ISSN 2181-3187

DASTURIY TA’MINOTNING ZAIFLIKLARI: TURLARI, SABABLARI VA HIMOYA CHORALAR

Fazliddinova Nigora Avaz qizi

Farg’ona davlat universuteti

Amaliy matematika yo’nalishi

1- bosqich magistranti

Mamajonova Dilnoza Dilshodjon qizi

Farg’ona davlat universuteti

Amaliy matematika yo’nalishi

1- bosqich magistranti

Annotatsiya. Ushbu maqolada dasturiy ta’minotning zaifliklari, ularning turlari, yuzaga kelish sabablari va xavfsizlikni ta’minlash usullari tahlil qilinadi. Dasturiy ta’minotdagi zaifliklar kiberhujumlar uchun asosiy imkoniyatlardan biri bo‘lib, ularning oldini olish dasturchilar va IT mutaxassislarining muhim vazifalaridan hisoblanadi. Maqolada SQL Injection, Cross-Site Scripting (XSS), autentifikatsiya va avtorizatsiya kamchiliklari, xotira zaifliklari kabi keng tarqalgan muammolar batafsil o‘rganilib, ularning oldini olish bo‘yicha kriptografik himoya, xavfsizlikni dasturlash bosqichida ta’minlash, testlash va xavfsizlik auditlari kabi samarali choralar tavsiya etiladi. Shuningdek, maqolada dasturiy ta’minot xavfsizligini oshirishda OWASP standartlari va zamonaviy xavfsizlik texnologiyalari ning ahamiyati yoritilgan. Ushbu tadqiqot dasturiy ta’minot ishlab chiquvchilari va axborot xavfsizligi bo‘yicha mutaxassislar uchun foydali metodik qo‘llanma bo‘lib xizmat qiladi.

ISSN 2181-3187

Kalit so‘zlar: dasturiy ta’minot xavfsizligi, zaifliklar, SQL Injection, Cross-Site Scripting (XSS), autentifikatsiya, avtorizatsiya, xotira zaifliklari, kriptografiya, xavfsizlik auditlari, OWASP, penetration testing, dasturiy kod tahlili, kiberxavfsizlik.

Kirish. **Axborot texnologiyalarining jadal rivojlanishi bilan dasturiy ta’minot xavfsizligi dolzarb masalaga aylandi. Dasturiy ta’minot zaifliklari (vulnerability) – bu dasturlarda mavjud bo‘lgan xatoliklar, noto‘g‘ri kod yozish natijasida yuzaga keladigan xavfsizlik kamchiliklari bo‘lib, ular kiberhujumlar uchun imkoniyat yaratadi. Ushbu maqolada dasturiy ta’minot zaifliklari turlari, ularning sabablarini aniqlash va ularni bartaraf etish usullari haqida batafsil ma’lumot beriladi.**

Mavzuga oid adabiyotlar sharhi. Dasturiy ta’minot xavfsizligi bo‘yicha ilmiy va texnik adabiyotlar keng ko‘lamni qamrab oladi. Quyida ushbu sohada muhim asarlar hamdamanbalar sharhi keltiriladi.

1. **R. Xamdamov, G. Jalilov – *Axborot xavfsizligi asoslari* (2019)** Ushbu o‘quv qo‘llanmada axborot xavfsizligi tushunchasi, uning huquqiy va texnik asoslari, dasturiy ta’minot xavfsizligini ta’minlash bo‘yicha muhim tamoyillar bayon etilgan. Mualliflar kiberhujumlarning oldini olish, shifrlash usullari va dasturiy ta’minot himoyasi haqida batafsil tushuntirishlar beradi.
2. **I. Ismoilov – *Kompyuter tarmoqlari va axborot xavfsizligi* (2021)** Kitobda kompyuter tarmoqlari xavfsizligi, kriptografik himoya usullari va dasturiy ta’minotning zaif tomonlarini aniqlash bo‘yicha muhim usullar keltirilgan. Shuningdek, zamonaviy xavfsizlik protokollari va tarmoq hujumlariga qarshi choralar bo‘yicha tavsiyalar berilgan.
3. **Sh. Ergashev – *Dasturiy ta’minot xavfsizligi va tahlili* (2020)** Kitobda dasturiy ta’minot xavfsizligi bo‘yicha asosiy tahdidlar, ularning yuzaga kelish sabablari va oldini olish usullari yoritilgan. Ayniqsa, **SQL**

ISSN 2181-3187

Injection, Cross-Site Scripting (XSS), buffer overflow kabi zaifliklarning real misollar asosida tushuntirilgani kitobning amaliy ahamiyatini oshiradi.

4. **X. Karimov – *Kriptografiya asoslari* (2018)**

Kriptografiya va uning dasturiy ta’minot xavfsizligidagi o‘rni haqida asosiy bilimlarni taqdim etadi. AES, RSA, SHA-256 kabi shifrlash algoritmlarining ishlash prinsiplari, ularning dasturiy ta’minot himoyasidagi roli haqida batafsil ma’lumot berilgan.

1. Dasturiy Ta’minot Zaifliklarining Turlari

Dasturiy ta’minotdagi zaifliklar turli ko‘rinishda namoyon bo‘lishi mumkin. Ularni quyidagi asosiy toifalarga ajratish mumkin:

1.1. Kiritilgan ma’lumotlarni noto‘g‘ri ishlov berish

SQL Injection (SQLi) – bu dasturiy ta’minotdagi zaiflik bo‘lib, unda hujumchi foydalanuvchi tomonidan kiritilgan malumotlar orqali SQL so‘rovlarini manipulyatsiya qilib, ma’lumotlar bazasiga ruxsatsiz kirish yoki uni o‘zgartirish imkoniyatiga ega bo‘ladi. SQL Injection hujumlari veb-sayt yoki dasturiy ta’minotning ma’lumotlar bazasiga so‘rov yuboradigan qismlarida amalga oshiriladi.

Oddiy SQL so‘rovi quyidagi ko‘rinishda bo‘lishi mumkin:

*SELECT * FROM users WHERE username = 'admin' AND password = '123456';*

Bu hujum odatda veb-ilovalarda uchraydi va foydalanuvchi kiritgan ma’lumotlarni tekshirish va filtratsiya qilmaslik natijasida yuzaga keladi.

Cross-Site Scripting (XSS) – bu veb-saytlarga zararli kod joylashtirish orqali foydalanuvchilarga hujum qilish usuli. Ushbu hujumda hujumchi veb-sayt kodiga

ISSN 2181-3187

zararli JavaScript kodlarini kiritib, foydalanuvchilarning shaxsiy ma’lumotlarini o‘g‘irlashi, seanslarini buzishi, yoki veb-sayt interfeysini o‘zgartirishi mumkin.

XSS hujumlari odatda noto‘g‘ri sanitizatsiya qilingan foydalanuvchi kiritmalari orqali amalga oshiriladi. Ya’ni, sayt foydalanuvchidan matn kiritishini talab qilsa, lekin kiritilgan HTML va JavaScript kodlarni to‘g‘ri filtrlamasa, XSS zaifligi yuzaga keladi.

SS hujumi odatda quyidagi bosqichlarda amalga oshiriladi:

1. Foydalanuvchi ma’lumot kiritishi mumkin bo‘lgan maydon aniqlanadi (kommentariylar, chat, qidiruv qutisi va h.k.).
2. Zararli JavaScript kodi ushbu maydonga kiritiladi.
3. Sayt foydalanuvchining zararli kodini filtrlamasdan chiqaradi.
4. Zararli kod boshqa foydalanuvchilarning brauzerlarida ishga tushadi.

1.2. Autentifikatsiya va Avtorizatsiya Kamchiliklari

Zaif parollar va ularning noto‘g‘ri boshqarilishi – foydalanuvchilarning oddiy yoki oldindan ma’lum bo‘lgan parollardan foydalanishi. Zaif parollar oson taxmin qilinadigan, qisqa yoki oddiy kombinatsiyalardan iborat bo‘lgan parollar. Ko‘pincha, foydalanuvchilar qulaylik uchun juda oddiy yoki bir necha joyda bir xil paroldan foydalanadi, bu esa xavfsizlikni pasaytiradi.

Eng keng tarqalgan zaif parollar:

- 123456
- password
- qwerty
- 123456789

ISSN 2181-3187

- admin
- ilovatingiz nomi yoki foydalanuvchi ismi

Bu kabi parollarni bruteforce hujumlar orqali juda tez buzish mumkin.

Token o‘g‘irlash – sessiya identifikatorlarini ekspluatatsiya qilish orqali foydalanuvchi ma’lumotlariga noqonuniy kirish. **Token o‘g‘irlash** hujumchilar foydalanuvchilarning autentifikatsiya tokenlarini qo‘lga kiritish orqali ularning akkauntlariga **ruxsatsiz kirish** qilish usulidir. Agar token o‘g‘irlansa, hujumchi foydalanuvchi nomidan tizimga kira oladi va **parolni bilmasdan ham to‘liq nazorat o‘rnatishi mumkin.**

1.3. Xotira Zaifliklari

Buffer Overflow – dasturiy kod belgilangan chegaradan ortiq ma’lumotni qabul qilganda, dasturiy ta’minot noto‘g‘ri ishlashiga olib keladi va bu zaiflik ekspluatatsiya qilinishi mumkin. Buffer Overflow xotira buferining chegarasidan tashqariga chiqib yozish natijasida yuzaga keladigan dasturiy zaiflik. Ushbu hujum natijasida hujumchi dasturning ish faoliyatini buzishi, kodni ekspluatatsiya qilish yoki imtiyozlarni oshirish imkoniyatiga ega bo‘lishi mumkin.

Bu zaiflik asosan C, C++ kabi xotira boshqaruvini o‘zi amalga oshiradigan dasturlash tillarida ko‘p uchraydi. Chunki bu tillarda avtomatik xotira himoyasi mavjud emas.

Use-After-Free – allaqachon bo‘shatilgan xotiraga murojaat qilish natijasida kiberjinoyatchilar tomonidan suiste’mol qilinishi mumkin. Use-After-Free (UAF) – bu xotira buzilishi bilan bog‘liq zaiflik bo‘lib, dastur allaqachon bo‘shatilgan xotira manziliga murojaat qilish natijasida yuzaga keladi.

ISSN 2181-3187

Bu zaiflik ko‘pincha C va C++ tillarida uchraydi, chunki ushbu tillarda xotira boshqaruvi qo‘lda amalga oshiriladi va avtomatik xotira tozalash (garbage collection) yo‘q.

Muammo: Agar dastur bo‘shatilgan xotira hududidan foydalanishda davom etsa, bu dastur ishdan chiqishiga, ma’lumotlarning buzilishiga, yoki hatto xavfli kod bajarilishiga olib kelishi mumkin.

Quyidagi bosqichlarda UAF zaifligi yuzaga kelishi mumkin:

- Dastur xotira ajratadi (malloc(), new yoki boshqa funksiyalar orqali).
- Xotira ishlatiladi.
- Xotira bo‘shatiladi (free(), delete kabi funksiyalar yordamida).
- Bo‘shatilgan xotira maydoniga murojaat qilinadi (Use-After-Free sodir bo‘ladi).

1.4. Kriptografik Zaifliklar

Zaif shifrlash algoritmlari – eskirgan yoki zaif shifrlash metodlarini ishlatish (masalan, MD5, SHA1). Zaif shifrlash o‘zining kriptografik mustahkamligini yo‘qotgan, zamonaviy hujumlar oldida himoya qila olmaydigan yoki buzilishi oson bo‘lgan shifrlash algoritmlaridir.

Bunday shifrlash usullari hujumchilar tomonidan parollarni o‘g‘irlash, maxfiy ma’lumotlarni deshifrlash, yoki soxta autentifikatsiya hujumlari uchun ekspluatatsiya qilinishi mumkin.

Zaif shifrlash algoritmlari bugungi kunda hujumlar oldida zaif bo‘lib qolgan va ularni ishlatish xavfsizlikka jiddiy tahdid soladi.

🔻 Eng xavfli zaif shifrlash algoritmlariga quyidagilar kiradi:

✅ MD5 va SHA-1 (Hash algoritmlari)

ISSN 2181-3187

✅ DES va RC4 (Simmetrik shifrlash algoritmlari)

✅ WEP (Wi-Fi shifrlash standarti)

🔹 Zamonaviy himoya choralari:

✅ SHA-256, SHA-3, AES-256 kabi algoritmlardan foydalanish

✅ Parollar uchun bcrypt, Argon2, yoki PBKDF2 ishlatish

✅ TLS 1.3 va WPA3 kabi zamonaviy protokollardan foydalanish

Agar ushbu tavsiyalarga rioya qilinsa, kriptografik xavfsizlik sezilarli darajada oshadi va zaif shifrlash tufayli yuzaga keladigan xatarlardan himoyalanish mumkin!

Tasodifiy sonlar generatsiyasidagi kamchiliklar – kriptografik operatsiyalar uchun foydalaniladigan tasodifiy sonlar noto‘g‘ri ishlatilganda, tizim xavfsizligi pasayadi. Tasodifiy sonlar generatsiyasi (Random Number Generation – RNG) – bu dasturlar tomonidan oldindan bashorat qilib bo‘lmaydigan raqamlarni yaratish jarayoni. RNG kriptografiya, o‘yinlar, simulyatsiyalar, lotereya tizimlari, parol yaratish, va boshqa ko‘plab sohalarda ishlatiladi.

Ikki asosiy tur mavjud:

1. True Random Number Generator (TRNG) – haqiqiy tasodifiylik (fizik hodisalarga asoslangan).
2. Pseudo-Random Number Generator (PRNG) – taxminiy tasodifiylik (matematik algoritmlar yordamida generatsiya qilinadi).

Muammo: Agar RNG noto‘g‘ri ishlasa yoki zaif bo‘lsa, hujumchilar tasodifiy sonlarni bashorat qilib, maxfiy ma’lumotlarni buzishlari mumkin!

2. Dasturiy Ta’minot Zaifliklarining Sabablari

Dasturiy ta’minotdagi zaifliklar turli omillar natijasida yuzaga keladi. Asosiy sabablarini quyidagicha ajratish mumkin:

ISSN 2181-3187

Dasturchilar xatolari – kod yozishda yo‘l qo‘yiladigan xatolar va himoya choralarining yetarli darajada qo‘llanilmasligi. Dasturchilar tomonidan qilingan xatolar **dasturiy ta’minotning zaiflashishiga, ma’lumotlarning o‘g‘irlanishiga yoki tizimning ishlamay qolishiga olib kelishi mumkin.** Ushbu xatolar ko‘pincha **kodlash jarayonida, xavfsizlik choralariga e’tiborsizlik qilganda,** yoki **tajribasizlik tufayli** sodir bo‘ladi.

Kam tajribali dasturchilar – xavfsizlikka oid amaliyotlar bilan tanish bo‘lmagan dasturchilar tomonidan ishlab chiqilgan dasturlar zaif bo‘lishi mumkin. Kam tajribali dasturchilar ko‘plab xavfsizlik muammolariga olib kelishi mumkin. Ular odatda xavfsizlikni ta’minlash, kodni samarali va xavfsiz yozish bo‘yicha yetarli bilimga ega bo‘lmaydilar. Bu esa tizimning zaiflashishiga, ma'lumotlarning o‘g‘irlanishiga yoki hatto tizimning ishlamay qolishiga olib kelishi mumkin. Kam tajribali dasturchilar tomonidan yo‘l qo‘yiladigan xatolarni aniqlash va ularga qarshi kurashish, xavfsiz dasturlashni ta'minlash uchun juda muhimdir.

Vaqt va resurs yetishmovchiligi – tezkor ishlab chiqish jarayoni (agile development) xavfsizlik tekshiruvlariga yetarlicha e’tibor berilmasligiga olib kelishi mumkin. Dasturiy ta'minot ishlab chiqishda **vaqt va resurs yetishmovchiligi** ko'pincha sifat va xavfsizlikka salbiy ta'sir ko'rsatadi. Tezkor ishlab chiqish muddatlari, cheklangan resurslar va budjetlar, ishlab chiqish jarayonini tezlashtirish zarurati kabi omillar xavfsiz, mustahkam va barqaror dastur yaratishga to'sqinlik qilishi mumkin. Dasturiy ta'minotni yaratish jarayonida bu muammolar xavfsizlik zaifliklarining paydo bo'lishiga olib kelishi va tizimlarning zaiflashishiga sabab bo'lishi mumkin.

Buzilgan yoki eskirgan dasturiy ta’minot – yangilanmagan va xavfsizlik yamoqlari qo‘llanilmagan dasturlar zaif bo‘lib qoladi. Dasturiy ta’minotning buzilishi yoki eskirishi — bu **yaxshi ishlashini to‘xtatgan, xavfsizlik zaifliklari va muammolarni yuzaga keltirgan** tizimlarning mavjudligini anglatadi. Bunday

ISSN 2181-3187

tizimlar foydalanuvchilarga xizmat ko‘rsatishda to‘liq ishlamasligi mumkin, shuningdek, tizimga kirishni osonlashtirgan yoki ma'lumotlarni yo‘qotishga olib keladigan xavfsizlik zaifliklarini yaratadi. Dasturiy ta'minotning eskirishiga va buzilishiga olib keluvchi omillar ko‘p, lekin eng ko‘p uchraydiganlari — **yangilanishlar va patchlar qo‘llanmasligi, komponentlarning eskirishi,** va **qator texnologik zaifliklar** hisoblanadi.

3. Dasturiy Ta’minot Zaifliklaridan Himoyalanish Choralar

Dasturiy ta’minotni xavfsiz qilish uchun turli usullar mavjud. Quyida asosiy himoya choralarini ko‘rib chiqamiz:

3.1. Xavfsizlikni dasturlash bosqichida ta’minlash

Secure Coding Practices – OWASP va boshqa xavfsizlik standartlariga rioya qilish. Xavfsiz kodlash amaliyotlari (Secure Coding Practices) — dasturiy ta'minot ishlab chiqishda xavfsizlikni ta'minlash va potentsial xavfsizlik zaifliklarini oldini olish uchun qo'llaniladigan usullar va yondashuvlar to'plamidir. Dasturchilar, ishlab chiqish jarayonida xavfsiz kod yozish orqali, tizimni zararli hujumlardan himoya qilishadi va foydalanuvchilarning shaxsiy ma'lumotlarini saqlashni ta'minlaydilar.

Kodning xavfsizligi dasturiy ta'minotning umumiy xavfsizligini ta'minlashda muhim rol o'ynaydi. Quyida xavfsiz kodlash amaliyotlari haqida batafsil ma'lumot berilgan.

Kirish ma’lumotlarini tekshirish – foydalanuvchi kiritgan barcha ma’lumotlarni sanitizatsiya qilish va tekshirish. Kirish ma'lumotlarini tekshirish (Input Validation) — bu dasturiy ta'minotda foydalanuvchidan yoki boshqa tizimlardan olingan ma'lumotlarni to‘g‘ri formatda va kutilgan qiymatlarda ekanligini tekshirish jarayonidir. Ushbu amaliyot, tizimni xavfsizlik zaifliklari, masalan, SQL Injection, Cross-Site Scripting (XSS) va buffer overflow kabi hujumlardan himoya qilish uchun muhimdir.

ISSN 2181-3187

Kirish ma'lumotlarini tekshirish dasturga kirish uchun ishlatiladigan barcha foydalanuvchi ma'lumotlarini, kirish ma'lumotlarini, URL parametrlarini, va foydalanuvchi kiritgan boshqa ma'lumotlarni, tizimni himoya qilish uchun xavfsizlik nuqtai nazaridan tasdiqlashni talab qiladi.

Zaifliklarni avtomatik skanerlash – dastur kodini analiz qilish uchun SAST (Static Application Security Testing) va DAST (Dynamic Application Security Testing) vositalaridan foydalanish. Zaifliklarni avtomatik skanerlash (Automated Vulnerability Scanning) — dasturiy ta'minot yoki tizimni xavfsizlik zaifliklariga qarshi skanerlashni avtomatlashtirish jarayonidir. Bu usul tizimlarni tez va samarali ravishda tekshirishga yordam beradi, xavfsizlik xatoliklarini aniqlashni soddalashtiradi va dasturchilarga ularni tuzatish uchun zarur bo‘lgan vaqtni tejash imkoniyatini yaratadi.

Zaifliklarni avtomatik skanerlash yordamida tizimlarning potentsial zaifliklari aniqlanadi va tizim xavfsizligini oshirish uchun kerakli choralar ko‘riladi. Keng tarqalgan zaifliklar, masalan, SQL Injection, Cross-Site Scripting (XSS), buffer overflow, credential stuffing va boshqa ko‘plab muammolarni avtomatik ravishda topish mumkin.

3.2. Xavfsizlikni kuchaytirish bo‘yicha texnik chora-tadbirlar

Kriptografik himoya – kuchli shifrlash algoritmlaridan foydalanish (AES, SHA-256). **Kriptografik himoya** – ma'lumotlarni himoya qilish, xavfsiz saqlash va uzatish uchun kriptografiya usullaridan foydalanish jarayonidir. Kriptografiya ma'lumotlarni shifrlash, imzolash, autentifikatsiya qilish va ma'lumotlar yaxlitligini ta'minlash orqali ma'lumotlar xavfsizligini oshiradi. Dasturiy ta'minot va tizimlar uchun kriptografik himoya, foydalanuvchi ma'lumotlarini o'g'irlashdan, yomon niyatli hujumchilardan himoya qilishda muhim ahamiyatga ega.

ISSN 2181-3187

Xotira xavfsizligi – Address Space Layout Randomization (ASLR) va Data Execution Prevention (DEP) texnologiyalaridan foydalanish**. Xotira xavfsizligi** — bu tizimda ishlayotgan dastur yoki ilovalarning xotira (RAM) resurslarini xavfsiz va samarali boshqarish, ma'lumotlarni himoya qilish, va tizimni yomon niyatli hujumlardan himoya qilish jarayonidir. Xotira xavfsizligi, tizimni noto‘g‘ri ishlashdan, xatolardan, ma'lumotlarning o‘g‘irlanishidan yoki halokatli zaifliklardan himoya qilishga yordam beradi. Xotira bilan bog'liq zaifliklar ko‘pincha dasturchilar tomonidan bajariladigan kodlardagi xatolar, noto‘g‘ri xotira boshqaruvi, va yomon kodlash amaliyotlari natijasida yuzaga keladi.

Dasturiy ta’minotni muntazam yangilash – xavfsizlik yamoqlarini o‘z vaqtida o‘rnatish. **Dasturiy ta'minotni muntazam yangilash** tizimlar va dasturlarni doimiy ravishda yangilab turish jarayonidir. Bu jarayon, dasturdagi xatolarni tuzatish, xavfsizlik zaifliklarini bartaraf etish, yangi funksiyalar qo‘shish va umumiy ishlashni yaxshilash maqsadida amalga oshiriladi. Muntazam yangilanishlar yordamida dasturiy ta’minotning samaradorligi va xavfsizligi ta'minlanadi.

3.3. Testlash va xavfsizlik auditlari

Pentestr sinovlari (penetration testing) – dastur xavfsizligini real hujum sharoitida sinash. **Pentest** (Penetration Testing) yoki **penetratsion sinovlar** — tizimlar, tarmoqlar, dasturlar yoki tashkilotlar infratuzilmasidagi xavfsizlikni baholash uchun amalga oshiriladigan hujum simulyatsiyasidir. Pentestning maqsadi, tizimni hujumchilarga qarshi himoyasizligini aniqlash va uning zaifliklarini kashf qilishdir. Pentestda, hujumchilar kabi tizimga kirish, zaifliklarni topish va undan foydalanish uchun turli metodlar va vositalar qo‘llaniladi.

Automatizatsiyalashgan zaiflik skanerlari – Nessus, Burp Suite, ZAP kabi vositalardan foydalanish. Avtomatizatsiyalashgan zaiflik skanerlari (Automated Vulnerability Scanners) — bu dasturiy ta’minot vositalaridir, ular tizim, tarmoq yoki dasturiy ilovalarini xavfsizlik zaifliklari va zaifliklarni avtomatik tarzda aniqlash

ISSN 2181-3187

uchun ishlatiladi. Bu vositalar xavfsizlik teshiklarini tez va samarali aniqlashga yordam beradi, shuningdek, tizimni hujumchilardan himoya qilish uchun zarur choralarni ko‘rishga imkon beradi.

Avtomatizatsiyalashgan zaiflik skanerlari asosan quyidagi vazifalarni bajaradi:

1. **Tizimni skanerlash**: Avtomatik ravishda tizim yoki tarmoqdagi barcha xavfsizlik nuqsonlarini skanerlash.
2. **Zaifliklarni aniqlash**: Tizimdagi yoki dasturdagi potentsial xavfsizlik zaifliklarini aniqlash.
3. **Xavfsizlikni tekshirish**: Dasturlar va tarmoqlarni turli xavfsizlik hujumlariga qarshi sinovdan o‘tkazish.
4. **Hisobot yaratish**: Aniqlangan zaifliklar, xavfsizlik teshiklari va muammolar haqida to‘liq hisobotlar tayyorlash.

Xodimlarni o‘qitish – dasturchilar va foydalanuvchilarning xavfsizlik xabardorligini oshirish. **Xodimlarni o‘qitish** — tashkilotning xavfsizlik siyosatlarini amalga oshirish va tizimlarining himoyasini mustahkamlashdagi eng muhim choralardan biridir. Dasturiy ta'minot va tizimlar xavfsizligini ta'minlash faqat texnik vositalar va tizimlar orqali amalga oshirilmaydi, balki xodimlar ham xavfsizlikni saqlashda muhim rol o‘ynaydi. Xodimlar o‘qitilsa, ular xavfsizlik tahdidlarini aniqlash va ularga qarshi choralar ko‘rishda samarali bo‘lishadi.

Xulosa. **Dasturiy ta'minot xavfsizligi bo‘yicha ilmiy va amaliy adabiyotlar tizimli yondashuvni o‘z ichiga oladi. Ushbu adabiyotlar, dasturiy ta'minotning xavfsizligini ta'minlash, tizimlar va ilovalarda mavjud bo‘lgan zaifliklarni aniqlash hamda ularni samarali bartaraf etish bo‘yicha muhim bilimlarni taqdim etadi. Xususan, OWASP (Open Web Application Security Project) standartlari, kriptografik usullar, zamonaviy zaifliklarni aniqlash texnikalari va dasturiy audit metodologiyalari xavfsizlik sohasida professional faoliyat yuritish uchun zarur bilimlar bazasini yaratadi.**

ISSN 2181-3187

Dasturiy ta'minot xavfsizligini ta'minlashda zamonaviy texnologiyalar va metodologiyalar muhim ahamiyatga ega. Bular, nafaqat xavfsizlik zaifliklarini aniqlash va tuzatish, balki dasturiy ta'minot ishlab chiqish jarayonida xavfsizlikni inobatga olishni ta'minlashga yordam beradi. Xodimlarni xavfsizlikka oid o‘qitish, xavfsizlikni doimiy ravishda monitor qilish, va avtomatizatsiyalashgan zaiflik skanerlari kabi vositalarni qo‘llash orqali tizimlarni himoya qilishda muvaffaqiyatga erishish mumkin.

Dasturiy ta'minot xavfsizligini doimiy ravishda oshirish va rivojlantirish tashkilotlar uchun zarurdir, chunki bu jarayon tizimlar va ma'lumotlarni himoya qilish, foydalanuvchi ma'lumotlarini xavfsiz saqlash va axborot xavfsizligi bo‘yicha yuqori talablarni qondirishga imkon beradi. Shuning uchun, dasturiy ta'minot xavfsizligi sohasidagi bilimlarni chuqurlashtirish va eng yangi texnologiyalarga asoslangan metodlarni qo‘llash tashkilotlar va dasturchilar uchun muhim ahamiyatga ega.

Foydalanilgan adabiyotlar

1 Xamdamov R., Jalilov G. *Axborot xavfsizligi asoslari.* – Toshkent: TATU nashriyoti, 2019.

2 Ismoilov I. *Kompyuter tarmoqlari va axborot xavfsizligi.* – Toshkent: Fan va texnologiya, 2021.

3 Ergashev Sh. *Dasturiy ta’minot xavfsizligi va tahlili.* – Toshkent: Innovatsiya, 2020.

4 Karimov X. *Kriptografiya asoslari.* – Toshkent: Axborot texnologiyalari, 2018.

5 O‘zbekiston Respublikasi Axborot texnologiyalari va kommunikatsiyalarini rivojlantirish vazirligi. *Axborot xavfsizligi bo‘yicha normativ hujjatlar to‘plami.* – Toshkent, 2022.