FAOLIYATNI CHEKLASH USULLARI. SANDBOXING, CONTAINER ISOLATION, RESURS CHEKLASH, NETWORK MICRO-SEGMENTATION VA AVTOMATIK BLOKLASH

Akbarova Madinabonu Shavkatbek qizi
Andijon shahar 56-sonli umumiy o'rta ta'lim maktabi o'qituvchisi

Kalit so'zlar: Bulutli xavfsizlik, faoliyatni cheklash, Sandboxing, konteyner izolyatsiyasi, resurslarni cheklash, tarmoq mikrosegmentatsiyasi, avtomatik bloklash, SOAR, Zero Trust.

Annotatsiya; Mazkur ishda bulutli virtual mashinalarda aniqlangan zararli dasturlarning salbiy ta'sirini kamaytirish va ularning tarmoq bo'ylab tarqalishining oldini olishga qaratilgan faoliyatni cheklash usullari tahlil qilinadi. Tadqiqotda Sandboxing (qumdon muhiti), konteynerlarni izolyatsiya qilish, resurslarni cheklash, tarmoq mikrosegmentatsiyasi hamda avtomatik bloklash mexanizmlarining texnik xususiyatlari va ularning o'zaro integratsiyasi ko'rib chiqilgan. Natijalar shuni ko'rsatadiki, *Zero Trust* tamoyiliga asoslangan mikrosegmentatsiya va mashina o'rganishi modellariga biriktirilgan avtomatik javob qaytarish (SOAR) vositalari zararli faoliyatni bir necha millisoniyalar ichida jilovlash va bulutli infratuzilmaning barqarorligini ta'minlashda yuqori samaradorlik ko'rsatadi.

Zamonaviy axborot xavfsizligi tizimlarida faoliyatni cheklash usullari muhim o'rin tutadi. Dasturiy ta'minot va tarmoq infratuzilmasini himoya qilishda qo'llaniladigan mazkur usullar, ayniqsa taqsimlangan hisoblash muhitlarida, kiberhujumlar va ruxsatsiz kirishlarning oldini olishda samarali vosita hisoblanadi. Ushbu paragrafda sandboxing, container isolation (Docker/Kubernetes), resurs cheklash (cgroups), network micro-segmentation va avtomatik bloklash usullari ilmiy-nazariy jihatdan tahlil etiladi.

Sandboxing — bu dasturiy ta'minotni izolyatsiyalangan muhitda ishga tushirish texnologiyasi bo'lib, uning asosiy maqsadi zararli yoki ishonchsiz kodni tizimning boshqa qismlaridan ajratib saqlashdan iborat. Sandbox muhiti dasturga cheklangan resurslar to'plami va operatsion tizim xizmatlariga kirish imkoniyatini taqdim etadi, lekin to'liq huquqlarni bermaydi. Sandboxing arxitekturasi quyidagi asosiy komponentlarni o'z ichiga oladi: izolyatsiya qatlami (isolation layer), monitoring va nazorat mexanizmlari, shuningdek, cheklangan API interfeyslari. Linux operatsion tizimida sandboxing namespaces, seccomp (Secure Computing Mode) va capabilities mexanizmlari orqali amalga oshiriladi. Seccomp filtrlari yordamida jarayonlarga ruxsat etilgan tizim chaqiruvlari (system calls) ro'yxati aniqlanib, noqonuniy operatsiyalar avtomatik ravishda bloklanadi.

Kubernetes orkestratsiya platformasi konteyner izolyatsiyasini yanada rivojlantiradi. Pod Security Policies (PSP) va ularga o'rin bosgan Pod Security Admission (PSA) mexanizmlari orqali konteyner imtiyozlari, fayl tizimi o'qish/yozish huquqlari va tarmoqqa kirish qoidalari markazlashgan holda boshqariladi. Kubernetes Network Policies esa konteynerlar o'rtasidagi tarmoq trafigini granular darajada filtrlash imkonini beradi. Konteyner izolyatsiyasining asosiy cheklovlari orasida yadro (kernel) bilan umumiy foydalanish masalasi muhim ahamiyat kasb etadi. Virtual mashinalardan farqli o'laroq, konteynerlar host tizimining yadrosini ulashadi, bu esa kernel-level zaifliklar orqali izolyatsiyani buzish imkoniyatini saqlab qoladi. Shu sababli, yuqori xavfsizlik talablari qo'yiladigan muhitlarda gVisor (Google), Kata Containers yoki Firecracker kabi yuksakroq izolyatsiya ta'minlovchi sandbox runtime'lardan foydalanish tavsiya etiladi.

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

Control Groups (cgroups) — Linux kernelining jarayonlar tomonidan ishlatiladigan tizim resurslarini (CPU, xotira, disk I/O, tarmoq o'tkazuvchanligini) boshqarish uchun mo'ljallangan asosiy mexanizmi. Cgroups v1 Linux 2.6.24 versiyasida (2008 yil), takomillashtirilgan cgroups v2 esa 2016 yilda kiritilgan. Resurs cheklashning maqsadlari ikki tomonlama hisoblanadi: birinchidan, tizim resurslarini adolatli taqsimlash (Quality of Service); ikkinchidan, xavfsizlik nuqtai nazaridan DOS (Denial of Service) hujumlarining oldini olish. Cgroups orqali quyidagi cheklovlar o'rnatilishi mumkin: CPU kvotasi va ulushi (cpu.max, cpu.weight parametrlari), xotira chegarasi va swap ishlatilishi (memory.max, memory.swap.max), I/O o'tkazuvchanlik chegarasi (io.max), shuningdek, tarmoq paketlari ustidagi prioritet.

Jadval 1. Kibertahdid aniqlash yechimlarining qiyosiy tahlili

Yechim	Turi	False Positive	Overhead	Evasion zaiflik
AWS GuardDuty	Bulutli SIEM	O'rtacha-Yuqori	Past (serverless)	O'rtacha
Azure Defender	Bulutli XDR	O'rtacha	Past-O'rtacha	O'rtacha
Google Chronicle	Bulutli SIEM	Past-O'rtacha	Past (bulut)	Past-O'rtacha
OSSEC	HIDS (open-source)	Yuqori	O'rtacha	Yuqori
Suricata	NIDS (open-source)	O'rtacha	Yuqori	Yuqori
ClamAV	Antivirus (open-source)	O'rtacha	Past-O'rtacha	Juda yuqori

XULOSA

Xulos qilib aytganda, amaliy misol sifatida Dockerning --cpus va --memory buyruq qatori bayroqchalari cgroups v2 orqali konteyner resurslarini cheklaydi. Kubernetes'da ResourceQuota va LimitRange ob'ektlari namespace darajasida va pod darajasida resurs cheklovlarini boshqaradi. Tadqiqotlar shuni ko'rsatadiki, resurs cheklovlari to'g'ri sozlanmagan hollarda, bitta zaiflik tizimning to'liq resurslarini egallab, xizmatni to'xtatib qo'yishi mumkin. Bu holat 2018-yilda Kubernetes klasterlarida kuzatilgan "noisy neighbor" muammosida yaqqol namoyon bo'lgan. Avtomatik bloklashning muhim muammosi noto'g'ri ijobiy natijalar (false positives) hisoblanadi. Noto'g'ri sozlangan tizimlar qonuniy foydalanuvchilarni ham bloklashi mumkin. Bu muammoni bartaraf etish uchun progressiv bloklash strategiyasi qo'llaniladi: avval ogohlantirish, so'ng vaqtinchalik, keyin esa doimiy bloklash. CAPTCHA va ko'p faktorli autentifikatsiya ham noto'g'ri bloklash holatlarini kamaytiruvchi qo'shimcha mexanizmlar sifatida qo'llaniladi. Yuqorida ko'rib chiqilgan beshta faoliyatni cheklash usuli bir-birini to'ldiruvchi qatlamli himoya arxitekturasini tashkil etadi. Defense-in-Depth tamoyiliga ko'ra, xavfsizlikning yagona qatlami etarli emas — har bir usulning o'z zaif tomonlari mavjud.

Sandboxing dastur darajasida izolyatsiyani ta'minlaydi, lekin host tizimiga yuqori darajadagi kirish imkoni mavjud hollarda yetarli himoya bermaydi. Container isolation infratuzilma darajasidagi ajratishni amalga oshiradi, ammo yadro zaifliklariga nisbatan himoyasiz qolishi mumkin. Cgroups resurs cheklovi DoS hujumlarini kamaytiradi, lekin mantiqiy hujumlarga qarshi samarasiz. Mikro-segmentatsiya lateral harakatlanishni cheklaydi, biroq dastlabki infiltratsiyaning oldini ololmaydi. Avtomatik bloklash esa real vaqt himoyasini ta'minlaydi, lekin false positive muammosiga duch keladi. Zamonaviy xavfsizlik tizimlarida ushbu usullarning kombinatsiyasi qo'llaniladi. Masalan, Kubernetes muhitida: Kata Containers orqali konteyner izolyatsiyasi, cgroups v2 orqali resurs cheklash, Cilium/eBPF orqali mikro-segmentatsiya va Falco orqali avtomatik monitoring va bloklash birgalikda

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.

ishlaydi. Bunday ko'p qatlamli yondashuv yagona texnologiyaga nisbatan ancha mustahkam xavfsizlik mexanizmini vujudga keltiradi.

FOYDALANILGAN ADABIYOTLAR

1. Althobaiti, M. M., & Qureshi, M. S. (2024). *Deep learning approaches for anomaly and malware detection in cloud computing virtual machines: A comprehensive survey*. IEEE Access, 12, 18450–18468.
2. Zhang, Y., Chen, X., & Liu, Z. (2025). *Hybrid CNN-LSTM models for temporal anomaly detection in virtualized cloud environments*. Future Generation Computer Systems, 150, 112–125.
3. Kaur, R., & Singh, S. (2023). *Evaluating Random Forest and Deep Learning architectures for hypervisor-level malware detection*. Journal of Information Security and Applications, 74, 103480.
4. Al-Ahmad, M., & Hassan, S. (2024). *Automated incident response and micro-segmentation in Zero Trust cloud environments*. IEEE Transactions on Cloud Computing, 12(2), 512–525.
5. Kovacs, P., & Varga, T. (2025). *Container isolation and resource throttling mechanisms for ransomware mitigation in Kubernetes*. Journal of Information Security and Applications, 79, 103620.
6. Guzman, A., & Rodriguez, M. (2023). *Dynamic sandboxing and machine learning-driven threat containment in virtualized infrastructures*. Computers & Security, 132, 103340.
7. Sharma, R., & Patel, K. (2025). *Evaluating SOAR platforms for hypervisor-level automated malware containment*. Cybersecurity and Infrastructure Protection, 9(1), 88–104.
8. National Institute of Standards and Technology (NIST). (2024). *Zero Trust Architecture and Network Micro-Segmentation Guide for Cloud Systems* (NIST SP 800-207A).
9. Chen, H., Wang, J., & Li, D. (2024). *Behavioral analysis of cloud malware using attention-based LSTM networks*. *Computers & Electrical Engineering*, 114, 109052.
10. Smith, J. A., & Brown, L. K. (2023). *Ensemble learning methods: Combining Random Forest and Gradient Boosting for intrusion detection in cloud VMs*. *Journal of Network and Computer Applications*, 210, 103534.
11. Patel, V., & Kumar, R. (2025). *Lightweight convolutional neural networks for real-time memory introspection and malware classification in hypervisors*. *IEEE Transactions on Dependable and Secure Computing*, 22(3), 1450–1464.
12. Kumar, A., & Gupta, S. (2022). *Comparative analysis of machine learning algorithms for anomaly detection in cloud data centers*. *International Journal of Information Security*, 21(4), 815–830.
13. Rodriguez, E., & Gomez, F. (2024). *Resource throttling and CPU quota enforcement against cryptojacking attacks in multi-tenant clouds*. *Cloud Computing Review*, 18(2), 101–115.

© Ushbu material Creative Commons Attribution 4.0 International (CC BY 4.0) litsenziyasi asosida tarqatiladi.