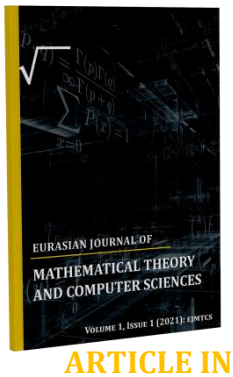
EURASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
Innovative Academy Research Support Center
UIF = 8.3 | SJIF = 5.916 www.in-academy.uz



DJANGO FREYMVORK ASOSIDA WEB ILOVALAR YARATISHDA AXBOROT XAVFSIZLIGINI TA'MINLASH

Normurodov Asliddin Alijon o'g'li
Termiz Davlat Univeristeti Kompyuter tizimlari va ularning dasturiy ta'minoti 1 - kurs magistranti
E-mail: normurodovasliddin7@gmail.com
https://www.doi.org/10.5281/zenodo.10516620

ARTICLE INFO

Received: 08th January 2024
Accepted: 15th January 2024
Online: 16th January 2024

KEY WORDS

Django, DEBUG, freymvork, csrf_cookie, HTTPS, SQLsql.

ABSTRACT

Djangoning eng so'nggi versiyasini o'rnatish va yangi versiyalarda xavfsizlikga yo'naltirishlar, maslahatlar va to'liqroq uskunalarni qo'llab-quvvatlash uchun yangilanish uchun quyidagi buyruq amalga oshiriladi. ***pip install --upgrade*** *Django Ilova ishlayotgan davrda DEBUG sozlamasini Falsega o'zgartirish xavfsizlikni oshiradi.*

 # settings.py
DEBUG = False

Qo'shimcha ravishda, ALLOWED_HOSTS sozlamasini to'ldirish kerak: ALLOWED_HOSTS = ['your_domain.com']. HTTP xavfsizlik siyosati (HTTP Security Headers) orqali xavfsizlikni oshirish uchun, X-Content-Type-Options hederni qo'llang. SECURE_CONTENT_TYPE_NOSNIFF = True. Ilova HTML sarlavhasining ijarachiligini cheklash uchun X-Frame-Options hederni sozlash.

# settings.py
X_FRAME_OPTIONS = 'DENY'.

Saytga ulangan ma'lumotlar uchun HTTPS protokolini ishlatish juda muhim. Ushbu xavfsizlikni oshirish uchun, SSL sertifikat olish va sozlash kerak.

Django ilovasining ishlashi uchun, barcha POST so'rovlari uchun CSRF tokenlarini tekshiradi. Agar siz shu tokenlarni ishlatish orqali ilovaning barcha so'rovlari optimal yechimga erishadi. Shuning uchun shu kodni ilova asosiy ma'lumotlar papkasidagi views.py fayliga qo'shish orqali yechimga erishamiz:

from django.views.decorators.csrf
import ensure_csrf_cookie
@ensure_csrf_cookie
def your_view(request):
 # your view logic here

Bu qadamlar sizga Django ilovalaringizni xavfsiz ishlatishda yordam bera oladi. Xavfsizlikning o'zini ta'minlashda davom etish, maslahatlar olish va so'ngi yangilanishlarni kuzatish juda muhim hisoblanadi.

Djangoda xavfsizlik. Ushbu hujjat Djangoning xavfsizlik xususiyatlarining umumiy ko'rinishidir. U Django-da ishlaydigan saytni himoya qilish bo'yicha maslahatlarni o'z ichiga oladi.

Saytlararo skript (XSS) himoyasi. XSS hujumlari foydalanuvchiga mijoz tomonidagi skriptlarni boshqa foydalanuvchilarning brauzerlariga kiritish imkonini beradi. Bunga odatda zararli skriptlarni ma'lumotlar bazasida saqlash va u boshqa foydalanuvchilarga ko'rsatish yoki tajovuzkorning JavaScript-ni foydalanuvchi brauzeri tomonidan bajarilishiga olib keladigan havolani bosish orqali erishiladi. Biroq, XSS hujumlari har qanday ishonchsiz ma'lumotlar manbalaridan, masalan, cookie-fayllar yoki veb-xizmatlardan kelib chiqishi mumkin, agar ma'lumotlar sahifaga kiritishdan oldin etarli darajada tozalanmagan bo'lsa.

Django shablonlaridan foydalanish sizni ko'pgina XSS hujumlaridan himoya qiladi. Biroq, u qanday himoyani va uning cheklovlarini tushunish muhimdir.

Django shablonlari HTML uchun ayniqsa xavfli bo'lgan maxsus belgilardan qochadi . Bu foydalanuvchilarni ko'pgina zararli kirishlardan himoya qilsa-da, bu mutlaqo bema'ni emas. Masalan, u quyidagilarni himoya qilmaydi:

<style class={{ var }}>...</style>

Agar varga o‘rnatilgan bo‘lsa , bu brauzerning nomukammal HTMLni qanday ko‘rsatishiga qarab, ruxsatsiz JavaScript-ning bajarilishiga olib kelishi mumkin. (Atribut qiymatini keltirish bu holatni hal qiladi.)

'class1 onmouseover=javascript:func()'

Bundan tashqari , maxsus shablon teglari, safeshablon yorlig'i, mark_safeva autoescape o'chirilganida foydalanishda ayniqsa ehtiyot bo'lish kerak .

Bundan tashqari, agar siz HTML-dan boshqa biror narsani chiqarish uchun shablon tizimidan foydalanayotgan bo'lsangiz, butunlay alohida belgilar va so'zlar bo'lishi mumkin, ular qochishni talab qiladi.

Bundan tashqari, HTMLni ma'lumotlar bazasida saqlashda juda ehtiyot bo'lishingiz kerak, ayniqsa bu HTML olinadi va ko'rsatilganda.

Saytlararo so'rovlarni qalbakilashtirish (CSRF) himoyasi CSRF hujumlari zararli foydalanuvchiga boshqa foydalanuvchining hisob ma'lumotlaridan foydalangan holda, uning ma'lumoti yoki roziligisiz harakatlarni amalga oshirishga imkon beradi.

Django CSRF hujumlarining ko'p turlaridan o'rnatilgan himoyaga ega, agar siz uni yoqsangiz va kerak bo'lganda foydalansangiz. Biroq, har qanday yumshatish texnikasi kabi, cheklovlar mavjud. Masalan, CSRF modulini global yoki alohida ko'rinishlar uchun o'chirib qo'yish mumkin. Buni faqat nima qilayotganingizni bilsangizgina qilishingiz kerak. Agar saytingizda sizning nazoratingizdan tashqarida bo'lgan subdomenlar bo'lsa, boshqa cheklovlar mavjud .

CSRF himoyasi har bir POST so'rovida sir borligini tekshirish orqali ishlaydi. Bu zararli foydalanuvchi POST shaklini veb-saytingizga “qayta o‘ynata” olmasligini va boshqa tizimga kirgan foydalanuvchini o‘zi bilmagan holda yuborishini ta’minlaydi. Zararli foydalanuvchi foydalanuvchiga xos bo'lgan sirni bilishi kerak (cookie yordamida).

HTTPS bilan o'rnatilganda CsrfViewMiddlewareHTTP havolasi sarlavhasi bir xil manbadagi URL manziliga o'rnatilganligini tekshiradi (shu jumladan subdomen va port). HTTPS qo'shimcha xavfsizlikni ta'minlaganligi sababli, xavfsiz ulanish so'rovlarini yo'naltirish va qo'llab-quvvatlanadigan brauzerlar uchun HSTS dan foydalanish orqali ulanishlar mavjud bo'lgan joyda HTTPS-dan foydalanishini ta'minlash zarur.

csrf_exemptAgar juda zarur bo'lmasa, dekorativ bilan ko'rinishlarni belgilashda juda ehtiyot bo'ling .

SQL qarshi himoyasi

SQL in'ektsiyasi - bu zararli foydalanuvchi ma'lumotlar bazasida o'zboshimchalik bilan SQL kodini bajarishi mumkin bo'lgan hujum turi. Bu yozuvlarning o'chirilishi yoki ma'lumotlarning sizib chiqishiga olib kelishi mumkin.

Django so'rovlari to'plamlari SQL in'ektsiyasidan himoyalangan, chunki ularning so'rovlari so'rov parametrlari yordamida tuzilgan. So'rovning SQL kodi so'rov parametrlaridan alohida aniqlanadi. Parametrlar foydalanuvchi tomonidan taqdim etilgan va shuning uchun xavfli bo'lishi mumkinligi sababli, ular asosiy ma'lumotlar bazasi drayveri tomonidan qochib ketadi.

Django shuningdek, ishlab chiquvchilarga xom so'rovlarni yozish yoki maxsus sql-ni bajarish uchun kuch beradi . Ushbu imkoniyatlardan tejamkorlik bilan foydalanish kerak va siz har doim foydalanuvchi nazorat qila oladigan parametrlardan to'g'ri qochish uchun ehtiyot bo'lishingiz kerak. extra() Bundan tashqari, va dan foydalanganda ehtiyot bo'lishingiz kerak RawSQL.

Clickjacking - bu zararli sayt boshqa saytni ramkaga o'raydigan hujum turi. Ushbu hujum, shubhasiz foydalanuvchini maqsadli saytda ko'zda tutilmagan harakatlarni amalga oshirish uchun aldashga olib kelishi mumkin.

Django qo'llab-quvvatlovchi brauzerda saytni freym ichida ko'rsatilishiga to'sqinlik qilishi mumkin bo'lgan kliklardan himoyani o'z ichiga oladi. Har bir ko'rish asosida himoyani o'chirib qo'yish yoki yuborilgan sarlavhaning aniq qiymatini sozlash mumkin.X-Frame-Options middleware

O'rta dastur o'z sahifalarini uchinchi tomon saytlari tomonidan ramkaga o'rashga hojat yo'q yoki faqat saytning kichik bo'limiga ruxsat berishi kerak bo'lgan har qanday sayt uchun tavsiya etiladi.

SSL/HTTPS

Xavfsizlik uchun saytingizni HTTPS orqasida joylashtirish har doim yaxshiroqdir. Busiz, zararli tarmoq foydalanuvchilari autentifikatsiya hisob ma'lumotlarini yoki mijoz va server o'rtasida uzatiladigan boshqa ma'lumotlarni, ba'zi hollarda - faol tarmoq tajovuzkorlari - har ikki yo'nalishda yuborilgan ma'lumotlarni o'zgartirishi mumkin .

Agar siz HTTPS taqdim etadigan himoyani xohlasangiz va uni serveringizda yoqgan bo'lsangiz, sizga kerak bo'lishi mumkin bo'lgan qo'shimcha qadamlar mavjud:

Agar kerak bo'lsa, SECURE_PROXY_SSL_HEADERu yerdagi ogohlantirishlarni to'liq tushunganingizga ishonch hosil qilib, sozlang. Buni bajarmaslik CSRF zaifliklariga olib kelishi mumkin va uni to'g'ri bajarmaslik ham xavfli bo'lishi mumkin!

HTTP orqali so‘rovlar HTTPSga yo‘naltirilishi uchun SECURE_SSL_REDIRECTga o‘rnating .True

Iltimos, ostidagi ogohlantirishlarga e'tibor bering SECURE_PROXY_SSL_HEADER. Teskari proksi-server uchun asosiy veb-serverni HTTPS-ga yo'naltirishni amalga oshirish uchun sozlash osonroq yoki xavfsizroq bo'lishi mumkin.

"Xavfsiz" cookie-fayllardan foydalaning.

Agar brauzer dastlab HTTP orqali ulansa, bu ko'pchilik brauzerlar uchun sukut bo'yicha, mavjud cookie fayllari sizib ketishi mumkin. Shuning uchun siz

SESSION_COOKIE_SECUREvaCSRF_COOKIE_SECURE sozlamalaringizni ga o'rnatishingiz kerak True. Bu brauzerga ushbu cookie-fayllarni faqat HTTPS ulanishlari orqali yuborishni buyuradi. Shuni esda tutingki, bu seanslar HTTP orqali ishlamasligini anglatadi va CSRF himoyasi har qanday POST ma'lumotlarini HTTP orqali qabul qilinishiga yo'l qo'ymaydi (agar siz barcha HTTP trafigini HTTPSga yo'naltirayotgan bo'lsangiz, bu yaxshi bo'ladi).

HTTP qat'iy transport xavfsizligi (HSTS) dan foydalaning

HSTS HTTP sarlavhasi bo'lib, brauzerga ma'lum bir saytga bo'lajak barcha ulanishlar har doim HTTPSdan foydalanishi kerakligi haqida xabar beradi. HTTP orqali HTTPS ga qayta yo‘naltirish so‘rovlari bilan birgalikda bu ulanishlar SSL ning qo‘shimcha xavfsizligidan bahramand bo‘lishini ta’minlaydi, agar bitta muvaffaqiyatli ulanish amalga oshirilsa. SECURE_HSTS_SECONDSHSTS , SECURE_HSTS_INCLUDE_SUBDOMAINS, va bilan SECURE_ HSTS_ PRELOAD yoki veb-serverda sozlanishi mumkin .

Xost sarlavhasini tekshirish

Django Hostmuayyan holatlarda URL manzillarini yaratish uchun mijoz tomonidan taqdim etilgan sarlavhadan foydalanadi. Ushbu qiymatlar saytlararo skript hujumlarining oldini olish uchun tozalangan bo'lsa-da, soxta Hostqiymatdan saytlararo so'rovlarni qalbakilashtirish, keshni zaharlash hujumlari va elektron pochta xabarlarida zaharlanish havolalari uchun foydalanish mumkin.

Hatto xavfsiz ko'rinadigan veb-server konfiguratsiyalari ham soxta Hostsarlavhalarga moyil bo'lganligi sababli, Django Hostsarlavhalarni ALLOWED_HOSTSusuldagi sozlamaga qarshi tasdiqlaydi django.http.HttpRequest.get_host().

Bu tekshirish faqat orqali amal qiladi get_host(); agar sizning kodingiz sarlavhaga Hostto'g'ridan-to'g'ri request. METAsizdan kirsa, bu xavfsizlik himoyasini chetlab o'tgan bo'lasiz.

Sozlashlar fayli (`settings.py`): Django ilovangizning asosiy sozlashlarini saqlash uchun `settings.py` faylining ichida kerakli sozlashlarni o'rnating. Bu orqali maxfiy klyuchlar, bazaga kirish ma'lumotlari, ilova nomi va boshqa sozlashlarni boshqarishingiz mumkin.

Masalan:

```
DEBUG = False  # Debug rejimini o'chirish
ALLOWED_HOSTS = ['example.com']  # Faqat ruxsat etilgan hostlar
```

Boshqaruv panelini himoya qilish: Django boshqaruv paneli (`/admin/`) uchun maxfiylikni ta'minlash juda muhimdir. Agar siz boshqaruv panelini o'zgartirganingizda, uni to'liq maxfiy qilish uchun yangi foydalanuvchi nomlari va klyuchlarni sozlashni unutmang.

Masalan:

```
# settings.py
# Boshqaruv panelini maxfiylikka rioya qilish
from decouple import config, Csv
SECRET_KEY = config('SECRET_KEY')
```

HTTPS ishlatish: Sayt tashrif buyuruvchilarning ma'lumiyatlarini himoya qilish uchun HTTPS ishlatish juda muhimdir. Bu uchun SSL sertifikatni o'rnatishingiz kerak. Django ilova

uchun `django-sslserver` yoki `django-secure` kabi keng qamrovli kutubxonalarni ishlatishingiz mumkin.

Django `csrf` himoyasi: Django freymvorki avtomatik ravishda `csrf` himoyasini amalga oshiradi. Foydalanuvchilarning saytlarga xavfsiz kirishini ta'minlash uchun bu kerakli va muhimdir.

SQL injektsiyalarga qarshi himoya: Django freymvorki SQL injektsiyalarga qarshi maxfiylikni ta'minlash uchun yaxshi imkoniyatlarga ega. Agar siz Django'nun ORM qo'llanmoqda bo'lsangiz, sizni SQL injektsiyalariga qarshi himoya qilish bo'yicha ko'p tajriba mavjud bo'ladi.

Fayl yuklash kabi operatsiyalarga cheksizlik: Ilovalar foydalanuvchidan fayllar qabul qilib, saqlab turadi va ularga kirish huquqini boshqarishni ta'minlash muhimdir. Fayllarni saqlash orqali ilova serverining xavfsizligini ta'minlash juda muhimdir.

Ehtiyojga mos ravishda kutubxonalardan foydalanish: Django hamda Python ekosistemasi o'zida keng ko'lamli kutubxonalarga ega. Maxfiylik va xavfsizlik uchun dasturlash jarayonida ehtiyojga mos ravishda kutubxonalardan foydalanish sizning ishingizni osonlashtiradi.

Auditoriyani chegaralash: Foydalanuvchilarning amaliyotlarini kuzatish va rad etish uchun Django logging tizimidan foydalanish juda muhimdir. Xatoliklarni va xavfsizlik muammoalarini aniqlab chiqish uchun logging sozlashlarini o'zgartiring.

Bu qadamlar sizning Django ilovangizni xavfsiz qilish uchun asosiy qadamlardir. Xavfsizlikni ta'minlashda unutilmaydigan narsa, ilovaning dasturchisi sifatida xavfsiz dasturlash prinsiplarini tushunish va ularni amalga oshirishdir.

O'zingizning Django ilovangizni yaratish uchun asosiy axborot xavfsizligi prinsiplarini o'rganish uchun dastur tuzish maqbuldir. Bu dastur quyidagi qadamlarni o'z ichiga oladi:

1-Qadam

1. Django O'rnatish:

- Django o'rnatishingizni tekshirib, o'rnating.

pip install django

2. Django Proyekti Yaratish:

- Yangi Django proyekti yaratish uchun terminalda quyidagi buyruqni ishga tushiring.

django-admin startproject myproject

3. Proyektga Kirish:

- Proyektni o'zi ichiga olib, proyekt papkasiga kiring.

cd myproject

2- Qadam

Axborot Xavfsizligi Qo'llash

1. ALLOWED_HOSTS Sozlashi:

- `settings.py` faylida `ALLOWED_HOSTS` sozlashini to'g'ri qiymatlar bilan sozlang.

```
# settings.py
ALLOWED_HOSTS = ['namuna.uz']
```

2. DEBUG Rejimi:

- `settings.py` faylida `DEBUG` rejimini o'chiring.

```
# settings.py
```

DEBUG = False

3. HTTPS Ishlatish:

- SSL sertifikatini o'rnatib, `SECURE_SSL_REDIRECT` sozlashi orqali barcha HTTP so'rovlarni HTTPS ga yo'naltiring.

 # settings.py
SECURE_SSL_REDIRECT = True

4. Django `csrf` Himoyasi:

- Formalarda `csrf_token` ishlatishni unutmang.

<!-- HTML formasi -->
<form method="post" action="{% url 'some_view' %}">
{% csrf_token %}
<!-- Qolgan formani yozing -->
</form>

5. SQL Injektsiyalarga Qarshi Himoya:

- ORM-ni foydalanib, raw SQL so'rovlarni ishlatmaslikni ta'minlang.

3-Qadam : Django Security Middleware Qo'llash

1. Middleware Qo'llash:

- `settings.py` faylida `SecurityMiddleware` ni qo'llash.

```
# settings.py
MIDDLEWARE = [
    # Boshqa middleware lar...
    'django.middleware.security.SecurityMiddleware',
    # Boshqa middleware lar...
]
```

4- Qadam : Ilovaning Logging Tizimi

1. Logging Sozlashi:

- `settings.py` faylida logging sozlashlarini qo'llash.

```
# settings.py
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'handlers': {
        'file': {
            'level': 'ERROR',
            'class': 'logging.FileHandler',
            'filename': '/path/to/your/logs/error.log',
        },
    },
    'loggers': {
        'django': {
            'handlers': ['file'],
            'level': 'ERROR',
            'propagate': True,
```

```
            },
        },
    }
```

5-Qadam: Fayl Yuklash Xavfsizligi

1. Fayl Yaratish:

- Ilova ichida foydalanuvchidan qabul qilinadigan fayllarni yuklashda xavfsizlikni ta'minlash.

```
# views.py
from django.conf import settings
from django.core.files.storage import FileSystemStorage
def upload_file(request):
    if request.method == 'POST' and request.FILES['myfile']:
        myfile = request.FILES['myfile']
        fs = FileSystemStorage(location=settings.MEDIA_ROOT)
        filename = fs.save(myfile.name, myfile)
        return HttpResponse(f'File "{filename}" successfully uploaded.')
    return render(request, 'upload.html')
```

Bu dastur o'zingizning Django ilovangizni yaratish va xavfsizlikni ta'minlashda bir boshlang'ich nuqtani ta'minlash uchun yordam bera oladi. Unutmang, ilova logicasi va foydalanuvchidan olingan ma'lumotlar bilan ishlashda xavfsizlikni oshirish juda muhimdir.

References:

1. Rubio, D. (2017). Beginning Django. Apress..
2. Liawatimena, S., Warnars, H. L. H. S., Trisetyarso, A., Abdurahman, E., Soewito, B., Wibowo, A., ... & Abbas, B. S. (2018, September). Django web framework software metrics measurement using radon and pylint. In 2018 Indonesian Association for Pattern Recognition International Conference (INAPR) (pp. 218-222). IEEE.
3. Vamsi, K. M., Lokesh, P., Reddy, K. N., & Swetha, P. (2021). Visualization of real world enterprise data using python Django framework. In IOP Conference Series: Materials Science and Engineering (Vol. 1042, No. 1, p. 012019). IOP Publishing.
4. Forcier, J., Bissex, P., & Chun, W. J. (2008). Python web development with Django. Addison-Wesley Professional.
5. Karimova, Z., Xaydarov, S., & Doniyorova, G. (2023). МАНИПУЛИРОВАНИЕ ИЗОБРАЖЕНИЯМИ С ПОМОЩЬЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ ПРИ РАСПОЗНАВАНИИ ДОРОЖНЫХ ЗНАКОВ. Предпринимательства и педагогика, 5(1), 112-119.
6. Хайдаров, Ш. (2023). ФОРМИРОВАНИЕ БАЗЫ ДАННЫХ ПРЕДУПРЕЖДЕНИЙ О ДОРОЖНЫХ ЗНАКАХ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА РАСПОЗНАВАНИЯ ОБЪЕКТОВ. Предпринимательства и педагогика, 5(1), 103-112.
7. Ziyatovich, M. F., & Islom o'g'li, X. S. (2023). SUN'IY INTELLEKT VA UNING TA'LIM SOHASIGA ALOHIDA MUROJAAT QILGAN HOLDA TURLI SOHALARDAGI QAMROVI. ОБРАЗОВАНИЕ НАУКА И ИННОВАЦИОННЫЕ ИДЕИ В МИРЕ, 16(3), 16-19.

8. Мамажанов, Р. Я., & Хайдаров, Ш. И. (2022). РАЗРАБОТКА ПЛАТФОРМЫ ИННОВАЦИОННОГО ПОДХОДА ПРИ ПОДГОТОВКЕ НАУЧНЫХ СТАТЕЙ НА МЕЖДУНАРОДНОМ УРОВНЕ. CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES, 3(12), 267-269.
9. Мамажанов, Р., & Хайдаров, Ш. (2022). ЦИФРОВАЯ ЭКОНОМИКА В ПОВЫШЕНИИ ЭКОНОМИКИ И ВЛИЯНИЯ ВУЗОВ РАСПОЛОЖЕНИЕ В ЦЕНТРЕ. CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES, 3(12), 270-275.