УДК 621.372

О.Ф.Нікулін, С.О.Нікулін

АЛГОРИТМИ РОБОТИ ЗАСОБІВ ЗАХИСТУ СПЕЦІАЛІЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ

Система захисту спеціалізованої системи управління є досить складною, а процеси, що в ній реалізуються, в багатьох випадках функціонують паралельно один до одного. Тому представити систему управління або процес функціонування всієї системи захисту SZ, в цілому, як деякий загальний процес, що природним чином може розглядатися як послідовний в часі недоцільно. У зв'язку з цим перш за все необхідно розробити алгоритм функціонування окремих складових систем, яким можна прописати певну функціональну орієнтацію, а потім об'єднати всі складові системи і описати їх сумісне функціонування на більш зональному рівні, синхронізуючи процеси кожної складової системи таким чином, щоб така синхронізація допускала несуперечну інтерпретацію всіх процесів в предметній області системи захисту. Відомо, що алгоритм роботи системи можна описувати різними способами, наприклад, у вигляді функціональних блок-схем, в яких використовуються функціональні блоки та блоки аналізу ситуацій, які визначають різні варіанти продовження функціонування відповідного алгоритму у вигляді фрагментів програмної реалізації окремих складових систем, у вигляді формального опису математичних моделей відповідних процесів і т. д. [1, 2].

Формальний опис математичними моделями процесів функціонування окремих фрагментів SZ є досить проблематичним, оскільки особливістю предметної області, в якій описується SZ, є те, що в ній існує досить багато компонент, які формально описати з необхідною точністю їх відображення неможливо, по крайній мірі, в рамках одного математичного апарату. Тому використання математичних моделей для опису окремих фрагментів функціонування SZ недоцільне і в більшості випадків неможливе. Використання програмно подібних методів для опису алгоритмів функціонування вимагає використання певної мови програмування і через такий опис буде менш доступний для осіб, які б хотіли скористатися відповідними алгоритмами при проведенні простих робіт по створенню відповідних фрагментів системи захисту. Більш того, така форма опису є візуально менш придатна для сприйняття логіки функціонування відповідного алгоритму, оскільки в цьому випадку логіка описується окремими командами мови, які відображають логічні зв'язки у вигляді адрес, на які передається управління в тому чи іншому випадку. Якщо логіка алгоритму є досить складна, то прослідкувати відповідні логічні зв'язки в цілому при використанні такої форми опису є досить важко.

Тому вданому випадку будемо використовувати градову форму опису алгоритмів функціонування окремих систем, однією з різновидностей якої є блок-схеми. Така форма опису алгоритмів є максимально візуалізована, що суттєво полегшує її сприйняття. В рамках таких засобів можна описувати алгоритми з різною мірою деталізації, оскільки окремі, зрозумілі функції розміщаються в окремих функціональних блоках. Від такої форми опису досить легко переходити до його програмної реалізації, оскільки вона не залежить від вибраної мови програмування, яка вибирається як базова для опису алгоритму. Розглянемо алгоритм функціонування системи захисту OZ від атак зі сторони мережі, яка складається з модулів:

- Zh_i — мобільний засіб захисту типу honeypot,
- AZh_i — модуль адаптації засобу захисту Zh_i,
- $ARZh_i$ — модуль аналізу процесу функціонування засобу захисту Zh_i.

В цілому, таку систему будемо позначати символом MZH. Перш ніж проектувати відповідну функціональну блок-схему, приймаючи до уваги, що MZH складається з кількох окремих модулів, необхідно визначитися з мірою деталізації відображення з кожного блоку у відповідній блок-схемі даної системи. Необхідність аналізу цього аспекту при побудові системи MZH обумовлюється наступними причинами:

- різні блоки, що входять до складу MZH реалізують функції, що мають різну міру оригінальності, яка визначається задачею, що є основним предметом дослідження,
- при проектуванні алгоритмів у вигляді блок-схеми необхідно забезпечити останні можливістю детально зображати найбільш складні фрагменти процесу функціонування відповідної частини алгоритму,
- оскільки складові системи використовують в процесі свого функціонування інформацію від інших складових систем, то в точках відповідних входів необхідно більш детально відобразити процеси аналізу або способу використання зовнішньої інформації для відповідної системи.

При проектуванні програмних засобів, особливо при проектуванні складних систем рекомендується не обмежуватися блок-схемами, які в певній мірі є зональними, а рекомендується використовувати різні діаграми, що відображають процеси, що повинні відбуватися в системі в достатньо повній мірі [3, 4]. До таких діаграм відносяться діаграми часу, функційні діаграми класу і т. д. Відповідні способи відображення дозволяють у традиційній формі описати всі особливості процесу функціонування системи, що є надзвичайно важливим при формуванні відповідної проектної документації.

У випадку проектування алгоритму системи MZH базовим блоком, що відображають засоби захисту типу honeypot, являються найбільш оригінальними. Тому їх робота в рамках алгоритму буде описуватися максимально детально. В рамках алгоритму AZh_i буде описуватися максимально детально. В блоках Zh_i і $ARZh_i$ більш детально будуть відображатися ті

фрагменти, які відносяться до менш відомих методів реалізації відповідних функцій та фрагменти, що представляють собою в певній мірі оригінальні розв'язки. Функціональна блок-схема алгоритму складової системи виявлення атак зі сторони мережі MZH приведена на рис.4.2.

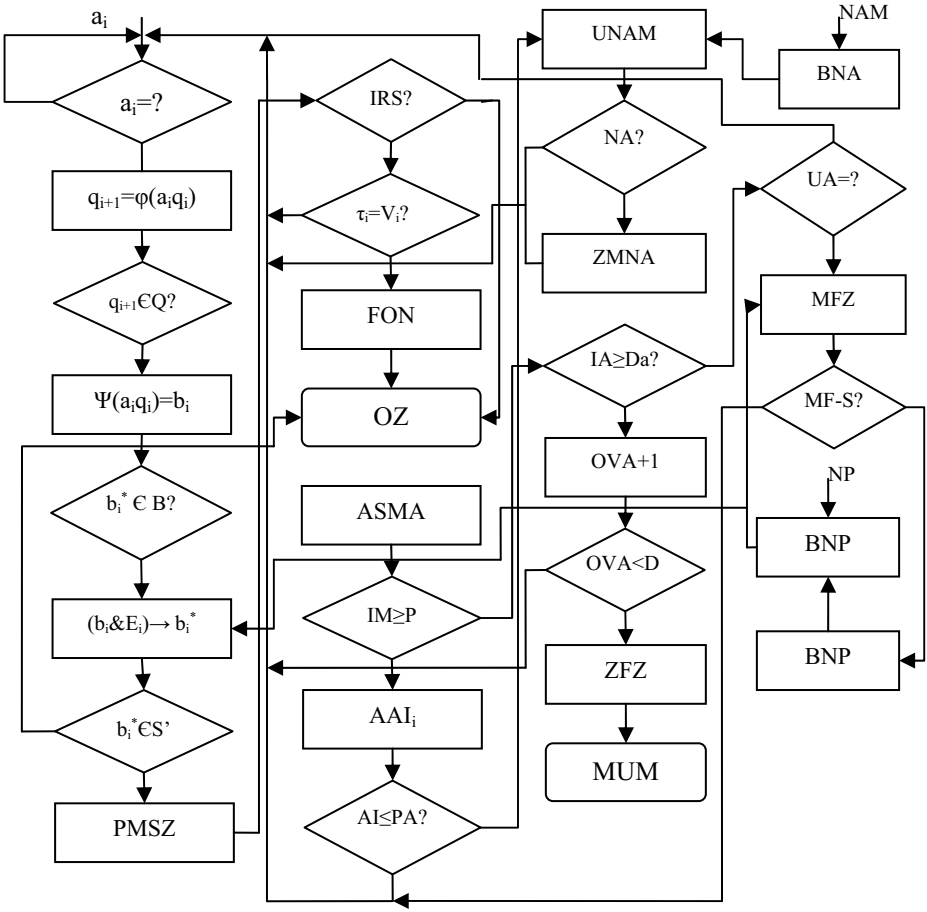

Рис. 4.2. На рисунку приведена блок-схема алгоритму складової системи захисту OZ від мережевих атак

На рис.4.2 використовуються наступні скорочення та позначення:

- a_i — вхідна компонента або пакет, поступаючими в мережі як об'єкт захисту формула, визначення зміни внутрішнього стану або внутрішнього параметру Zh_i в результаті аналізу a_i, що еквівалентно розпізнаванню параметра вхідного потоку,
- $q_{i+1}=\varphi(a_iq_i)$ — розпізнавання чи параметр відноситься до можливих

описів атак,

- $\Psi(aiqi)=bi$ — визначення параметру, який в рамках автоматної інтерпретації являється елементом вихідного алфавіту автомату, а інтерпретації виявлення атак являється зовнішньою реакцією Zh_i на розпізнаваний елемент атаки a_i,
- b_i Є B — перевірка, чи визначений параметр, є допустимою в Zh_i вихідною компонентою,
- $(b_i\&E_i)= b_i^*$ — визначення b* як елемента атаки,
- PMSZ — збереження відповідного елемента атаки в магазинній пам'яті автомату або в пам'яті Zh_i,
- IRS? — перевірка чи виявлений елемент атаки є її останньою компонентою,
- $\tau_i=V_i$? — перевірка, чи необхідно формувати імітацію реакції системи, що атакується для небезпеки, що ініціювала відповідну атаку,
- FOV — формування повідомлення до джерела пакету, яким атакована система, що імітує необхідну реакцію на атаку небезпеки,
- OZ — спеціалізована система управління, що захищається з точки зору взаємодії останньої з Zh_i,
- ASMA — аналіз стану магазинної пам'яті Zh_i,
- IM≥P — перевірка міри активності Zh_i, що здійснюється на основі аналізу магазинної пам'яті, в якій зберігаються виявлені атаки,
- AAI_i — аналіз опису фрагменту текучої атаки, що виявлена, засобом захисту Zh_i,
- AI≤PA — перевірка чи фрагмент опису атаки, що знаходиться в пам'яті Zh_i, завершує опис атаки,
- UNAM — визначення активності відповідної атаки та видалення з магазинної пам'яті Zh_i атаки, яка є недостатньо активна,
- NA? — чи наступна аналізована атака є новою?
- ZMNA — запис в магазин Zh_i нової атаки?
- IA≥Da? — чи достатньо багато поступало з мережі атак аналізованого типу ?
- OVA+1 — збільшення кількості активних атак в пам'яті засобу захисту Zh_i,
- OVA<D — перевірка, чи міра активності є достатньою, щоб Zh_i не повинен був емігрувати,
- ZFZ — формування даних про активність атак в точці простору, де знаходиться Zh_i,
- MUM — передача даних про умови необхідності міграції в модуль управління міграцією.
- BNA — блок формування нових описів атак на основі даних, що сформовані в блоці моделювання атак небезпеками,
- UA=? — перевірка, чи усунена або додана атака приводить до

необхідності модифікації Zh_i,

- MFZ — блок модифікації Zh_i,
- MF-S — перевірка, чи проведена модифікація Zh_i не привела до виникнення суперечностей,
- BAVS — блок аналізу й усунення суперечностей,
- NP — дані про аксіоми правил перетворень, що передаються з боку базових алгоритмів аналізу подій, що використовуються при розпізнаванні атак на модифікації засобів захисту.

Розглянемо алгоритм функціонування складової системи моніторингу засобів захисту. Відповідна блок-схема приведена на рис.4.3.

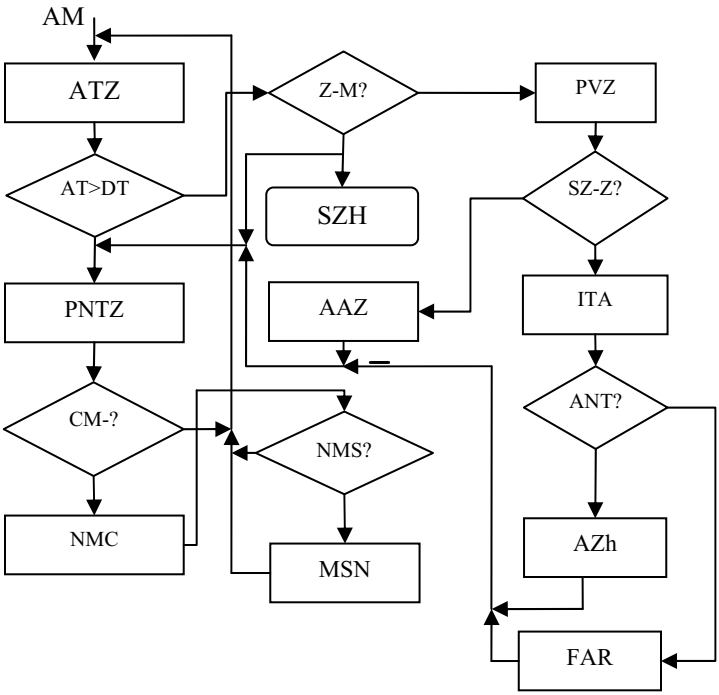

Рис. 4.3. Блок-схема алгоритму функціонування складової системи моніторингу засобів захисту

На рис.4.3 використовуються наступні скорочення та позначення:

- AM — активізація системи моніторингу, яка здійснюється системою управління SUZH,
- ATZ — блок аналізу засобу захисту, який знаходиться у вибраній точці простору мережі, що відповідає певному адресу входу системи, що

охороняється,

- Формула — перевірка, чи рівень активності засобу захисту в текучій точці простору більший від заданого мінімально допустимого порогу,
- PNTZ — перехід системи моніторингу до наступного засобу захисту, який знаходиться в іншій точці простору мережі, що захищається,
- CM? — перевірка, чи завершився один цикл моніторингу,
- MNS — аналіз результатів одного циклу моніторування та визначення міжциклового інтервалу.
- MNS? — визначення, чи необхідна модифікація стратегії моніторування,
- MSM — модифікація стратегії моніторування засобів захисту Zhi,
- Z-M? — визначення, чи засіб захисту Zhi потребує модифікації в межах циклу моніторингу,
- SZH — система управління захистом у частині, яка приймає участь у функціонуванні системи моніторингу,
- AZZ — архівізація засобу захисту, активність якого менша заданого рівня,
- PVS — реалізація процедури вигасання засобу захисту, активність якого є нижча від заданого порогу,
- SZ-Z? — визначення, чи засіб захисту потрібний системі захисту у випадках, коли їх достатньо багато,
- IIA — інсталяція нового засобу захисту, що орієнтований на інший тип атаки,
- ANT — визначення, чи адрес точки розміщення визначено,
- AZh — активізація засобу захисту по новому адресу,
- FAR — формування віртуального адресу для засобів захисту на основі аналізу типу атаки, на яку орієнтований інстальований засіб захисту.

Наступною системою, що входить до складу системи захисту є система моделювання небезпек. Вона складається з двох блоків — блоку моделювання небезпек та блоку аналізу та виявлення загроз об’єкту, який охороняється. Блок моделювання небезпек представляє собою в певному розумінні модель небезпеки, при цьому її частину, що представляє собою інформаційну систему, за допомогою якої формуються і ініціюються атаки на інші інформаційні системи. В більшості випадків небезпека в повному розумінні цього терміну включає наступні компоненти:

- інформаційні засоби, можливо, спеціалізовані, за допомогою яких формується сам інтруз, відсилається в мережу з ціллю здійснення атаки на визначену інформаційну систему,
- користувач, який зацікавлений у несанкціонованому втручанні в систему, яку передбачається атакувати відповідним інтрузом,
- інформаційна система, яка обумовлює зацікавленість користувача у несанкціонованому втручанні у систему, яку передбачається атакувати.

Очевидно, що можливі випадки, коли небезпека визначається тільки

користувачем, випадки, коли небезпека представляє собою випадковий фактор, випадки, коли небезпека є опосередненою та цілий ряд інших можливостей. В даному випадку обмежимося випадком, який визначає небезпеку приведеними вище компонентами.

Будь-яка модель передбачає певний опис або певне відображення деякого процесу або ряду процесів, стосовно яких передбачається проводити ті, чи інші дослідження. В даному випадку такими процесами є наступні:

- процес проектування інтруза,
- процес визначення цілі атаки,
- процес виявлення або визначення засобів реалізації атаки, яка може здійснюватися інтрузом, який проектується.

Визначення цілі атаки в рамках даного підходу ґрунтується на існуванні інформаційної системи, що є подібною або аналогічною до системи, яку передбачається атакувати. Це означає, що ціль формується на основі існування можливості використання результатів успішної атаки для роботи інформаційної системи аналога. Таку інформаційну систему будемо називати поглинаючою системою. На сучасному етапі розвитку інформаційних технологій для ініціації процесу проектування інтруза і для реалізації самого процесу проектування необхідна участь зацікавленого користувача. При цьому користувач може використовувати фахівця у галузі створення інтрузів, яких на сьогоднішній день досить багато, і відомі вони як хакери. Включення до складу моделі небезпеки спеціалізовані інструментальні інформаційні засоби, що орієнтовані на проектування інтрузів та їхню ініціацію або завантаження в мережу є необхідною умовою, оскільки ті чи інші засоби переважно орієнтовані на проектування певних типів інтрузів. Наприклад, відомі інструментальні засоби для реалізації станів для серверів електронної пошти, відомі спеціалізовані генерації системи вірусів і т. д.

Зупинимося більш детально на особливостях проектування інтрузів. Інтруз в багатьох випадках представляє собою досить складну структуру, оскільки він включає в себе цілий ряд програмних компонент, їх використання може представляти собою певну стратегію об’єктами реалізації такого інтруза, крім самої системи, що атакується, можуть бути цілий ряд компонент, що використовуються в цифровій мережі. Наприклад, такими об’єктами можуть бути окремі хости, локальні мережі, комутатори та цілий ряд інших компонент, що забезпечують процес функціонування комп’ютерної мережі.

Тому у випадку, коли сформована ціль атаки на деяку інформаційну систему проектується стратегія її реалізації. В даному випадку обмежимося тільки тими частинами інтрузів, які безпосередньо повинні співпрацювати з елементами системи, що охороняється.

Очевидно, що чим більше факторів включати в склад моделі, що повинна описувати небезпеку для системи, що охороняється, тим більш

адекватним будуть результати моделювання і більш однозначно можна було б виявляти атаки систем і захисту. Але це може привести до неоправданого ускладнення відповідної моделі. Тому в рамках даної роботи модель небезпеки буде обмежуватися наступними компонентами або факторам, що її визначають:

- генератор послідовності подій, що безпосередньо ініціюються програмними засобами, які представляють собою реалізацію частини інтруза, що безпосередньо співпрацює з атакованою системою,
- загрози, що існують у об'єкта охорони і можуть використовуватися для впровадження відповідних інтрузів, що представляють собою окремі програми,
- сукупність компонент або функцій системи, що охороняється, які можуть складати зацікавленість зі сторони аналогічних систем, і тому можуть бути ціллю тієї чи іншої атаки.

Загрози в широкому розумінні цього терміну представляють собою цілий спектр особливостей системи, починаючи від помилок, що допущені в реалізації функціональної частини програм структури системи та закінчуючи недостатньо повно реалізованими засобами захисту. В даному випадку під загрозами будемо розуміти властивості програми, що обумовлені якістю виготовлення системи, в яку включається наявність помилок різних типів. Приймемо, що система захисту формується у відповідності з вимогами до рівня безпеки функціонування системи, що задається як вихідний параметр при її проектуванню, і якщо виявляється, що засоби захисту не відповідають вимогам, то цей випадок не будемо розглядати як загрозу, оскільки останні завжди можна розширити.

Більшість атак, що реалізується по відношенню до компонент в мережі є відомими або представляють собою певну модифікацію відомих атак, оскільки модифікації орієнтовані на використання особливостей кожної системи, то в рамках моделей небезпек можна їх враховувати, що розширяє можливості формування більш адекватних еталонних зразків атак, які необхідно виявляти засобам захисту.

1. *Кнут Д.Э.* Исскуство программирования. Т1. Основные алгоритмы. — М.: Издательский Дом «Вильямс», 2000.
2. *Керниган Б., Пайк Р.* Практика программирования. — СПБ.: Невский Диалект, 2001.
3. *Ройс У.* Управление проектами по созданию программного обеспечения. Унифицированный подход. — М.: «Лори», 2002.
4. *Booch Gready* Object Solutions: Managing the Objekt Oriented Projekt. Addison — Wesli Publishing Company, Menlo Park, California, 1996

Поступила 29.09.2010р.