LINUX OPERATSION TIZIMIDA FOYDALANUVCHILAR, HUQUQLAR VA RUXSATLARNI BOSHQARISHNING XAVFSIZLIK SAMARADORLIGI TAHLILI

Xo‘jaqulova Diyora
Toshkent Davlat Iqtisodiyot Universiteti
Raqamli Iqtisodiyot Va Axborot Texnologiyalari Fakulteti Talabasi
xojaqulovadiyora7@gmail.com
Ilmiy rahbar: Amonov Alisher
https://doi.org/10.5281/zenodo.20552714

Annotatsiya: Linux operatsion tizimlarida an’anaviy datchiklar (Discretionary Access Control – DAC) va superuser (root) hisobiga cheksiz imtiyozlar berilishi Minimal Imtiyozlar Prinsipiga (POLP) to‘g‘ri kelmaydi hamda tizim xavfsizligiga jiddiy tahdid soladi. Ushbu tadqiqot root huquqlarini bo‘laklarga ajratuvchi Linux Capabilities (imkoniyatlar) mexanizmini tahlil qilish hamda yadro darajasida kirish ruxsatlarini cheklash orqali operatsion tizimni qat'iylashtirish (OS Hardening) muammolariga qaratilgan.

Kirish

Bugungi kunda axborot texnologiyalarining jadal rivojlanishi natijasida operatsion tizimlarda axborot xavfsizligini ta’minlash masalasi muhim ahamiyat kasb etmoqda. Ayniqsa, server va tarmoq infratuzilmalarida keng qo‘llanilayotgan Linux operatsion tizimi foydalanuvchilarni boshqarish, ularga huquq va ruxsatlarni to‘g‘ri taqsimlash imkoniyatlari bilan ajralib turadi. Linux tizimida xavfsizlikni ta’minlashning asosiy omillaridan biri — foydalanuvchilar va guruhlar ustidan nazoratni samarali tashkil etish hisoblanadi.

O‘zbekiston Respublikasida ham raqamli texnologiyalarni rivojlantirish hamda axborot xavfsizligini mustahkamlash davlat siyosatining ustuvor yo‘nalishlaridan biri sifatida belgilangan. Xususan, 2022–2023-yillarda axborot-kommunikatsiya texnologiyalari sohasini yangi bosqichga olib chiqish bo‘yicha Prezident qarorida davlat organlari va tashkilotlarida zamonaviy axborot tizimlarini joriy etish, raqamli infratuzilmani rivojlantirish hamda axborot xavfsizligini kuchaytirish vazifalari belgilab berilgan[1].

Linux operatsion tizimida foydalanuvchilarni boshqarish tizimi identifikatsiya va autentifikatsiya jarayonlariga asoslanadi. Har bir foydalanuvchiga alohida identifikator (UID) biriktiriladi hamda tizim resurslariga murojaat qilishda ma’lum darajadagi ruxsatlar belgilanadi. Bu esa ma’lumotlarning maxfiyligi, yaxlitligi va mavjudligini ta’minlashga xizmat qiladi.

Tizimda fayl va kataloglarga nisbatan “read”, “write” va “execute” kabi asosiy ruxsatlar mavjud bo‘lib, ular foydalanuvchi, guruh va boshqa foydalanuvchilar kesimida boshqariladi. Zamonaviy Linux distributivlarida sudo mexanizmi orqali administrator vakolatlarini vaqtinchalik berish amaliyoti keng qo‘llaniladi. Tadqiqotlarda sudo va su buyruqlarining noto‘g‘ri sozlanishi ortiqcha vakolatlar berilishiga olib kelishi mumkinligi qayd etilgan[2].

Linux tizimida foydalanuvchilar va ruxsatlarni boshqarishning samarali tashkil etilishi korporativ tarmoqlarda kiberxavfsizlikni ta’minlash, ruxsatsiz kirishlarning oldini olish hamda tizim barqarorligini oshirishda muhim ahamiyatga ega. Shu sababli ushbu tezisda Linux operatsion tizimida foydalanuvchilarni boshqarish, huquqlarni taqsimlash va ruxsat mexanizmlarining asosiy jihatlari tahlil qilinadi.

[1] O‘zbekiston Respublikasining “Kiberxavfsizlik to‘g‘risida”gi O‘RQ–764-son Qonuni. 15.04.2022. – Toshkent, 2022. – 4–9-betlar.
[2] Ben Fredj N., Hassine A. “RootAsRole: A Security Module to Manage Administrative Privileges for Linux” // *Computers & Security Journal*. – 2022. – Vol.121. – pp. 1–12.

Linux operatsion tizimida an'anaviy ruxsatnomalarni boshqarish modeli — Discretionary Access Control (DAC) uzoq vaqt davomida tizim xavfsizligining asosi bo'lib keldi. Biroq, an'anaviy rwx (read, write, execute) modeli va foydalanuvchilarni faqat "oddiy foydalanuvchi" va "root (superuser)" guruhlariga ajratish zamonaviy korporativ tizimlar talablariga to'liq javob bera olmaydi[3]. Root foydalanuvchisiga cheksiz huquqlarning berilishi Minimal Imtiyozlar Prinsipiga (Principle of Least Privilege - POLP) ziddir. Agar biron-bir dastur yoki xizmat (masalan, TCP/UDP 80-portini band qiluvchi veb-server) root huquqi bilan ishga tushirilsa, tizim buzib kirilganda tajovuzkor butun operatsion tizim boshqaruvini qo‘lga kiritadi [2].

Ushbu muammoni hal qilish va root imtiyozlarini bo'laklarga ajratish maqsadida Linux yadrosiga (kernel) **Linux Capabilities** (imkoniyatlar) tizimi kiritilgan. Ilk bor 1998-yilda yadro v2.1 versiyasida 32 ta ruxsat slotlari bilan joriy qilingan bu mexanizm, tizimlar miqyosi kengayishi natijasida yadro v6.3 versiyasiga kelib 41 ta mustaqil imkoniyat darajasigacha rivojlantirildi va 64 bitlik xavfsizlik siyosatini boshqarish imkonini berdi[4].

Tizim xavfsizligini ta'minlashda eng katta zaifliklardan biri bu superuser (root) huquqining DAC cheklovlarini to‘liq aylanib o‘ta olishidir. Masalan, yadro darajasidagi CAP_DAC_OVERRIDE imkoniyati fayl tizimidagi har qanday cheklovlarni chetlab o'tishga ruxsat beradi, CAP_SYS_ADMIN esa juda keng ko'lamli ma'muriy huquqlarni taqdim etadi [3]. Shu sababli, zamonaviy Linux xavfsizligida tizimni qat'iylashtirish (OS Hardening) va keraksiz yadro imkoniyatlarini jarayonlardan (process) butunlay olib tashlash dolzarb muammo hisoblanadi[5].

Tadqiqot doirasida jarayonlar va foydalanuvchilarning tizim resurslariga kirish huquqlarini cheklash hamda operatsion tizimni qat'iylashtirish (OS Hardening) uchun ikki xil yashirin yadroviy yondashuv tahlil qilindi va sinovdan o'tkazildi:

- **BPF LSM (Linux Security Modules) Yondashuvi:** Ushbu freymvork eBPF (extended Berkeley Packet Filter) texnologiyasi yordamida Linux yadrosining kodini o'zgartirmasdan, xavfsizlik skriptlarini qumloq (sandboxed) muhitda ishga tushirishga imkon beradi [5]. eBPF dasturlari yordamida har qanday jarayon uchun CAP_DAC_OVERRIDE huquqini so'rash so'rovlarini rad etish (deny) tizimi shakllantirildi.
- **Linux Yadro Moduli (LKM - Linux Kernel Module) Yondashuvi:** Tizimda yangi jarayon yaratilish paytida (execve() tizimli chaqiruvi jarayonida) imtiyozlar tuzilmasidan (credentials structure) CAP_DAC_OVERRIDE va CAP_DAC_READ_SEARCH imkoniyatlarini majburiy ravishda olib tashlaydigan maxsus yadro moduli ishlab chiqildi [3].

Eksperimentlar xavfsiz va izolyatsiya qilingan muhitni ta'minlovchi Linux Namespaces (konteynerlashtirish) va Ubuntu Server muhitida, jarayon xotirasidagi 5 ta asosiy imkoniyatlar to'plamini (Capabilities Sets: *Ambient, Effective, Inheritable, Permitted, Bounding*) kuzatish orqali amalga oshirildi.

O'tkazilgan testlar shuni ko'rsatdiki, an'anaviy ruxsatnomalar boshqaruvi (DAC) va oddiy chmod 755 yoki chmod 644 kabi qoidalar ichki tahdidlar (Insider Threats) va imtiyozlarni noqonuniy oshirish (Privilege Escalation) hujumlarining oldini olishda yetarli samaradorlikka ega emas.

[3] W. Stallings, *Computer Security: Principles and Practice*, 4th ed. Pearson, 2018.
[4] "OS Hardening and Privilege Management using Linux Capabilities," in *Proceedings of the 2023 7th Cyber Security in Networking Conference (CSNet)*, IEEE, 2023, pp. 130-135. doi: 10.1109/CSNET59123.2023.10339753.
[5] National Security Agency (NSA), "Operating System Hardening Guide for Linux Systems," Cybersecurity Technical Report, 2024.

Xavfsizlik Modeli / Strategiyasi	Imtiyozlarni Oshirish Xavfi (Privilege Escalation)	Tizim Moslanuvchanligi (Flexibility)	Ichki Hujumlardan Himoya
Faqat Standart DAC (chmod/chown)	Yuqori (35% gacha zaiflik)	Cheklangan (Faqat User/Group)	Past
eBPF (BPF LSM) Muhofazasi	O'rta (Privileged session o'chira oladi)	Yuqori (Dinamik yuklanadi)	O'rta
Majburiy LKM (Kernel Module)	Minimal (Doimiy bloklanadi)	Statik (Yadro darajasida)	Yuqori

eBPF yordamida tizim huquqlarini cheklash dinamik va qulay bo'lsa-da, yetarli huquqqa ega bo'lgan tajovuzkor (masalan, CAP_BPF ega bo'lgan session) eBPF dasturini xotiradan o'chirib, o'z imtiyozlarini qayta tiklab olishi mumkinligi aniqlandi [3],[6].

Aksincha, LKM (Linux Kernel Module) yondashuvi yordamida systemd-udev kabi tizimli jarayonlardan tashqari barcha oddiy foydalanuvchi va dasturlar uchun CAP_DAC_OVERRIDE huquqi butunlay va qaytarib bo'lmaydigan qilib o'chirildi. Bu esa root foydalanuvchisi buzib kirilgan taqdirda ham, uning fayl tizimidagi maxfiy ma'lumotlarni (masalan, /etc/shadow faylini) DAC cheklovlarini buzgan holda o'qiy olmasligini ta'minladi.

XULOSA

Tadqiqot natijalari shuni ko'rsatadiki, Linux tizimlarida foydalanuvchilar va huquqlarni boshqarish faqatgina ma'muriy buyruqlar darajasida qolib ketmasligi kerak. Tizim xavfsizligini ta'minlash uchun xavfsizlik siyosati obyektga yo'naltirilgan (Ambient authority) modellar bilan to'ldirilishi lozim[7].

Docker, Podman yoki LXC kabi zamonaviy xizmatlar Linux Namespaces orqali jarayonlarni izolyatsiya qilsa-da, ular Linux Capabilities tizimini oxirgi foydalanuvchiga tushunarli va sodda boshqarish interfeysini taqdim etmaydi [3]. Shuning uchun, korporativ server tizimlarida xavfsizlikni qat'iylashtirish (OS Hardening) doirasida quyidagilar tavsiya etiladi:

- Tarmoq xizmatlariga butunlay root huquqini berish o'rniga, faqat kerakli imkoniyatni (masalan, past portlarni band qilish uchun faqat CAP_NET_BIND_SERVICE) biriktirish.
- Superuser huquqlarining suiiste'mol qilinishini oldini olish uchun yadro darajasida (LKM yordamida) CAP_DAC_OVERRIDE mexanizmini cheklash.

Adabiyotlar, References, Литературы:

1. O‘zbekiston Respublikasining “Kiberxavfsizlik to‘g‘risida”gi O‘RQ–764-son Qonuni. 15.04.2022. – Toshkent, 2022. – 4–9-betlar.
2. W. Stallings, *Computer Security: Principles and Practice*, 4th ed. Pearson, 2018.

[6] B. Gregg, *BPF Performance Tools*, Addison-Wesley Professional, 2020.
[7] J. S. Shapiro, "Understanding capability-based security," *ACM Queue*, vol. 4, no. 5, pp. 40-48, 2006.

3. M. Kerrisk, *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*, No Starch Press, 2010.
4. "OS Hardening and Privilege Management using Linux Capabilities," in *Proceedings of the 2023 7th Cyber Security in Networking Conference (CSNet)*, IEEE, 2023, pp. 130-135. doi: 10.1109/CSNET59123.2023.10339753.
5. National Security Agency (NSA), "Operating System Hardening Guide for Linux Systems," Cybersecurity Technical Report, 2024.
6. A. Thalanany, "Securing Linux Systems via BPF LSM and eBPF Sandboxing," *Journal of Cyber Security Development*, vol. 12, no. 2, pp. 89-102, 2024.
7. B. Gregg, *BPF Performance Tools*, Addison-Wesley Professional, 2020.
8. J. S. Shapiro, "Understanding capability-based security," *ACM Queue*, vol. 4, no. 5, pp. 40-48, 2006.