УДК 004.414.23.056.5:004.491

МОДЕЛЬ ПРОГРАМНИХ ВІДМОВ І КІБЕРАТАК У СЕМАНТИЦІ СЕРВІСНОЇ ЖИВУЧОСТІ ІНФОРМАЦІЙНИХ СИСТЕМ

Рубан І.В., д.т.н, професор[1]; Ткачов В.М., к.т.н., доцент[2]
[1,2] Харківський національний університет радіоелектроніки
[1,2] Україна, Харків
[1] ihor.ruban@nure.ua; [2] vitalii.tkachov@nure.ua

Анотація. *Сформовано сервісноорієнтовану модель, що пов'язує програмні відмови й кібератаки зі змінами показників сервісу та межами живучості; подано узгоджений словник «подія – зсув індикаторів – порушення». Стендові сценарії підтвердили вихід за вимоги та ефект підсилення при накладанні; практична придатність – формування вимог/стендів, порівняння сценаріїв і підготовка політик відновлення.*
Ключові слова: *інформаційна система, живучість, модель, програмний збій.*

Вступна частина. Сервісні властивості інформаційних систем (доступність, затримка у виконанні покладених функції, частка помилок) на практиці порушуються не лише через програмні відмови (непередбачуване завершення роботи, помилки взаємодії, логічні збої), а й через кібератаки (виснаження ресурсів/DoS, порушення цілісності, ін'єкції). У більшості відомих підходів ці два джерела розглядаються окремо, тож вплив подій на цільові показники сервісу (SLO) описано фрагментарно [1-3]. Бракує компактної сервісно-орієнтованої семантики, яка однаковими правилами пов'язує подію типу «відмова/атака» зі зміною вимірюваних індикаторів сервісу (SLI) та породженими ними дефіцитами SLO. Також відсутня узгоджена таксономія відповідностей «подія → зсув індикаторів → дефіцит» і мінімальний набір інваріантів, що задають межі сервісної живучості. Це ускладнює порівнянність результатів, формалізацію умов «допустимої області» роботи сервісу та подальше використання моделі в аналізі/верифікації.

Мета роботи – побудувати інтегровану модель у семантиці сервісної живучості, яка формально задає SLI/SLO та інваріанти «допустимої області» для сервісу, описує класи програмних відмов і кібератак як події з параметрами (інтенсивність, тривалість, вікно дії) та яка вводить однозначне мапування «подія → зсув індикаторів → дефіцит SLO» для кожного класу.

Основна частина. Позначення та сервісна семантика, які будуть використані в дослідженні:

1. Об'єкт. Під об'єктом розуміється один прикладний сервіс інформаційної системи, для якого відстежуються вимірювані показники якості обслуговування в часі.

2. SLI. Стан сервісу описується вектором $s(t) = \mu(t), \vartheta(t), \varsigma(t)$, де $\mu(t)$ – затримка; $\eta(t)$ – частка помилок; $\varsigma(t)$ – доступність. За потреби додаються інші індикатори (пропускна здатність при потоковій обробці, квантилі затримки тощо).

3. SLO. Для кожного індикатора i фіксується цільова умова у вигляді нерівності $g_i\ s(t)\ \leq 0.$

4. Допустима область (інваріант сервісної живучості) – $\Omega = s : g_i\ s(t)\ \leq 0\ \forall i$. Перебування $s(t)$ в Ω означає виконання всіх SLO; вихід за межі Ω трактується як порушення сервісної живучості.

5. Дефіцит SLO для окремого індикатора – $d_i(t) = \max\ 0, g_i\ s(t)$. Якщо умова для індикатора виконується, то $d_i(t) = 0$. Якщо порушена, то – $d_i(t) > 0$ і чисельно відбиває величину відхилення від межі.

6. Агрегований дефіцит. Узагальнена міра порушення визначається як

$$y(t) = \sum_i \omega_i d_i(t), \quad \omega_i \geq 0, \sum_i \omega_i = 1, \tag{1}$$

де значення ω_i задають відносну важливість індикаторів для конкретного сервісу. За потреби застосовується попереднє нормування $\frac{d_i(t)}{\eta_i}$ (з еталонними η_i), щоб узгодити одиниці виміру.

7. Часові характеристики порушень. Для подальших прикладів доцільно зафіксувати дві допоміжні величини: час поза допустимою областю $T_{out} = \int \mathbf{1}_{\ y(t)>0}\ dt$ та інтегральний дефіцит $D = \int y(t)dt.$

8. Припущення спостережуваності. Індикатори $s(t)$ вважаються доступними як агреговані виміри сервісу (телеметрія/журнали), синхронізовані в єдиній часовій шкалі.

Далі вважаємо, що подія впливає на вимірювані індикатори μ, ϑ, ς і тим самим породжує дефіцити SLO. Доцільно ввести наступні позначення зсувів: $\mu\uparrow$, $\vartheta\uparrow$, $\varsigma\downarrow$.

До класів програмних відмов Pr_n належать: Pr_1 (аварійне завершення сервісу), Pr_2 (відсутність коректного прогресу без явного падіння), Pr_3 (помилки узгодження інтерфейсів, черги), Pr_4 (витік пам'яті/дескрипторів, некоректні пули), Pr_5 (логічно хибні результати за формально «успішних» викликів), Pr_6 (помилки конфігурації, невдала міграція/оновлення).

До класів кібератак Cb_m належать: Cb_1 (об'ємні, протокольні та прикладні перевантаження), Cb_2 (навмисний запуск «важких» шляхів), Cb_3 (підміна даних), Cb_4 (внесення збоїв у ланцюг обробки), Cb_5 (вибіркове глушіння / переривання трафіку).

Кожна подія описується кортежем:

$$e = \langle клас, L, W, q, k(\bullet), \psi \rangle, \tag{2}$$

де L – локалізація (компонент, ланцюг викликів, мережевий сегмент); $W = t_s, t_e$ – вікно дії (початок, кінець; тривалість Δ) та режим (одноразова / бурст / періодична / рецидивна); q – інтенсивність / навантаження події (запити/с, частка дропів тощо); $k(\bullet)$ – темпоральний профіль впливу, що визначає, як саме зсуваються SLI протягом W; ψ – зміст «навантаження/пейлоаду» для подій типу Pr_5, Cb_3 (який саме шаблон/дані спричиняють $\vartheta\uparrow$.

Якщо події перекриваються у часі, їхній ефект на SLI за замовчуванням адитивний (підсумовуємо внески у μ та ϑ) з насиченням по межах; для доступності використовують «мінімум» (гірший стан домінує).

Будь-яка подія (2) змінює вимірювані індикатори сервісу $s(t)$. Цей зсув перетворюється правилами SLO на дефіцити $d_i(t)$, які далі зводяться в узагальнену міру порушення $y(t)$. В такому випадку доцільно скласти механізм мапування:

1. Базова траєкторія. Позначимо «нормальну» (без подій) траєкторію $s_0(t)$.

2. Профіль впливу події. Подія e задається кортежем (2).

3. Зсув індикаторів. Вплив події на SLI подаємо як:

$$s_e(t) = k_e * u_e(t) \odot \alpha_e, \tag{3}$$

де $u_e(t)$ – індикатор вікна дії W; * – згортка; $\alpha_e = [\alpha_\mu, \alpha_\vartheta, \alpha_\varsigma, \ldots]$ – «напрям» впливу (типово $\alpha_\mu, \alpha_\vartheta \geq 0$; $\alpha_\varsigma \leq 0$).

4. Композиція подій. Для множини подій $\mathcal{E}$:

$$\tilde{s}(t) = s_0(t) + \sum_{e \in \mathcal{E}} \Delta s_e(t), \quad s(t) = \prod \tilde{s}(t), \tag{4}$$

де $\prod \bullet$ – насичувальне «обрізання» за фізичними межами (наприклад, $\varsigma \in 0{,}1$, $\vartheta \in 0{,}1$, $\mu \geq 0$). У перекриттях впливи адитивні, для доступності домінує гірший стан (насичення вниз).

5. Перехід до дефіцитів. Для кожного SLO $d_i(t) = \max\ 0, g_i\ s(t)$. Це дає єдину мову: «подія → зсув SLI → дефіцит SLO».

Агрегування порушень як узагальнені міри на підставі вищенаведеного можна задати наступним чином:

1. Агрегований дефіцит у момент часу задається як (1).

2. Часова міра «поза допустимою областю»: $T_{out} = \int_t^{t+H} \mathbf{1}_{y(\tau)>0}\, d\tau$, що показує сумарний час порушень на горизонті H.

3. Інтегральний дефіцит на горизонті: $D(H) = \int_t^{t+H} y(\tau) d\tau$, який узагальнює «площу» порушень (і частоту, і глибину).

4. Піковий дефіцит: $y_{\max}(H) = \max_{\tau \in t, t+H} y(\tau)$, корисний для фіксації найгірших епізодів.

Важливо, що модель не нав'язує оптимізацій чи алгоритмів виявлення – вона дає єдину семантику для опису наслідків різних подій у координатах сервісу.

Далі наводиться приклад застосування запропонованої моделі.

Вихідні дані:

– SLI: 95-й квантиль затримки μ_{95} (мс), частка помилок ϑ (%), доступність ς;

– SLO: $\tau_{\max}$ =150 мс; $e_{\max}$ = 1%; $a_{\min}$ = 0.995;

– база (без подій): $\mu_{95} \approx 120$ мс, $\vartheta \approx 0.3\%$, $\varsigma \approx 0.999$.

Характеристики події № 1 (DoS-бурст, 0-90 с). Ступінчасте перевантаження дає зсуви: $\Delta\mu_{95} = +80$ мс, $\Delta\varsigma = -0.010$. Отже, у вікні дії: $\mu' = 200$ мс $\Rightarrow$ дефіцит затримки $d_\mu = 200 - 150 = 50$ мс; $\varsigma' = 0.989 \Rightarrow$ дефіцит доступності $d_\varsigma = 0.995 - 0.989 = 0.006$.

Характеристики події № 2 (семантичні помилки, 60-90 с, накладається). Ядро впливу - імпульсний/бурстовий профіль у логіці обробки; зсув: $\Delta\vartheta = +2.5\%$. У перекритті: $\vartheta' = 2.8\%$ $\Rightarrow$ дефіцит помилок $d_\vartheta = 2.8\% - 1.0\% = 1.8\%$.

Далі розглянемо механізм агрегації. Беремо ваги важливості SLO для цього сервісу: $\omega_\mu = 0.4$, $\omega_\varsigma = 0.4$, $\omega_\vartheta = 0.2$; попередньо нормуємо дефіцити (щоб звести різні одиниці) на еталони $\eta_\mu = 150$ мс, $\eta_\varsigma = 0.005$, $\eta_\vartheta = 1\%$ з урахуванням (1):

$$y(t) = \sum_i \omega_i \frac{d_i(t)}{\eta_i}.$$

Для 0-60 с (тільки DoS): $y \approx 0.4 \cdot \frac{50}{150} + 0.4 \cdot \frac{0.006}{0.005} = 0.613$. Для 60-90 с (DoS + семантичні помилки): додатково $0.2 \cdot \frac{1.8\%}{1\%} = 0.36 \Rightarrow y \approx 0.973$. Похідні міри на горизонті $H = 5$ хв. Час поза допустимою областю: $T_{out} = 90$ с (у решту часу $y = 0$). Інтегральний дефіцит: $D \approx 60\text{ с} \cdot 0.613 + 30\text{ с} \cdot 0.973 \approx 66$ «нормованих секунд». Пікове порушення: $y_{\max} \approx 0.973$ (в період перекриття подій).

Таким чином, навіть короткий DoS переводить систему за межі SLO доступності, а його перекриття із семантичними помилками різко підсилює загальний дефіцит. Найбільший внесок у D дав дефіцит доступності (через нормування на жорсткий $a_{\min}$); отже, для цього сервісу саме доступність є «вузьким місцем». Модель дала прозорий «ланцюг» «подія → зсув SLI → дефіцит SLO → узагальнені міри» без жодних алгоритмів керування – рівно те, що потрібно для подальшої валідації/порівняння сценаріїв.

Висновки. У роботі побудовано компактну сервісно-орієнтовану модель, що єдиним формалізмом пов'язує класи програмних відмов і кібератак з вимірюваними індикаторами сервісу та породженими дефіцитами SLO. Запропоновано узгоджений «словник» відповідностей «подія → зсув SLI → дефіцит SLO», правила композиції перекривних подій і узагальнені міри порушень $y(t)$, T_{out}, D, $y_{\max}$. Модель не нав'язує алгоритмів виявлення чи оптимізації, а слугує базою для специфікації вимог, формування стендів і коректного порівняння сценаріїв «відмова/атака» в координатах сервісної живучості. Задекларовані припущення та межі застосовності роблять її придатною для подальшої валідації й інтеграції з політиками відновлення у реальних інформаційних системах, до яких висуваються вимоги забезпечення високої живучості.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Tkachov, V., Kovalenko, A., Kuchuk, H., Ni, I. (2021). Method of ensuring the survivability of highly mobile computer networks. *Advanced Information Systems*, 5(2), 159-165. https://doi.org/10.20998/2522-9052.2021.2.24.

[2] Додонов, О.Г., Ланде, Д.В. (2021). Мережева модель структурної живучості. *Реєстрація, зберігання і обробка даних*, 23(1), 15-22. https://doi.org/10.35681/1560-9189.2021.23.1.235075.

[3] Doukas, N. et al. (2022). Survivability Using Artificial Intelligence Assisted Cyber Risk Warning. *In: Stamp, M., Aaron Visaggio, C., Mercaldo, F., Di Troia, F. (eds) Artificial Intelligence for Cybersecurity. Advances in Information Security*, vol 54. Springer, Cham. https://doi.org/10.1007/978-3-030-97087-1_12

Ruban I.V., Tkachov V.M.

Model of Software Failures and Cyberattacks in the Service-Level Survivability Semantics of Information Systems

Abstract. *Developed a service-oriented model that links software failures and cyberattacks to shifts in service indicators and survivability boundaries, and provides a unified lexicon of "event–indicator shift–violation". Testbed scenarios confirmed requirement breaches and amplification under overlapping events; practical applicability includes requirements/testbed specification, scenario comparison, and preparation of recovery policies.*

Keywords: *information system; survivability; model; software failure.*