PYTHON ASOSIDA REAL VAQTDA TRAFIK ANOMALIYALARINI ANIQLASH MODULINI YARATISH

Ramazonova Madina Shavkatovna

Murodov Ma’murjon Ma’rupovich

Ahmadov Ulug‘bek Said o‘g‘li

Muxammad al – Xorazmiy nomidagi TATU “Kiberxavfsizlik va kriminalistika” kafedrasi assistentlari,

Abdullayev Xushnud Raxmatulla o‘g‘li

Muxammad al – Xorazmiy nomidagi TATU talabasi

Annotatsiya: Ushbu maqolada Python dasturlash tili asosida real vaqtda tarmoq trafigi anomaliyalarini aniqlash modulining yaratilishi batafsil yoritilgan. Tizim Scapy kutubxonasining past darajadagi paket ushlash imkoniyatlaridan foydalanib, SYN Flood, UDP Flood va HTTP Flood kabi DDoS hujum turlarini aniqlash qobiliyatiga ega. Anomaliyalarni aniqlash uchun sliding window algoritmi va Shannon entropiyasi kabi statistik usullar qo‘llanilgan. Tizim arxitekturasi modular tamoyilga asoslangan bo‘lib, to‘rtta asosiy moduldan iborat: anomaliya aniqlash, paket ushlash, avtomatik bloklash va boshqaruv moduli. Maqolada har bir modulning ishlash printsipi, iptables bilan integratsiya, multithreading orqali parallel jarayonlarni boshqarish va tizimning umumiy ma’lumot oqimi chuqur tahlil qilingan.

Kalit so‘zlar: tarmoq xavfsizligi, anomaliya aniqlash, DDoS hujum, Python, Scapy, sliding window algoritmi, Shannon entropiyasi, SYN Flood, UDP Flood, HTTP Flood, iptables, modular arxitektura, real-time monitoring, paket tahlili, avtomatik bloklash, multithreading.

DEVELOPMENT OF A REAL-TIME NETWORK TRAFFIC ANOMALY DETECTION MODULE USING PYTHON

Ramazonova Madina Shavkatovna

Murodov Ma'murjon Ma'rupovich

Ahmadov Ulug‘bek Said o‘g‘li

Assistants of the Department of Cybersecurity and Digital Forensics, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Abdullayev Xushnud Raxmatulla o‘g‘li

Student of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Abstract: This paper presents the development of a real-time network traffic anomaly detection module using Python. The proposed system utilizes the Scapy library to capture and analyze network packets and detect DDoS attacks such as SYN Flood, UDP Flood, and HTTP Flood. Statistical methods, including the sliding window algorithm and Shannon entropy, are applied to identify abnormal traffic behavior. The system consists of packet capture, anomaly detection, automatic blocking, and management modules. Integration with Linux iptables enables automatic threat mitigation, while multithreading improves performance and scalability.

Keywords: network security, anomaly detection, DDoS attack, Python, Scapy, sliding window algorithm, Shannon entropy, SYN Flood, UDP Flood, HTTP Flood, iptables, real-time monitoring.

РАЗРАБОТКА МОДУЛЯ ОБНАРУЖЕНИЯ АНОМАЛИЙ СЕТЕВОГО ТРАФИКА В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ PYTHON

Рамазанова Мадина Шавкатовна

Муродов Маъмуржон Марупович

Ахмадов Улугбек Саид угли

Ассистенты кафедры «Кибербезопасность и цифровая криминалистика» Ташкентского университета информационных технологий имени Мухаммада аль-Хорезми

Абдуллаев Хушнуд Рахматулла угли

Студент Ташкентского университета информационных технологий имени Мухаммада аль-Хорезми

Аннотация: В статье представлена разработка модуля обнаружения аномалий сетевого трафика в режиме реального времени на основе языка программирования Python. Предлагаемая система использует библиотеку Scapy для перехвата и анализа сетевых пакетов и обнаружения DDoS-атак типов SYN Flood, UDP Flood и HTTP Flood. Для выявления аномального поведения трафика применяются статистические методы, включая алгоритм скользящего окна и энтропию Шеннона. Система состоит из модулей захвата пакетов, обнаружения аномалий, автоматической блокировки и управления. Интеграция с Linux iptables обеспечивает автоматическое противодействие угрозам, а многопоточность повышает производительность и масштабируемость решения.

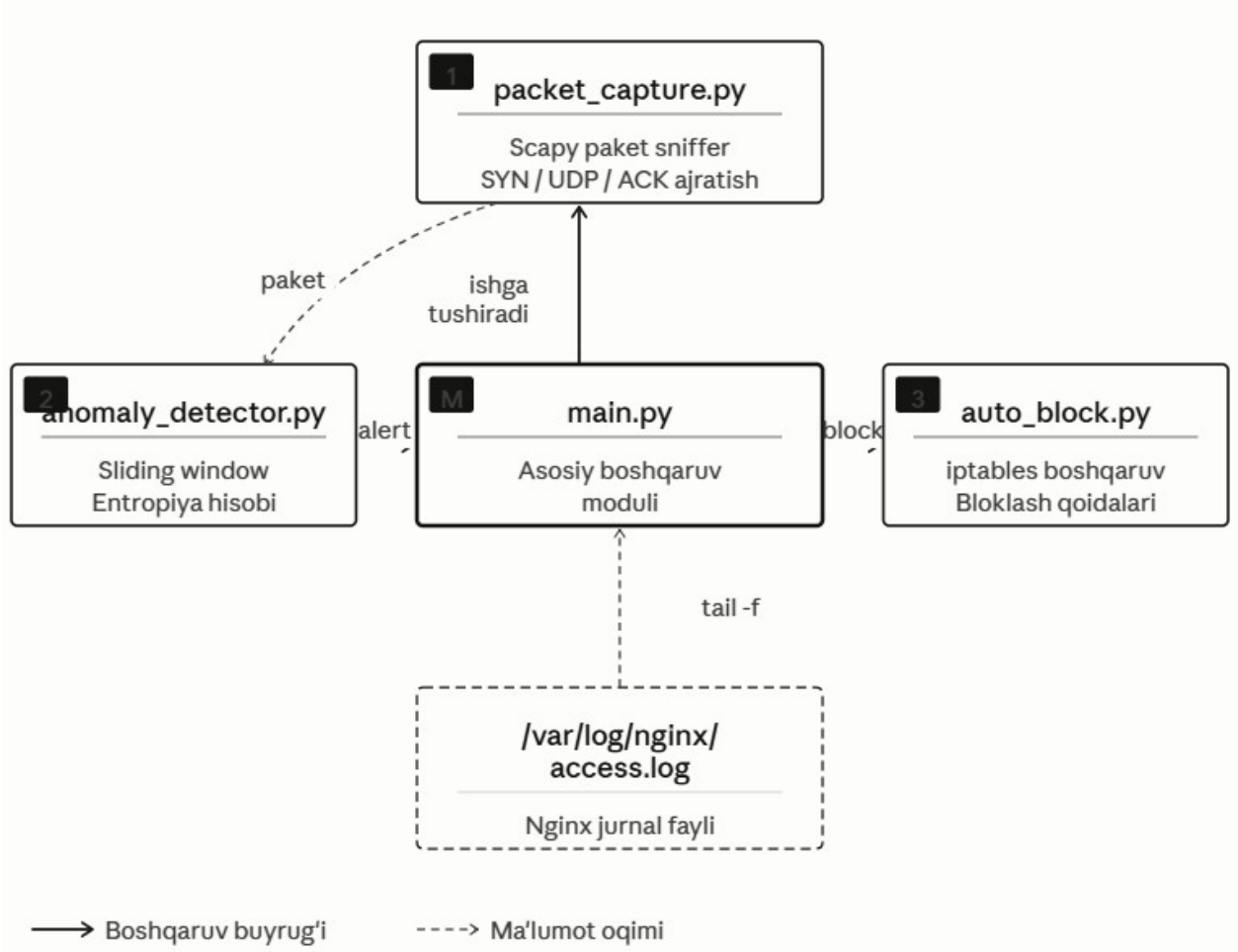
Ключевые слова: сетевая безопасность, обнаружение аномалий, DDoS-атака, Python, Scapy, алгоритм скользящего окна, энтропия Шеннона, SYN Flood, UDP Flood, HTTP Flood, iptables, мониторинг в реальном времени.

Ishlab chiqilgan dasturiy tizimning markaziy vazifasi tarmoq trafigini real vaqtda kuzatish, undagi anomal naqshlarni aniqlash va aniqlangan tahdidlar haqida tegishli komponentlarni xabardor qilishdan iborat. Tizim Python 3.12 dasturlash tilida yaratilib, Scapy kutubxonasining past darajadagi paket ushlash imkoniyatlaridan foydalanadi. Anomaliyalarni aniqlash uchun sliding window algoritmi va Shannon entropiyasi kabi statistik usullar qo‘llanildi. Tizim arxitekturasi modular tamoyilga asoslangan bo‘lib, har bir funksional vazifa alohida Python moduliga ajratildi: paket ushlash, anomaliya aniqlash, avtomatik bloklash va yuqori darajadagi orkestratsiya. Bunday yondashuv har bir

komponentni alohida sinovdan o‘tkazish, mustaqil takomillashtirish va boshqa loyihalarda qayta ishlatish imkonini beradi.

Tizim arxitekturasi modular tamoyil asosida qurilgan bo‘lib, har bir funksional vazifa mustaqil Python modullariga ajratilgan. Jumladan, paketlarni ushlash, trafikni tahlil qilish, anomal IP-manzillarni aniqlash, avtomatik bloklash va tizim orkestratsiyasini boshqarish alohida komponentlar ko‘rinishida tashkil etildi. Ushbu yondashuv tizimning moslashuvchanligini oshiradi, har bir modulni alohida sinovdan o‘tkazish va mustaqil ravishda takomillashtirish imkonini beradi. Bundan tashqari, yaratilgan komponentlarni boshqa axborot xavfsizligi loyihalarida qayta foydalanish ham mumkin.

Modulli arxitekturaning tanlanish sabablari. Dasturiy ta’minot ishlab chiqishda barcha mantig‘ini bitta katta faylga joylash – eng keng tarqalgan amaliy xato hisoblanadi. Bunday yondashuv qisqa muddatda tezlik beradi, ammo loyiha kattalashishi bilan kodni qo‘llab-quvvatlash deyarli imkonsiz holatga keladi. Modulli arxitektura esa har bir vazifa uchun alohida fayl yaratish orqali kodni o‘qishni osonlashtiradi, xatoliklarni topish vaqtini qisqartiradi va birgalikda ishlash imkonini beradi. Ushbu loyihada to‘rtta asosiy modul ajratildi: anomaliya aniqlash mantig‘i, paket ushlash mexanizmi, avtomatik bloklash funksiyasi va asosiy boshqaruv moduli. Har bir modul aniq vazifaga ega bo‘lib, boshqalardan mustaqil ishlashi mumkin. Modullar o‘rtasidagi o‘zaro bog‘liqliklar va ma’lumot oqimida keltirilgan(2.5-rasm).



1.1-rasm. Python tizimining modulli arxitekturasi

Anomaliya aniqlash moduli – tizimning intellektual qismi. Anomaliya aniqlash moduli tizimning markaziy elementi hisoblanadi va uchta hujum turini – SYN Flood, UDP Flood va HTTP Flood – aniqlash algoritmlarini o‘z ichiga oladi. Modulning asosiy ma’lumot tuzilmasi sifatida har bir manba IP-manzil uchun alohida vaqt tamg‘alari navbati saqlanadi. Bu navbat qisqa vaqt oralig‘idagi paketlar oqimini tahlil qilish imkonini beradi va yangi paket kelganda eski yozuvlarni avtomatik o‘chiradi. Bunday tuzilma xotirada ixcham bo‘lib, juda tez ishlaydi – millionlab paketlarni qayta ishlashda ham sezilarli sekinlashuv kuzatilmaydi.

Modul tarkibida belgilangan asosiy konfiguratsiya parametrlari quyidagilardan iborat: SYN paketlar uchun chegara qiymati 10 soniyada 100 paket; UDP paketlar uchun chegara 10 soniyada 200 paket; HTTP so‘rovlar uchun chegara 10 soniyada 50 so‘rov; sliding window kattaligi 10 soniya. Bu chegara

qiymatlari tajribalar asosida tanlanib, qonuniy foydalanuvchi xulq-atvori va bot trafigi orasidagi farqni aniq ajratib koʻrsatadi. Oddiy foydalanuvchi sekundiga 1–5 ta HTTP soʻrov yuborsa, bot tizimlari sekundiga oʻnlab yoki yuzlab soʻrov generatsiya qiladi – bu ikki tartib kattaligidagi farqni tashkil etadi.

Sliding window algoritmi – vaqt asosidagi tahlil. Sliding window – anomaliya aniqlash sohasida eng keng qoʻllaniladigan algoritmlardan biri boʻlib, vaqt boʻyicha oʻzgaruvchi voqealar oqimini tahlil qilish uchun moʻljallangan. Algoritmning ishlash mantigʻi quyidagicha: har bir yangi paket kelganda uning kelish vaqti navbatga qoʻshiladi; navbatning boshidan belgilangan oyna kattaligidan (10 soniya) eski barcha yozuvlar avtomatik oʻchiriladi; navbatda qolgan elementlar soni soʻnggi 10 soniya ichidagi paketlar tezligini koʻrsatadi.

Ushbu yondashuv an’anaviy oʻrtacha tezlik hisoblashdan jiddiy ustun, chunki u trafikning aniq oʻzgarishlariga sezgir reaksiya beradi. An’anaviy oʻrtacha hisoblashda butun ishlash davri boʻyicha hisoblangan oʻrtacha qiymat qisqa muddatli pikni “boʻgʻib qoʻyadi” – ya’ni 5 soniyalik kuchli hujum bir soatlik oʻrtachada deyarli sezilmaydi. Sliding window esa aynan soʻnggi 10 soniyadagi holatga e’tibor qaratadi va keskin oʻsishlarni darhol aniqlaydi. Bundan tashqari, bu algoritm xotira sarfini tabiiy ravishda chegaralaydi – eski ma’lumotlar avtomatik oʻchirilgani sababli xotira toʻlib ketmaydi.

Shannon entropiyasi orqali IP xilma-xilligini baholash. Bot tarmoqlari (botnet) yordamida amalga oshirilgan DDoS hujumlarining muhim belgilaridan biri – manba IP-manzillarning gʻayrioddiy taqsimoti. Oddiy holatda veb-server trafigi koʻplab turli foydalanuvchilardan keladi va manba manzillar yuqori xilma-xillikka ega boʻladi. Hujum vaqtida esa trafik bir necha oʻnlab yoki yuzlab bot manzillaridan kelishi mumkin va xilma-xillik keskin pasayadi. Bu holatni miqdoriy baholash uchun ma’lumot nazariyasi sohasidan olingan Shannon entropiyasi tushunchasi qoʻllaniladi.

Shannon entropiyasi matematik formula asosida hisoblanadi va manba IP-manzillarning umumiy trafikdagi taqsimotini miqdoriy baholaydi. Hisoblash uchun har bir noyob IP-manzilning umumiy trafikdagi ulushi aniqlanadi va ushbu

ulushlar asosida entropiya qiymati shakllantiriladi. Entropiya qiymati nol bitdan sakkiz bitgacha o‘zgaradi: yuqori qiymat trafik ko‘plab turli manbalardan kelayotganini, past qiymat esa trafik kam sonli manbalardan kelayotganini bildiradi. Tajribalar asosida 1.5 bit entropiya chegara qiymati sifatida belgilandi – undan past qiymatlar amplifikatsiya hujumi yoki kichik botnet faoliyatining ko‘rsatkichi sifatida qabul qilinadi.

SYN Flood aniqlash mexanizmi. TCP protokolida ulanish o‘rnatish uchun uch bosqichli qo‘l siqish jarayoni qo‘llaniladi: mijoz SYN paketi yuboradi, server SYN-ACK bilan javob beradi, mijoz ACK paketi bilan ulanishni yakunlaydi. SYN Flood hujumida hujumchi minglab SYN paketi yuboradi, ammo ACK javobini hech qachon yubormaydi – natijada server xotirasi yarim ochiq ulanishlar bilan to‘lib ketadi. Bu hujumni aniqlash uchun har bir SYN paketi kelganda manba IP-manzili sliding window orqali tekshiriladi. Agar so‘nggi 10 soniya ichida bir xil IP-manzildan kelgan SYN paketlar soni belgilangan chegarani oshsa, tizim ogohlantirish chiqaradi va bloklash jarayonini ishga tushiradi.

Bundan tashqari, modul SYN va ACK paketlari soni o‘rtasidagi nisbatni ham doimiy ravishda kuzatib boradi. Normal trafikda bu nisbat 1 ga yaqin bo‘ladi, chunki har bir SYN paketiga ACK javob keladi. SYN Flood hujumi paytida esa bu nisbat 10 yoki undan yuqori qiymatlarga ko‘tariladi. Ushbu metrika protokol darajasidagi qo‘shimcha aniqlash mezoni sifatida xizmat qiladi va false positive holatlarini kamaytirishga yordam beradi.

UDP Flood aniqlash xususiyatlari. UDP protokoli ulanishsiz protokol bo‘lganligi sababli, undagi anomaliyalarni aniqlash faqat paket tezligi mezoniga asoslanadi. Hujumchi UDP paketlarni juda yuqori tezlikda yuborganda, har biriga server “port unreachable” xabari bilan javob qaytarishga harakat qiladi va o‘z resurslarini tez sarflaydi. UDP Flood uchun chegara qiymati SYN Flood ga nisbatan ikki barobar yuqori belgilangan, chunki UDP paketlar tabiatan tezroq yuboriladi va qonuniy UDP trafigi (DNS so‘rovlari, video oqimlari) ham yuqori paket tezligiga ega bo‘lishi mumkin.

HTTP Flood aniqlashning o‘ziga xos qiyinchiliklari. HTTP Flood hujumini aniqlash boshqa ikkita hujum turidan jiddiy farq qiladi va bir qator qo‘shimcha qiyinchiliklarga ega. Birinchidan, HTTP so‘rovlari tashqi ko‘rinishi jihatidan qonuniy brauzer trafigiga juda o‘xshaydi – har bir so‘rov to‘g‘ri shakllantirilgan va standart sarlavhalarga ega bo‘lishi mumkin. Ikkinchidan, paket darajasidagi tahlil yetarli emas, chunki har bir HTTP so‘rov bir nechta TCP paketdan iborat bo‘lishi mumkin. Uchinchidan, qonuniy foydalanuvchilar ham vaqti-vaqti bilan tez so‘rov yuborishlari mumkin (sahifani yangilash, AJAX so‘rovlari).

Bu qiyinchiliklarni hal qilish uchun HTTP Flood aniqlash Nginx serverining access.log faylini real vaqtda kuzatish orqali amalga oshiriladi. Tizim Linux dunyosidagi mashhur “tail -f” yondashuvini Python da takrorlaydi: fayl oxiriga o‘tib, yangi qatorlar paydo bo‘lishini kutadi. Har yangi yozuvda manba IP-manzili ajratib olinadi va sliding window orqali tekshiriladi. Bunday yondashuv Nginx ning ichki buferlash mexanizmlaridan foydalanadi va tizimga hech qanday qo‘shimcha yuk tushirmaydi.

Avtomatik bloklash moduli – iptables bilan integratsiya. Avtomatik bloklash moduli aniqlangan zararli IP-manzillarni Linux iptables xavfsizlik devori orqali bloklash vazifasini bajaradi. Modulning asosiy ishlash printsipi shundaki, Python kodi `subprocess` mexanizmi orqali tizim buyruqlarini chaqiradi va iptables ga yangi qoida qo‘shadi. Qoida quyidagi formada yaratiladi: belgilangan manba IP-manzilidan kelayotgan barcha paketlarni javobsiz yo‘q qilish (DROP). DROP maqsadi REJECT dan farqli ravishda hujumchiga hech qanday javob yubormaslik sababli ma’qulroq – REJECT holida server “port mavjud emas” xabari yuboradi va bu hujumchiga server hali ishlayotganini bildiradi.

Bloklash muvaffaqiyatli amalga oshirilgandan so‘ng IP-manzil ikkita joyga yoziladi. Birinchisi – `blocked_ips.txt` matn fayli bo‘lib, bloklangan manzillarning doimiy ro‘yxatini saqlaydi. Bu fayl tizim qayta ishga tushirilganda avval bloklangan manzillarni qayta yuklash uchun ishlatiladi. Ikkinchisi –

`/tmp/ddos_blocks.log` jurnal fayli bo‘lib, har bir bloklash hodisasini vaqt tamg‘asi bilan qayd etadi va keyingi tahlillar uchun manba bo‘lib xizmat qiladi.

Paket ushlash moduli – Scapy bilan past darajadagi tarmoq tahlili. Paket ushlash moduli Scapy kutubxonasining `sniff` funksiyasidan foydalanib, tarmoq interfeysidan o‘tayotgan barcha paketlarni real vaqtda ushlaydi. Har bir paket uch bosqichli tahlildan o‘tkaziladi: avval paketda IP qatlamining mavjudligi tekshiriladi; keyin TCP yoki UDP qatlami aniqlanadi; nihoyat, TCP paket bo‘lsa flags maydoni o‘qiladi va paket SYN, ACK yoki SYN-ACK ekanligi aniqlanadi. SYN paket bo‘lsa SYN Flood tekshirish jarayoni ishga tushadi, UDP paket bo‘lsa UDP Flood tekshirish boshlanadi, ACK paket bo‘lsa nisbat hisoblagichi yangilanadi.

Modulning muhim optimallashtirilgan jihati – ushlangan paketlar xotirada saqlanmaydi. Scapy ning standart sozlamasi har bir ushlangan paketni xotirada saqlash bo‘lib, bu uzoq muddatli ishlashda xotira sarfining tez o‘sishiga olib keladi. Tizimda esa `store=False` parametri o‘rnatilib, paketlar darhol qayta ishlanib tashlanadi. Bu yondashuv tizimning bir necha kun davomida uzluksiz ishlashini ta’minlaydi.

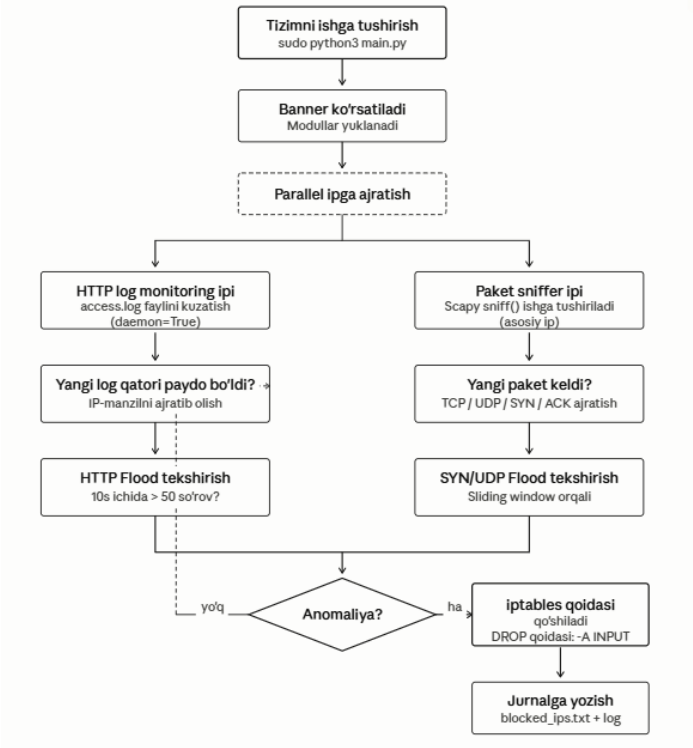
Asosiy boshqaruv moduli va parallel jarayonlar. Asosiy boshqaruv moduli – `main.py` – tizimning yuqori darajadagi orkestratori vazifasini bajaradi. U barcha modullarni birlashtiradi, tizimning bosh bannerini chiqaradi va ikki parallel jarayonni ishga tushiradi. Birinchi jarayon – Nginx access.log faylining real vaqt monitoringi – alohida ipda (thread) ishga tushiriladi va doimiy ravishda yangi log yozuvlarini kuzatadi. Ikkinchi jarayon – Scapy paket sniffer – asosiy ipda ishlab, tarmoq darajasidagi paketlarni qayta ishlaydi.

Ikki jarayonning bir vaqtda ishlashi multithreading texnologiyasi orqali ta’minlanadi. Har ikki jarayon umumiy anomaliya aniqlash va bloklash ob’yektlaridan foydalanadi, bu esa barcha aniqlangan tahdidlarning yagona logging va bloklash tizimi orqali qayta ishlanishini kafolatlaydi. HTTP monitoring ipi `daemon=True` parametri bilan yaratiladi – bu asosiy dastur to‘xtatilganda HTTP

monitoring ham avtomatik to‘xtashini ta’minlaydi va orqada qolgan (“zombie”) jarayonlarning paydo bo‘lishining oldini oladi.

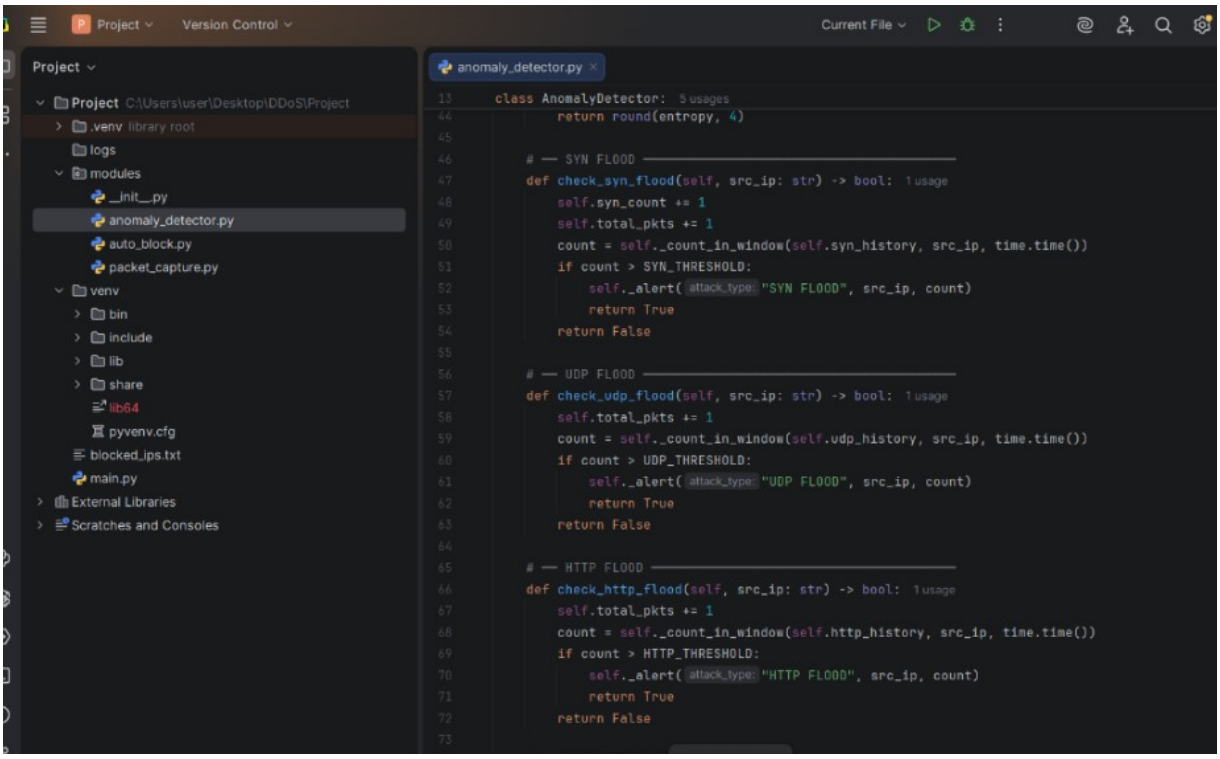
Modullar o‘rtasidagi ma’lumot oqimi. Tizim arxitekturasi kompozitsiya naqshiga (composition pattern) asoslangan bo‘lib, har bir yuqori darajadagi komponent quyi darajadagi komponentlarni o‘z ichiga oladi. Paket ushlash moduli o‘z navbatida anomaliya aniqlash va bloklash modullarini o‘z ichiga oladi. Anomaliya aniqlanganda boshqaruv jarayonida quyidagi ketma-ket harakatlar amalga oshiriladi: anomaliya aniqlash moduli ogohlantirish chiqaradi va bloklangan IP-manzillar to‘plamiga yangi manzilni qo‘shadi; paket ushlash moduli bu signalni qabul qilib, bloklash moduliga IP-manzilni uzatadi; bloklash moduli iptables qoidasini shakllantiradi va tegishli fayllarga yozuv qo‘shadi. Bunday bir yo‘nalishli ma’lumot oqimi tizim murakkabligini kamaytiradi va xatoliklarni topish jarayonini sezilarli darajada osonlashtiradi.

Tizim ishga tushishidan so‘nggi xulq-atvor. Tizim to‘g‘ri sozlangan sharoitda ishga tushganda terminalda bosh banner chiqadi, “HTTP log kuzatilmoqda” va “Paketlar ushlanmoqda” xabarlari paydo bo‘ladi. Shu lahzadan boshlab tizim to‘liq monitoring rejimida ishlaydi. Har bir kiruvchi paket Scapy snifferi orqali qayta ishlanadi, har bir yangi log yozuvi alohida ipda tahlil qilinadi. Hujum aniqlanganda terminalda darhol ogohlantirish xabari ko‘rinadi, IP-manzil iptables orqali bloklanadi va hujum trafigi to‘liq to‘xtaydi. Tizim ishga tushirilgandan to‘xtatilgunga qadar bo‘lgan jarayonning umumiy ish sxemasida keltirilgan(2.6-rasm).



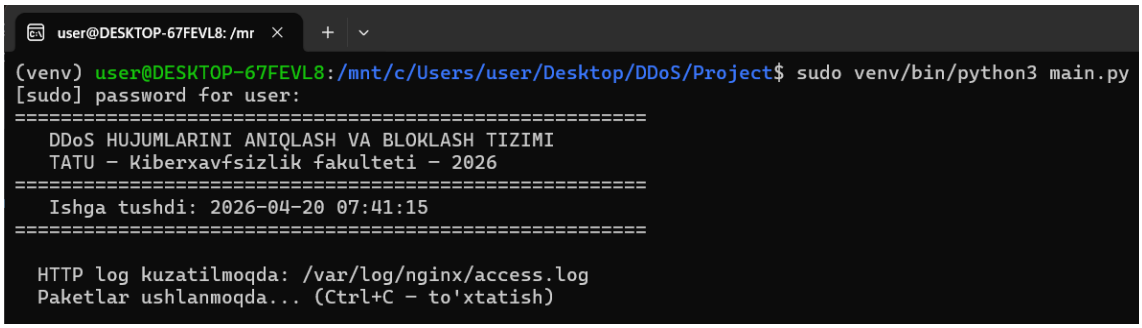
1.2-rasm. Python tizimining ish jarayoni sxemasi

Modullarning kodi PyCharm IDE da yozilib, hujjatlash izohlari bilan to‘liq ta’minlandi. Har bir funksiya va sinf maqsadi, kirish parametrlari va qaytariluvchi qiymati haqida qisqa izoh bilan birga keladi. Bunday yondashuv kelajakda kodni qo‘llab-quvvatlash, boshqa dasturchilarga loyihani tushuntirish va tadqiqot natijalarini hujjatlashda qulaylik tug‘diradi. Asosiy modullarning kodi PyCharm muharririda (2.7-rasm) da keltirilgan.



2.7-rasm. anomaly_detector.py moduli kodi PyCharm muharririda

Yaratilgan tizim sinov uchun ishga tushirilgandan so‘ng terminal interfeysida o‘zining boshlang‘ich holatini ko‘rsatadi: tizim nomi, ishga tushgan vaqti va monitoring rejimida ekanligi haqida xabar beradi. Ushbu boshlang‘ich holat (2.8-rasm) da namoyish etilgan.



2.8-rasm. Python tizimining ishga tushirilgan holati

Ushbu bo‘limda ishlab chiqilgan to‘rt modul birgalikda to‘liq funksional, kengaytiriluvchi va sinov uchun tayyor DDoS aniqlash tizimini tashkil etadi. Tizim Scapy ning past darajadagi tarmoq tahlil imkoniyatlarini, sliding window algoritmining tezligini, Shannon entropiyasining matematik aniqligini va iptables ning ishonchli bloklash mexanizmini birlashtirib, professional darajadagi himoya yechimini namoyish etadi. Tizim samaradorligining miqdoriy baholanishi, real

hujum senariylari asosidagi sinov natijalari va aniqlash aniqligi ko‘rsatkichlari keyingi bo‘limda batafsil tahlil qilinadi.

XULOSA

Ushbu maqolada Python dasturlash tili asosida real vaqtda tarmoq trafigi anomaliyalarini aniqlash tizimining yaratilishi har tomonlama o‘rganildi. Tadqiqot natijalaridan kelib chiqqan holda quyidagi xulosalarga kelish mumkin:

Birinchidan, modular arxitektura yondashuvining qo‘llanilishi tizimning kengaytiriluvchanligi, sinovdan o‘tkaziluvchanligi va kodni qayta ishlatish imkoniyatini sezilarli darajada oshirdi. To‘rtta mustaqil modul – anomaliya aniqlash, paket ushlash, avtomatik bloklash va boshqaruv moduli – bir-biridan mustaqil ravishda ishlab chiqildi va sinovdan o‘tkazildi.

Ikkinchidan, sliding window algoritmi va Shannon entropiyasi kabi statistik usullarning birgalikda qo‘llanilishi turli xil DDoS hujum turlarini samarali aniqlash imkonini berdi. Sliding window algoritmi qisqa vaqt oralig‘idagi trafik o‘zgarishlariga tezkor reaksiya ko‘rsatadi, Shannon entropiyasi esa manba IP-manzillarning xilma-xilligini baholash orqali botnet faoliyatini aniqlashga yordam beradi.

Uchinchidan, Scapy kutubxonasining past darajadagi tarmoq tahlil imkoniyatlari bilan iptables xavfsizlik devorining avtomatik bloklash mexanizmining birlashtirilishi tizimga nafaqat tahdidlarni aniqlash, balki ularga real vaqtda javob qaytarish qobiliyatini ham berdi. store=False parametrining qo‘llanilishi esa tizimning uzoq muddatli barqaror ishlashini ta’minladi.

To‘rtinchidan, HTTP Flood hujumini aniqlashda Nginx access.log faylini real vaqtda kuzatish yondashuvining qo‘llanilishi paket darajasidagi tahlilning cheklovlarini bartaraf etdi va qonuniy brauzer trafigidan farqli bot trafigini samarali ajratish imkonini berdi.

Umuman olganda, yaratilgan tizim professional darajadagi DDoS himoya yechimi sifatida o‘zini namoyon etdi. Kelajakda tizimni yanada takomillashtirish uchun sun’iy intellekt va mashinaviy o‘rganish algoritmlarini qo‘llash, hujum

turlarini avtomatik klassifikatsiya qilish va taqsimlangan tarmoq muhitlarida ishlash imkoniyatlarini kengaytirish maqsadga muvofiqdir.

FOYDALANILGAN ADABIYOTLAR

1. Stallings, W. (2017). Network Security Essentials: Applications and Standards. 6th Edition. Pearson.
2. Sanders, C. (2017). Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. 3rd Edition. No Starch Press.
3. Bhuyan, M.H., Bhattacharyya, D.K., & Kalita, J.K. (2014). Network Anomaly Detection: Methods, Systems and Tools. IEEE Communications Surveys & Tutorials, 16(1), 303–336.
4. Mirkovic, J., & Reiher, P. (2004). A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39–53.
5. Zargar, S.T., Joshi, J., & Tipper, D. (2013). A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046–2069.
6. Shannon, C.E. (1948). A Mathematical Theory of Communication. The Bell System Technical Journal, 27(3), 379–423.
7. Lakhina, A., Crovella, M., & Diot, C. (2005). Mining Anomalies Using Traffic Feature Distributions. ACM SIGCOMM Computer Communication Review, 35(4), 217–228.
8. Scapy Project. (2024). Scapy: Packet crafting for Python2/3. https://scapy.net/
9. Noonan, W., & Dubrawsky, I. (2006). Firewall Fundamentals. Cisco Press.
10. Bejtlich, R. (2013). The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press.
11. Wang, H., Zhang, D., & Shin, K.G. (2002). Detecting SYN Flooding Attacks. Proceedings of IEEE INFOCOM, Vol. 3, pp. 1530–1539.

12. Carl, G., Kesidis, G., Brooks, R.R., & Rai, S. (2006). Denial-of-Service Attack-Detection Techniques. IEEE Internet Computing, 10(1), 82–89.
13. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), Article 15.
14. O‘zbekiston Respublikasi “Axborot texnologiyalari va kommunikatsiyalarini rivojlantirish to‘g‘risida”gi Qonuni. (2003). https://lex.uz/
15. Kurose, J.F., & Ross, K.W. (2021). Computer Networking: A Top-Down Approach. 8th Edition. Pearson.