Mundarija

[**Kirish 3**](#_Toc200692764)

[**I.bob. Axborot tizimlaridagi zaifliklar va ularni aniqlash usullari 5**](#_Toc200692765)

[**1.1. Axborot tizimlari, ularning zaifliklari va turlari 5**](#_Toc200692766)

[**1.2. Nmap vositasi va uning asosiy parametrlari. Tarmoq zaifliklari 6**](#_Toc200692767)

[**1.3. SQLmap vositasi va uning asosiy parametrlari. Veb-ilovalarning zaifliklari 8**](#_Toc200692768)

[**1.4. Nikto vositasi va uning asosiy parametrlari. Veb-serverlarning zaifliklari 9**](#_Toc200692769)

[**I bob bo‘yicha xulosa 10**](#_Toc200692770)

[**II.bob. Python asosida axborot tizimlaridagi zaifliklarni aniqlovchi avtomatlashtirilgan vosita yaratish va sinovdan o‘tkazish 12**](#_Toc200692771)

[**2.1. Ishchi muhitni tayyorlash va zaruriy vositalarni o’rnatish 12**](#_Toc200692772)

[**2.2. Dastur funksiyalarining umumiy tavsifi. Botni sinovdan o‘tkazish 13**](#_Toc200692773)

[**II bob bo‘yicha xulosa 16**](#_Toc200692774)

[**Xulosa 18**](#_Toc200692775)

[**Foydalanilgan adabiyotlar 19**](#_Toc200692776)

# Kirish

Bugungi kunda raqamli texnologiyalarning jadal rivojlanishi hayotimizning barcha sohalariga o‘z ta’sirini ko‘rsatmoqda. Ammo bu taraqqiyot bilan bir qatorda axborot tizimlariga tahdid soluvchi xavfsizlik muammolari ham keskin ortib bormoqda. Internetga ulangan har bir qurilma – mobil telefon, server, veb-ilova yoki oddiy foydalanuvchi kompyuteri – potensial zaifliklarga ega bo‘lishi mumkin. Ushbu zaifliklardan esa kiberjinoyatchilar foydalanib, turli hujumlarni amalga oshiradilar: maxfiy ma’lumotlarni o‘g‘irlash, tizim faoliyatini izdan chiqarish, zararli dasturlarni joylashtirish kabi holatlar bugungi kunda dolzarb muammoga aylangan.

Shu sababli, axborot xavfsizligini ta’minlashda tizimlarda mavjud bo‘lgan zaifliklarni erta aniqlash, ularni avtomatik tarzda tahlil qilish va ularga qarshi samarali choralar ko‘rish eng muhim vazifalardan biri sanaladi. Ayniqsa, bu jarayonlarni avtomatlashtirish – ya’ni tizimni doimiy ravishda nazorat qilib boruvchi, inson aralashuvisiz ishlay oladigan dasturiy vositalarni yaratish – zamonaviy axborot xavfsizligining ajralmas qismiga aylangan.

Mazkur malakaviy amaliyot ishida aynan shunday muammolarga yechim bo‘ladigan avtomatlashtirilgan vosita ishlab chiqishga e’tibor qaratilgan. Ya’ni, axborot tizimlaridagi zaifliklarni aniqlash uchun keng qo‘llaniladigan va samaradorligi amaliyotda isbotlangan Nmap, SQLmap va Nikto dasturlari asosida Python dasturlash tilida ishlovchi skanerlovchi bot yaratiladi. Ushbu bot belgilangan IP manzillar, domenlar yoki tarmoq diapazonlarini avtomatik tarzda skanerlab, aniqlangan zaifliklar haqida foydalanuvchiga aniq va tushunarli ma’lumotlar taqdim etadi. Shuningdek, u xavfsizlikni mustahkamlash bo‘yicha tavsiyalar ham beradi.

**Malakaviy amaliyot ishining maqsadi:** O‘rgangan nazariy bilimlarimizni amaliyotda mustahkamlash, axborot xavfsizligiga doir real muammolarga texnologik va avtomatlashtirilgan yechimlar topish. Xususan, Nmap, SQLmap va Nikto vositalaridan foydalangan holda Python dasturlash tilida zaifliklarni aniqlovchi skanerlovchi bot ishlab chiqish asosiy vazifa sifatida belgilangan.

**Amaliyot davomida bajarilishi kerak bo‘lgan asosiy vazifalar:**

1. Axborot xavfsizligi sohasida mavjud zaifliklar va ularni aniqlash usullarini nazariy jihatdan o‘rganish.

2. Nmap, SQLmap va Nikto vositalarining ishlashini tahlil qilish hamda ularning real tizimlarda qanday qo‘llanilishini o‘rganish.

3. Python dasturlash tilida avtomatlashtirilgan skanerlovchi bot yaratish:

* Nmap orqali portlar va xizmatlarni avtomatik aniqlash;
* SQLmap orqali veb-ilovalardagi SQL injeksiya zaifliklarini tahlil qilish;
* Nikto orqali veb-serverlarda mavjud xavfsizlik zaifliklarini aniqlash.

4. Aniqlangan zaifliklar haqida natijalarni qulay formatda (matnli fayl yoki JSON) chiqarish va xavfsizlikni yaxshilash bo‘yicha tavsiyalarni biriktirish.

5. Yakuniy xulosa va takliflarni ishlab chiqish, ya’ni yaratilgan vositaning afzalliklari, cheklovlari va uni takomillashtirish yo‘llarini tahlil qilish.

Ushbu malakaviy ish zamonaviy kiberxavfsizlik sohasida amaliyotga yo‘naltirilgan muhim qadam bo‘lib, biz kabi yosh mutaxassislar uchun chuqur bilim va tajriba manbai bo‘lib xizmat qiladi. Avtomatlashtirishga asoslangan yondashuv esa nafaqat ish unumdorligini oshiradi, balki zamonaviy axborot tizimlarining xavfsizligini ta’minlashda muhim rol o‘ynaydi.

# I.bob. Axborot tizimlaridagi zaifliklar va ularni aniqlash usullari

## 1.1. Axborot tizimlari, ularning zaifliklari va turlari

Axborot tizimlari – bu ma’lumotlarni yig‘ish, saqlash, qayta ishlash va uzatish orqali tashkilot yoki foydalanuvchining ehtiyojlarini qondirishga xizmat qiluvchi texnologik va insoniy resurslar yig‘indisidir. Bunday tizimlar har bir tashkilotning kundalik faoliyatida muhim rol o‘ynaydi. Axborot tizimlariga quyidagilar kiradi:

* Kompyuter va server qurilmalari;
* Tarmoq infratuzilmalari (router, switch, firewall va hokazo);
* Dasturiy ta’minotlar (operatsion tizimlar, ilovalar, veb-xizmatlar);
* Foydalanuvchi interfeysi va ma’lumotlar bazalari;
* Axborotni uzatish va qabul qilish mexanizmlari.

Bugungi kunda deyarli barcha tashkilotlar o‘z faoliyatini axborot tizimlarisiz tasavvur qila olmaydi. Shu sababli, ushbu tizimlarning xavfsizligini ta’minlash dolzarb masalaga aylangan.

Axborot xavfsizligi – bu axborot tizimlarining ishonchliligi, maxfiyligi va uzluksizligini ta’minlovchi chora-tadbirlar majmuasidir. U uchta asosiy tamoyilga asoslanadi:

Maxfiylik (Confidentiality) – axborot faqat ruxsat etilgan shaxslar tomonidan ko‘rilishi kerak;

Butunlik (Integrity) – axborot o‘zgarmasligi va asl nusxasi bilan bir xil bo‘lishi kerak;

Mavjudlik (Availability) – axborot doimiy va kerakli vaqtda foydalanishga tayyor bo‘lishi zarur.

Shu bilan birga, har qanday axborot tizimi zaifliklarga ega bo‘lishi mumkin. Zaiflik (vulnerability) – bu tizimdagi xavfsizlik kamchiliklari yoki xatoliklar bo‘lib, ular orqali hujumchi yoki zararli dastur tizimga ruxsatsiz kirish, ma’lumotlarni o‘g‘irlash yoki tizim faoliyatini buzish imkoniyatiga ega bo‘ladi.

Axborot tizimlarida uchraydigan asosiy zaifliklar quyidagilardir:

* Dasturiy zaifliklar: noto‘g‘ri yozilgan kodlar, SQL injection, XSS (Cross-site scripting), bufer to‘lishi kabi xatoliklar;
* Tarmoq zaifliklari: ochiq portlar, noto‘g‘ri sozlangan marshrutizator yoki xavfsizlik devorlari (firewall);
* Foydalanuvchi darajasidagi zaifliklar: oddiy yoki qayta ishlatilayotgan parollar, ijtimoiy muhandislik qurbonlari;
* Yangi xatolar (Zero-day): hali ishlab chiqaruvchilar tomonidan aniqlanmagan va patch chiqarilmagan zaifliklar;
* Konfiguratsiya zaifliklari: noto‘g‘ri yoki xavfsizlik talablariga javob bermaydigan tizim sozlamalari.

Bu zaifliklar xavfsizlikka jiddiy tahdid solishi mumkin va ularni aniqlash, monitoring qilish va bartaraf etish axborot xavfsizligi bo‘yicha muhim vazifa sanaladi. Ushbu jarayonni amalga oshirish uchun bir nechta asosiy usullar mavjud:

Qo‘lda tekshirish (Manual Testing) – mutaxassislar tomonidan tizimlar qo‘lda ko‘rib chiqiladi. Bu usul aniq, lekin vaqt va resurs talab qiladi;

Avtomatlashtirilgan skanerlash (Automated Scanning) – dasturiy vositalar yordamida tizim avtomatik tarzda skanerdan o‘tkaziladi va zaifliklar aniqlanadi;

Penetratsion test (Penetration Testing) – tizimga hujum qilish orqali real zaifliklar aniqlanadi, bu amaliyot ehtiyotkorlik bilan amalga oshiriladi;

Audit va monitoring – tizimlar muntazam ravishda kuzatilib, loglar tahlil qilinadi.

Keyingi bo‘limlarda biz yuqoridagi zaifliklarni aniqlashda keng qo‘llaniladigan Nmap, SQLmap va Nikto vositalarining ishlash mexanizmlarini batafsil tahlil qilamiz.

## 1.2. Nmap vositasi va uning asosiy parametrlari. Tarmoq zaifliklari

Nmap (Network Mapper) – bu tarmoqni skanerlash va xavfsizlikni tahlil qilish uchun mo‘ljallangan ochiq manbali kuchli vositadir. U dastlab 1997-yilda Gordon Lyon (Fyodor) tomonidan yaratilgan bo‘lib, bugungi kunga kelib kiberxavfsizlik sohasida keng qo‘llaniladigan standart vositalardan biriga aylangan. Nmap yordamida foydalanuvchi ma’lum IP manzillar, tarmoq diapazonlari yoki domenlarni tahlil qilib, ularga ulangan qurilmalarni, ochiq portlarni, xizmatlarni va potentsial zaifliklarni aniqlashi mumkin.

Nmap vositasi odatda quyidagi maqsadlarda ishlatiladi:

* Tarmoqdagi faol qurilmalarni aniqlash;
* Ochiq portlar va ularning xizmatlarini aniqlash;
* Operatsion tizim va versiyalarni taxmin qilish (OS Detection);
* Xostlarning mavjudligini tekshirish;
* Xavfsizlik zaifliklarini aniqlash va ekspluatatsiya qilish ehtimolini baholash.

Nmap o‘zi to‘g‘ridan-to‘g‘ri ekspluatatsiya qilmaydi, biroq u quyidagi potentsial zaifliklarni aniqlashga yordam beradi:

Ochiq portlar: Xavfsizlik jihatdan muhim bo‘lmagan xizmatlarning ochiq portlari, ular orqali kirish imkoni;

Versiyasi eskirgan xizmatlar: Eskirgan yoki zaifliklarga ega xizmat versiyalari (Apache, FTP, SSH, Telnet va hokazo);

Firewall va filtratsiya yo‘qligi: Tarmoq himoyasining yetishmasligi yoki noto‘g‘ri sozlanganligi;

Tarmoq tuzilmasining zaif joylari: Tarmoq topologiyasini ochiqlovchi ma’lumotlar orqali tarmoqdagi xavfli zonalarni aniqlash.

Nmap ning asosiy parametrlari:

-sS TCP SYN skan (tez va ancha kam zaifliklar aniqlanadi)

-sT TCP connect skan (standart, lekin sezilarli)

-sU UDP portlarni skanerlash

-p Maxsus port yoki portlar diapazonini skanerlash, masalan: -p 80,443

-O Operatsion tizimni aniqlash

-sV Xizmat versiyalarini aniqlash

-A Kengaytirilgan tahlil: OS, versiyalar, skriptlar

-Pn Ping orqali host mavjudligini tekshirmasdan to‘g‘ridan-to‘g‘ri skanerlash

-T4 Skan tezligini belgilash (T0 – juda sekin, T5 – juda tez)

-oN Natijalarni matn formatida saqlash

-oX XML formatda natijalarni eksport qilish

--script Nmap Scripting Engine (NSE) orqali zaifliklarni aniqlash ssenariylarini ishga tushurish.

Nmap vositasi axborot tizimlarining tashqi himoya darajasini tahlil qilishda muhim rol o‘ynaydi. U orqali ochiq portlar va xizmatlarni aniqlash, tarmoqdagi zaif qurilmalarni topish va umumiy xavfsizlikni baholash mumkin. Ayniqsa, dasturiy vositalarning versiyalarini aniqlash va ular orqali zaiflik ehtimolini baholash kiberxavfsizlikda dastlabki muhim qadamdir.

## 1.3. SQLmap vositasi va uning asosiy parametrlari. Veb-ilovalarning zaifliklari

SQLmap – bu avtomatlashtirilgan ochiq kodli vosita bo‘lib, u veb-ilovalarda mavjud bo‘lishi mumkin bo‘lgan SQL inyeksiya (SQL Injection) zaifliklarini aniqlash va ekspluatatsiya qilish uchun mo‘ljallangan. SQLmap foydalanuvchiga minimal harakat bilan ma’lumotlar bazasiga ruxsatsiz kirish, ma’lumotlarni ko‘rish, o‘zgartirish yoki yuklab olish imkonini beradi.

Ushbu vosita turli xil ma’lumotlar bazasi tizimlarini (MySQL, PostgreSQL, Oracle, Microsoft SQL Server, SQLite va boshqalar) aniqlash, foydalanuvchilar, parollar, jadvallar va ustunlar kabi tuzilmalarni chiqarib olish imkoniyatiga ega.

SQLmap asosan quyidagi SQL inyeksiya zaifliklarini aniqlaydi va ekspluatatsiya qiladi:

Boolean-based blind SQLi – mantiqiy shartlar orqali ma’lumot chiqarish;

Time-based blind SQLi – so‘rovning bajarilish vaqti asosida javobni aniqlash;

Error-based SQLi – serverning chiqarayotgan xatoliklari orqali ma’lumot olish;

Union-based SQLi – UNION operatori orqali jadval biriktirish va ma’lumot chiqarish;

Stacked queries – bir nechta SQL so‘rovlarini ketma-ket bajarish;

Out-of-band SQLi – tarmoq orqali boshqa kanal bilan javob qaytarish (kamdan-kam qo‘llaniladi).

SQLmap vositasining asosiy parametrlari:

-u Tekshirilayotgan URL manzilini ko‘rsatish

--cookie Sessiya uchun cookie ma’lumotlarini belgilash

--data POST so‘rovlaridagi ma’lumotlar bilan ishlash

--dbs Serverdagi barcha ma’lumotlar bazalarini chiqarish

--tables Tanlangan bazadagi barcha jadvallarni chiqarish

--columns Tanlangan jadvaldagi ustunlarni chiqarish

--dump Jadvaldagi real ma’lumotlarni chiqarish

--level Skan chuqurligini belgilash (1–5)

--risk So‘rov xavf darajasini belgilash (1–3)

--batch Foydalanuvchidan so‘rov qilmasdan default javoblar bilan ishlash

--threads Ko‘p oqimli (multithreading) skan qilish

--technique Aniq texnikalarni belgilash (B, E, U, S, T, Q).

SQLmap – bu veb-xavfsizlik tahlilchilari va penetratsion testchilar uchun muhim vosita bo‘lib, u SQL inyeksiya zaifliklarini aniqlash va ularni ekspluatatsiya qilishni avtomatlashtiradi. Ushbu vosita yordamida test qilinayotgan tizimdagi ma’lumotlar bazasiga qanday darajada ruxsatsiz kirish mumkinligini tahlil qilish mumkin. Shuning uchun SQLmap axborot tizimlarining xavfsizligini baholashda juda samarali vositadir.

## 1.4. Nikto vositasi va uning asosiy parametrlari. Veb-serverlarning zaifliklari

Nikto – bu ochiq manba kodli, tezkor va samarali veb-serverlarni skanerlash vositasi bo‘lib, u turli zaifliklarni aniqlash, noto‘g‘ri konfiguratsiyalarni ochib berish va xavfsizlikni baholash maqsadida foydalaniladi. U asosan veb-serverlar va ularning ilovalari (Apache, Nginx, IIS, va boshqalar) ustida ishlaydi.

Nikto vositasi har xil veb-server versiyalarini aniqlaydi, xavfsizlik xatolarini tekshiradi, xavfsiz bo‘lmagan fayllarni, skriptlarni va server konfiguratsiyalaridagi noto‘g‘ri sozlamalarni ko‘rsatadi.

Nikto orqali quyidagi xavfsizlik muammolarini aniqlash mumkin:

* Xavfsiz bo‘lmagan CGI skriptlar;
* Eski yoki yaroqsiz server versiyalari;
* Ma’lumotlar sizib chiqishiga olib keluvchi sozlamalar;
* Default (standart) sahifalar, test sahifalari, noto‘g‘ri konfiguratsiyalar;
* Keng tarqalgan zaifliklar (CVE asosida);
* HTTP Header sozlamalari (X-Frame-Options, X-XSS-Protection);
* Xavfsizlik patchlari o‘rnatilmagan serverlar.

Nikto vositasining asosiy parametrlari:

-h Maqsad server manzilini ko‘rsatish (IP yoki URL)

-port Port raqamini belgilash (default: 80)

-ssl HTTPS (SSL) orqali skanerlashni faollashtirish

-Display V To‘liq natijalarni ko‘rsatish (verbose)

-Tuning Maxsus zaiflik turlariga yo‘naltirilgan skan

-Format Hisobot formatini tanlash (txt, html, xml va h.k.)

-output Natijalarni faylga yozish

-update Nikto zaifliklar bazasini yangilash.

Nikto – veb-server xavfsizligini baholashda muhim vosita bo‘lib, u serverdagi eski versiyalar, noto‘g‘ri konfiguratsiyalar va xavfsizlik zaifliklarini tezda aniqlash imkonini beradi. U penetration testchilar va tizim administratorlari uchun dastlabki xavfsizlik tahlil vositasi sifatida keng qo‘llaniladi.

## I bob bo‘yicha xulosa

Mazkur bobda zamonaviy axborot tizimlarida uchrashi mumkin bo‘lgan asosiy zaifliklar, ularning turlari va ushbu zaifliklarni aniqlashda qo‘llaniladigan asosiy vositalar tahlil qilindi. Avvalo, axborot xavfsizligining asosiy tamoyillari – maxfiylik, butunlik va mavjudlik tushunchalari yoritildi hamda axborot tizimlaridagi xavf-xatarlar manbalari haqida fikr yuritildi.

Shundan so‘ng, axborot tizimlaridagi zaifliklarni aniqlashda keng qo‘llaniladigan bir nechta muhim vositalar o‘rganildi:

Nmap vositasi orqali tarmoqdagi qurilmalar va ularning portlari haqida ma’lumot olish, ochiq portlar orqali mumkin bo‘lgan zaifliklarni aniqlash usullari ko‘rib chiqildi.

SQLmap vositasi yordamida veb-ilovalarda uchraydigan SQL inyeksiya zaifliklarini avtomatik tarzda aniqlash imkoniyatlari tahlil qilindi.

Nikto vositasi orqali esa veb-serverlarda mavjud bo‘lishi mumkin bo‘lgan xavfli fayllar, noto‘g‘ri sozlamalar va eski versiyalarni aniqlash imkoniyati ko‘rsatib o‘tildi.

Ushbu vositalar axborot tizimlarining zaif tomonlarini aniqlash, ularni hujumlardan oldin bartaraf etish, tizim xavfsizligini mustahkamlashda muhim ahamiyatga ega. Ularning o‘ziga xos funksional imkoniyatlari, afzalliklari va qo‘llash doiralari tajriba va amaliyot asosida o‘rganilishi lozim. Keyingi bobda skanerlovchi bot yordamida ushbu vositalardan foydalangan holda zaifliklarni aniqlashga harakat qilamiz.

Shunday qilib, I bobda axborot tizimlarining zaifliklarini aniqlashning nazariy va amaliy jihatlari yoritilib, kelgusi boblarda ularni chuqur tahlil qilishga zamin yaratildi.

# II.bob. Python asosida axborot tizimlaridagi zaifliklarni aniqlovchi avtomatlashtirilgan vosita yaratish va sinovdan o‘tkazish

## 2.1. Ishchi muhitni tayyorlash va zaruriy vositalarni o’rnatish

Axborot tizimlaridagi zaifliklarni avtomatik aniqlovchi botni yaratish jarayoni samarali kechishi uchun avvalo to‘liq izolyatsiyalangan va nazorat qilinadigan ishchi muhit tashkil etilishi muhim hisoblanadi. Bunday yondashuv tizim xavfsizligini saqlash, xatoliklar ta'sirini kamaytirish va tajriba uchun qulay sharoit yaratishga xizmat qiladi. Shu maqsadda loyiha uchun VirtualBox yordamida virtual mashina ishga tushirildi va unga Ubuntu operatsion tizimi o‘rnatildi.

Ubuntu operatsion tizimi ochiq manbali ekanligi, soddaligi va ko‘plab xavfsizlik vositalarini qo‘llab-quvvatlashi bilan ajralib turadi. Ushbu tizim botni sinovdan o‘tkazish va xavfsizlik vositalaridan samarali foydalanish uchun optimal yechim bo‘ldi.

Virtual muhitni tayyorlash bosqichlari:

1. VirtualBox o‘rnatish va sozlash: VirtualBox rasmiy saytidan (https://www.virtualbox.org/wiki/Downloads) dastur yuklab olindi va o’rnatildi. VirtualBox yordamida yangi virtual mashina yaratildi. Operatsion tizim sifatida Ubuntu 22.04 versiyasi tanlandi. CPU va RAM resurslari bot ish faoliyatiga mos tarzda ajratildi (10GB RAM, 8 CPU, tavsiya etiladi: kamida 2 GB RAM, 2 CPU).
2. Ubuntu o‘rnatilishi: Ubuntu ISO fayli rasmiy saytdan (https://ubuntu.com/download/desktop) yuklab olindi. Virtual mashinaga Ubuntu operatsion tizimi o‘rnatilib, zaruriy yangilanishlar amalga oshirildi:

sudo apt update && sudo apt upgrade -y

1. Git o‘rnatish va loyiha klonlanishi: Git vositasi orqali tayyorlangan bot kodlari GitHub’dan yuklab olindi:

sudo apt install git -y

git clone https://github.com/samandar8/IPScanerbot

1. Python va virtual muhit (venv) o‘rnatilishi: Loyihada foydalaniladigan Python kutubxonalarini izolyatsiyalash uchun virtual muhit tashkil etildi:

sudo apt install python3 python3-venv python3-pip -y

cd IPScanerbot

python3 -m venv venv

source venv/bin/activate

1. Zaruriy vositalarni o‘rnatish: Nmap – tarmoqlarni skanerlash va ochiq portlarni aniqlash uchun. SQLmap – SQL inyeksiya zaifliklarini aniqlash va ekspluatatsiya qilish uchun. Nikto – Veb-serverlarda mavjud xavfsizlik zaifliklarini aniqlash uchun:

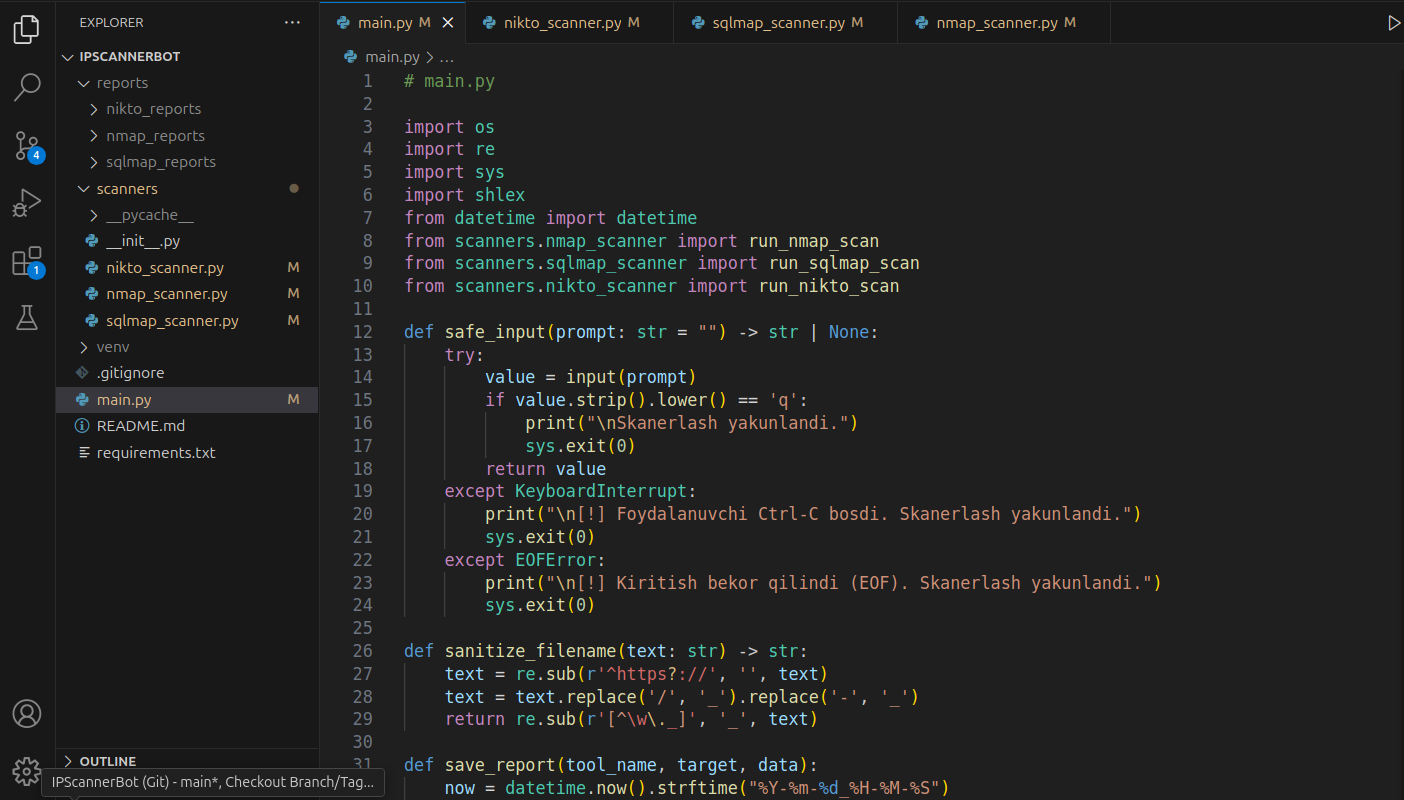
sudo apt install nmap sqlmap nikto

Yuqoridagi bosqichlar orqali xavfsiz va nazorat ostidagi virtual muhit tayyorlandi. Bu muhit botni ishlab chiqish, sinovdan o‘tkazish hamda turli zaifliklarni tahlil qilish uchun zarur bo‘lgan barcha vositalar bilan ta’minlandi.

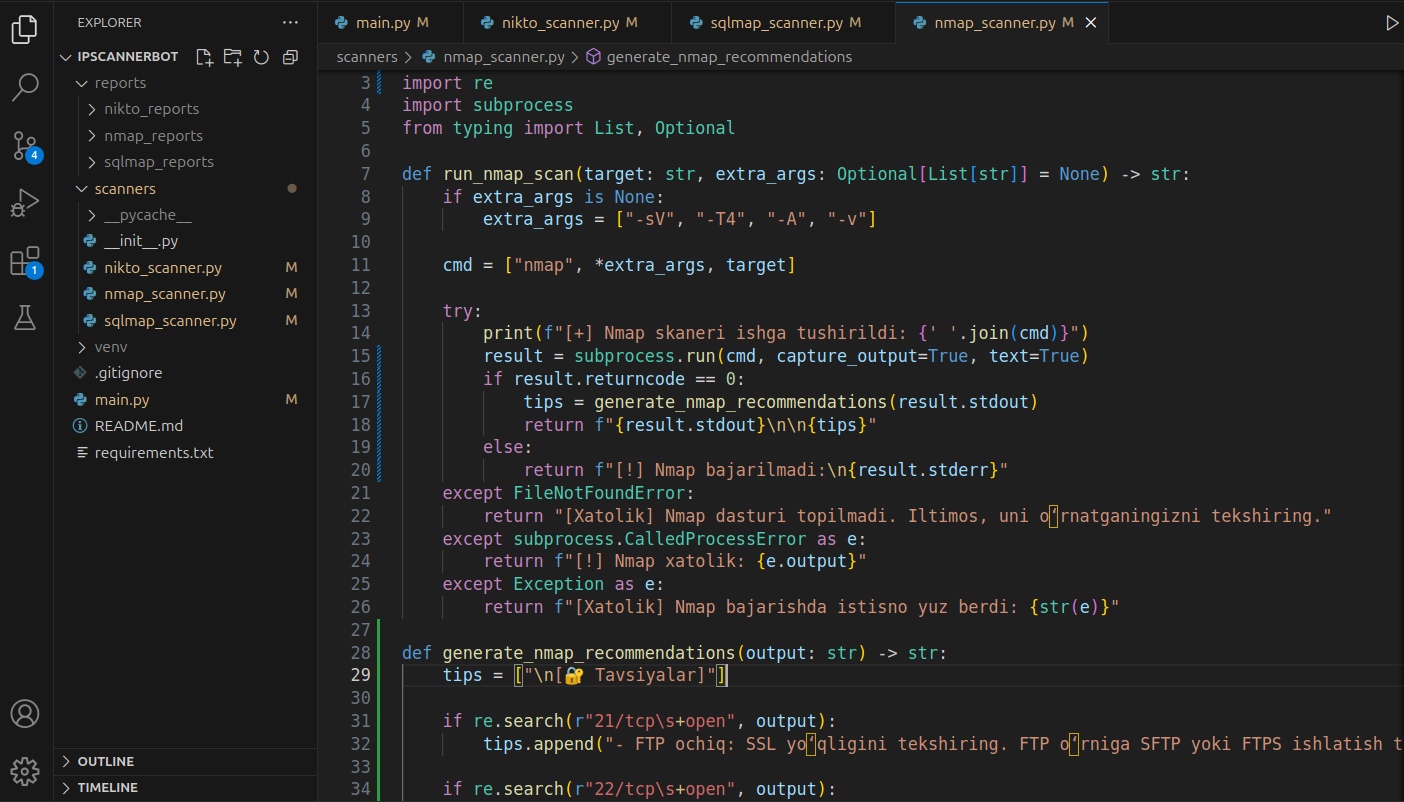
## 2.2. Dastur funksiyalarining umumiy tavsifi. Botni sinovdan o‘tkazish

Loyiha doirasida axborot tizimlaridagi zaifliklarni avtomatik aniqlovchi botning dasturiy qismi ishlab chiqildi va GitHub repozitoriyasida saqlandi. Bot Python dasturlash tilida ishlab chiqilgan bo‘lib, u Nmap, SQLmap, Nikto vositalari bilan integratsiya qilingan.

Asosiy fayl sifatida main.py (2.2.1-rasm) tanlab olingan. Barcha kerakli modullar ushbu faylda import qilinadi. Ushbu faylda skaner vositasini tanlash, parametrlarni sozlash, xavfsiz input’lar bilan ishlash, kiritilgan qiymatlarni tekshirish va jarayonni tugatish funksiyalari joylashgan. Funksiyalarni nomlashda foydalanuvchilarga tushunarli bo’lishi uchun skaner vositalarining nomlaridan foydalanilgan. Skaner vositalarining dastur kodlari scanners papkasi ichida joylashgan. Har bir scanner modulda skaner vositasini ishga tushiruvchi va tavsiyalarni avtomatik generatsiya qiluvchi ikkitadan funksiya mavjud (2.2.2-rasm).



2.2.1-rasm. main.py faylidagi dastur kodi.

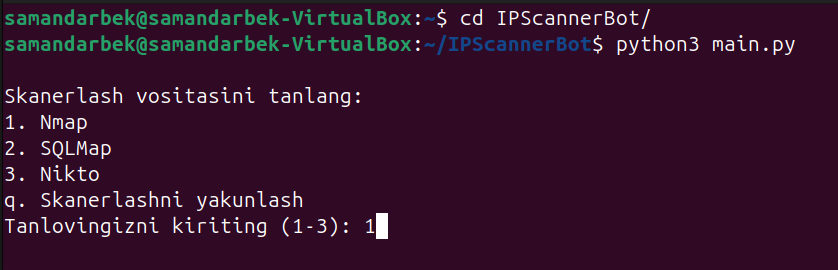


2.2.2-rasm. nmap\_scanner.py faylidagi Nmap vositasining dastur kodi.

Yuqoridagi rasmlarda barcha vositalarning dastur kodlari joylashgan fayllarni ham natijalar saqlanadigan papkalarni ko’rish mumkin. Ushbu python skriptni terminal orqali ishga tushirish mumkin.

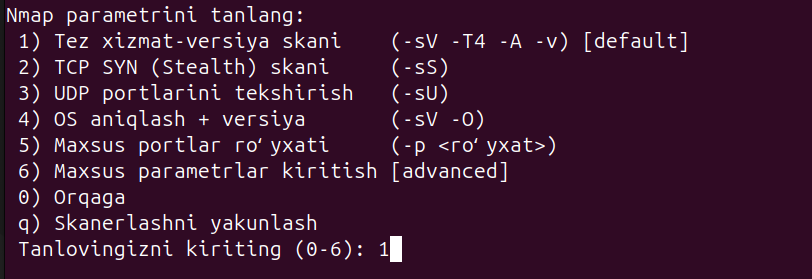
Bot quyidagi bosqichlarda ishlaydi (botni Nmap skaneri sifatida to’liq sinovdan o’tkazamiz):

1. Github orqali dastur kodi yuklab olingan. Terminal orqali IPScannerBot papkasiga o’tiladi va bot ishga tushiriladi:



2.2.3-rasm. Vositalar menyusi.

1. Vositalar menyusidan (2.2.3-rasm) Nmap’ni (1) tanlaymiz va parametrlar menyusidan (2.2.4-rasm) “Tez xizmat-versiya skani”ni (1) tanlaymiz:



2.2.4-rasm. Nmap vositasining parametrlar menyusi.

1. Vosita va parametr tanlanganidan keyin skaner qilinadigan nishon kiritiladi (<https://example.com>) va skanerlash jarayoni boshlanadi (2.2.5-rasm):



2.2.5-rasm. Skanerlash jarayonining boshlanishi.

1. Natija va tavsiyalar hisobot sifatida matnli (.txt) ko’rinishda saqlanadi va ekranda chop etiladi (2.2.6-rasm):



2.2.6-rasm. Nmap skaneri natijasining ekranda chop etilishi.

reports/<tool\_name>\_reports/<target>\_<tool\_name>\_time

Natija va tavsiyalar yuqoridagi ko’rinishda saqlanadi. Bu foydalanuvchilarga zarur hollarda skanning natijasidan foydalanish imkonini beradi. Barcha vositalardan shu tartibda foydalanish mumkin. Kerakli vosita va parametr tanlanadi hamda skaner qilinadigan nishon manzili kiritiladi. Natijani skanning tugashi bilanoq tavsiyalar bilan birga ekranda ko’rish mumkin bo’ladi.

Ushbu bo‘limda ko‘rib chiqilganidek, ishlab chiqilgan bot foydalanuvchiga real vaqt rejimida xavfsizlik tekshiruvlarini amalga oshirish, natijalarni saqlash va avtomatlashtirilgan tavsiyalar olish imkonini beradi. Bu esa axborot tizimlarida xavfsizlikni nazorat qilish jarayonini soddalashtiradi va tezlashtiradi.

## II bob bo‘yicha xulosa

Ushbu bobda axborot tizimlaridagi zaifliklarni avtomatik aniqlovchi botni ishlab chiqish uchun zarur bo‘lgan ishchi muhitni yaratish, dasturiy vositalarni o‘rnatish hamda botning funksiyalarini testdan o‘tkazish bosqichlari yoritib berildi. VirtualBox orqali ajratilgan izolyatsiyalangan muhitda Ubuntu operatsion tizimi asosida xavfsiz va barqaror ish muhiti tashkil qilindi. Bot Python tilida ishlab chiqildi va Nmap, SQLmap, Nikto kabi xavfsizlik vositalari bilan integratsiya qilindi. GitHub orqali boshqarilayotgan loyiha tuzilmasi modul ko‘rinishida tashkil etilib, har bir vosita alohida fayl orqali chaqiriladi.

Bot terminal orqali ishga tushiriladi va foydalanuvchi tomonidan tanlangan skanerlash vositasi, parametrlar va nishon asosida xavfsizlik tahlilini amalga oshiradi. Natijalar matnli fayl ko‘rinishida saqlanadi va avtomatik tarzda tavsiyalar bilan ta’minlanadi. Barcha jarayonlar real sharoitda sinovdan o‘tkazilib, botning funksional jihatdan ishlash samaradorligi amaliy isbotini topdi.

# Xulosa

Biz mazkur malakaviy amaliyot ishida axborot tizimlaridagi zaifliklarni avtomatik aniqlovchi bot ishlab chiqdik. Loyiha davomida axborot tizimlaridagi zaifliklar hamda ularni aniqlovchi vositalarning nazariy asoslarini o’rgandik. Shuningdek, kurs ishidan ko’zlangan kerakli ko’nikmalar ham hosil qilindi:

* Axborot xavfsizligi sohasida mavjud zaifliklar va ularni aniqlash usullarini nazariy jihatdan o‘rgandik.
* Nmap, SQLmap va Nikto vositalarini integratsiya qilgan holda Python dasturlash tilida zaifliklarni skanerlovchi bot yaratdik.
* Aniqlangan zaifliklar haqida natijalar va tavsiyalarni qulay formatda chop etdik va matnli fayl ko’rinishida saqladik.
* Kurs ishi davomida bajarilgan amaliyotlardan kerakli xulosalarni chiqardik.

Loyiha dastlab xavfsiz va izolyatsiyalangan virtual muhit tashkil etishdan boshlandi. VirtualBox asosida yaratilgan virtual mashinaga Ubuntu operatsion tizimi o‘rnatilib, test va ekspluatatsiya uchun qulay muhit yaratildi.

Bot Python dasturlash tilida yozilib, unga Nmap, SQLmap va Nikto kabi xavfsizlik vositalari integratsiya qilindi. Har bir skaner vositasi alohida modulda yozilib, umumiy boshqaruv main.py fayli orqali amalga oshirildi. Foydalanuvchi terminal orqali kerakli vosita, parametr va nishon manzilini tanlab, tizim zaifliklarini aniqlashi va hisobot shaklida tavsiyalar olishi mumkin. Natijalar avtomatik tarzda .txt fayl ko‘rinishida saqlanadi.

Kurs ishini bajarish davomida duch kelgan muammo va savollarimizga kurs ishi rahbarimiz tomonidan kerakli maslahat, tavsiya va amaliy yordamlar berildi. Ushbu kurs ishi davomida olgan bilimlarimiz kelajakdagi faoliyatimizda asqotishiga ishonamiz.

# Foydalanilgan adabiyotlar

1. Tanenbaum A.S., Wetherall D.J. **Computer Networks**. – Pearson Education, 2011.
2. Kurose J.F., Ross K.W. **Computer Networking: A Top-Down Approach**. – 7th Edition, Pearson, 2016.
3. Python Documentation – <https://docs.python.org/3/>
4. Ubuntu Linux Documentation – <https://ubuntu.com/docs>
5. VirtualBox Official Documentation – <https://www.virtualbox.org/manual/>
6. Nmap Network Scanning Guide – <https://nmap.org/book/>
7. SQLmap User Guide – <https://sqlmap.org/>
8. Nikto Web Scanner Documentation – https://cirt.net/Nikto2
9. <https://chatgpt.com/>
10. <https://github.com/nmap/nmap>
11. <https://github.com/sqlmapproject/sqlmap>
12. <https://github.com/sullo/nikto>
13. <https://github.com/samandar8283/IPScannerBot>