AXBOROT TARMOQLARINI BOSHQARISH VA MONITORING QILISHDA PROXY SERVERLARDAN FOYDALANISHNING AHAMIYATI

G‘aniyev Temur Komiljon o‘g‘li,
Guliston davlat pedagogika instituti doktoranti
Email: ganiyevtemur1@gmail.com

Annotatsiya: Ushbu maqolada proxy serverlarning tarmoq boshqaruvidagi va monitoring qilish ahamiyati, ularning turlari, qo‘llanilishi va ularsiz tizimning zaif tomonlari haqida batafsil tushunchalar beriladi. Shuningdek c# dasturlash tilidan foydalanib proxy server yaratish jarayoni ko‘rsatilib o‘tiladi.

Kalit so‘zlar: proxy server, axborot tarmoqlari, monitoring, tarmoq boshqaruvchi, keshlash, dasturlash, c#, trafik

Abstract: This article provides a detailed understanding of the importance of proxy servers in network management and monitoring, their types, applications, and the vulnerabilities of the system without them. It also shows the process of creating a proxy server using the C# programming language.

Keywords: proxy server, information networks, monitoring, network manager, caching, programming, C#, traffic

Аннотация: В данной статье подробно рассматривается роль прокси-серверов в управлении и мониторинге сетей, их типы, области применения и уязвимости системы без них. Также показан процесс создания прокси-сервера на языке программирования C#.

Ключевые слова: прокси-сервер, информационные сети, мониторинг, сетевой менеджер, кэширование, программирование, C#, трафик

Kirish

Zamonaviy axborot tarmoqlari doimiy rivojlanib borayotgan muhit bo‘lib, unda ma’lumotlar almashinuvi, xavfsizlik va samaradorlik juda muhim ahamiyatga ega.

Korporativ tarmoqlar, davlat idoralari va hatto kichik bizneslar ham o‘z tizimlarini himoya qilish va boshqarish uchun turli texnologiyalardan foydalanadilar. Shu texnologiyalar orasida **proxy serverlar** alohida o‘rin tutadi. Ular nafaqat tarmoq trafigini optimallashtirishda, balki foydalanuvchilarning harakatini monitoring qilish, ma’lumotlar xavfsizligini ta’minlash va ma’lumotlar oqimlarni boshqarishda ham muhim rol o‘ynaydi. [3]

Ushbu maqolada proxy serverlarning tarmoq boshqaruvidagi ahamiyati, ularning turlari, qo‘llanilishi va ularsiz tizimning zaif tomonlari haqida batafsil to‘xtalamiz.

Mavzuga oid adabiyotlarning tahlili

Proxy server – bu foydalanuvchi (mijoz) bilan internet resurslari o‘rtasida vositachi sifatida ishlaydigan maxsus server. Uning asosiy vazifasi mijozning so‘rovini qabul qilib, uni internetdagi maqsadli serverga yo‘naltirish va javobni foydalanuvchiga qaytarishdir. [2]

Proxy Serverning ish prinsipi:

1. Foydalanuvchi so‘rov yuboradi (masalan, veb-saytga kirish).
2. So‘rov avval proxy serverga boradi.
3. Proxy server so‘rovni tahlil qiladi (filtrlash, keshlash, loglash).
4. Agar so‘rov ruxsat etilgan bo‘lsa, u internetdagi serverga yo‘naltiriladi.
5. Javob proxy orqali foydalanuvchiga qaytariladi.

Bu jarayon foydalanuvchining haqiqiy IP-manzilini yashirish, trafikni nazorat qilish va tarmoq tezligini oshirish imkonini beradi.

Proxy serverlar tarmoqni boshqarishda quyidagi jihatlarda muhim rol o‘ynaydi:

a) Trafikni optimallashtirish va tezlikni oshirish

- Keshlash (Caching) – Proxy server tez-tez tashrif buyuriladigan veb-sahifalarni o‘z xotirasida saqlaydi, bu esa bir xil kontentni qayta yuklashni kamaytiradi. [2]

- Bandwidth ni samarali taqsimlash – Bir nechta foydalanuvchilar bir xil resursdan foydalanganda, proxy server yukni muvozanatlash (load balancing) orqali tarmoq ishini yaxshilaydi.

b) Xavfsizlikni Ta’minlash

- IP-manzilni yashirish – Foydalanuvchining haqiqiy manzili yashirinadi, bu esa kiberhujumlardan himoya qiladi.

- Malware va Phishingga Qarshi Himoya – Zararli saytlarga kirish bloklanadi. [3]

- DDoS-hujumlarini oldini olish– Proxy server so‘rovlarni tekshirib, yaroqsiz trafikni to‘sib qo‘yadi.

c) Monitoring va Hisobot Tuzish

- Foydalanuvchi faoliyatini kuzatish – Qaysi saytlarga kirilayotgani, qancha vaqt sarflanganligi kabi ma’lumotlar yig‘iladi.

- Audit loglari – Barcha tarmoq harakatlari saqlanib, qonuniy talablar va ichki nazorat uchun foydalaniladi[3].

Tadqiqot metodologiyasi (Research Methodology)

Proxy Serverlarning bir qancha turlari mavjud bo‘lib, har bir aniq vazifaga javob beradi.

Proxy Turi	Tavsifi	Qo‘llanilishi
Forward Proxy	Mijozlar internetga chiqishda ishlatiladi.	Korporativ tarmoqlarda foydalanuvchi trafigini boshqarish.
Reverse Proxy	Server oldida turib, yukni taqsimlash va himoya qilish.[8]	Veb-saytlar, cloud xizmatlari.
Transparent Proxy	Foydalanuvchi bilmagan holda trafikni yo‘naltiradi.	Internet provayderlari, kutubxonalar.
High Anonymity Proxy	IP-manzilni butunlay yashiradi.	Maxfiylikni talab qiladigan ishlar.

1-jadval. Proxy serverlarning turlari va ularning qo‘llanilishi

Proxy Serverlarsiz tizimning zaif jihatlari

Agar tashkilot proxy serverlardan foydalanmasa, quyidagi muammolar yuzaga kelishi mumkin:

Xavfsizlik zaifligi – Hackerlar to‘g‘ridan-to‘g‘ri tarmoqqa kirish imkoniyatiga ega bo‘lishi mumkin. [7]

Trafikni nazorat qilish qiyinlashadi – Bandwidth samarasiz ishlatilishi mumkin.

Monitoring imkoniyatlari cheklanadi – Foydalanuvchilar harakatini kuzatish qiyin bo‘ladi.

Tahlil va natijalar

C# dasturlash tilidan foydalangan holatda proxy server yaratishning eng oson va qulay usullaridan biri HttpListener sinfdan foydalanishdir. Quyida c# dasturlash tilidan proxy server yaratish kodlari keltirilgan:

```
using System.Net;
/* Yangi HttpListener klassidan obyekt yaratamiz va mos url manzilda so‘rovlarni qabul qilishni boshlaymiz */
HttpListener listener = new HttpListener();
listener.Prefixes.Add("http://localhost:8080/");
listener.Start();
// So‘rovlarni kutamiz.
HttpListenerContext context = listener.GetContext();
/* Kelgan so‘rovning tanasidan request va response obyektlarni alohida request va response o‘zgaruvchilariga o‘zlashtirib olamiz. */
HttpListenerRequest request = context.Request;
HttpListenerResponse response = context.Response;
/* Kelgan so‘rovga o‘zgartirish kiritish agar kerak bo‘lasa (headers qo‘shish, URL ni o‘zgartirish).
```

```
So‘ovni kerakli manzilga yo‘naltirish destinationRequest obyekti orqali */
HttpWebRequest destinationRequest = (HttpWebRequest)
WebRequest.Create(request.Url);
/* request obyektning mavjud headerlarni destinationRequest obyektini headerlariga
nusxa olish jarayoni */
foreach(string header in request.Headers) {
   destinationRequest.Headers[header] = request.Headers[header];
}
/* destinationRequest obyektidan kelgan javob so‘rovni destinationResponse obyektga
o‘zlashtiriladi. */
HttpWebResponse destinationResponse = (HttpWebResponse)
destinationRequest.GetResponse();
/* destinationResponse obyektida hosil bo‘lgan headerlarni response obyektini
headerlariga nusxa olish jarayoni */
foreach(string header in destinationResponse.Headers) {
   response.Headers[header] = destinationResponse.Headers[header];
}
/* destinationResponse obyektidan so‘rovga nisbatan javob oqimni yangi
destinationStream obyektiga o‘zlashtiramiz va oqimni response obyektini chiquvchi
oqimiga nusxalaymiz*/
using(Stream destinationStream = destinationResponse.GetResponseStream())
using(Stream outputStream = response.OutputStream) {
   destinationStream.CopyTo(outputStream);
}
/* response va listener obyektlarini barcha obyektlarga nisbatan bog‘lanishlarni
uzamiz va xotiradan bo‘shatamiz */
response.Close();
```

```
listener.Close();
```

Yuqoridagi dastur kodi http://localhost:8080 manzilida HttpListener sinfidan olingan obyekt yordamida foydalanuvchidan kelgan so‘rovlarni qabul qiladi va HttpWebRequest sinfdan obyekt yaratib ularni foydlanuvchi so‘rovida nazarda tutilgan manzilga yo‘naltiradi. Natijada HttpWebRequest sinfdan olingan obyektga javob sifatida kelgan ma’lumotni foydalanuvchiga yo‘naltiradi. Ushbu dastur kodida eng sodda proxy serverga namuna keltirildi va haqiqiy proxy serverlarda so‘rovlarni qayta ishlashni turli xil usullari, xatolarni ko‘rib chiqish, timeout kabi tushunchalar bilan ishlay olish funksiyalari qo‘shilishi mumkin. [4]

Proxy server dasturmizni sinab ko‘rish uchun c# dasturlash tilida xunit test yozib ko‘ramiz va bu orqali dasturmizni tekshirib ko‘ramiz. Quydida dastur kodi keltirilgan:

```
using System.Net;
namespace ProxyServer.Tests;
public class ProxyServerTests
   [Fact]
   public void ProxyServerResturnsOkWithURL()
   {
      /*  Bizga kerakli bo‘lgan URL manzilni proxy serverdan so‘raymiz va proxy server uni yo‘naltirib beradi. */
      string destinationUrl = "https://gspi.uz";
      // Proxy serverga yangi so‘rov yaratamiz.
      HttpWebRequest request = (HttpWebRequest)WebRequest.Create("http://localhost:5233/");
      /* X-Destination-Url header yordamida biz kerakli bo‘lgan manzilni berib yuboramiz.*/
      request.Headers["X-Destination-Url"] = destinationUrl;
      /* So‘rov  yuboramiz va so‘rovga kelgan javobni  proxy serverdan olamiz. */
```

```
        HttpWebResponse response = (HttpWebResponse)request.GetResponse();
        // Kelgan javobni biz xohlagan natija bilan solishtiramiz. */
        Assert.Equal(HttpStatusCode.OK, response.StatusCode);
        Assert.Equal(destinationUrl, response.ResponseUri.AbsoluteUri);
        /* response obyketida mavjud bog‘lanishlarni uzamiz */
        response.Close();
    }
}
```

Xulosa va takliflar (Conclusion/Recommendations)

Proxy serverlar – bu zamonaviy tarmoq infratuzilmasining ajralmas qismi. Ular nafaqat xavfsizlikni oshiradi, balki tarmoq samaradorligini yaxshilash, foydalanuvchi faoliyatini nazorat qilish va qonuniy talablarga javob berish imkonini beradi. [5]

Tashkilotlar uchun takliflar:

- ✔ Korporativ proxy yechimlarini joriy eting(masalan, Squid, Nginx). [6]
- ✔ Trafikni muntazam monitoring qiling.
- ✔ Xavfsizlik sozlamalarini yangilab turing.

Proxy serverlardan to‘g‘ri foydalanilsa, ular tarmoqning barqarorligi va xavfsizligini sezilarli darajada oshirishi mumkin.

Foydalanilgan Adabiyotlar

1. Field, J. (2013). The security considerations with Squid proxy server. International Journal of Engineering Research & Technology (IJERT), 2(5), 112–118. ijert.org
2. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext Transfer Protocol -- HTTP/1.1 (RFC 2616). Internet Engineering Task Force (IETF). ietf.org
3. G‘aniyev, T. (2025). Axborot tarmoqlarida ma’lumot oqimlarni boshqarish hamda monitoring qilishning ahamiyati. Ta’lim, fan va innovatsiya, (7), 45–52.

4. Kumar, A., & Sharma, P. (2025). Analysis of the effectiveness of Squid proxy in website filtering in a network environment. ResearchGate. researchgate.net
5. Santos, M. R. (2020). Bandwidth management analysis using proxy server. SciTePress. scitepress.org
6. Wenceslao, J. L. B., Ramos, J. M., & Castro, L. D. (2022). Network performance of proxy-enabled server using three configurations. SRS Journal, 14(2), 89–97.
7. Williams, S. (2019). Web-based proxy server filtering system. ResearchGate. researchgate.net
8. Zhang, Y. (Ed.). (2021). Proxy server research papers. Academia. academia.edu