Events - All Events' and 'Showing all 2000 latest items'. On the left, there is a 'FILTERS' sidebar with categories like 'Overview' and 'Security'. The main area displays a table of events with columns: NAME, EVENT INFO, DETECTION IP, and DETECTION TIME. The table lists various events such as WebTrafficAudit, MachineLogon, MachineLogoff, PolicyScopeChange, ServiceWarning, ConfigurationTrafficAudit, SystemStatus, TCPTrafficAudit, RegistryDelete, WebTrafficAudit, RegistryRead, and RegistryRead.

NAME	EVENT INFO	DETECTION IP	DETECTION TIME
WebTrafficAudit	URL Access By megatron.corp.trigeo.com	192.168.168.10	2019-06-20 15:24:01
MachineLogon	Network Logon "CORP\CTXS"	WALLACE	2019-06-20 15:24:01
MachineLogoff	Logoff "CORP\CTXS"	WALLACE	2019-06-20 15:24:01
PolicyScopeChange	Privilege assigned to "CTXS"	WALLACE	2019-06-20 15:24:01
ServiceWarning	duplex mismatch discovered on Fast	192.168.168.204	2019-06-20 15:23:59
ConfigurationTrafficAudit	DHCP: Renew from 192.168.168.48 ()	192.168.168.5	2019-06-20 15:23:55
SystemStatus	56 connections in use	192.168.167.1	2019-06-20 15:23:55
TCPTrafficAudit	Deny TCP (no connection)	192.168.167.1	2019-06-20 15:23:53
RegistryDelete	Registry Value Delete "REGISTRY..."	10.110.250.54	2019-06-20 15:23:53
WebTrafficAudit	Secure URL Access By scotty.corp.trigeo...	192.168.168.10	2019-06-20 15:23:47
RegistryRead	Registry Value Read "REGISTRY..."	10.110.250.54	2019-06-20 15:23:46
RegistryRead	Registry Key Read "REGISTRY..."	10.110.250.54	2019-06-20 15:23:45

Слика 5. SolarWinds Security Event Manager софтвер

Kismet

Кисмет је бежични IDS отвореног кода, што значи да се фокусира на бежичне протоколе као што су Bluetooth и WiFi. Овај програм прати и открива неовлашћене приступне тачке, које су чешће него што мислимо. Неовлашћене приступне тачке могу случајно да креирају запослени изненађујуће лако. Кисмет открива недостатке у конфигурацији и поставци мреже, прескачући канале како би проширио свој домет. Једна од најбољих функционалности у вези са Кисметом је проширена подршка за додатке. Могуће је коришћење додатака за проширење функционалности веб корисничког интерфејса. Kestrel је популаран додаток за Кисмет за додавање функција мапирања уживо у Кисмет интерфејс. Користи библиотеку летака и приказује локације уређаја и мреже на мапи уживо. Груписање пинова се користи за густе области на мапи.

IoD Screwdriver је још један користан додаток, иако садржи пуно реклама. Ту је и додаток за генерисање извештаја за груписање уређаја према SSID-у или BSSID-у, са могућношћу извоза у CSV или PDF. Као IDS отвореног кода, Кисмет има заједницу корисника који непрестано повећавају број инсталираних додатака. Могућност повезивања са заједницом је још једна предност Кисмета. Могуће је ћаскање са члановима заједнице Кисмет на Discord-у. То значи да је подршка лако доступна.

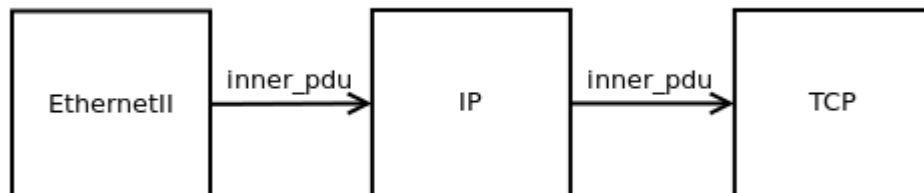
Главни недостатак Кисмета је то што може потрајати неко време за претрагу мрежа. Може да ради на Линуксу и на Windows 10 са WSL фрејмворком. Иако је импресиван на много начина, вреди напоменути да систем за откривање упада у мрежу отвореног кода не може да понуди исти ниво софистицираности и поузданости као алати затвореног кода. То обично значи да IDS алати отвореног кода нису посебно погодни за пословну употребу.

3.7 Libtins

Ово поглавље посветићемо једној од wrapper библиотека над libpcap библиотеком, Libtins. Libtins је мултиплатформска C++ апликација високог нивоа за детекцију и креирање пакета. Главна сврха ове библиотеке је пружање једноставног, ефикасног и независног начина за креирање алата који треба да шаљу, примају и манипулишу мрежним пакетима. Веома је једноставна за коришћење. Приказаћемо пример који штампа изворишне и одредишне адресе и портове сваког TCP пакета детектованог на неком интерфејсу.

```
#include <iostream>
#include <tins/tins.h>
using namespace Tins;
using namespace std;
bool callback(const PDU &pdu) {
    // Find the IP Layer
    const IP &ip = pdu.rfind_pdu<IP>();
    // Find the TCP Layer
    const TCP &tcp = pdu.rfind_pdu<TCP>();
    cout << ip.src_addr() << ':' << tcp.sport() << " -> "
         << ip.dst_addr() << ':' << tcp.dport() << endl;
    return true;
}
int main() {
    Sniffer("eth0").sniff_loop(callback);
}
```

Библиотека поседује класу Sniffer која ослушкује пакете на прослеђеном интерфејсу eth0. PDU је класа којом се представљају различита заглавља и помоћу њих се може креирати стек угњеждавањем. EthernetII, IP и TCP су класе које се користе за представљање тих заглавља. Оне наслеђују PDU класу.



Слика 6. Изглед једног угњежђеног PDU објекта

Libtins има више функционалности и подршку за различите протоколе. Неке од функционалности су:

- Креирање и слање мрежних пакета
- Packet sniffing и аутоматска интерпретација пакета
- Читање и креирање PCAP фајлова
- Праћење и састављање TCP токова у реалном времену.
- Дешифровање WEP и WPA2(TKIP and CCMP) криптираних 802.11 фрејмова података у реалном времену и интерпретација дешифрованог садржаја
- Мултиплатформска подршка: x86, x64, ARM and MIPS (probably more).

Подржани протоколи: IEEE 802.11, IEEE 802.3, IEEE 802.1q, Ethernet, ARP, IP, IPv6, ICMP, ICMPv6, TCP, UDP, DHCP, DHCPv6, DNS, RadioTap, MPLS, EAPOL, PPPoE, STP, LLC, LLC+SNAP, Linux Cooked Capture, PPI, PKTAP, NULL/Loopback.

Libtins библиотека је тестирана на огромном броју тестова, а такође је вршено и мерење брзине и поређење са сличним библиотекама које нуде исте функционалности. Показало се да је ова библиотека значајно бржа од свих осталих доступних библиотека осим libpcap библиотеке. На следећој слици су приказани резултати једног теста на којем је мерено процесирање TCP пакета.

Library	Time taken(seconds)	Packets per second
libpcap	0.141	3546099
libtins	0.273	1831501
pcapplusplus	0.38	1315789
dpkt	8.132	61485
libcrafter	12.209	40953
impacket	18.5	27027
scapy	187.082	2672

Слика 7. Benchmark резултати

```

F:\SS\Master\DigitalnaForenzika\Projekat\Debug\Projekat.exe
Welcome to Network Analyzer app!
What would you like to do?
1 - Trace Route
2 - Real-time sniffing
3 - Analyzing traffic from file
4 - Check interfaces status
5 - Exit
Your answer: 2
Real time sniffing selected!!
What would you like to do?
1 - Trace DNS queries
2 - Trace ARP
3 - Analyzing traffic from file
4 - Check interfaces status
5 - Exit
Your answer: 2
[INFO] 192.168.0.102 is at c4:36:6c:72:13:76
[INFO] 192.168.0.105 is at 24:0a:64:1e:f6:d3
  
```

Слика 8. Мултифункционална C++ апликација која користи Libtins библиотеку

3.7.1 Инсталација библиотеке Libtins

Libtins библиотека је доступна на следећем линку: <https://github.com/mfontanini/libtins>

Да би се библиотека користила потребно је да се прво билдује за одређену платформу. За то нам је потребан инсталиран CMake. Такође, потребно је да инсталирамо и прсар, као и прсар sdk. Након скидања свих алата, позиционирамо се у фолдер Libtins библиотеке. Отварамо конзолу и извршавамо редом следећи скуп наредби:

```
mkdir build
cd build
cmake ../ -DPCAP_ROOT_DIR=<Putanja do NPCAP SDK> -DLIBTINS_ENABLE_WPA2=0 -
DLIBTINS_BUILD_SHARED=0
```

У build фолдеру се креира пројекат. Отворити креирани пројекат у visual studio-u и билдовати ALL PROJECTS. Након билдовања креиран је lib фолдер. Позиционирати се у овај фолдер а затим у Debug фолдер. Из креираног фолдера ископирати све фајлове(tins.lib, tins.pdb) у наш нови пројекат. Из libtins пројекта ископирати све из include фолдера (tins фолдер) у наш пројекат у фолдер Include. Из прсар sdk пројекта из фолдера Include ископирати све у фолдер Include нашег пројекта, а такође и све из фолдера lib у фолдер lib. Сада вршимо подешавање нашег пројекта. У оквиру пројекта на почетку морамо укључити препроцесорску директиву #define TINS_STATIC која означава да у пројекту користимо статичку библиотеку.

Потребно је убацити Linker Dependencies у оквиру Project Properties/Linker/Input/Additional Dependencies:

- tins.lib
- Ws2_32.lib
- Iphlpapi.lib
- wpcap.lib

У оквиру Project Properties/C/C++/General/Additional Include Directories убацити Include фолдер. У оквиру Project Properties/Linker/General/Additional Library Directories убацити фолдер lib. Пројекат је сада конфигурисан и повезан са библиотеком и може се кренути са имплементацијом.

4 Закључак

Детекција упада наставља да буде активно поље истраживања. Систем за откривање упада је део одбрамбених операција који допуњује одбрану путем других механизма као што су заштитни зидови, UTM (Unified threat management) итд. Систем за откривање упада у основи детектује знакове напада и затим упозорава. Према методологији детекције, системи за откривање упада се обично категоришу као системи за откривање злоупотребе и откривање аномалија. Из перспективе имплементације, они се могу класификовати на IDS базиран на мрежи или на хосту. У тренутним системима за детекцију упада информације се прикупљају и са мрежних и са хост ресурса. Многи истраживачи и практичари се активно баве овим проблемима у смислу перформанси; систем за откривање упада постаје тачнији јер открива више напада и подиже мање лажно позитивних аларма.

5 Литература

- [1] <https://kompjutas.com/packet-sniffing-predstavljanje-uvod-u-tehnologiju-i-primer-snifovanja/>, Packet Sniffing
- [2] <https://en.wikipedia.org/wiki/Pcap>, Pcap формат
- [3] <http://libtins.github.io/>, Libtins
- [4] <https://resources.infosecinstitute.com/topic/suricata-what-is-it-and-how-can-we-use-it/>, Suricata
- [5] <https://suricata.io/features/>, Suricata
- [6] [https://en.wikipedia.org/wiki/Snort_\(software\)](https://en.wikipedia.org/wiki/Snort_(software)), Snort
- [7] [https://home.sophos.com/en-us/security-news/2020/what-is-a-signature#:~:text=In%20computer%20security%20terminology%2C%20a,byte%20sequence\)%20in%20network%20traffic.](https://home.sophos.com/en-us/security-news/2020/what-is-a-signature#:~:text=In%20computer%20security%20terminology%2C%20a,byte%20sequence)%20in%20network%20traffic.), Signature
- [8] <https://resources.infosecinstitute.com/topic/ids-ips-overview/>, IDS/IPS
- [9] <https://encyclopedia.kaspersky.com/glossary/ids-intrusion-detection-system/#:~:text=IDS%20classification&text=A%20network%20intrusion%20detection%20system,the%20operation%20of%20individual%20devices.>, IDS
- [10] https://en.wikipedia.org/wiki/Packet_analyzer, Packet analyzer
- [11] https://www.researchgate.net/publication/340655192_Classification_and_Importance_of_Intrusion_Detection_System, Класификација IDS система
- [12] <https://resources.infosecinstitute.com/topic/network-design-firewall-idsips/>, IDS
- [13] <https://www.dnsstuff.com/network-intrusion-detection-software>, IDS софтвер
- [14] <https://www.geeksforgeeks.org/intrusion-prevention-system-ips/?ref=lbp>, IPS
- [15] <https://doc.emergingthreats.net/bin/view/Main/SidAllocation>, Suricata sid allocation
- [16] <https://www.digitalocean.com/community/tutorials/understanding-suricata-signatures>, Suricata signatures
- [17] <https://suricata.readthedocs.io/en/suricata-6.0.4/rules/index.html>, Suricata rules