МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

DOI: 10.25586/RNU.V9187.20.01.P.003 УДК 004.056+519.688

В.А. Минаев, И.Д. Королев, С.А. Коноваленко,
Д.С. Васильев, В.Г. Секунов

СТРУКТУРНО-ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ИМИТАЦИИ КОМПЬЮТЕРНЫХ АТАК НА АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ

На основе методологии системного подхода и функционального моделирования IDEF0 представлена структурно-функциональная модель имитации компьютерных атак на автоматизированную систему, позволяющая обеспечивать возможность выбора наиболее рациональной структуры системы обнаружения, предупреждения и ликвидации их последствий, а также достичь необходимого уровня достоверности имитации. Описаны структура и процесс функционирования подсистемы имитации компьютерных атак, основу которого составляет формализованное представление множества уязвимостей автоматизированной системы в виде упорядоченных наборов тегов (идентификаторов), используемых для задания процедур формирования и конфигурирования набора эксплойтов, применяемых в имитации атак на автоматизированную систему. Делается вывод, что разработанная модель позволяет повысить оперативность процесса тестирования реальных автоматизированных систем путем имитации атак с помощью автоматизации процедур формирования и конфигурирования набора эксплойтов, обеспечить необходимый уровень достоверности процесса имитации атак за счет воспроизводства их различных типов, создавать рациональные системы обнаружения, предупреждения и ликвидации последствий компьютерных атак на автоматизированные системы за счет расширения области синтеза структурно-функциональной модели и ее формализованного описания нужного уровня детализации.

Ключевые слова: структурно-функциональная модель, имитация, компьютерная атака, автоматизированная система, уязвимость, эксплойт.

V.A. Minaev, I.D. Korolev, S.A. Konovalenko, D.S. Vasiliev, V.G. Sekunov

STRUCTURAL AND FUNCTIONAL SIMULATION MODEL OF COMPUTER ATTACKS ON AUTOMATED SYSTEMS

The article presents a structural and functional model of computer attacks simulation on automated systems based on the system approach methodology and functional modeling IDEF0. This model allows to choose the most rational structure of the system for detection, prevention and elimination of computer attacks consequences, as well as to achieve the necessary level of reliability of simulation. The article describes the structure and process of functioning of the computer attack simulation subsystem, which is based on a formalized representation of the set of vulnerabilities of an automated system in the form of ordered sets of tags (identifiers) used to specify the procedures for forming and configuring a set of exploits used in simulating attacks on an automated system. It is concluded that the developed model makes it possible to increase the efficiency of the process of testing real as by simulating attacks based on automation of procedures for forming and configuring a set of exploits, to provide the necessary level of reliability of the process of simulating attacks by reproducing their various types, to create rational systems for detecting, preventing and eliminating the consequences of computer attacks on automated systems by expanding the area of synthesis of the structural and functional model and its formalized description of the desired level of detail.

Keywords: structural and functional model, simulation, computer attack, automated system, vulnerability, exploit.

Введение

Основным условием построения эффективной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак (СОПКА) на автоматизированную систему (АС) является применение системного подхода при представлении сложных организационно-технических систем в виде взаимосвязанных функциональных подсистем, объединяющих в своем составе различные специализированные средства (СС) [8; 13]. Одной из основных функций СОПКА является контроль состояния защищенности АС, основанный на пассивном и активном методах [4; 7].

Модель подсистемы выявления уязвимостей (ПВУ) АС, реализующей пассивный метод контроля, построена в работе [13]. В развитие проведенных исследований в настоящей статье производится синтез модели подсистемы имитации компьютерных атак (ПИКА) на АС, которая обеспечивает реализацию активного метода.

Исследование существующих СС, способных функционировать в составе ПИКА, свидетельствует о том, что процесс имитации компьютерных атак (КА) на АС характеризуется низкой оперативностью и недостаточным уровнем достоверности, что, в свою очередь, обусловлено отсутствием универсального СС, способного на основе данных, предоставляемых ПВУ АС, в автоматизированном режиме формировать и проводить множество различных типов КА на АС [6; 7; 12; 24].

Стоит также отметить, что до настоящего времени в рамках решения задачи по построению рациональной структуры СОПКА на АС уделялось недостаточное внимание формализации конкретных вариантов ПИКА, дающей возможность облегчить их практическую реализацию. Данная статья восполняет существующий пробел в формальном описании модели ПИКА на АС, имея практическую направленность в целом на повышение эффективности СОПКА.

Построение модели ПИКА на АС осуществим в два этапа:

Первый этап – синтез структурной модели ПИКА.

Второй этап – описание процесса функционирования ПИКА.

Синтез структурной модели имитации компьютерных атак

Синтез структурной модели ПИКА (рис. 1) осуществим на основе концептуальной модели системы комплексного контроля защищенности АС, представленной в [7], посредством определения функциональных элементов (модулей), входящих в ее состав, и задания связей между ними, позволяющих устранить следующие недостатки, присущие процессу функционирования типовых СС имитации КА на АС:

1. При воздействии типового СС имитации КА на АС высока вероятность ее отказа.

2. Достаточно высока сложность конфигурирования типового СС имитации КА на АС и формирования пользовательских сигнатур КА.

3. Принятие решения о выборе типа КА на АС, планируемой к имитации, во многом зависит от уровня квалификации специалиста по информационной безопасности (ИБ) и в достаточной мере субъективизировано [6; 7; 12; 24].

В структурной модели ПИКА представлены следующие компоненты [1; 2; 9; 10; 15; 16; 17; 18; 19; 20; 21; 22; 23; 25; 27]:

1. База данных (БД) эксплойтов – предназначена для хранения множества эксплойтов, представляемых в виде тегов (идентификаторов) и программных кодов, написанных на одном из языков программирования (например, C/C++, Python, Ruby, PHP).

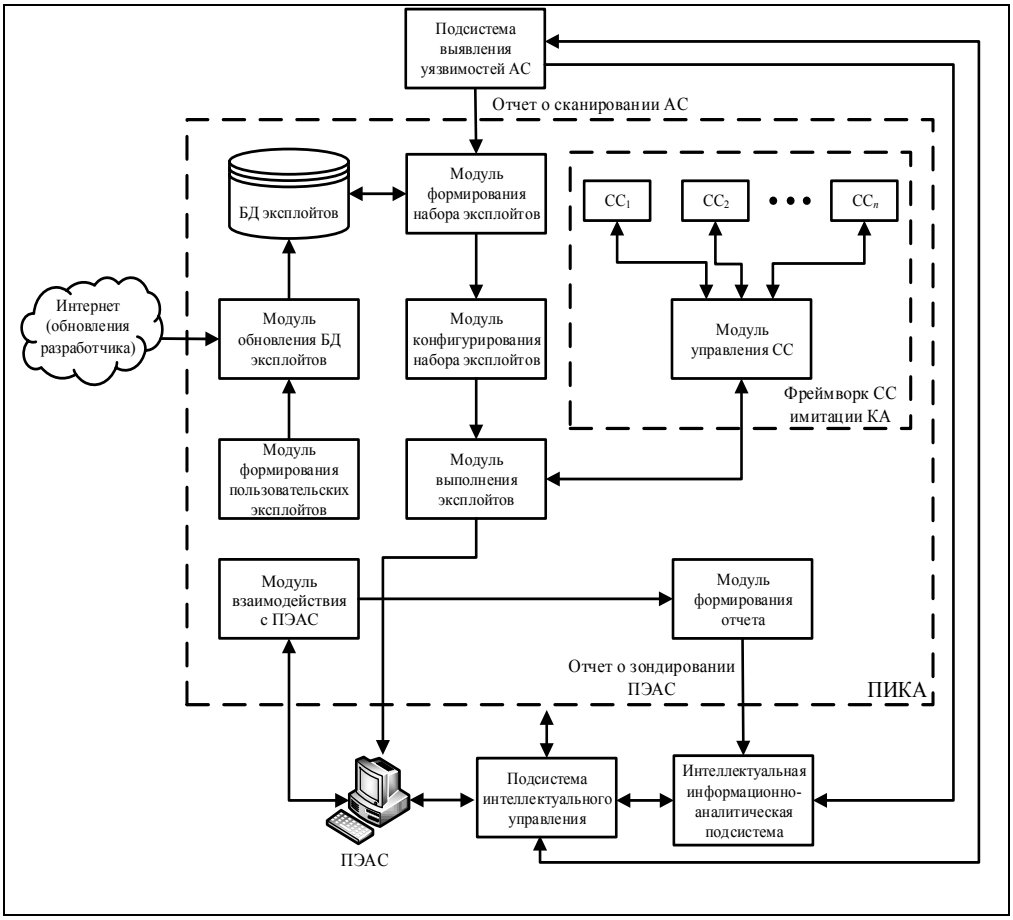

Рис. 1. Структурная модель ПИКА

2. Модуль формирования пользовательских эксплойтов – предназначен для создания нестандартных эксплойтов, не содержащихся в БД, но необходимых специалисту по ИБ для реализации всех процессов имитации КА на подсистему эмуляции АС (ПЭАС).

3. Модуль обновления БД эксплойтов – предназначен для получения обновлений, поступающих от разработчика СС имитации КА и специалиста по ИБ, с последующим их внесением в БД эксплойтов.

4. Модуль взаимодействия с ПЭАС – предназначен для установления надежной взаимосвязи с ПЭАС на программном уровне, а также для мониторинга ее защищенности в момент реализации процесса имитации КА и после его завершения.

5. Модуль формирования набора эксплойтов – предназначен для автоматического формирования набора эксплойтов, планируемых к имитации на ПЭАС, на основе отчета о сканировании АС, предоставляемого ПВУ, и множества эксплойтов, хранящихся в БД.

6. Модуль конфигурирования эксплойтов – предназначен для автоматического формирования упорядоченного набора эксплойтов.

7. Модуль выполнения эксплойтов – предназначен для зондирования ПЭАС с помощью СС имитации КА, входящих в состав фреймворка и реализующих соответствующие эксплойты, включенные в их упорядоченный набор.

8. Фреймворк СС имитации КА – состоит из 1) резидентной части, представленной в виде модуля управления всеми СС имитации КА, их конфигурирования и запуска, а также осуществления взаимодействия с модулем выполнения эксплойтов; 2) точек расширения, представленных в виде совокупности узкоспециализированных СС имитации КА, предназначенных для реализации эксплойтов определенного типа.

9. Модуль формирования отчета – предназначен для построения отчета о состоянии защищенности ПЭАС вследствие проведенной имитации КА.

10. Такие подсистемы, как ПВУ АС, ПЭАС, подсистема интеллектуального управления, интеллектуальная информационно-аналитическая подсистема, – предназначение определено в [7].

Синтез функциональной модели имитации компьютерных атак

Описание процесса функционирования ПИКА осуществим при следующих допущениях:

- БД эксплойтов является актуальной на момент реализации процесса имитации КА на ПЭАС;
- процессы функционирования модулей обновления БД эксплойтов, формирования пользовательских эксплойтов, выполнения эксплойтов, взаимодействия с ПЭАС, формирования отчета и фреймворка СС имитации КА полностью обеспечивают выполнение своих функций и детально не рассматриваются в статье.

С учетом указанных допущений в рамках описания процесса функционирования ПИКА на основе методологии функционального моделирования IDEF0 [11], а также с учетом требований, предъявляемых к проведению зондирования ПЭАС и формированию отчета о его выполнении [3; 4; 5], построим ее функциональную модель, представленную в виде диаграмм узлов А-0 и А-1 (рис. 2–3).

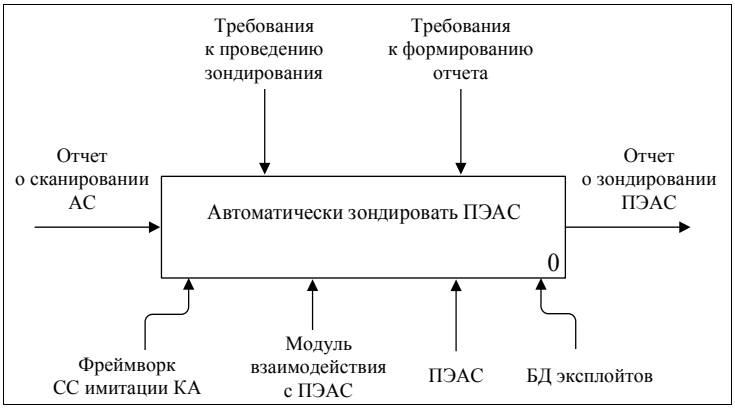

Рис. 2. Диаграмма «Автоматически зондировать ПЭАС», узел А-0

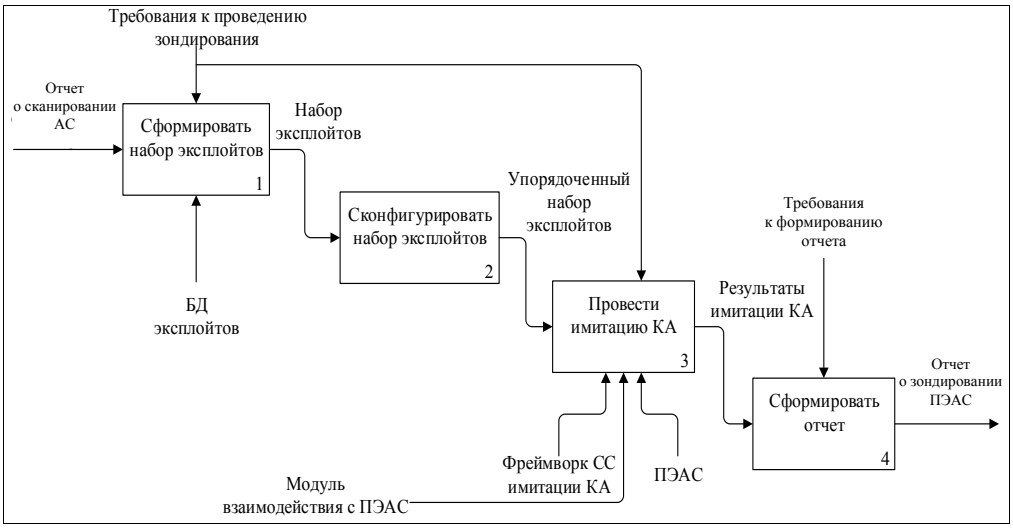

Рис. 3. Диаграмма «Автоматически зондировать ПЭАС», узел А-1

На основе функциональной модели (см. рис. 2–3) в целях дальнейшей ее формализации определим исходные данные, необходимые для моделирования:

1. Отчет о сканировании АС, предоставляемый ПВУ и содержащий множество уязвимостей, обнаруженных в АС, R [1; 2; 19]:

$$R = \left\{ \Omega_i \mid i = \left(\overline{1, \alpha} \right) \right\}, \tag{1}$$

где Ω_i – произвольная уязвимость АС; α – общее количество уязвимостей АС.

2. Множество эксплойтов X, хранящихся в БД [16; 22; 25]:

$$X = \left\{ \Psi_j \mid j = \overline{1, \xi} \right\}, \tag{2}$$

где Ψ_j – произвольный эксплойт; ξ – общее количество эксплойтов, хранящихся в БД эксплойтов.

Далее представим Ω_i в виде упорядоченного набора тегов (идентификаторов), характеризующих уязвимые элементы сканируемой АС [10; 20; 21; 23]:

$$\Omega_i = (B^{\Omega_i} G^{\Omega_i} D^{\Omega_i} E^{\Omega_i} CVSS^{\Omega_i}),\ i = \left(\overline{1, \alpha} \right), \tag{3}$$

где B^{Ω_i} – конечное множество операционных систем (ОС) АС, в которых реализуется Ω_i, например B^{Ω_i} = {*Windows, …, Linux*}; G^{Ω_i} – конечное множество программного обеспечения (ПО) АС, в котором реализуется Ω_i, например G^{Ω_i} = {*Aimp, …, Chrome*}; D^{Ω_i} – конечное множество аппаратного обеспечения (АО) АС, в котором реализуется Ω_i, например D^{Ω_i} = {*Zyxel, …, Intel*}; E^{Ω_i} – конечное множество сетевых служб АС, в которых реализуется Ω_i, например E^{Ω_i} = {*FTP, …, HTTP*}; $CVSS^{\Omega_i}$ – вектор опасности Ω_i.

На основе результатов, полученных в [26], представим вектор $CVSS^{\Omega_i}$ в виде упорядоченного набора следующих тегов (идентификаторов):

$$CVSS^{\Omega_i} = (AV^{CVSS^{\Omega_i}} AC^{CVSS^{\Omega_i}} AU^{CVSS^{\Omega_i}} C^{CVSS^{\Omega_i}} I^{CVSS^{\Omega_i}} A^{CVSS^{\Omega_i}}),\ i = \overline{(1,\alpha)}, \quad (4)$$

где $AV^{CVSS^{\Omega_i}}$ – степень удаленности потенциального атакующего от АС, использующей Ω_i; $AC^{CVSS^{\Omega_i}}$ – сложность эксплуатации Ω_i на АС; $AU^{CVSS^{\Omega_i}}$ – требуемый уровень привилегий (прав), необходимый для проведения атаки, использующей Ω_i; $C^{CVSS^{\Omega_i}}$, $I^{CVSS^{\Omega_i}}$, $A^{CVSS^{\Omega_i}}$ – оценка степени влияния атаки, использующей Ω_i, на конфиденциальность, целостность и доступность информации, содержащейся в АС, соответственно.

Кроме того, указанные теги принимают значения [26]:

$$AV^{CVSS^{\Omega_i}} = \{N, L\}, \quad (5)$$

где N – Network, т.е. потенциальный атакующий проводит атаку, используя Ω_i через глобальную сеть; L – Local, т.е. потенциальный атакующий проводит атаку, используя Ω_i через локальную сеть.

$$AC^{CVSS^{\Omega_i}} = \{L, M, H\}, \quad (6)$$

где L – Low, т.е. низкий уровень сложности эксплуатации Ω_i на АС; M – Medium, т.е. средний уровень сложности эксплуатации Ω_i на АС; H – High, т.е. высокий уровень сложности эксплуатации Ω_i на АС.

$$AU^{CVSS^{\Omega_i}} = \{N, S, M\}, \quad (7)$$

где N – None, т.е. при проведении атаки, использующей Ω_i, не требуется расширенных прав доступа к АС; S – Single, т.е. при проведении атаки, использующей Ω_i, требуются права пользователя АС; M – Multiple, т.е. при проведении атаки, использующей Ω_i, требуются права администратора АС.

$$C^{CVSS^{\Omega_i}} = \{N, P, C\},\ I^{CVSS^{\Omega_i}} = \{N, P, C\},\ A^{CVSS^{\Omega_i}} = \{N, P, C\}, \quad (8)$$

где N – None, т.е. атака, использующая Ω_i, не оказывает воздействия на конфиденциальность, целостность и доступность информации, содержащейся в АС; P – Partial, т.е. атака, использующая Ω_i, частично оказывает воздействие на конфиденциальность, целостность и доступность информации, содержащейся в АС; C – Complete, т.е. атака, использующая Ω_i, оказывает полное воздействие на конфиденциальность, целостность и доступность информации, содержащейся в АС.

Далее, с учетом (4)–(8) определим общий индекс опасности Ω_i $\left(CVSS_{ind}^{\Omega_i}\right)$ в виде предложенного в [26]:

$$CVSS_{ind}^{\Omega_i} = r(CVSS^{\Omega_i}),\ i = \overline{(1,\alpha)}, \quad (9)$$

где $r\left(CVSS^{\Omega_i}\right)$ – функция, задаваемая на основе подходов, описанных в [26].

Представим Ψ_j в виде упорядоченного набора тегов (идентификаторов), характеризующих возможность его применения:

$$\Psi_j = (z^{\Psi_j} B^{\Psi_j} G^{\Psi_j} D^{\Psi_j} E^{\Psi_j} u^{\Psi_j}),\ j = \overline{1, \xi}, \tag{10}$$

где z^{Ψ_j} – тип атаки Ψ_j, например z^{Ψ_j} = "DoS"; B^{Ψ_j} – конечное множество ОС, на которых выполняется Ψ_j; G^{Ψ_j} – конечное множество ПО, на которое направлено воздействие Ψ_j; D^{Ψ_j} – конечное множество АО, на которое направлено воздействие Ψ_j; E^{Ψ_j} – конечное множество сетевых служб АС, на которые направлено воздействие Ψ_j; u^{Ψ_j} – тег (идентификатор) принадлежности Ψ_j к эксплойтам, предоставляемым специалистом по ИБ (пользовательские эксплойты).

Далее осуществим моделирование процесса формирования набора эксплойтов J, направленность которых соответствует тегам $\Omega_i \in R$:

$$J = \left\{ \Phi_\lambda \,|\, \lambda = \left(\overline{1, \phi}\right) \right\}, \tag{11}$$

где Φ_λ – произвольный эксплойт, принадлежащий J; ϕ – общее количество эксплойтов, направленность которых соответствует тегам $\Omega_i \in R$.

Формирование J осуществим посредством определения Φ_λ на основании соответствия тегов Ψ_j и Ω_i:

$$\Phi_\lambda = \begin{cases} \Psi_j & \text{при } f_0(\Psi_j, \Omega_i) = 1; \\ 0 & \text{при } f_0(\Psi_j, \Omega_i) = 0; \end{cases} \quad j = \overline{1, \xi},\ i = \overline{1, \alpha}, \tag{12}$$

где $f_0(\Psi_j, \Omega_i)$ – функция, определяющая соответствие тегов, присущих Ψ_j и Ω_i.

Определим $f_0(\Psi_j, \Omega_i)$ в виде

$$f_0(\Psi_j, \Omega_i) = f_1(B^{\Psi_j}, B^{\Omega_i}) \wedge f_1(G^{\Psi_j}, G^{\Omega_i}) \wedge (f_1(D^{\Psi_j}, D^{\Omega_i}) \vee f_1(E^{\Psi_j}, E^{\Omega_i})), \tag{13}$$

$$j = \overline{1, \xi},\ i = \overline{1, \alpha},$$

где $f_1(Y^{\Psi_j}, Y^{\Omega_i})$ – функция, определяющая соответствие значений тегов, характеризующих ОС, ПО, АО и сетевые службы АС, в которых выполняется воздействие Ψ_j и реализуется Ω_i.

Определим $f_1(Y^{\Psi_j}, Y^{\Omega_i})$ в виде

$$f_1(Y^{\Psi_j}, Y^{\Omega_i}) = \begin{cases} 1, & \text{если} (Y^{\Psi_j} \cap Y^{\Omega_i}) \neq \varnothing; \\ 0, & \text{если} (Y^{\Psi_j} \cap Y^{\Omega_i}) = \varnothing; \end{cases} \quad j = \overline{1, \xi},\ i = \overline{1, \alpha}, \tag{14}$$

где Y^{Ψ_j} – произвольный тег, принимающий значения, соответствующие характеристикам B^{Ψ_j}, G^{Ψ_j}, D^{Ψ_j}, E^{Ψ}; Y^{Ω_i} – произвольный тег, принимающий значения, соответствующие характеристикам B^{Ω_i}, G^{Ω_i}, D^{Ω_i}, E^{Ω_i}.

Исходя из (12)–(14), отметим, что если нескольким Ω_i соответствует один и тот же Ψ_j, то он добавляется в множество J только один раз.

Далее для каждого $\Phi_\lambda \in J$, реализующего соответствующую Ω_i, определим вектор опасности $CVSS^{\Phi_\lambda}$ в виде

$$CVSS^{\Phi_\lambda} \triangleq CVSS^{\Omega_i}. \tag{15}$$

С учетом (9), (15) определим общий индекс опасности $\Phi_\lambda(CVSS_{ind}^{\Phi_\lambda})$ в виде [26]:

$$CVSS_{ind}^{\Phi_\lambda} = r(CVSS^{\Phi_\lambda}),\ \ \lambda = \overline{(1,\phi)}, \tag{16}$$

где $r(CVSS^{\Phi_\lambda})$ – функция, задаваемая на основе подходов, представленных в [26].

Учитывая (11)–(16), осуществим моделирование процесса конфигурирования, в результате которого сформируем упорядоченный набор эксплойтов W

$$W = \left\{\Xi_h \,|\, h = \overline{(1,\nu)}\right\}, \tag{17}$$

где Ξ_h – произвольный набор эксплойтов, принадлежащий W; ν – общее количество наборов эксплойтов, содержащихся в упорядоченном наборе эксплойтов.

Формирование W осуществим на основе следующих принципов:

Принцип 1. Эксплойты, позволяющие обеспечить расширение привилегий (прав), выполняются в первую очередь.

Принцип 2. Эксплойты, нарушающие доступность информации, содержащейся в АС, выполняются в последнюю очередь.

Принцип 3. Эксплойты, требующие меньше привилегий (прав), выполняются с большим приоритетом.

Принцип 4. Эксплойты, нарушающие конфиденциальность информации, содержащейся в АС, выполняются с большим приоритетом, чем эксплойты, нарушающие целостность информации.

Принцип 5. Пользовательские эксплойты выполняются с большим приоритетом, чем эксплойты, предоставляемые разработчиком СС имитации КА.

Принцип 6. Эксплойты с большим индексом $CVSS_{ind}^{\Phi_\lambda}$ выполняются с большим приоритетом.

На основе принципов 1–6 процесс конфигурирования представим в виде последовательного выполнения следующих этапов:

Первый этап – выделение эксплойтов по принципам 1–3.

Второй этап – выделение эксплойтов по принципу 4.

Третий этап – выделение эксплойтов по принципам 1–6.

В рамках первого этапа процесса конфигурирования J на основе принципов 1–3 и тегов (идентификаторов) $CVSS^{\Phi_\lambda}$ [см. (4)–(8), (15)], а также тега z^{Φ_λ} [см. (10)] множество J декомпозируем на следующие подмножества:

1. Множество эксплойтов, позволяющих обеспечить расширение привилегий (прав), Q_1:

$$Q_1 = \left\{\Delta_\beta \,|\, \beta = \overline{(1,\chi)}\right\}, \tag{18}$$

где Δ_β – произвольный эксплойт, принадлежащий Q_1; χ – общее количество эксплойтов, позволяющих обеспечить расширение привилегий (прав).

2. Множество эксплойтов, нарушающих доступность информации, содержащейся в АС, Q_2:

$$Q_2 = \left\{\Gamma_\delta \,|\, \delta = \overline{(1,\varepsilon)}\right\}, \tag{19}$$

где Γ_δ – произвольный эксплойт, принадлежащий Q_2; ε – общее количество эксплойтов, нарушающих доступность информации, содержащейся в АС.

3. Множество эксплойтов, не требующих расширенных прав доступа к АС, Q_3:

$$Q_3 = \left\{ O_\varphi \middle| \varphi = (\overline{1,\gamma}) \right\}, \tag{20}$$

где O_φ – произвольный эксплойт, принадлежащий Q_3; γ – общее количество эксплойтов, не требующих расширенных прав доступа к АС.

4. Множество эксплойтов, требующих права пользователя АС, Q_4:

$$Q_4 = \left\{ \Pi_\eta \middle| \eta = (\overline{1,\iota}) \right\}, \tag{21}$$

где Π_η – произвольный эксплойт, принадлежащий Q_4; ι – общее количество эксплойтов, требующих права пользователя АС.

5. Множество эксплойтов, требующих права администратора АС, Q_5:

$$Q_5 = \left\{ \Lambda_\kappa \middle| \kappa = (\overline{1,\mu}) \right\}, \tag{22}$$

где Λ_κ – произвольный эксплойт, принадлежащий Q_5; μ – общее количество эксплойтов, требующих права администратора АС.

Формирование Q_1 осуществим посредством определения Δ_β в J:

$$\Delta_\beta = \begin{cases} \Phi_\lambda & \text{при } z^{\Phi_\lambda} = \text{«расширение привилегий»}; \\ 0, & \text{иначе} \end{cases} \quad \lambda = (\overline{1,\phi}), \tag{23}$$

где z^{Φ_λ} – тег (идентификатор), определяющий тип атаки $\mathbf{\Phi}_\lambda$.

Формирование Q_2 осуществим посредством определения Γ_δ в J:

$$\Gamma_\delta = \begin{cases} \Phi_\lambda & \text{при } A^{CVSS^{\Phi_\lambda}} \neq N; \\ 0 & \text{при } A^{CVSS^{\Phi_\lambda}} = N; \end{cases} \lambda = (\overline{1,\phi}), \tag{24}$$

где $A^{CVSS^{\Phi_\lambda}}$ – оценка степени влияния $\mathbf{\Phi}_\lambda$ на доступность информации, содержащейся в АС, значение которой определяется в соответствии с (8), (15).

Формирование Q_3 осуществим посредством определения O_φ в J:

$$O_\varphi = \begin{cases} \Phi_\lambda & \text{при } AU^{CVSS^{\Phi_\lambda}} = N; \\ 0 & \text{при } AU^{CVSS^{\Phi_\lambda}} \neq N, \end{cases} \lambda = (\overline{1,\phi}), \tag{25}$$

где $AU^{CVSS^{\Phi_\lambda}}$ – требуемый уровень привилегий (прав), необходимый для выполнения $\mathbf{\Phi}_\lambda$, значение которого определяется в соответствии с (7), (15).

Формирование Q_4 осуществим посредством определения Π_η в J:

$$\Pi_\eta = \begin{cases} \Phi_\lambda & \text{при } AU^{CVSS^{\Phi_\lambda}} = S; \\ 0 & \text{при } AU^{CVSS^{\Phi_\lambda}} \neq S; \end{cases} \lambda = (\overline{1,\phi}). \tag{26}$$

Формирование Q_5 осуществим посредством определения Λ_κ в J:

$$\Lambda_\kappa = \begin{cases} \Phi_\lambda & \text{при } AU^{CVSS^{\Phi_\lambda}} = M; \\ 0 & \text{при } AU^{CVSS^{\Phi_\lambda}} \neq M; \end{cases} \lambda = (\overline{1,\phi}). \tag{27}$$

Затем в рамках второго этапа процесса конфигурирования J на основе принципа 4 и тегов (идентификаторов) $CVSS^{\Phi_\lambda}$ [см. (4)–(8), (15)] множества Q_3, Q_4, Q_5 композируем на следующие подмножества:

1. Множество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и не требующих расширенных прав доступа к АС, $K_1^{Q_3}$:

$$K_1^{Q_3} = \left\{ \Theta_o \,|\, o = \left(\overline{1, \pi} \right) \right\}, \tag{28}$$

где Θ_o – произвольный эксплойт, принадлежащий $K_1^{Q_3}$; π – общее количество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и не требующих расширенных прав доступа к АС.

2. Множество эксплойтов, нарушающих целостность информации, содержащейся в АС, и не требующих расширенных прав доступа к АС, $K_2^{Q_3}$:

$$K_2^{Q_3} = \left\{ \Sigma_\varpi \,|\, \varpi = (\overline{1, \theta}) \right\}, \tag{29}$$

где Σ_ϖ – произвольный эксплойт, принадлежащий $K_2^{Q_3}$; θ – общее количество эксплойтов, нарушающих целостность информации, содержащейся в АС, и не требующих расширенных прав доступа к АС.

3. Множество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и требующих права пользователя АС, $K_1^{Q_4}$:

$$K_1^{Q_4} = \left\{ \mathrm{T}_\vartheta \,|\, \vartheta = (\overline{1, \rho}) \right\}, \tag{30}$$

где T_ϑ – произвольный эксплойт, принадлежащий $K_1^{Q_4}$; ρ – общее количество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и требующих права пользователя АС.

4. Множество эксплойтов, нарушающих целостность информации, содержащейся в АС, и требующих права пользователя АС, $K_2^{Q_4}$:

$$K_2^{Q_4} = \left\{ Z_\sigma \,|\, \sigma = (\overline{1, \varsigma}) \right\}, \tag{31}$$

где Z_σ – произвольный эксплойт, принадлежащий $K_2^{Q_4}$; ς – общее количество эксплойтов, нарушающих целостность информации, содержащейся в АС, и требующих права пользователя АС.

5. Множество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и требующих права администратора АС, $K_1^{Q_5}$:

$$K_1^{Q_5} = \left\{ F_\tau \,|\, \tau = (\overline{1, \upsilon}) \right\}, \tag{32}$$

где F_τ – произвольный эксплойт, принадлежащий $K_1^{Q_5}$; υ – общее количество эксплойтов, нарушающих конфиденциальность информации, содержащейся в АС, и требующих права администратора АС.

6. Множество эксплойтов, нарушающих целостность информации, содержащейся в АС, и требующих права администратора АС, $K_2^{Q_5}$:

$$K_2^{Q_5} = \left\{ S_\omega \,|\, \omega = (\overline{1, \psi}) \right\}, \tag{33}$$

где S_ω – произвольный эксплойт, принадлежащий $K_2^{Q_5}$; ψ – общее количество эксплойтов, нарушающих целостность информации, содержащейся в АС, и требующих права администратора АС.

Формирование $K_1^{Q_3}$, $K_1^{Q_4}$, $K_1^{Q_5}$ осуществим посредством определения O_φ [см. (20)], Π_η [см. (21)], Λ_κ [см. (22)] в соответствующие $K_1^{Q_3}$ [см. (28)], $K_1^{Q_4}$ [см. (30)], $K_1^{Q_5}$ [см. (32)] в общем виде:

$$U = \begin{cases} g \ \text{ при } C^{CVSS^g} \neq N; \\ 0 \ \text{ при } C^{CVSS^g} = N, \end{cases} \tag{34}$$

где g – произвольный эксплойт, соответствующий $O_\varphi \in Q_3$ [см. (20)], либо $\Pi_\eta \in Q_4$ [см. (21)], либо $\Lambda_\kappa \in Q_5$ [см. (22)]; U – произвольный эксплойт, соответствующий $\Theta_o \in K_1^{Q_3}$ [см. (28)], либо $\mathrm{T}_\vartheta \in K_1^{Q_4}$ [см. (30)], либо $F_\tau \in K_1^{Q_5}$ [см. (32)]; C^{CVSS^g} – оценка степени влияния g на конфиденциальность информации, содержащейся в АС, значение которой определяется в соответствии с (8), (15).

Формирование $K_2^{Q_3}$, $K_2^{Q_4}$, $K_2^{Q_5}$ осуществим посредством определения O_φ [см. (20)], Π_η [см. (21)], Λ_κ [см. (22)] в соответствующие $K_2^{Q_3}$ [см. (29)], $K_2^{Q_4}$ [см. (31)], $K_2^{Q_5}$ [см. (33)] в общем виде:

$$V = \begin{cases} g \ \text{ при } I^{CVSS^g} \neq N; \\ 0 \ \text{ при } I^{CVSS^g} = N, \end{cases} \tag{35}$$

где V – произвольный эксплойт, соответствующий $\Sigma_\varpi \in K_2^{Q_3}$ [см. (29)], либо $Z_\sigma \in K_2^{Q_4}$ [см. (31)], либо $S_\omega \in K_2^{Q_5}$ [см. (33)]; I^{CVSS^g} – оценка степени влияния g на целостность информации, содержащейся в АС, значение которой определяется в соответствии с (8), (15).

В рамках третьего этапа процесса конфигурирования J на основе принципов 1–6, а также с учетом (18)–(35) представим Ξ_h в следующем виде:

$$\Xi_h = (\Delta_\beta\ \Theta_o\ \Sigma_\nu\ T_J\ Z_\sigma\ F_\tau\ S_\omega\ \Gamma_\delta),$$

$$\beta = \overline{1,\chi},\ O = \overline{1,\pi},\ \varpi = \overline{1,\theta},\ \vartheta = \overline{1,\rho},\ \sigma = \overline{1,\varsigma},\ \tau = \overline{1,\upsilon},\ \omega = \overline{1,\psi},\ \delta = \overline{1,\varepsilon}. \tag{36}$$

Отметим, что эксплойты, образующие Ξ_h [см. (36)], выбираются из соответствующих множеств на основе принципов 5–6, т.е. наибольшим приоритетом при реализации процедуры выбора обладают пользовательские эксплойты [см. (10)] с наибольшим значением индекса $CVSS_{ind}^{\Phi_\lambda}$ [см. (16)].

Кроме того, в целях недопущения случаев повторного добавления Φ_λ в W [см. (17)] осуществляется его исключение из множества J [см. (11)] после каждого формирования Ξ_h [см. (36)], в который входит Φ_λ.

Выводы

Разработанная модель подсистемы имитации КА на АС позволяет:

1. Повысить оперативность процесса тестирования реальных АС путем имитации атак с помощью автоматизации процедур формирования и конфигурирования набора эксплойтов [14].

2. Обеспечить необходимый уровень достоверности процесса имитации атак за счет задания структуры ПИКА, в составе которой выделен фреймворк СС имитации КА, способный воспроизводить множество необходимых типов атак.

3. Создавать рациональные системы обнаружения, предупреждения и ликвидации последствий КА на АС за счет расширения области синтеза структурно-функциональной модели и ее формализованного описания.

Литература

1. *Брэгг Р., Родс-Оусли М., Страссберг К.* Безопасность сетей: полное руководство. М.: Эком, 2011. 912 с.
2. *Галицкий А.В., Рябко С.Д., Шаньгин В.Ф.* Защита информации в сети – анализ технологий и синтез решений. М.: ДМК Пресс, 2010. 615 с.
3. ГОСТ Р 56546–2015. Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем. М.: Стандартинформ, 2015. 17 с.
4. ГОСТ Р 58143–2018. Информационная технология. Методы и средства обеспечения безопасности. Детализация анализа уязвимостей программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 15408 и ГОСТ Р ИСО/МЭК 18045. Ч. 2: Тестирование проникновения. М.: Стандартинформ, 2018. 22 с.
5. ГОСТ Р ИСО/МЭК 18045–2013. Информационная технология. Методы и средства обеспечения безопасности. Методология оценки безопасности информационных технологий. М.: Стандартинформ, 2014. 249 с.
6. Инструменты Kali Linux. URL: https://kali.tools/ (дата обращения: 17.01.2020).
7. *Коноваленко С.А., Королев И.Д., Васильев Д.С., Антоненко С.А.* Концептуальная модель системы комплексного контроля состояния защищенности автоматизированных систем // Современные научные исследования и разработки. 2018. № 10 (27). С. 447–454.
8. Концепция государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации: указ Президента Российской Федерации от 12 декабря 2014 г. № 1274 // Федеральная служба безопасности Российской Федерации. URL: http://www.fsb.ru/files/PDF/Vipiska_iz_koncepcii.pdf (дата обращения: 17.01.2020).
9. *Малюк А.А.* Информационная безопасность: концептуальные и методологические основы защиты информации. М.: ГЛТ, 2016. 280 с.
10. *Медведовский И.Д. и др.* Атака из Internet. М.: Солон-Р, 2009. 368 с.
11. Методология функционального моделирования IDEF0. М.: Госстандарт России, 2000. 75 с.
12. *Милосердов А., Гриднев Д.* Тестирование на проникновение с помощью Kali Linux 2.0 // CoderNet. URL: https://codernet.ru/category/hacking/files/kniga_testirovanie_na proniknovenie_s_kali_linux.php/ (дата обращения: 17.01.2020).
13. *Минаев В.А., Королев И.Д., Мазин А.В., Коноваленко С.А.* Модель выявления уязвимостей при нестабильных сетевых взаимодействиях с автоматизированной системой // Радиопромышленность. 2018. № 2. С. 48–57.
14. Модуль формирования и конфигурирования эксплойтов: свидетельство о государственной регистрации программы для ЭВМ 2019667790 Российская Федерация / Коноваленко С.А. и др.; заявитель и правообладатель Коноваленко С.А. № 2019666779; заявл. 16.12.19; опубл. 27.12.19, Реестр программ для ЭВМ. 1 с.

15. *Соколов А.В., Шаньгин В.Ф.* Защита информации в распределенных корпоративных сетях и системах. М.: ДМК Пресс, 2010. 656 с.
16. *Фостер Дж.С.* Защита от взлома: сокеты, эксплойты, shell-код: выявление уязвимостей операционных систем и прикладных программ к атакам хакеров. М.: ДМК, 2013. 784 с.
17. *Чипига А.Ф.* Информационная безопасность автоматизированных систем. М.: Гелиос АРВ, 2017. 336 с.
18. *Чирилло Д.* Обнаружение хакерских атак. СПб.: Питер, 2017. 864 с.
19. *Шаньгин В.Ф.* Защита информации в компьютерных системах и сетях. М.: ДМК Пресс, 2012. 592 с.
20. *Шелухин О.И.* Обнаружение вторжений в компьютерные сети (сетевые аномалии). М.: ГЛТ, 2013. 220 с.
21. *Engebretson P.* The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Madiso: Dakota State University Press: Syngress, 2011. 159 p.
22. *Gibson D.* CompTIA Security+: Get Certified Get Ahead. North Charleston: Create Space, 2011. 874 p.
23. *Harris Sh., Eagle Ch.* Gray Hat Hacking: The Ethical Hacker's Handbook. N. Y.: McGraw-Hill/Osborne Media, 2004. 434 p.
24. *Hertzog R., O'Gorman J.* Kali Linux Revealed. Cornelius: Offensive Security, 2017. 314 p.
25. *Kennedy D. et al.* Metasploit: The Penetration Tester's Guide. San Francisco: No Starch Press, 2011. 328 p.
26. *Mell P., Scarfone K.* A Complete Guide to the Common Vulnerability Scoring System. Version 2.0 // First. URL: https://www.first.org/cvss/v2/guide (date of the application: 17.01.2020).
27. *Weidman G.* Penetration Testing: A Hands-On Introduction to Hacking. San Francisco: No Starch Press, 2014. 528 p.

Literatura

1. *Bregg R., Rods-Ousli M., Strassberg K.* Bezopasnost' setej: polnoe rukovodstvo. M.: Ekom, 2011. 912 c.
2. *Galitskij A.V., Ryabko S.D., Shan'gin V.F.* Zashchita informatsii v seti – analiz tekhnologij i sintez reshenij. M.: DMK Press, 2010. 615 c.
3. GOST R 56546–2015. Zashchita informatsii. Uyazvimosti informatsionnykh sistem. Klassifikatsiya uyazvimostej informatsionnykh sistem. M.: Standartinform, 2015. 17 s.
4. GOST R 58143–2018. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Detalizatsiya analiza uyazvimostej programmnogo obespecheniya v sootvetstvii s GOST R ISO/MEK 15408 i GOST R ISO/MEK 18045. Ch. 2: Testirovanie proniknoveniya. M.: Standartinform, 2018. 22 s.
5. GOST R ISO/MEK 18045–2013. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Metodologiya otsenki bezopasnosti informatsionnykh tekhnologij. M.: Standartinform, 2014. 249 s.
6. Instrumenty Kali Linux. URL: https://kali.tools/ (data obrashcheniya: 17.01.2020).
7. *Konovalenko S.A., Korolev I.D., Vasil'ev D.S., Antonenko S.A.* Kontseptual'naya model' sistemy kompleksnogo kontrolya sostoyaniya zashchishchennosti avtomatizirovannykh sistem // Sovremennye nauchnye issledovaniya i razrabotki. 2018. № 10 (27). S. 447–454.
8. Kontseptsiya gosudarstvennoj sistemy obnaruzheniya, preduprezhdeniya i likvidatsii posledstvij komp'yuternykh atak na informatsionnye resursy Rossijskoj Federatsii: ukaz

Prezidenta Rossijskoj Federatsii ot 12 dekabrya 2014 g. № 1274 // Federal'naya sluzhba bezopasnosti Rossijskoj Federatsii. URL: http://www.fsb.ru/files/PDF/Vipiska_iz_koncepcii.pdf (data obrashcheniya: 17.01.2020).
9. *Malyuk A.A.* Informatsionnaya bezopasnost': kontseptual'nye i metodologicheskie osnovy zashchity informatsii. M.: GLT, 2016. 280 c.
10. *Medvedovskij I.D. i dr.* Ataka iz Internet. M.: Solon-R, 2009. 368 s.
11. Metodologiya funktsional'nogo modelirovaniya IDEF0. M.: Gosstandart Rossii, 2000. 75 s.
12. *Miloserdov A., Gridnev D.* Testirovanie na proniknovenie s pomoshch'yu Kali Linux 2.0 // CoderNet. URL: https://codernet.ru/category/hacking/files/kniga_testirovanie_na proniknovenie_s_kali_linux.php/ (data obrashcheniya: 17.01.2020).
13. *Minaev V.A., Korolev I.D., Mazin A.V., Konovalenko S.A.* Model' vyyavleniya uyazvimostej pri nestabil'nykh setevykh vzaimodejstviyakh s avtomatizirovannoj sistemoj // Radiopromyshlennost'. 2018. № 2. S. 48–57.
14. Modul' formirovaniya i konfigurirovaniya eksplojtov: svidetel'stvo o gosudarstvennoj registratsii programmy dlya EVM 2019667790 Rossijskaya Federatsiya / Konovalenko S.A. i dr.; zayavitel' i pravoobladatel' Konovalenko S.A. № 2019666779; zayavl. 16.12.19; opubl. 27.12.19, Reestr programm dlya EVM. 1 s.
15. *Sokolov A.V., Shan'gin V.F.* Zashchita informatsii v raspredelennykh korporativnykh setyakh i sistemakh. M.: DMK Press, 2010. 656 c.
16. *Foster Dzh.S.* Zashchita ot vzloma: sokety, eksplojty, shell-kod: vyyavlenie uyazvimostej operatsionnykh sistem i prikladnykh programm k atakam khakerov. M.: DMK, 2013. 784 c.
17. *Chipiga A.F.* Informatsionnaya bezopasnost' avtomatizirovannykh sistem. M.: Gelios ARV, 2017. 336 c.
18. *Chirillo D.* Obnaruzhenie khakerskikh atak. SPb.: Piter, 2017. 864 c.
19. *Shan'gin V.F.* Zashchita informatsii v komp'yuternykh sistemakh i setyakh. M.: DMK Press, 2012. 592 c.
20. *Shelukhin O.I.* Obnaruzhenie vtorzhenij v komp'yuternye seti (setevye anomalii). M.: GLT, 2013. 220 c.
21. *Engebretson P.* The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Madiso: Dakota State University Press: Syngress, 2011. 159 p.
22. *Gibson D.* CompTIA Security+: Get Certified Get Ahead. North Charleston: Create Space, 2011. 874 p.
23. *Harris Sh., Eagle Ch.* Gray Hat Hacking: The Ethical Hacker's Handbook. N. Y.: McGraw-Hill/Osborne Media, 2004. 434 p.
24. *Hertzog R., O'Gorman J.* Kali Linux Revealed. Cornelius: Offensive Security, 2017. 314 p.
25. *Kennedy D. et al.* Metasploit: The Penetration Tester's Guide. San Francisco: No Starch Press, 2011. 328 p.
26. *Mell P., Scarfone K.* A Complete Guide to the Common Vulnerability Scoring System. Version 2.0 // First. URL: https://www.first.org/cvss/v2/guide (date of the application: 17.01.2020).
27. *Weidman G.* Penetration Testing: A Hands-On Introduction to Hacking. San Francisco: No Starch Press, 2014. 528 p.