ISSN 2519-2310

CS&CS Journal

KARAZIN UNIVERSITY
CLASSICS AHEAD OF TIME

3(15) 2019

COMPUTER SCIENCE AND CYBERSECURITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В.Н.КАРАЗІНА
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ имени В.Н. КАРАЗИНА
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
КОМПЬЮТЕРНЫЕ НАУКИ И КИБЕРБЕЗОПАСНОСТЬ
COMPUTER SCIENCE AND CYBERSECURITY
(CS&CS)

Issue 3(15) 2019

Заснований 2015 року

Міжнародний електронний науково-теоретичний журнал
Международный электронный научно-теоретический журнал
International electronic scientific journal

The journal publishes research articles on theoretical, scientific and technical problems of effective facilities development for computer information communication systems and on information security problems based on advanced mathematical methods, information technologies and technical means.

Journal is published quarterly.
Approved for placement on the Internet by Academic Council of the Karazin Kharkiv National University (April 27, 2020, protocol No.8)

The journal has Digital Object Identifier: **10.26565/2519-2310**.

Editorial office:
V.N. Karazin Kharkiv National University
Svobody Sq., 6, office 315a, Kharkiv, 61022, Ukraine *(North building of University, 3th floor)*
Phone: +38 (057) 705-10-83
E-mail: *cscsjournal@karazin.ua*
Web-page: *http://periodicals.karazin.ua/cscs (Open Journal System)*

Published articles have been internally and externally peer reviewed

TABLE OF CONTENTS

Issue 3(15) 2019

УДК 519.111: 621.391 DOI: 10.26565/2519-2310-2019-3-01

ФОРМУЛИРОВКА И РЕШЕНИЕ ЗАДАЧИ ОПТИМАЛЬНОГО РЕЗЕРВИРОВАНИЯ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Анна Кононченко, Виктория Попенко, Виктор Краснобаев

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
akononpro@gmail.com, vita.popenko@gmail.com, v.a.krasnobaev@gmail.com

Рецензент: Николай Карпинский, д.т.н., проф., Университет Бельсько-Бяла,
ул. Виллова 2, 43-309 Бельсько-Бяла, Польша
mkarpinski@ath.bielsko.pl

Поступила: Ноябрь 2019.

Аннотация. *Объектом исследования являются процессы, протекающие в специализированных компьютерных системах (СКС) обработки цифровых данных, которые представлены непозиционной системой счисления в остаточных классах (СОК). Рассмотрены возможности повышение надёжности высокопроизводительных процессоров, которые функционируют на базе СОК, за счёт решения задачи оптимального резервирования их структуры. Для достижения поставленной цели, в статье формулируются и решаются прямая и обратная задачи оптимального резервирования структуры в СОК. Решение задач оптимального резервирования осуществляется путём применения метода наискорейшего покоординатного спуска, который является достаточно точным для проведения инженерных расчётов. Представлены результаты расчётов и сравнительного анализа надёжности (по вероятности безотказной работы) для трехканальной мажоритарной системы обработки информации в позиционной системе счисления и для резервированной структуры в СОК.*

Ключевые слова: *надёжность, система счисления, система остаточных классов, система обработки информации, специализированные компьютерные системы.*

1 Введение

Одним из основных требований, предъявляемых к современным системам обработки информации (СОИ), которые работают в реальном времени, являются обеспечение заданного уровня надёжности. Это связано, прежде всего, с тем, что выход из строя или даже кратковременный сбой в работе таких специализированных компьютерных систем (СКС) может привести к авариям или нанести серьезный экономический ущерб.

Целью статьи является, рассмотрение возможностей повышения надёжности высокопроизводительных процессоров в системе остаточных классов (СОК) за счёт решения задачи оптимального резервирования их структуры.

2 Анализ последних исследований

Существует два основных метода повышения надежности СКС, функционирующих в ПСС [1-3]: повышение надежности отдельных логических элементов (использование новой элементной базы) и введение различных типов избыточности (применения различных видов резервирования, влияющих как на конструктивную, так и на функциональную надежность СКС). Поскольку надежность логических элементов СКС определяется уровнем развития технологии, то очевидно, что введение избыточности при использовании любой элементной базы является наиболее эффективным путем повышения надежности СКС. Один из эффективных практических методов повышения надежности СКС является структурное резервирование, например, на уровне троированной мажоритарной структуры. Однако применение структурного резервирования в ПСС усложняет структуру вычислительного комплекса, повышает его энергопотребление, увеличивает массогабаритные и другие характеристики, что в конечном итоге повышает стоимость его создания и эксплуатации, а также ограничивает сферу его применения в различных технических системах. В связи с этими обстоятельствами возникает необходимость применения новых методов повышения надежности СКС, и, в частности, методов основанных на применении кодов в СОК. Возможность повышения

надежности при этом обуславливается свойствами СОК (малоразрядность, равноправность и независимость остатков), что позволяет более эффективно применять структурное резервирование по сравнению с кодами в ПСС.

3 Основные материалы исследований

При применении резервирования возможна формулировка задачи оптимального резервирования в двух вариантах [4-6].

1) *Прямая задача оптимального резервирования.* Требуется обеспечить вероятность безотказной работы СОИ не менее заданной при минимальных затратах:

$$\begin{cases} V_{SRC}^{(l)} \to \min; \\ P_{SRC}^{(l)}(t) \ge P_{зад}(t) \ \left[t = const\right]. \end{cases} \quad (1)$$

2) *Обратная задача оптимального резервирования.* Требуется обеспечить максимально возможную вероятность безотказной работы СОИ при заданных затратах:

$$\begin{cases} P_{SRC}^{(l)}(t) \ \left[t = const\right] \to \max; \\ V_{SRC}^{(l)} \le V_{зад}^{(l)}. \end{cases} \quad (2)$$

где: $P_{SRC}^{(l)}(t)$ – вероятность безотказной работы ℓ-байтовой СОИ в СОК;

$P_{зад}(t)$ – заданное значение вероятности безотказной работы;

$V_{COИ}^{(l)}$ – количество оборудования («стоимость затрат») СОИ в СОК;

$V_{зад}^{(l)}(t)$ – заданное количество оборудования.

Перечислим некоторые из известных методов решения задачи оптимального резервирования [7-10]:

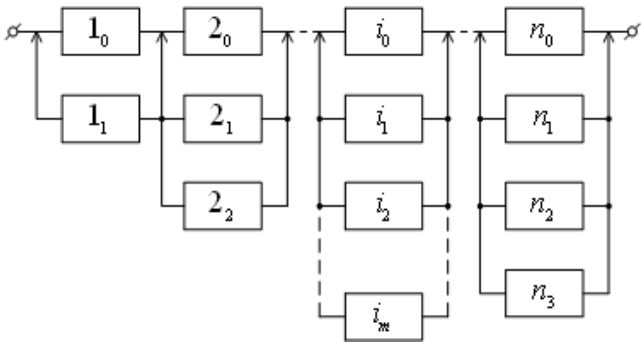

Рис. 1 – Надёжностная схема СОИ в СОК

1. Метод динамического программирования;
2. Модифицированный метод динамического программирования;
3. Метод наискорейшего покоординатного спуска;
4. Метод частной оптимизации с контролем ограничений;
5. Метод «отражающего экрана»;
6. Метод выбора наиболее «жёсткого» ограничения.

Каждый из перечисленных методов имеет свои достоинства и недостатки. Поэтому рассмотрим наиболее удобный и сравнительно простой для проведения инженерных расчётов метод наискорейшего покоординатного спуска. Это многошаговый процесс, который представлен на рис. 1.

1. Рассматриваемая система, состоит из n участков. Для каждого *i*-го участка необходимо определить (при различных кратностях резервирования m_i) значения вероятности безотказной работы $P_i(t, m_i)$ для некоторого фиксированного интервала времени при соответствующем способе резервирования, который возможен для данного участка системы.

2. Составляется сводная таблица (см. Табл. 1) значений $P_i(t,m_i)$ для всех практически возможных m_i и различных $i=1,2,\ldots,n$ (*для различных участков системы*), полученных по указанным формулам.

Таблица 1 – Сводная таблица переменных значений

m_i	$P_1(t)$	$P_2(t)$	…	$P_i(t)$	…	$P_n(t)$
0	$P_1(t,0)$	$P_2(t,0)$	…	$P_i(t,0)$	…	$P_n(t,0)$
1	$P_1(t,1)$	$P_2(t,1)$	…	$P_i(t,1)$	…	$P_n(t,1)$
2	$P_1(t,2)$	$P_2(t,2)$	…	$P_i(t,2)$	…	$P_n(t,2)$
…	…	…	…	…	…	…
m	$P_1(t,m)$	$P_2(t,m)$	…	$P_i(t,m)$	…	$P_n(t,m)$
…	…	…	…	…	…	…

3. На основании полученных значений $P_i(t,m_i)$, сведенных в таблицу 1, и известных значений "стоимости" элементов ω_i рассчитывается $\gamma_i(m_i)$ по формуле (3) для всех значений *i* и различных значений m_i, после чего составляется таблица 2.

$$\gamma_i(m_i+1)=\frac{P_i(t,m_i+1)-P_i(t,m_i)}{\omega_i P_i(t,m_i)}. \qquad (3)$$

4. Все значения $\gamma_i(m_i)$ таблицы 2 перенумеровываются в каждом из столбцов в порядке убывания, а также далее нумеруем все значения $\gamma'(k)$ в порядке их убывания, и исследуется следующий многошаговый процесс.

Таблица 2 – Сводная таблица полученных данных

m_i	γ_1	γ_2	…	γ_i	…	γ_n
0	–	–	…	–	…	–
1	$\gamma_1(1)$	$\gamma_2(1)$	…	$\gamma_i(1)$	…	$\gamma_n(1)$
2	$\gamma_1(2)$	$\gamma_2(2)$	…	$\gamma_i(2)$	…	$\gamma_n(2)$
…	…	…	…	…	…	…
m	$\gamma_1(m)$	$\gamma_2(m)$	…	$\gamma_i(m)$	…	$\gamma_n(m)$
…	…	…	…	…	…	…

На первом шаге:

- выбирается γ с номером 1 (максимальная из величин $\gamma_i(1)$);
- по таблице 1 отыскивается соответствующая величина $P_i(t,1)$;
- вычисляется значение:

$$P_c^{(1)}(t)=\frac{P_i(t,1)}{P_i(t,0)}P_c^{(0)}(t), \qquad (4)$$

где $P_c^{(0)}(t)=\prod_{k=1}^{n}P_k(t,0)$ – начальное значение вероятности безотказной работы исходной системы (без резервирования);

- вычисляется значение:

$$W_c^{(1)} = W_c^{(0)} + \omega_i, \tag{5}$$

где: - $W_c^{(0)}$ – начальная "стоимость" системы; - ω_i – "стоимость" 1-го элемента, который предназначен для резервирования *i*-го участка.

На втором шаге:

- выбирается γ с номером 2 (max среди оставшихся $\gamma_k(1)$ для $k \neq 1$ или $\gamma_i(2)$);
- по таблице 1 отыскивается соответствующая величина $P_k(t,1)$ (или $P_i(t,2)$, если номер 2 имеет $\gamma_i(2)$);
- вычисляется значение:

$$P_c^{(2)}(t) = \frac{P_k(t,1)}{P_k(t,0)} P_c^{(1)}(t), \tag{6}$$

или

$$P_c^{(2)}(t) = \frac{P_i(t,2)}{P_i(t,1)} P_c^{(1)}(t), \tag{7}$$

если номер 2 имеет $\gamma_i(2)$;

- вычисляется значение:

$$W_c^{(2)} = W_c^{(1)} + \omega_k, \tag{8}$$

или имеем

$$W_c^{(2)} = W_c^{(1)} + \omega_i, \tag{9}$$

если номер 2 имеет $\gamma_i(2)$. Затем этот многошаговый процесс продолжается при использовании выражений

$$P_c^M(t) = \frac{P_i(t,m_i)}{P_i(t,m_i-1)} P^{M-1}; \tag{10}$$

или

$$W_c^{(M)} = W_c^{(M-1)} + \omega_i, \tag{11}$$

при $m_i \geq 0$; $M \geq 0$.

Полученные данные сводятся в таблицу 3. Указанный процесс прекращается на шаге

$$M\left(M = \sum_{i=1}^{n} m_i\right). \tag{12}$$

Когда для первой задачи выполняется условие:

$$P_c^{(M-1)}(t) < P_{mp}(t) \leq P_c^M(t). \tag{13}$$

Для второй задачи выполняется условие:

$$W_c^M \leq W_{mp} \leq W_c^{(M+1)}. \tag{14}$$

Из анализа данных таблицы 4 *(решения задачи оптимального резервирования)* следует, что независимо от того, какая задача решается, каков процесс решения, а также условия вы-

бора γ_i , на каждом шаге автоматически обеспечивается минимальная "стоимость" системы при решении 1-й задачи и max возможная вероятность безотказной работы системы при ре шении 2-й задачи.

Таблица 3 – Совокупность промежуточных значений

M	m_1	m_2	···	m_n	$P_i^{(M)}(t)$	$W_c^{(M)}$
0	0	0	…	0	$P_i^{(0)}(t)$	$W_c^{(0)}$
1	1	0	…	0	$P_i^{(1)}(t)$	$W_c^{(1)}$
2	1	0	…	0	$P_i^{(2)}(t)$	$W_c^{(2)}$
3	1	1	…	0	$P_i^{(3)}(t)$	$W_c^{(3)}$
4	1	2	…	0	$P_i^{(4)}(t)$	$W_c^{(4)}$
5	1	3	…	0	$P_i^{(5)}(t)$	$W_c^{(5)}$
6	2	3	…	1	$P_i^{(6)}(t)$	$W_c^{(6)}$
…	…	…	…	1	…	…

В соответствии с приведённой методикой в таблице 5 представлены расчётные значения вероятности безотказной работы $P_i^{(M)}(t)$ и "стоимости" системы $W_c^{(M)}$, для однобайтового вычислительного устройства в СОК при интенсивности отказов $\lambda = 10^{-8}$ и времени работы $t=1$ [час].

Вероятность безотказной работы резервированной СОИ в СОК $P_{COK}^{(M)}(t)$ определяется для каждого фиксированного интервала времени ее работы.

Таблица 4 – Таблица решений

M	$P_i^{(M)}(t)$	$W_c^{(M)}$
0	$P_i^{(0)}(t)$	$W_c^{(0)}$
1	$P_i^{(1)}(t)$	$W_c^{(1)}$
2	$P_i^{(2)}(t)$	$W_c^{(2)}$
3	$P_i^{(3)}(t)$	$W_c^{(3)}$
4	$P_i^{(4)}(t)$	$W_c^{(4)}$
5	$P_i^{(5)}(t)$	$W_c^{(5)}$
6	$P_i^{(6)}(t)$	$W_c^{(6)}$
…	…	…

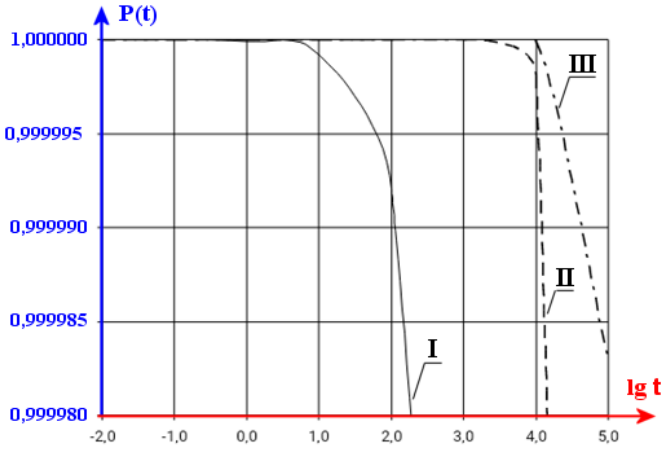

Рис. 2 – Графики функции $y=P(t)$ *(для* $\ell=1$, $\lambda_э=10^{-8}$*)*

Данные табл. 6 представляют результаты расчёта вероятностей безотказной работы однобайтовой СОИ при $\lambda = 10^{-8}$ для: – нерезервированной одноканальной структуры СОИ в ПСС (I); – трехканальной мажоритарной структуры СОИ в ПСС (II); – резервированной СОИ в СОК (III). В соответствии с данными таблицы 5 на рис. 2 приведены зависимости вероятности безотказной работы от времени *P(t)* для каждой рассмотренной структуры СОИ.

Таблица 5 – Предварительные результаты вычислений

Основания СОК				**Произведение**	$W_c^{(M)}$
$m_1 = 3$	$m_2 = 4$	$m_3 = 5$	$m_4 = 7$	$P_{SRC}^{(M)}(t)$	**[y. e.]**
0,9999999800000	0,9999999800000	0,9999999700000	0,9999999700000	0,9999999000000	10
0,9999999800000	0,9999999800000	0,9999999999999	0,9999999700000	0,9999999300000	13
0,9999999800000	0,9999999800000	0,9999999999999	0,9999999999999	0,9999999599999	16
1,0000000000000	0,9999999800000	0,9999999999999	0,9999999999999	0,9999999799999	18
1,0000000000000	1,0000000000000	0,9999999999999	0,9999999999999	0,9999999999999	20
1,0000000000000	1,0000000000000	1,0000000000000	0,9999999999999	0,9999999999999	23
1,0000000000000	1,0000000000000	1,0000000000000	1,0000000000000	0,9999999999999	26

Таблица 6 – Вероятности безотказной работы однобайтовой СОИ

Время работы СОИ (*t*)	***lg t***	**ПСС**		**СОК**
		Одноканальная СОИ	***Трёхканальная мажоритарная СОИ***	
		I	***II***	***III***
0,01	-2	0,9999999992	1,0000000000	1,0000000000
0,1	-1	0,9999999920	1,0000000000	1,0000000000
1	0	0,9999999200	0,9999999999	0,9999999999
10	1	0,9999992000	0,9999999999	0,9999999999
100	2	0,9999920000	0,9999999998	0,9999999999
1000	3	0,9999200032	0,9999999808	0,9999999983
10000	4	0,9992003199	0,9999980826	0,9999998300
100000	5	0,9920319148	0,9998105407	0,9999830161

5 Выводы

Представленные результаты расчёта и сравнительного анализа надёжности СОИ в СОК и ПСС, получены путём решения задачи оптимального резервирования методом наискорейшего покоординатного спуска, свидетельствуют о том, что использование непозиционного кодирования позволяет достичь требуемого уровня надёжности при меньшем количестве дополнительного оборудования, чем широко известный в ПСС метод мажоритарного троирования.

Ссылки

[1] I. Ya. Akushskii and D. I. Yuditskii, *Machine Arithmetic in Residual Classes* [in Russian]. Moscow: Sov. Radio, 1968.

[2] V. Krasnobayev, A. Yanko and S. Koshman, “A Method for arithmetic comparison of data represented in a residue number system”, *Cybernetics and Systems Analysis*, vol. 52, Issue 1, pp. 145-150, 2016.

[3] O. Karpenko, A. Kuznetsov, V. Sai and Yu. Stasev, “Discrete Signals with Multi-Level Correlation Function”, *Telecommunications and Radio Engineering*, vol. 71, Issue 1, pp. 91-98, 2012.

[4] V. Krasnobayev, S. Koshman and M. Mavrina, “A method for increasing the reliability of verification of data represented in a residue number system”, *Cybernetics and Systems Analysis*, vol. 50, Issue 6, pp. 969-976, 2014.

[5] S. Kavun, A. Zamula and I. Mikheev, “Calculation of expense for local computer networks”, *4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkov, pp. 146-151, 2017.

[6] V. Krasnobayev and S. Koshman, “A method for operational diagnosis of data represented in a residue number system”, *Cybernetics and Systems Analysis*, vol. 54, Issue 2, pp. 336-344, 2018.

[7] O. Kazymyrov, R. Oliynykov and H. Raddum, “Influence of addition modulo 2n on algebraic attacks”, *Cryptography and Communications*, vol. 8, Issue 2, pp. 277-289, April 2016.

[8] D. A. Patterson , *The Morgan Kaufmann Series in Computer Architecture and Design*. Morgan Kaufmann, 2016.

[9] A. Yanko, S. Koshman, V. Krasnobayev, “Algorithms of data processing in the residual classes system”, *4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, pp. 117-121, 2017.

[10] O. Kuznetsov, Yu. Gorbenko, I. Bilozertsev, A. Andrushkevych and O. Narizhnyi, “Algebraic Immunity of Non-linear Blocks of Symmetric Ciphers”, *Telecommunications and Radio Engineering*, vol. 77, Issue 4, pp. 309-325, 2018.

[11] A. Kuznetsov, I. Kolovanova and T. Kuznetsova, “Periodic characteristics of output feedback encryption mode”, *4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkov, pp. 193-198, 2017.

[12] Yu.V. Stasev, A.A. Kuznetsov, A.M. Nosik, “Formation of pseudorandom sequences with improved autocorrelation properties”, *Cybernetics and Systems Analysis*, vol. 43, Issue 1, pp. 1-11, January 2007.

[13] N. Naumenko, Yu. Stasev, A. Kuznetsov, “Methods of synthesis of signals with prescribed properties”, *Cybernetics and Systems Analysis*, vol. 43, Issue 3, pp. 321-326, May 2007.

[14] V. Ruzhentsev, R. Oliynykov, “Properties of Linear Transformations for Symmetric Block Ciphers on the basis of MDS-codes”, *Proceedings of the 6th International Conference on Network Architecture and Information System Security SAR-SSI*, pp. 193-196, 2011.

[15] O. Potii, O. Illiashenko, D. Komin, “Advanced Security Assurance Case Based on ISO/IEC 15408”, *Theory and Engineering of Complex Systems and Dependability Advances in Intelligent Systems and Computing*, vol. 365, pp. 391-401, 2015.

[16] A. S. Molahosseini, L.Seabra de Sousa, Ch.-H.Chang. *Embedded Systems Design with Special Arithmetic and Number Systems*, Springer International Publishing, 2017.

[17] V. Dolgov, I. Lisitska, K. Lisitskyi, “The new concept of block symmetric ciphers design”, *Telecommunications and Radio Engineering*, vol. 76, Issue 2, pp. 157-184, 2017.

[18] A. Kuznetsov, A. Smirnov, D. Danilenko, A. Berezovsky, “The statistical analysis of a network traffic for the intrusion detection and prevention systems”, *Telecommunications and Radio Engineering*, vol. 74, Issue 1, pp. 61-78, 2015.
[19] R. Gavrylko, Yu. Gorbenko, “A physical quantum random number generator based on splitting a beam of photons”, *Telecommunications and Radio Engineering*, vol. 75, Issue 2, pp. 179-188, 2016.

Reviewer: Mikołaj Karpiński, Dr. of Sciences (Eng.), Full Prof., University of Bielsko-Biala, Bielsko-Biala, Poland.
E-mail: mkarpinski@ath.bielsko.pl

Received: November 2019.

Authors:
Anna Kononchenko, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: akononpro@gmail.com
Victoria Popenko, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: vita.popenko@gmail.com
Victor Krasnobaev, Doctor of Sciences (Eng.), Full Prof., V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: v.a.krasnobaev@gmail.com

The formulation and solution of the task of the optimum reservation in the system of residual classes.

Abstract. Processes in specialized computer systems of the handling of the digitized data, which are representing non-positional notation in the system of residual classes (NRC), serve as the object of research. The considered capabilities of increasing the reliability of the high-performance processors, functioning on NRC's base as a result of solving the task of the optimum redundancy of their structure. In order to achieve the goal, direct and inverse tasks of the optimum reservation of the structure in NRC are formulated and solved in the article. The solution of problems of the optimum reservation is carried out by application of a method of the fastest coordinate descent, which is quite exact for carrying out engineering calculations. The results of calculations and comparative analysis of reliability (on the probability of trouble-free operation) for a three-channel majority information processing system are presented, in positional notation and for reservation of the structure in NRC are represented.

Keywords: Reliability; Notation; Residue classes system; Information handling system; Specialized computer system.

Рецензент: Микола Карпінський, д.т.н., проф., університет Бельсько-Бяла, Бельсько-Бяла, Польща.
E-mail: mkarpinski@ath.bielsko.pl

Надійшло: Листопад 2019.

Автори:
Анна Кононченко, студентка факультету комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, Харків, 61022, Україна.
E-mail: akononpro@gmail.com
Вікторія Попенко, студентка факультету комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, Харків, 61022, Україна.
E-mail: vita.popenko@gmail.com
Віктор Краснобаєв, д.т.н., проф., Харківський національний університет імені В. Н. Каразіна, Харків, 61022, Україна.
E-mail: v.a.krasnobaev@gmail.com

Формулювання та рішення завдання оптимального резервування в системі залишкових класів.

Анотація. Об'єктом дослідження є процеси, що тривають в спеціалізованих комп'ютерних системах обробки цифрових даних, які представлені непозиційною системою числення в залишкових класах (СЗК). Розглянуті можливості підвищення надійності високопродуктивних процесорів, які функціонують на базі СЗК, за рахунок рішення задачі оптимального резервування їх структури. Для досягнення поставленої мети, в статті формулюються і вирішуються пряма і зворотна задачі оптимального резервування структури в СЗК. Рішення задач оптимального резервування здійснюється шляхом застосування методу найшвидшого покоординатного спуску, який є досить точним для проведення інженерних розрахунків. Представлені результати розрахунків і порівняльного аналізу надійності (за ймовірністю безвідмовної роботи) для 3-х канальної мажоритарної системи обробки інформації в позиційній системі числення і для резервованої структури в СЗК.

Ключові слова: надійність; система числення; система залишкових класів; система обробки інформації; спеціалізовані комп'ютерні системи.

УДК 004.056.5 DOI: 10.26565/2519-2310-2019-3-02

ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ КОРПОРАТИВНОГО МІЖМЕРЕЖЕВОГО ЕКРАНУ ТА ПИТАННЯ ВЗАЄМОДІЇ З СИСТЕМОЮ IDS

Денис Рондалєв, Ольга Мелкозьорова, Олексій Нарєжній

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
denisrondalev@gmail.com, olja.mex@gmail.com, o.nariezhnii@karazin.ua

Рецензент: Роман Олійников, д.т.н., проф., ПАО "ІІТ", вул. Бакулина, 12, Харків, 61166, Україна
roliynykov@gmail.com

Надійшла: Листопад 2019.

Анотація: *Запропоновано стислий огляд особливостей використання корпоративного міжмережевого екрану та питань взаємодії з елементами системи виявлення вторгнень. Розглянуто деякі важливі особливості синтезу моделі загроз. Привернено увагу важливості коректного настроювання системи виявлення вторгнень (Snort). Виділено основні етапи налаштувань та деякі особливості оцінки рівня захисту корпоративного міжмережевого екрану. Звернено увагу на важливість питань сегментації мережевих ресурсів та розміщення датчиків системи виявлення вторгнень.*

Ключові слова: *Snort; IDS; IPS; DLP; міжмережевий екран.*

1 Вступ

Міжмережевий екран – це програмний або програмно-апаратний елемент комп'ютерної мережі, який здійснює фільтрацію і контроль поточного мережевого трафіку відповідно до заданих правил [1]. В більшості випадків, спеціальне програмне забезпечення (ПЗ), до якого можливо віднести програмний міжмережевий екран (ММЕ), встановлюється на серверній частині або на окремому хості «внутрішньої» частині комп'ютерної мережі.

Сучасні ММЕ корпоративного рівня, що взаємодіють з програмним модулем системи виявлення вторгнень (*англ. Intrusion Detection System – IDS*) або системи захисту від витоку даних (*англ. Data Leak Prevention - DLP*), за умови їх правильного налаштування, дозволяють забезпечувати потрібний рівень захисту від несанкціонованого доступу (НСД) до відповідних інформаційних та апаратних ресурсів з використанням різного типу вразливостей протоколів і ПЗ. Як і у всіх інших випадках, що пов'язані з використанням сучасних Інтернет-технологій, не менш важливими є питання збереження конфіденційності даних та уникнення впливу шкідливого ПЗ на функціонування «чутливих» корпоративних сервісів і елементів мережевої інфраструктури.

Метою роботи є аналіз та узагальнення відомостей, щодо можливостей покращення рівня інформаційної безпеки (ІБ) ресурсів локальної обчислювальної мережі при використанні ММЕ корпоративного рівня. Основними завданнями, в межах зазначеної мети, слід вважати:

- визначення основних функцій та аналіз особливостей настроювань корпоративного ММЕ;
- огляд основних можливостей підвищення ефективності захисту корпоративних ресурсів шляхом інтеграції декількох програмно-апаратних рішень *(ММЕ, IDS, DLP та ін.*);
- аналіз відомих загроз та формування рекомендацій щодо підвищення рівня захисту корпоративних ресурсів.

2 Аналіз проблематики ММЕ та питання взаємодії з IDS

Головною метою пошуку і виявлення потенційних вторгнень є моніторинг наявних мережевих активів для визначення характеру і цілей аномальної поведінки або нештатного вико-

ристання ресурсів мережі, яка захищається. Ця концепція існує більше двадцяти років, але не так давно відбулося різке зростання популярності та її включення в загальну інфраструктуру інформаційної безпеки [2]. З великим ступенем впевненості можна стверджувати, що починаючи з роботи 1980 року, «Computer Security Threat Monitoring and Surveillance» [3], з'явилася ідея стосовно можливостей виявлення «зовнішніх» зловмисних вторгнень. З того часу спільними зусиллями профільних фахівців та компаній розробників відповідного ПЗ вдалося значно вдосконалили технології та методики виявлення вторгнень до її поточного стану. Фактично, цей звіт, написаний для урядової організації NIST, визначив, що аудиторські записи містять надважливу (*перш за все, з точки зору питань забезпечення інформаційної безпеки - ІБ*) інформацію, яка може бути вкрай цінною для розуміння поведінки користувачів та відстеження невірного або нелегітимного використання наявних ресурсів мережі (*інформаційних та апаратних*), яка захищається. Практично, з появою цього документу оформилися основні ідеї концепції «виявлення» нецільового (не декларованого) використання наявних мережевих ресурсів, що призвело до величезного вдосконалення аудиторських підсистем, практично кожної операційної системи. Таким чином, робота [3], практично, стала початком створення систем виявлення вторгнень на основі хоста та систем виявлення вторгнень (IDS) загалом.

У 1983 році «SRI International» почала працювати над урядовим проектом [4], який започаткував нові зусилля, в межах розробки напрямку систем виявлення вторгнень. Метою було проаналізувати аудиторські записи з урядових комп'ютерів та створити характерні профілі користувачів на основі їх типової мережевої діяльності. Приблизно через рік, була розроблена перша система для виявлення вторгнень, так звана «Експертна система виявлення вторгнень» (IDES), яка послужила основою для розвитку технології IDS в майбутньому.

Помітний комерційний розвиток технологій виявлення вторгнень розпочався на початку 1990-х. «Haystack Labs» був першим комерційним постачальником постачальників інструментів IDS з лінійкою «Stalker» [5]. «SAIC» також розробляла способи виявлення вторгнень на основі хоста, що мала назву «Система виявлення комп'ютерних зловживань» (CMDS). Одночасно «Криптологічний центр підтримки ВПС США» розробив автоматизовану систему вимірювання безпеки (ASIM) для моніторингу поточного мережевого трафіку в мережі військово-повітряних сил [6]. В цілому, ASIM домоглася значного прогресу у подоланні виниклих проблем з масштабуванням та портативністю, які раніше торкалися минулих продуктів. Крім того, ASIM стала першим рішенням, яке одночасно включало до себе як апаратне, так і програмне рішення для виявлення нелегітимних вторгнень в мережу. ASIM до цих пір використовується «Командою управління оперативного реагування на надзвичайні ситуації ВПС» (AFCERT) у місцях по всьому світу [1]. Як це часто траплялося, група розробників за проектом ASIM у 1994 році утворила комерційну компанію «Wheel Group», а їх продукт «Net Ranger»[1] став першим комерційно спроможним мережевим пристроєм виявлення зовнішніх вторгнень.

В загальному випадку, виявлення не декларованих мережевих вторгнень, в переважній кількості випадків, стосується процесів передачі даних між кількома хостами (вузлами). Так звані «сніфери пакетів» перехоплюють відповідні пакети даних, що циркулюють у мережі з використанням різних комунікаційних середовищ (кабельні та бездротові мережі) та протоколів передачі даних, як правило стеку TCP/IP. Після перехоплення пакети аналізуються багатьма різними способами. Так, наприклад, деякі рішення IDS порівнюють отримані пакети з відповідною базою даних, що містить характерний опис відомих зловмисних атак та їх цифрових відбитків (сигнатур), а інші виявляють аномальну активність пакету, яка може вказувати на шкідливу поведінку відповідного коду, або прояви не декларованої (в т.ч. не авторизованої) мережевої активності [7].

В переважної більшості випадків IDS відстежує мережевий трафік на предмет існування будь-якої не передбаченої або забороненої мережевої активності. При цьому, основна функція IDS – сповіщення адміністраторів мережі або персоналу відділу ІБ, у разі фіксації подій, що входять до сфери компетенцій відповідних систем (*де рівень чутливості системи визначається її попередніми налаштуваннями*). В якості можливих автоматичних реакцій системи

захисту можуть виступати, певні коригувальні дії (*що змінюють окремі параметри роботи мережевих пристроїв та захисного ПЗ*), блокуючи дії (*наприклад, заборона певного виду трафіку та/або запит підтверджень*), сигнальні або інформуючи дії (*формування та доведення до потрібного персоналу відповідних сповіщень безпеки*), та взагалі сукупність всіх зазначених дій. В любому випадку, активація автоматичних дій системи безпеки (*без втручання персоналу*) дозволяє значно скоротити час її захисних реакцій та створити умови для привертання уваги відповідних фахівців на аналізі саме випадків нетипової мережевої активності (*у разі детектування ознак її існування*). В якості передбачених автоматичних реакцій можуть бути [8-11]: - «закриття» декількох портів; - заборона вхідного і/або вихідного трафіку у визначеному діапазоні IP-адресів; - тимчасове припинення будь-якої мережевої активності у визначеному сегменті корпоративної мережевої архітектури (*наприклад, блокування взаємодії з бездротовою частиною корпоративної мережі*); - активація алгоритму санкціонування критичних процедур (*наприклад, підтвердження доступу до певної інформації/пристрою або підтвердження спроби «зовнішнього службового входу» до системи*); - активація *Honeypot* і таке інше. В любому випадку, всі заздалегідь передбачені автоматичні реакції системи захисту, це «зброя» оперативного реагування, ефективність застосування якої, базується на ретельному та систематичному аналізі особливостей мережевої активності, що притаманна для кожної конкретної інформаційної структури (локальної мережі). В межах періодичного аудиту мережевої активності повинні бути отримані та формалізовані відомості, які в сукупності з попередньо-визначеними цифровими відбитками відомих атак, використовуються для парирування спроб нелегітимного (*в т.ч. недекларованого*) впливу на вразливі або критичні «місця» (*сегменти інфраструктури*) та об'єкти (*інформаційні і апаратні*) інформаційної структури, яка захищається [2,8].

Станом на сьогоднішній день існує досить широкий спектр методів та інструментів для модифікації і генерації шкідливого мережевого трафіку для наступного проведення атак із зовні [8] на різні мережеві структури. Використання на зовнішньому периметрі безпеки мережі будь-якого ММЕ (*програмного, апаратного, програмно-апаратного*) дозволяє досить ефективно здійснювати фільтрацію такого трафіку, при умові, якщо цій ММЕ здатний його розпізнати (*залежить від коректності і актуальності встановлених правил його роботи*). В іншому випадку, ММЕ помітно втрачає свою ефективність, оскільки не здатний в автоматичному режимі однозначно прийняти самостійне рішення про те, що саме робити з нерозпізнаним мережевім трафіком [9]. При цьому, як свідчить відомий досвід [8,9], перший корпоративний ММЕ, зазвичай, встановлюється окремо від основного сегменту мережі саме для того, щоб підозрілі вхідні запити не потрапляли безпосередньо «всередину» корпоративної мережі (*перша лінія периметру безпеки*).

На відміну від ММЕ, система виявлення вторгнень (IDS) відрізняється тим, що вона спроможна виявляти не тільки зовнішні вторгнення, але і внутрішні атаки. Так, міжмережеві екрани обмежують доступ/трафік між різними мережами (*або частинами однієї мережі*) для запобігання можливого вторгнення, але не сигналізують про напад (*в т.ч. нелегітимний виток даних*) зсередини мережі. При цьому, IDS відстежує внутрішні атаки, шляхом вивчення особливостей поточних мережевих комунікацій, виявлення сигнатур відомих мережевих загроз, а в разі необхідності вживає передбачені заходи, щодо оповіщення персоналу, зазвичай, взаємодіючі зі службою ОС «*netfilter*» [10]. В певному сенсі IDS є однією з форм реалізації ММЕ прикладного (*англ. Application layer*) рівня OSI, що застосовується як додатковий та більш інтелектуальний компонент сучасних міжмережевих екранів [15].

Зазвичай IDS класифікуються за місцем виявлення вторгнення та методу виявлення, який в них впроваджено, при цьому більшість відповідних систем використовують один із 3-х відомих методів виявлення вторгнень [16, 17]:

- *аналіз сигнатур*. Відносно простий та дуже ефективний метод проти відомих різновидів мережевих атак. Його ефективність безпосередньо залежить від регулярного оновлення бази сигнатур, та не може виявляти невідомі або модифіковані версії загроз. Е першим методом, який було застосовано для виявлення вторгнень. Його принцип ро-

боти базується на збігу послідовності зі зразком шкідливого трафіку. У разі такого збігу ініціюється тривога;

- *аналіз протоколів*. Сутність цього методу полягає в розгляді вмісту та структури даних трафіку, що циркулює (*які суворо визначені відповідними вимогами*), та узгодження фактичних даних з програмою [16]. Як відомо, кожен протокол має кілька полів з очікуваними або нормальними значеннями, і якщо ці стандарти десь порушуються, то IDS оголошується тривога;
- *метод виявлення аномалій*. Робота даного методу базується на правилах або евристиці (не на шаблонах), через що така система може виявляти і раніше невідомі загрози. Для впровадження цієї системи потрібно створити модель штатної роботи мережі, з якою (моделлю) потім буде порівнюватися вся нова (нетипова) мережева поведінка.

Станом на сьогоднішній день, по місту виявлення потенційного вторгнення системи IDS діляться на три основні типи: - хостова IDS (*англ. Host-based IDS - HIDS*); - мережева IDS (*англ. Network-based IDS - NIDS*) та гібридна IDS [2]:

- HIDS розміщується на одному з мережевих пристроїв (*сервер чи робоча станція*), де дані аналізуються локально. Прикладом хостової IDS є «AIDE» та «Tripwire»;
- NIDS розміщується в «чутливій» точці мережевої інфраструктури, де є можливість доступу до всіх пристроїв мережі, або пристроїв в межах її окремого сегменту [8,9]. Наприклад, її можливо встановити у підмережі, де розташовано MME, щоб запобігти можливого втручання до нього. NIDS також називають «сніфером пакетів», оскільки він фіксує пакети, що проходять через усі комунікаційні носії. Характерним прикладом мережевої IDS є «Snort» ver. 2.9.15 [18], якій може аналізувати на предмет можливих аномалій різні типи протоколів (TCP, UDP, ICMP, IP);
- гібридна IDS забезпечує логічне доповнення до NIDS та HIDS, і є точкою центрального управління процедурами виявлення вторгнень [19].

3 Особливості синтезу моделі загроз

Модель загроз ІБ повинна містити загальний опис інформаційної системи (ІС) і її структурно-функціональних характеристик, а також опис загроз безпеки інформації, що включає опис можливостей порушників (модель порушника), опис найбільш характерних вразливостей даної ІС та відповідних способів реалізації загроз безпеці інформації. Таким чином модель загроз ІБ містить формалізований опис методів та засобів здійснення загроз для інформації [20]. В любому випадку, принциповим є те, що практично ніяка ІС не може вважатися абсолютно захищеною в повному розумінні цього ствердження, а тільки в рамках конкретної моделі загроз та впродовж заздалегідь визначеного терміну часу.

Як правило в рамках моделі загроз для кожної конкретної ІС, вирішуються наступні завдання:

- контроль забезпечення рівня захищеності даних ІС;
- попередження впливу на технічні засоби ІС, в результаті якого може бути порушено або змінено їх функціонування;
- аналіз захищеності від загроз безпеки ІС на корпоративному рівні, та виконання робіт щодо підтримки потрібного (*визначеного*) рівня ІБ;
- синтез сигнатур моделей загроз безпеки з урахуванням їх призначення, умов та особливостей функціонування ІС.

Зазвичай серверна частина ІС, фактично, являє собою фізичну електронну обчислювальну машину (або кластер відповідних машин), що розташована в точці мережевої інфраструктури, де є доступ до всіх пристроїв у мережі. Прикладне ПЗ, в більшості випадків, є серверним додатком. Адміністратори мережі або фахівці підрозділу ІБ можуть здійснювати доступ до потрібних елементів та інформаційних ресурсів ІС, як, безпосередньо, з вузлів локальної мережі [8], так і через «зовнішню» мережу Інтернет з використанням захищених каналів зв'язку (*не у всіх випадках, та ні для всіх ІС, що регламентується відповідними вимогами ІБ для кожної окремої ІС* [9-10]).

Стислий перелік найбільш типових вразливостей що притаманні для більшості сучасних ІС можна представити у наступному вигляді (в межах складання моделі загроз):

- сканування мереж, за допомогою якого можливе проведення нелегітимне дослідження безпеки мережі та виявлення активних мережевих сервісів;
- атаки на відмову в обслуговуванні, що спрямовані на обмеження доступу легітимних користувачів до загальнодоступних мережевих служб або ресурсів;
- DNS-запити на сторонні DNS-сервери, які можуть бути порушенням вимог корпоративної політики. Обхід відповідних обмежень може бути наслідком роботи шкідливого ПЗ, що є серйозною загрозою для безпеки корпоративної мережі;
- атаки методом «грубої сили», які полягають в незаконному отриманні автентифікаційних пар (ім'я користувача і пароль) шляхом спроби перебору різних варіантів для входу (отримання доступу) в мережеві служби;
- шкідливий або незвичайний трафік, який потенційно може вказувати на скомпрометовану систему [21].

Модель зловмисника інформаційної безпеки – це набір припущень про одного або декількох можливих порушників інформаційної безпеки, їх кваліфікацію, та їх технічні і матеріальні засоби і таке ін. Загальний рівень потенціалу порушника згідно нормативному акту – перший, порівнянний з групою хакерів [22]. За замовчуванням передбачається, що потенційний порушник має канал зовнішнього доступу з мережі Інтернет, за допомогою якого він намагається отримати доступ до корпоративних ресурсів, які захищаються. При цьому порушник може дотримуватися як пасивної стратегії, без порушень інформаційного обміну (здійснювати непомітну розвідку), так і активної, коли він намагається вплинути на інформаційні процеси, що відбуваються в кожної ІС. Як свідчить досвід відомих інцидентів, в більшої частині цих випадків, порушників крім використання типових вразливостей, що були зазначені вище, поєднує деяка обмеженість їх фінансових ресурсів. Це факт потрібно враховувати при побудові системи захисту, крім того, за рідким виключенням, потенційному порушнику достовірно не відомо повний опис порядку, послідовності і параметрів процедур, що виконуються на хості (або мережі), якій він атакує. Як правило обчислювальна потужність технічних засобів порушника поєднує його персональний комп'ютер (як засіб управління), та розгалужену систему чужих обчислювальних засобів або навіть мереж (так званих бот-систем), та «власні» канали зв'язку з високою пропускною здатністю. Для адекватності оцінки рівня потенційного збитку, якій може завдати порушник, рівень його знань в області ІТ повинно враховувати, як високий. Слід мати на увазі, що при підготовці до атаки порушником може використовуватися пошук нових або відомих вразливостей, а може бути синтезована нове шкідливе ПЗ, яке втілює результати попередньої розвідки (*нелегітимного спостереження*) мережі. В цьому контексті можуть бути докладені певні зусилля для отримання уявлення про принципи функціонування системи захисту [8], та внесені потайні адресні зміни в роботу системи. В цілому порушник діє приховано, звичайно до моменту досягнення поставленої мети або появи, частиш всього, не врахованої їм будь-якої серйозної перешкоди [8,9,11].

В цілому, постановка задачі щодо захисту ресурсів конкретної корпоративної системи полягає у виборі місця розташування і наступного конфігурування ММЕ корпоративного рівня, а саме складання відповідних правил його роботи в різних умовах функціонування мережі (*або певного сегменту цієї мережі*) та налаштування алгоритму взаємодії з елементами систем виявлення вторгнень та/або захисту від витоку даних (*в разі їх використання*).

4 Специфіка настроювання «Snort»

Як вже вказувалося раніше, NIDS «Snort» [18,21] може бути налаштована для роботи в декількох різних режимах:

- як IDS – режим аналізу та документування пакетів («інформер»). В цьому випадку сніфер пакетів буде зчитувати мережеві пакети та відображати їх у консолі, відповідно до встановлених правил;

- як IPS – режим «у розрив», який доповнює режим IDS та взаємодіє зі службою операційної системи (ОС) «netfilter».

У загальних рисах «Snort» функціонує, як це показано на рис. 1. Мережева IDS «Snort» буде відслідковувати вторгнення, які орієнтовані на внутрішні сервери, тому найкращім місцем розміщення для її датчика є точка, поза MME (зі сторони внутрішньої мережі), що надає можливість аналізувати трафік у відповідному сегменті мережі та оперативно оновлювати списки актуальних блокувань [23], чим і забезпечується парирування атак.

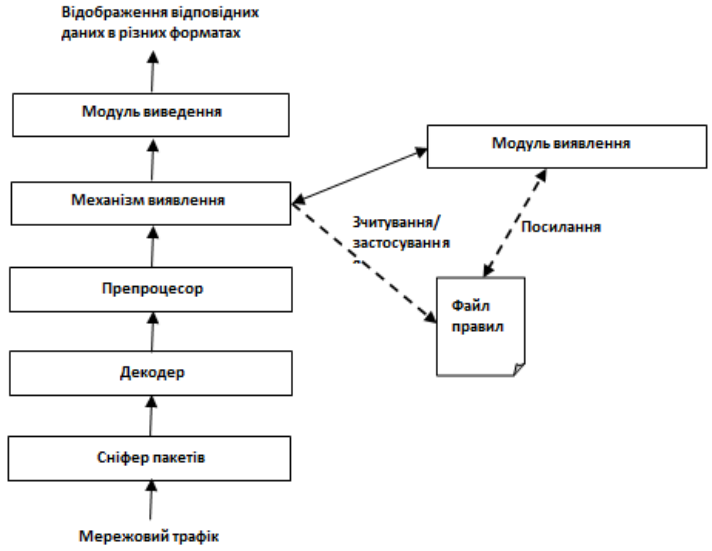

Рис. 1 – Спрощена схема функціонування IDS «Snort»

Важливо зазначити, що модулі декодування пакетів не можуть повноцінно здійснювати розбір зашифрованого трафіку, тому для покращення ефективності роботи систем IDS слід використовувати сервіси які здійснюють зовнішнє розшифрування трафіку [24]. Мережева IDS «Snort» використовує правила, написані простою, та в той же час гнучкою мовою. В основі правил його логіки роботи використовується булева алгебра, де під істиною розуміється легітимний пакет даних, а під хибністю – факт ймовірного вторгнення.

В загальному випадку, структура правил для NIDS «Snort» виглядає наступним чином:

```
<Режим дії> <Протокол> <IP-адреса джерела> <Порти джерела> <Оператор напряму> <IP-адреса одержувача> <Порти одержувача> (ключ_1 : значення_1; ключ_2 : значення_2; ... ключ_N : значення_N;).
```

За замовченням, усі правила встановленні в «режимі оповіщення» (*Alert*), але в разі виникнення потреби можливо їх перемикання в «режим блокування пакетів» (*Drop* або *Reject*). В разі збігу сигнатури з відомостями сигнатури загроз, інцидент буде зареєстрований, та відбудеться дія (*або їх послідовність*), що передбачена відповідним правилом «Snort». Приклад характерних правил роботи «Snort» представлено в таблиці 1.

В табл. 1 символ $ позначає діапазон портів та IP-адрес за замовчуванням, які встановлені у файлі конфігурації «Snort». Для деяких правил потрібно встановити додаткові опції, наприклад такі як:

- для #101000001 встановлюється ключ «*detection_filter*» зі значеннями «*track by_src*», «*count 1000*» та «*seconds 5*»;
- для #101000002 встановлюється ключ «*detection_filter*» зі значеннями «*track by_src*», «*count 100*» та «*seconds 5*»;
- для #101000003, #101000004, #101000005, #101000006, #101000007, #101000008 та #101000009 встановлюється ключ «*flags*» зі значенням «S»;
- для #100000100 встановлюється ключ «*detection_filter*» зі значеннями «*track by_src*», «*count 100*» та «*seconds 10*», ключ «*flags*» зі значенням «S»;
- для #100000200 встановлюється ключ «*detection_filter*» зі значеннями «*track by_src*», «count 100» та «seconds 10», ключ «*flags*» зі значенням «A»;
- для #101100002 встановлюється ключ «*content*» зі значенням «|68 C6 0E 34|»;
- для #101100003 встановлюється ключ «*content*» зі значенням «malicious|04|site»;

Таблиця 1 – Правила роботи Snort *(варіант)*

Type	SID	Prot.	Source IP	Source Port	Recipient IP	Recipient Port	Message
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Network-scan	101000001	TCP	$external	Any	$home	Any	tcp scan
	101000002	UDP	$external	Any	$home	Any	udp scan
	101000003	TCP	$external	Any	$home	$ssh	tcp-syn to ssh
	101000004		$external	Any	$home	139	tcp-syn to netbios
	101000005		$external	Any	$home	$smtp	tcp-syn to smtp
	101000006		$external	Any	$home	110	tcp-syn to pop3
	101000007		$external	Any	$home	143	tcp-syn to imap
	101000008		$external	Any	$home	$ftp	tcp-syn to ftp
	101000009		$external	Any	$home	$sip	tcp-syn to sip
Attempted-dos	100000100		$external	Any	$home	$http	dos syn-flood attempt
	100000200		$external	Any	$home	$http	dos attempt
Policy-violation	101100001	UDP	$dns	53	$home	Any	dns from blocked server
	101100002		$dns	53	$home	Any	dns from blocked server
	101100003		$dns	53	$home	Any	dns from blocked server
Suspicious-login	100000010	TCP	$external	Any	$home	$ssh	ssh auth brute force attempt
	100000020		$external	Any	$home	$smtp	smtp auth brute force attempt
	100000030		$home	$http	$external	Any	directory brute force attempt
	100000040		$home	$http	$external	Any	directory brute force attempt
Bad-unknown	100000001	ICMP	$external	Any	$home	Any	large icmp packet

- для #100000010 встановлюється ключ «*detection_filter*» зі значеннями «*track by_dst*», «count 10» та «seconds 60»;
- для #100000020 встановлюється ключ «detection_filter» зі значеннями «track by_dst», «count 10» та «seconds 60»;
- для #100000030 встановлюється ключ «*content*» зі значенням «HTTP/1.1 403», ключ «*depth*» зі значенням «12»;
- для #100000040 встановлюється ключ «*content*» зі значенням «HTTP/1.1 405», ключ «*depth*» зі значенням «12»;
- для #100000001 встановлюється ключ «*dsize*» зі значенням «>800».

5 Етапи налаштувань та деякі особливості оцінки рівня захисту ММЕ

Для оцінки рівня захисту корпоративного ММЕ, необхідно передбачити виконання наступних етапів:

1. адміністратори мережі або фахівці з ІБ виконують підключення до корпоративної системи з локальної мережі, або через мережу Інтернет з використанням захищених каналів зв'язку. Після цього виконується первинне конфігурування «Snort» [18, 21], реда-

гування файлу конфігурації бібліотек DAQ та налаштування діапазонів портів і IP-адрес за замовчуванням, а також встановлюються можливі режими дії правил;

2. виконується запуск системи «Snort» у режимі IPS, та відбувається налаштування взаємодії зі службою міжмережевого екрану:
   - *запуск «Snort» здійснюється командою*:
   `snort -A console --daq-var queue=0 -u snort -g snort -c /etc/snort/snort.conf -Q`;
   - *взаємодія зі службою MME забезпечується командою*:
   `iptables -t nat -I PREROUTING -j NFQUEUE --queue-num 0 && iptables -I FORWARD -j NFQUEUE --queue-num 0`;
3. в разі збігу сигнатури правил з файлом правил, системою генерується та передається до модулю виведення відповідне інформаційне сповіщення, що містить наступні елементи:
   - *дату та час події;*
   - *виконану дію (реакцію NIDS) в залежності від здійснених налаштувань системи;*
   - *ідентифікатор та опції застосованого правила.*
4. якщо система налаштована на режим роботи IDS, то персонал, що відповідає за питання ІБ, буде отримувати відповідні сповіщення про виявлену загрозу, а хибний та нелегітимний трафік при цьому не блокується. При цьому для запобігання визначеної загрози необхідні певні дії персоналу, у відповідності з рішенням адміністратора мережі або іншої відповідальної особи;
5. якщо система налаштована на режим IPS (*автоматичний режим проведення захисних реакцій*), то визначені співробітники додатково отримують сповіщення про поточний стан виконаного блокування трафіку. В такому випадку легітимний трафік не блокується, а нелегітимний трафік блокується відповідно до заздалегідь встановлених правил. В цьому разі подальші дії персоналу не потрібні.

Для оцінки рівня захисту MME корпоративного рівня за допомогою певних показників необхідно оцінити загальний рівень захищеності ІС. Результат відповідної оцінки повинен містити наступні елементи [25]:

- визначення середовища загрози;
- оцінка загрози (*об'єкт/об'єкти можливої атаки*);
- можливий сценарій реалізації загрози;
- обробка загрози (*здійснювані заходи щодо відбиття загрози*);
- ризик реалізації загрози (*пріоритети можливих атак*).

Для попередньої «розвідки» і дослідження поточного рівня безпеки мережі-жертви, а також виявлення активних мережевих процесів, потенційний комп'ютерний порушник може використовувати повністю легальні програмні інструменти для мережевого адміністрування, наприклад, такий як «*nmap*» (див. табл. 2).

Таблиця 2 – Поширені типи загроз

Середовище загрози	№	Сценарій загрози	Обробка загрози (SID)	Об'єкт атаки	Режим дії	Пріоритет атаки
Сканування мереж	1	nmap -T4 -A -v [IP]	101000001	Всеспрямовані	Alert/Drop	Середній
	2	nmap -sS -sU -T4 -A -v [IP]	101000002	Всеспрямовані	Alert/Drop	Середній
	3	nmap -T4 -F [IP]	101000003+	Всеспрямовані	Alert	Середній
	4	nmap [IP]	101000003+	Всеспрямовані	Alert	Середній
Атаки на відмову в обслуговуванні	1	hping3 -S -d 200 -p 80 --flood [IP]	100000100	HTTP-сервери	Alert/Drop	Високий
	2	hping3 -S -d 65495 -p 80 --flood [IP]	100000100	HTTP-сервери	Alert/Drop	Середній
	3	hping3 -i u1000 -A -d 200 -p 80 [IP]	100000200	HTTP-сервери	Alert/Drop	Середній

Продовження Табл. 3

Середовище загрози	№	Сценарій загрози	Обробка загрози (SID)	Об'єкт атаки	Режим дії	Пріоритет атаки
Виявлення DNS-запитів	1	nslookup www.malicious.site 1.1.1.1	101100003+	Всеспрямовані	Alert/Drop	Низький
	2	nslookup www.malicious.site 8.8.8.8	101100000+	Всеспрямовані	Alert/Drop	Низький
Атаки методом «грубої сили»	1	patator smtp_login host=[IP] user=admin password=FILE0 0=passlist.txt	100000020	SMTP-сервери	Alert/Drop	Середній
	2	patator ssh_login host=[IP] user=admin password=FILE0 0=passlist.txt -x ignore:mesg='Authentication failed.'	100000010	SSH-сервери	Alert/Drop	Високий
	3	hydra -l admin -p passlist.txt [IP] -t 4 ssh	100000010	SSH-сервери	Alert/Drop	Високий
	4	curl [IP]/server-status/	100000030	HTTP-сервери	Alert	Низький
	5	curl -X DELETE [IP]/	100000040	HTTP-сервери	Alert	Низький
Шкідливий трафік	1	hping3 -1 -c 1 -d 1000 [IP]	100000001	Всеспрямовані	Alert	Низький

В разі підготовки та проведення атаки на відмову в обслуговуванні та генерації шкідливого трафіку зловмисник може використовувати інструмент генерації пакетів «*hping3*». Для здійснення спроби проведення нелегітимних DNS-запитів на сторонні DNS-серверу – інструмент мережевого адміністрування «*nslookup*». Для проведення атаки методом «грубої сили» порушник може використовувати відповідні інструменти для перебору пар «логін-пароль», наприклад такі як ,«*hydra*» та «*patator*», або інструмент мережевого адміністрування «*cURL*» [26-31]. Порівняльний аналіз найбільш поширених/характерних загроз безпеки наведено в таблиці 2.

3 Висновки

На даний час існує досить широкий спектр методів та інструментів для зміни та генерації шкідливого мережевого трафіку, що дозволяє здійснювати атаки на різні ІС, як із зовні, так і ініціюючи його всередині периметра безпеки окремої структури.

При вірної побудові та коректному настроюванні сучасних систем захисту, зловмисний трафік «відстежується» відповідними датчиками системи IDS, що щільно взаємодіє з MME, які розташовані в різних сегментах корпоративної мережі.

Зазначено, що для найбільш поширених загроз безпеки, рівень захисту корпоративного MME в значної мірі грунтується на відповідних правилах системи виявлення вторгнень (*наприклад «Snort», у разі її використання*).

Підкреслено, що MME корпоративного рівня, якій в межах заходів відбиття загроз безпеки, щільно взаємодіє із IDS (*наприклад «Snort»*), дозволяє досить ефективно розпізнавати не тільки зовнішні вторгнення, але й внутрішні атаки, а також виявляти загрози, які неможливо розпізнати звичайним MME (*в разі його поодинокого використання, тобто без залучення IDS, DLP та Honeypot рішень*).

Вдале розміщення датчиків системи IDS та її інтеграція в з одним чи кількома корпоративними MME [8, 9] дозволяє в реальному часі виявляти такі загрози, як: - сканування мереж; - атака на відмову в обслуговуванні; - атаки методом «грубої сили»; - несанкціонований виток даних; - породження та циркуляція нелегітимного трафіку та ін.

З метою зменшення рівня та масштабів потенційних втрат, що можуть статися в наслідок недбалого ставлення персоналу з питань ІБ [10], в межах діючої комплексної системи безпеки (MME_n + *IDS/DLP*, де *n* – кількість впроваджених корпоративних MME), необхідно передбачити і активувати алгоритм автоматичних захисних реакцій системи захисту при кратності випадків бездіяльності персоналу. Перелік відповідних індикативних критеріїв та часо-

вих тайм-аутів повинен формулюватися і постійно коригуватися с залученням керівного складу компанії (установи). В такому випадку можливо забезпечити цілісність реалізації задуму стосовно стратегії та тактики парирування загроз і мінімізувати час бездіяльності системи, навіть в умовах самоусунення (*як умисного так і ненавмисного*) уповноваженого персоналу.

Для додаткового посилення захисту корпоративних ресурсів, актуальним напрямом діяльності слід вважати впровадження деяких «адресних» рішень, що передбачає розробку більш локалізованих, та не поширених серед інших користувачів, алгоритмів і механізмів протидії нелегітимному впливу (*наприклад, децентралізація окремих функцій адміністрування безпеки або впровадження алгоритмів санкціонування певних процедур/дій*).

Посилання

[1] О. Р. Лапонина, *Межсетевое экранирование*, Бином, 2014.
[2] A. Sh.Ashoor, "Importance of Intrusion Detection System", *International Journal of Scientific Engineering Researc*. – pp. 7.
[3] J. P. Anderson, "Computer Security Threat Monitoring And Surveillance", pp. 56, 1980.
[4] T. F. Lunt et al., *A Real-Time Intrusion-Detection Expert System (IDES)*, SRI International, pp. 166,1992.
[5] *National Information Systems Security '95*, 1996
[6] The Evolution of Intrusion Detection Systems, 2001.[Online]. Available: https://www.symantec.com/connect/articles/evolution-intrusion-detection-systems.
[7] "Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems", *Independent Study*, pp. 17, 2003.
[8] Дж. Маллери и др., *Безопасная сеть вашей компании*, Москва: НТ Пресс, 2007.
[9] О.С. Ріпний, О.О. Дьяченко, С.В. Малахов, "Особливості функціонування систем IDS та IPS при реалізації спроб несанкціонованого доступу до корпоративних ресурсів", *Матеріали IX міжнародної НТК. 11-12.04.2019*, Харків: НТУ "ХПІ", с.95, 2019.
[10] В.В. Сербин, С.В. Малахов, "Захист від несанкціонованих дій в сучасних інформаційних системах", *Проблеми інформатизації: Матеріали VII міжнародній НТК. 13-15.11.2019*, т.1: секції 1-3, Ч: ЧДТУ, 2019, с.119.
[11] А.Тарасенко, *Технология Honeypot*, Ч.1: Назначение Honeypot. [Online]. Available: https://www.securitylab.ru/analytics/275420.php
[12] "Global number of cyber security incidents from 2009 to 2015", *Statista Research Department*. – 2015. [Online]. Available: https://www.statista.com/statistics/387857/number-cyber-security-incidents-worldwide/. Accessed on 24.12.2018.
[13] Д. В.Чепмен-мл., Э. Фокс, *Брандмауэры Cisco Secure PIX*, Вильямс, 2003.
[14] What is netfilter.org? [Online]. Available: https://www.netfilter.org/. Accessed on 21.12.2019.
[15] *Ethical Hacking and Countermeasures: Secure Network Infrastructures*, 2009.
[16] M. E. Whitman, *Principles of Information Security*, 2009.
[17] E. Kirda et al., "Recent Advances in Intrusion Detection", *12th International Symposium*, 2009.
[18] *What is Snort?* [Online]. Available: https://www.snort.org/faq/what-is-snort. Accessed on 27.11.2019.
[19] *Prelude Log Monitoring Lackey Manual*. [Online]. Available: https://www.prelude-siem.org/projects/prelude/wiki/PreludeLml . Accessed on 12.11.2019.
[20] *Про затвердження Положення про державний контроль за станом технічного захисту інформації*, 2007. [Online]. Available: https://zakon.rada.gov.ua/laws/show/z0785-07. Accessed on 12.08.2019.
[21] *Snort Users Manual*. Node 29, 2019. [Online]. Available: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node29.html. Accessed on 12.11.2019.
[22] *Про затвердження Положення про порядок розроблення, виробництва та експлуатації засобів криптографічного захисту інформації*, 2007. [Online]. Available: https://zakon.rada.gov.ua/laws/show/z0862-07. Accessed on 05.09.2019.
[23] *Snort IPS With NFQ (nfqueue) Routing on Ubuntu*, 2017. [Online]. Available: http://sublimerobots.com/2017/06/snort-ips-with-nfq-routing-on-ubuntu/. Accessed on 16.09.2019.
[24] "Encrypted Traffic Analytics with the New Cisco Network and Stealthwatch", *Cisco public*, pp. 52, 2019.
[25] "NIST Special Publication 800-30", *National Institute of Standards and Technology*, pp. 95, 2012.
[26] *Patator Github page*. [Online]. Available: https://github.com/lanjelot/patator. Accessed on 02.11.2019.
[27] *Hydra Github page*. [Online]. Available: https://github.com/vanhauser-thc/thc-hydra. Accessed on 02.11.2019.
[28] *cURL Man Page*. [Online]. Available: https://curl.haxx.se/docs/manpage.html. Accessed on 02.11.2019.
[29] *nslookup Man Page*. [Online]. Available: https://manpages.debian.org/jessie/dnsutils/nslookup.1.en.html. Accessed on 02.11.2019.
[30] Getting started with hping 3. [Online]. Available: http://wiki.hping.org/94. Accessed on 02.11.2019.
[31] nmap Man Page. [Online]. Available: https://nmap.org/book/man.html#man-description. Accessed on 02.11.2019.

Reviewer: Roman Oliynikov, Doctor of Sciences (Engineering), Full Prof., JSC "Institute of Information Technologies", 12 Bakulin St., Kharkiv, 61166, Ukraine.
E-mail: roliynykov@gmail.com

Received: November 2019.

Authors:
Denis Rondalev, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: denisrondalev@gmail.com

Olga Melkozerova, Ph.D., Senior Lecturer, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: olja.mex@gmail.com
Oleksii Nariezhnii, Ph.D., Associate Professor, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: o.nariezhnii@karazin.ua

Features of the functioning of the corporate firewall and issues of interaction with the IDS system.

Abstract. Offers a brief overview of the features of using a corporate firewall and the issues of interaction with the elements of the IDS system. Some important features of synthesis of the threat model are considered. Attention was drawn to the importance of correct IDS system setup (*Snort*). The basic stages of configuration and some features of the assessment of the level of protection of the corporate firewall are highlighted. Attention is drawn to the importance of network resource segmentation and IDS system sensor placement.

Keywords: Snort; IDS; IPS; DLP; Firewall.

Рецензент: Роман Олейников, д.т.н., проф., ЧАО "Институт информационных технологий", ул. Бакулина, 12, Харьков, 61166, Украина. E-mail: roliynykov@gmail.com

Поступила: Ноябрь 2019.

Авторы:

Денис Рондалев, студент факультета компьютерных наук, Харьковский национальный университет им. В. Н. Каразина, Харьков, Украина.
E-mail: denisrondalev@gmail.com
Ольга Мелкозерова, к.т.н., ст. преподаватель, каф. безопасности информационных систем и технологий, Харьковский национальный университет им. В. Н. Каразина, Харьков, Украина.
E-mail: olja.mex@gmail.com
Алексей Нарежний, к.т.н., доцент, каф. безопасности информационных систем и технологий, Харьковский национальный университет им. В. Н. Каразина, Харьков, Украина.
E-mail: o.nariezhnii@karazin.ua

Особенности функционирования корпоративного брандмауэра и вопросы взаимодействия с системой IDS.

Аннотация. Предложен краткий обзор особенностей использования корпоративного межсетевого экрана и вопросов взаимодействия с элементами системы обнаружения вторжений. Рассмотрены некоторые важные особенности синтеза модели угроз. Обращено внимание на важность корректной настройки системы обнаружения вторжений (Snort). Выделены основные этапы настройки и некоторые особенности оценки уровня защиты корпоративного брандмауэра. Обращено внимание на важность вопросов сегментации сетевых ресурсов и размещения датчиков системы обнаружения вторжений.

Ключевые слова: Snort; IDS; IPS; DLP; межсетевой экран.

УДК 004.056.5 DOI: 10.26565/2519-2310-2019-3-03

SQL-ИНЪЕКЦИИ: ОБЗОР ПОТЕНЦИАЛЬНЫХ СПОСОБОВ ЗАЩИТЫ

Юрий Попов, Сабина Рузудженк, Карина Погорелая

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
yuriipopov18@gmail.com, ruzudzhenk.jb@gmail.com, karina.pogorelka@gmail.com

Рецензент: Ирина Лисицкая, д.т.н., проф., ХНУ имени В.Н. Каразина, пл. Свободы, 6, Харьков, 61022, Украина
lisitska@karazin.ua

Поступила: Ноябрь 2019.

Аннотация. *В статье представлен краткий обзор известных техник взлома программ и веб-сайтов, работающих с базами данных. На основе проведенного анализа основных разновидностей SQL-атак выделены наиболее серьезные типы угроз: – внутриполосная, слепая и внеполосная. Утверждается, что по совокупности характеристик, внеполосная SQL-атака является наиболее опасной. Обращено внимание на необходимость периодического тестирования и мониторинга веб-сайтов, что является актуальным средством защиты от SQL-инъекций. Отмечено, что наилучший метод тестирования – попытка подвергнуть код SQL-инъекции. Рассмотренные способы защиты способны повысить общий уровень безопасности программных продуктов от атак типа SQL-инъекция, обеспечивают корректную работу приложений и целостность пользовательских данных.*

Ключевые слова*: SQL-инъекция; базы данных; внеполосная SQL-атака; внутриполосная SQL-атака.*

1 Введение

В последние годы практически во всех современных компаниях, работающих в сфере высоких технологий, всё большую популярность приобретает тенденция использования в бизнес-процессах разнообразных web-приложений, информационные ресурсы которых, обрабатывают и хранят персональные данные клиентов, компаний подрядчиков и, непосредственно, владельцев компаний [1]. Использование web-приложений наделяет производственные и бизнес-процессы новыми качествами, прежде всего, такими как: - высокая мобильность бизнеса; - доступность сервисов; - непрерывность бизнес-процессов; - масштабируемость получаемого эффекта и т.п.. Учитывая все эти обстоятельства, вопросы обеспечения информационной безопасности (ИБ) при обработке и хранении персонифицированной и «чувствительной» корпоративной информации, сохраняют высочайший приоритет и являются крайне актуальным направлением деятельности, как для специалистов соответствующих подразделений компаний (отделов и служб ИБ), так и для профильных специалистов отрасли ИБ. Так, согласно статистике Positive Technologies [2], около 70% веб-сайтов поддаются различным атакам, среди которых одно из первых мест занимают атаки типа SQL-инъекция.

SQL-инъекция – одна из самых распространённых техник взлома программ и веб-сайтов, работающих с различными базами данных [3]. Атака, как правило, производится на основе внедрения в различные типы запросов некорректных SQL операторов, что позволяет злоумышленнику получить, практически, полный несанкционированный доступ к соответствующей базе данных (БД), локальным файлам, а также возможность удалённого выполнения произвольных операций на сервере. Кроме того, SQL-атаки, зачастую, являются результатом неэкранированного ввода, передаваемого сайту и используемого как часть запроса к БД [3].

Таким образом, вопросы организации противодействия атакам типа «SQL-инъекция» являются актуальным направлением деятельности и требуют постоянной модификации уже существующих и разработки новых способов защиты и методик противодействия.

2 Механизм атаки на основе SQL-инъекции

Атака на основе SQL-инъекции производится путем запросов к БД на основе вводимых пользователем данных с применением некорректно фильтруемых escape-символов.

С точки зрения архитектуры самого приложения пользователь взаимодействует с веб-сервером, посредством веб-клиента, который взаимодействует с SQL-сервером, по протоколу HTTP (Рис. 1). Как правило, веб-сервер требует от пользователя аутентификации в системе (пользователю необходимо предоставить *name* и *password*). Для этого веб-сервер выполняет операцию: *SELECT * FROM Users WHERE userid='name' AND userpass='password'*, где *Users* – таблица БД, содержащая персональные данные пользователей; *name* – имя пользователя; *password* – пароль, соответствующий имени пользователя. При определённом подборе злоумышленником информации для ввода, есть вероятность обхода действующего механизма идентификации в системе и возможность последующей модификации исполняемого запроса (*например, использование комментариев дает возможность деактивировать проверку пароля*) [4].

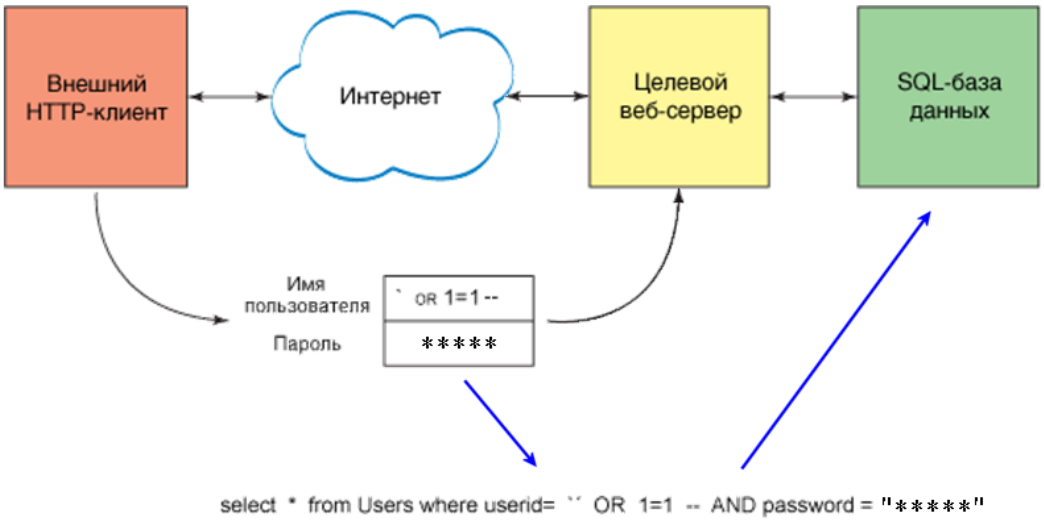

Рис. 1 – Механизм атаки на основе SQL-инъекции

Таким образом, данные, вводимые на веб-странице, способны использовать уязвимости при обмене с БД. SQL-инъекция использует введенные пользователем некорректные данные для получения разрешения на прямое взаимодействие с внутренней БД, в результате чего атакующий (хакер) получает доступ ко всем записям данной таблицы БД.

2.2 Основные разновидности SQL-атак

В настоящий момент известно несколько типов атак, применяющих SQL-инъекции. Необходимо отметить, что, в целом, уязвимость типа SQL-инъекция известна уже на протяжении семнадцати лет, однако продолжает интересовать специалистов по безопасности до сих пор. Эта разновидность уязвимости впервые была описана в декабре 1998 года на примере сервера Microsoft SQL, в котором было возможно получение конфиденциальных данных посредством использования команд в обычных пользовательских вводах, таких как «имя» или «номер телефона» (rain.forest.puppy, 1998) [8]. Хотя это событие и было впервые задокументировано в 1998, однако SQL-инъекция не привлекала большого внимания в сообществе информационной безопасности, вплоть до 2002 года. По мере развития SQL-инъекций, появлялись их новые типы, основное отличие которых заключается в способах и сложности их внедрения, а также в используемых способах защиты (будут рассмотрены ниже). Коротко рассмотрим 3 основных типа атак, применяющих технику SQL-инъекции [5].

Внутриполосная SQLi (классическая SQLi). Данный тип является самым распространенным, инъекция, в основном, происходит, когда потенциальный злоумышленник использует один и тот же канал связи для запуска атаки и последующего сбора результатов.

При этом внутриполосные SQL-инъекции делятся на две разновидности:

- SQL-инъекция на основе ошибок. Основана на сообщении об ошибке, выдаваемом сервером БД, для получения информации о её структуре;

- внедрение SQL на основе объединения. Основана на использовании оператора SQL UNION для объединения результатов двух или более операторов SELECT в один результат, который затем возвращается, как HTTP ответ [6].

Инференциальная SQL-инъекция (слепая или Blind SQLi). При атаках с использованием слепой SQL-инъекции злоумышленник не может увидеть результат своей атаки внутри группы, поскольку данные не передаются через веб-приложение. По этой причине она также называется Blind SQLi [6]. Инференциальные SQL-инъекции бывают двух типов:

- булевая слепая SQLi. Основана на отправке SQL-запроса в базу данных, что вынуждает приложение возвращать другой результат в зависимости от того, возвращает ли запрос результат «TRUE» или «FALSE»;

- базирующаяся на времени слепая SQLi. Основана на отправке соответствующего SQL-запроса к БД, что заставляет базу данных ждать определенное время (в секундах), прежде чем ответить. Собственно, время ответа и укажет злоумышленнику, является ли результатом запроса «ИСТИНА» или «ЛОЖЬ».

Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах.

Внеполосная SQLi. Внедрение SQL-инъекции происходит, когда злоумышленник не может использовать один и тот же канал для запуска атаки и сбора результатов.

Внеполосные методы предлагают злоумышленнику альтернативу логическим методам SQL-инъекции, основанным на времени, особенно если ответы сервера не очень стабильны, делая вывод, основанный на времени, ненадежным [6].

Эти методы зависят от способности сервера БД делать DNS (*Domain Name Server*) или HTTP-запросы для доставки необходимых данных хакеру.

Анализ проблематики по данному направлению позволяет утверждать, что наибольшую угрозу целостности информации представляют именно вышеперечисленные типы SQL-атак. При комплексном использовании данных разновидностей атак (*интеграции или коллаборации друг с другом*) практически не остается возможности сохранить конфиденциальность персональных данных. Соответственно, так как утечка информации подобного рода крайне недопустима, то владельцам различных web-приложений необходимо продумывать адекватную стратегию защиты, демпфирующих возможный эффект от подобного рода атак.

3 Способы защиты программных продуктов от атак типа SQL-инъекция

Во избежание подобного рода атак необходимо максимально ограничить программному продукту доступ к серверным данным, т.е. разработать и внедрить приложения, работающие исключительно с параметризованными запросами. Таким образом, при атаках соответствующих приложений, не имеющих достаточных прав доступа к исходным таблицам, исключается возможность получения злоумышленником нелегитимного доступа к локальным данным или БД. В этой связи коротко рассмотрим наиболее известные способы защиты от атак типа SQL-инъекция, а также выделим рекомендуемые меры защиты при разработке компонентов БД и сформируем общие рекомендации для разработки web-ориентированных систем, использующих сервера БД.

Первый способ предусматривает необходимость обеспечить фильтрацию данных, поступающих на сервер: - т.е. специальные символы должны экранироваться, а численная информация – подвергаться проверке на вводимый тип. Кроме того необходимо ограничивать вход (*например, количество вводимой информации, после проверки на сервере; запросы, превышающие установленное количество – отклоняются*).

Кроме того, важным аспектом при защите от подобного рода атак является безопасность самого процесса хранения конфиденциальных данных. Так, например, используемая БД не должна содержать такие данные в виде простого текста или таблицы (*пароли должны быть*

хэшированы, а также содержать случайным образом генерируемую строку, добавляемую перед шифрованием и др.) [7].

Вторым способом обеспечения безопасности является использование серверами БД параметризованных запросов. В общем случае параметризованные запросы представляют собой способ передачи данных, при котором внешние параметры передаются серверу отдельно от SQL-запросов. В большинстве языков программирования реализация данных функций уже предусмотрена [7]:

1. *Delphi* – свойство *TQuery.Params;*
2. *Java* – класс *PraparedStatement;*
3. *C#* – свойство *SqlCommand.Parameters;*
4. *PHP* – свойство *MySQLi.*

Третий способ – это максимальное ограничение отображения сообщений об ошибках пользователей (*отображаются общие сообщения об ошибках, возможные для всех сбоев*). Однако, при этом, на стороне сервера необходимо отслеживать все неудачные запросы для возможности их последующего просмотра и анализа (*аудита инцидента*) в случае атаки.

Периодическое тестирование и мониторинг, также, можно смело отнести к довольно эффективным *способам защиты* от SQL-инъекций. При этом наилучшим способом тестирования, является попытка подвергнуть код SQL-инъекции. Существует множество сканеров для обнаружения подобных атак, которые находят уязвимые места, а также тестируют различные разновидности атак.

В целом, следует акцентировать внимание на том, что для обеспечения потенциально более высокого уровня защиты системы, необходимо использовать (*по возможности*) сочетание всех приведенных выше способов защиты. При настройке/программировании таких систем, нужно проверить программный код на выявление уязвимостей и подвергнуть код SQL-инъекции, что в последующем поможет практически мгновенно отслеживать реакцию программы на подобные атаки.

5 Выводы

Анализ информации об известных инцидентах безопасности, обзор соответствующей периодики и обобщение мнения соответствующих отраслевых специалистов позволяет утверждать, что сокрытие уязвимостей и защита конфиденциальных данных являются важнейшими направлениями обеспечения информационной безопасности, которые не теряют своей актуальности и до настоящего времени. Вследствие этого при разработке программных продуктов не стоит пренебрегать приемами проверки и фильтрации данных.

Выделенные в работе способы защиты могут существенно обезопасить программные системы от атак, основанных на механизме SQL-инъекции, уменьшить общую подверженность атакам, а также обеспечить корректную работу приложений и целостность пользовательских данных.

Опираясь на обзор известных способов защиты от атак типа SQL-инъекция и анализ эффективности возможных мер противодействия, следует констатировать, что:

1- к основным способам защиты от атак типа «SQL-инъекция» можно отнести:

а) обеспечение фильтрации данных, поступающих на сервер БД;

б) использование серверами БД параметризованных запросов;

2 - к рекомендуемым мерам предосторожности при разработке компонентов БД, следует отнести:

а) интеграция дополнительных мер безопасности для всей хранимой информации (*например, хэширование паролей или использование ЭЦП*);

б) выявление устранение потенциальных уязвимостей зависящих/независящих от данных (*например, путем периодического тестирования и мониторинга*);

3 - для обеспечения более высокого уровня защиты системы, крайне необходимо комплексирование нескольких способов защиты.

Ссылки

[1] *Популярность языков программирования: рейтинг 2018*. [Электронный ресурс]. Режим доступу: https://techrocks.ru/2018/07/29/programming-languages-popularity-2018/
[2] Статистика Positive Technologies. [Электронный ресурс]. Режим доступу: https://www.ptsecurity.com/ru-ru/research/analytics/
[3] Д. Евтеев, *SQL Injection от А до Я*. [Электронный ресурс]. Режим доступу: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/PT-devteev-AdvancedSQL-Injection.pdf.
[4] П. Яновски та Є.Бурмакин, *Основы веб-хакинга*. [Электронный ресурс]. Режим доступу: white-hat-hacking-ru-sample.pdf.
[5] М. Егоров, "Выявление и эксплуатация SQL-инъекций в приложениях", *Защита информации. INSIDE*, № 2, с. 2-8, 2011.
[6] *SQL инъекции. Проверка, взлом, защита*. [Электронный ресурс]. Режим доступа: https://habr.com/ru/post/130826/.
[7] К.И. Колесникова, Ю.И.Кошкарёва, *SQL-инъекции*. [Электронный ресурс]. Режим доступа: https://docplayer.ru/46070414-Sql-inekcii-nauchnyy-rukovoditel-gapanyuk-yu-e-k-t-n-docent.html
[8] *SQL Injection: The Longest Running Sequel in Programming History*. [Электронный ресурс]. Режим доступа: https://www.researchgate.net/publication/324227697_SQL_Injection_The_Longest_Running_Sequel_in_Programming_History/link/5ac6a25d4585151e80a37b27/download

Reviewer: Irina Lisitska, Doctor of Sciences (Engineering), Full Prof., Academician of the Academy of Sciences of Applied Radio Electronics, V. N. Karazin Kharkiv National University, Kharkiv, 61022, Ukraine.
E-mail: lisitska@karazin.ua

Received: November 2019.

Authors:
Yuri Popov, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine. E-mail: yuriipopov18@gmail.com
Sabina Ruzudzhenk, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: ruzudzhenk.jb@gmail.com
Karina Pogorelaya, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: karina.pogorelka@gmail.com

SQL-injections: an overview of potential protection methods.

Abstract. This work exposes a brief review of well-known hacking techniques for programs and websites working with databases. Based on a comprehensive analysis of the main types of SQL attacks, the most profound threats are identified. They include in-band, blind and out-of-band types of SQL injections. An out-of-band SQL attack is considered to be the most dangerous because of its characteristics' combination. Attention was also paid on the need of periodic testing and monitoring, which is an actual method of protection against SQL injections. It is emphasized, that the best testing method is undertaking code by the SQL injection. The protection methods, reviewed above, can increase the overall security of software products from attacks such as SQL injection, ensure the correct functionality of applications and the integrity of user data.

Keywords: SQL-injection; Protection methods; Data bases (DB); In-band SQLi; Out-Of-Band SQLi.

Рецензент: Ірина Лисицька, д-р тех. наук, проф., Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: lisitska@karazin.ua

Надійшло: Листопад 2019.

Автори:
Юрій Попов, студент факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: yuriipopov18@gmail.com
Сабіна Рузудженк, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: ruzudzhenk.jb@gmail.com
Карина Погоріла, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: karina.pogorelka@gmail.com

SQL-ін'єкції: огляд потенційних способів захисту.

Анотація. У статті надано короткий огляд відомих технік злому програм і веб-сайтів, що працюють з базами даних. На основі проведеного аналізу основних різновидів SQL-атак виділені найбільш серйозні типи загроз: - внутріполосна, сліпа та позасмугова. Стверджується, що за сукупністю характеристик, позасмугова SQL-атака є найбільш небезпечною. Звернуто увагу на необхідність періодичного тестування і моніторингу веб-сайтів, що є актуальним засобом захисту від SQL-ін'єкцій. Відзначено, що найкращий метод тестування - спроба піддати код SQL-ін'єкції. Розглянуті способи захисту здатні підвищити загальний рівень безпеки програмних продуктів від атак типу SQL-ін'єкція, забезпечують коректну роботу додатків та цілісність призначених для користувача даних.

Ключові слова: SQL-ін'єкція; бази даних (БД); внутріполосна SQL-атака; позасмугова SQL-атака.

УДК 519.111: 621.391 DOI: 10.26565/2519-2310-2019-3-04

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ МЕТОДА КОРРЕКЦИИ ОШИБОК ДАННЫХ, ПРЕДСТАВЛЕННЫХ В КЛАССЕ ВЫЧЕТОВ

Сергей Кошман, Виктория Попенко, Анна Кононченко

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
s.koshman@karazin.ua, vita.popenko@gmail.com, akononpro@gmail.com

Рецензент: Алексей Стахов, д.т.н., проф., академик Академии инженерных наук Украины, Международный Клуб Золотого Сечения, Онтарио, Канада
goldenmuseum@rogers.com

Надійшло: Листопад 2019.

Аннотация: *В статье рассмотрен метод исправления однократных ошибок в классе вычетов (КВ). Результаты анализа корректирующих возможностей арифметического кода показали высокую эффективность использования непозиционных кодовых структур в КВ. Для исправления однократных ошибок требуется проведение дополнительных процедур обработки данных, т.е. применение, дополнительно к информационному резервированию, еще и временного резервирования. Приведены примеры исправления однократных ошибок данных, представленных кодом КВ. Рассмотренные примеры подтверждают практическую реализуемость данного метода коррекции ошибок. Использование системы остаточных классов может быть полезно для реализации быстрых компьютерных вычислений с возможностью распараллеливания некоторых процессов. Кроме того, её можно использовать для создания надежных и отказоустойчивых компонентов компьютерных систем.*

Ключевые слова: *непозиционная система счисления в классе вычетов; исправление однократных ошибок данных; арифметическое непозиционное кодирование информации.*

1 Введение

Известно, что для коррекции (*исправления*) ошибок данных необходимо, чтобы представленная кодовая структура обладала определенной (необходимой) корректирующей способностью [1-3]. Для этого, естественно, в кодовую структуру вводится информационную избыточность. Это в полной мере относится и к непозиционным кодовым структурам (НКС), представленным в классе вычетов (КВ) [4-11].

В общем случае процесс коррекции ошибок данных в КВ, как и в двоичной позиционной системе счисления (ПСС), состоит из трёх этапов. Первый этап – контроль данных (определение правильности или неправильности исходного числа A_{KB}). Второй этап - это диагностика неправильного $\tilde{A}_{KB}$ числа (определение одного искажённого остатка $\tilde{a}_i$ по основанию m_i КВ числа $\tilde{A}_{KB}$). И, наконец, третий этап. Это исправление неправильного остатка $\tilde{a}_i$ на истинное a_i число, т.е. исправления неправильного $\tilde{A}_{KB}$ числа (получение правильного числа $A_{KB} = \tilde{A}_{исп.}$). Степень R информационной избыточности (корректирующая способность кода) оценивается величиной минимального кодового расстояния (МКР) $d_{\min}^{(ПСС)}$.

Известно, что в КВ значение МКР определяется соотношением $d_{\min}^{(KB)} = k+1$, где k – количество контрольных оснований в упорядоченном КВ.

2 Основная часть

В рамках данной статьи рассмотрим примеры реализации метода коррекции ошибок НКС $A_{KB} = (a_1 \| a_2 \| ... \| a_{i-1} \| a_i \| a_{i+1} \| ... \| a_n \| ... \| a_{n+k})$ в КВ с минимальной $(k=1)$ дополнительной информационной избыточностью. В этом случае МКР равно $d_{\min}^{(KB)} = 2$.

Рассмотрим соотношение, с помощью которого можно исправить ошибку в данном остатке $\tilde{a}_i$ [11-13].

Пусть в неправильном числе $\tilde{A}=(a_1 \| a_2 \| ... \| a_{i-1} \| \tilde{a}_i \| a_{i+1} \| ... \| a_n \| a_{n+1})$, где $\tilde{A} \geq M$, ошибка $\tilde{a}_i = (a_i + \Delta a_i) \bmod m_i$ достоверно содержится в остатке a_i по модулю m_i. Очевидно, что

$$\tilde{A} = \left(A + \Delta A \right) \bmod M_0 . \tag{1}$$

С учетом того, что величину ошибки можно представить в виде $\Delta A = (0 \| 0 \| ... \| 0 \| \Delta a_i \| 0 \| ... \| 0 \| 0)$, тогда правильное $(A < M)$ число A можно определить в следующем виде:

$$A = (\tilde{A} - \Delta A) \bmod M_0 = \left[\left(a_1 \| a_2 \| ... \| a_{i-1} \| \tilde{a}_i \| a_{i+1} \| ... \| a_n \| a_{n+1} \right) - \left(0 \| 0 \| ... \| 0 \| \Delta a_i \| 0 \| ... \right.\right.$$

$$\left.\left. ... \| 0 \| 0 \right) \right] \bmod M_0 = [a_1 \| a_2 \| ... \| a_{i-1} \| (\tilde{a}_i - \Delta a_i) \bmod m_i \| a_{i+1} \| ... \| a_n \| a_{n+1}] \bmod M_0 .$$

Количественно оценим значение A. Так как число A правильное, т.е. находится в числовом интервале [0, M), тогда должно выполняться следующее неравенство

$$A = \left(\tilde{A} - \Delta A \right) \bmod M_0 < M . \tag{2}$$

С учетом того, что величина ΔA ошибки равняется значению $\Delta A = \Delta a_i \cdot B_i$, то неравенство (2) будет иметь следующий вид:

$$\begin{gathered} \tilde{A} - \Delta a_i \cdot B_i - r \cdot M_0 < M, \text{ или } \tilde{A} - \Delta a_i \cdot B_i - r \cdot M_0 < M_0 / m_{n+1} (r = 1, 2, 3, ...), \\ \tilde{A} - (\tilde{a}_i - a_i) \cdot B_i - r \cdot M_0 < M_0 / m_{n+1}, \quad \tilde{A} - (a_i - \tilde{a}_i) \cdot B_i - r \cdot M_0 < M_0 / m_{n+1}, \\ \left(a_i - \tilde{a}_i \right) \cdot B_i < M_0 / m_{n+1} - \tilde{A} + r \cdot M_0, \quad a_i - \tilde{a}_i < (M_0 / m_{n+1}) / B_i - \tilde{A} / B_i + r \cdot M_0 / B_i, \\ a_i < \tilde{a}_i + (M_0 / m_{n+1}) / B_i - \tilde{A} / B_i + r \cdot M_0 / B_i . \end{gathered} \tag{3}$$

С учетом того, что ортогональный базис для модуля m_i КВ представляется в виде $B_i = \bar{m}_i \cdot M_0 / m_i$, то выражение (3) примет вид:

$$a_i < \tilde{a}_i + (m_i + r \cdot m_i \cdot m_{n+1}) / (\bar{m}_i \cdot m_{n+1}) - \tilde{A} / B_i \quad \text{или}$$

$$a_i < \tilde{a}_i + m_i \left(1 + r \cdot m_{n+1} \right) / (\bar{m}_i \cdot m_{n+1}) - \tilde{A} / B_i \ . \tag{4}$$

Так как значение остатка a_i есть натуральное число, то значение $m_i (1 + r \cdot m_{n+1}) / (\bar{m}_i \cdot m_{n+1}) - \tilde{A} / B_i$ в выражении (4) должно быть целым числом. Поэтому взяв целую часть последнего соотношения, получим формулу для исправления ошибки в остатке $\tilde{a}_i$ числа $\tilde{A}$ в виде

$$a_i = (\tilde{a}_i + [m_i \cdot \left(1 + r \cdot m_{n+1} \right) / (\bar{m}_i \cdot m_{n+1}) - \tilde{A} / B_i) \bmod m_i] . \tag{5}$$

Рассмотрим примеры реализации процесса коррекции данных в КВ.

Пример 1. Осуществить контроль и, при необходимости, провести коррекцию числа $A_{KB} = (0 \| 0 \| 0 \| 0 \| 5)$, заданного в КВ с информационными $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_5 = 7$ и контрольным $m_k = m_5 = 11$ основаниями.

При этом $M=\prod_{i=1}^{n} m_i=\prod_{i=1}^{4} m_i=420$ и $M_0=M\cdot m_{n+1}=420\cdot 11=4620$. Ортогональные базисы B_i $(i=\overline{1,n+1})$ КВ представлены в таблице 1.

Таблица 1 – Ортогональные базисы B_i КВ $(l=1)$

$B_1=(1\|\|0\|\|0\|\|0)=1540,\ \ \bar{m}_1=1$
$B_2=(0\|\|1\|\|0\|\|0)=3465,\ \ \bar{m}_2=3$
$B_3=(0\|\|0\|\|1\|\|0)=3696,\ \ \bar{m}_3=4$
$B_4=(0\|\|0\|\|0\|\|1)=2640,\ \ \bar{m}_4=4$
$B_5=(0\|\|0\|\|0\|\|0)=2520,\ \ \bar{m}_5=6$

I. Контроль данных $A_{KB}=(0\|\|0\|\|0\|\|5)$. В соответствии с процедурой контроля [1, 4] определим значение

$$A_{ПСС}=\left(\sum_{i=1}^{n+1} a_i\cdot B_i\right)\bmod M_0=\left(\sum_{i=1}^{5} a_i\cdot B_i\right)\bmod M_0=(a_1\cdot B_1+a_2\cdot B_2+a_3\cdot B_3+$$

$$a_4\cdot B_4+a_5\cdot B_5)\bmod M_0=(0\cdot 1540+0\cdot 3465+0\cdot 3696+0\cdot 2640+5\cdot 2520)\bmod 4620=$$

$$=(5\cdot 2520)\bmod 4620=12600(\bmod 4620)=3360>420\cdot$$

Таким образом, в процессе контроля определено, что $A_{KB}=3360>M=420$. В этом случае, при возможности возникновений только однократных ошибок, делается вывод о том, что рассматриваемое число $\tilde{A}_{3360}=(0\|\|0\|\|0\|\|5)$ неправильное $(3360>M=420)$.

Для исправления числа $\tilde{A}_{3360}=(0\|\|0\|\|0\|\|5)$ вначале необходимо провести диагностику данных, т.е. определить искажённый $\tilde{a}_i$ остаток. После чего необходимо определить истинное значение a_i остатка по модулю m_i и после чего провести исправление искажённого $\tilde{a}_i$ остатка.

II. Диагностика данных $\tilde{A}_{3360}=(0\|\|0\|\|0\|\|5)$. В соответствии с методом проекций [4], составим возможные проекции $\tilde{A}_j$ числа $\tilde{A}_{3360}=(0\|\|0\|\|0\|\|5)$: $\tilde{A}_1=(0\|\|0\|\|0\|\|5)$, $\tilde{A}_2=(0\|\|0\|\|0\|\|5)$, $\tilde{A}_3=(0\|\|0\|\|0\|\|5)$, $\tilde{A}_4=(0\|\|0\|\|0\|\|5)$ и $\tilde{A}_5=(0\|\|0\|\|0\|\|0)$.

Формула для вычисления значений $\tilde{A}_{j\,ПСС}$ проекций числа в ПСС имеет вид [1]

$$\tilde{A}_{j\,ПСС}=\left(\sum_{\substack{i=1;\\ j=\overline{1,n+1.}}}^{n} a_i\cdot B_{ij}\right)\bmod M_j=(a_1\cdot B_{1j}+a_2\cdot B_{2j}+\ldots+a_n\cdot B_{nj})\bmod M_j\ . \qquad (6)$$

В соответствии с формулой (6) вычислим все значения $\tilde{A}_{j\,ПСС}$. Далее проводим $(n+1)$ сравнение: чисел $\tilde{A}_{j\,ПСС}$ с числом $M=M_0/m_{n+1}$. Если среди проекций $\tilde{A}_i$ есть числа не находящиеся внутри информационного $[0,M)$ числового интервала (т.е. $\tilde{A}_k\geq M$), содержащего k правильных чисел, то делается вывод о том, что эти k остатков числа A не искажены. Ошибочными могут быть только остатки, находящиеся среди остальных $[(n+1)-k]$ остатков числа $\tilde{A}_{KB}$.

Набор частных рабочих оснований для заданного КВ и совокупность частных B_{ij} ортогональных базисов представлены соответственно в табл. 2 и табл. 3.

Таблица 2 – Набор частных рабочих оснований КВ ($l=1$)

i / j	m_1	m_2	m_3	m_4	M_j
1	4	5	7	11	1540
2	3	5	7	11	1155
3	3	4	7	11	924
4	3	4	5	11	660
5	3	4	5	7	420

Таблица 3 – Совокупность частных ортогональных базисов B_{ij} КВ ($l=1$)

B_{ij} i / j	1	2	3	4
1	385	616	1100	980
2	385	231	330	210
3	616	693	792	672
4	220	165	396	540
5	280	105	336	120

Итак, имеем, что (см. табл. 2)

$$\tilde{A}_{1ПСС}=\left(\sum_{i=1}^{4}a_i\cdot B_{i1}\right)\bmod M_1=(a_1\cdot B_{11}+a_2\cdot B_{21}+a_3\cdot B_{31}+a_4\cdot B_{41})\bmod M_1=$$
$$=(0\cdot 385+0\cdot 616+0\cdot 1100+5\cdot 980)\bmod 1540=280<420.$$

Делаем вывод, что остаток a_1 числа $\tilde{A}_1$ – возможно $\bar{a}_1$ искажённый остаток;

$$\tilde{A}_{2ПСС}=\left(\sum_{i=1}^{4}a_i\cdot B_{i2}\right)\bmod M_2=(a_1\cdot B_{12}+a_2\cdot B_{22}+a_3\cdot B_{32}+a_4\cdot B_{42})\bmod M_2=$$
$$=(0\cdot 385+0\cdot 231+0\cdot 330+5\cdot 210)\bmod 1155=1050>420.$$

Таким образом, получим, что a_2 достоверно не искажённый остаток;

$$\tilde{A}_{3ПСС}=\left(\sum_{i=1}^{4}a_i\cdot B_{i3}\right)\bmod M_3=(a_1\cdot B_{13}+a_2\cdot B_{23}+a_3\cdot B_{33}+a_4\cdot B_{43})\bmod M_3=$$
$$=(0\cdot 616+0\cdot 693+0\cdot 792+5\cdot 672)\bmod 924=588>420.$$

Получим, что a_3 достоверно не искажённый остаток;

$$\tilde{A}_{4ПСС}=\left(\sum_{i=1}^{4}a_i\cdot B_{i4}\right)\bmod M_4=(a_1\cdot B_{14}+a_2\cdot B_{24}+a_3\cdot B_{34}+a_4\cdot B_{44})\bmod M_4=$$
$$(0\cdot 220+0\cdot 165+0\cdot 369+5\cdot 540)\bmod 660=60<420.$$

Вывод: остаток a_4 по модулю m_4 числа $\tilde{A}_4$ – возможно $\bar{a}_4$ искажённый остаток;

$\tilde{A}_{5ПСС}=\left(\sum_{i=1}^{4}a_i\cdot B_{i5}\right)\bmod M_5$. Так как $M_5=M=420$, то остаток $\bar{a}_5$ по контрольному модулю $m_k=m_5$ всегда будет в совокупности возможных $\bar{a}_i$ искажённых остатков числа в КВ.

Общий вывод. В процессе диагностики данных, представленных НКС $\tilde{A}=(0\|0\|0\|0\|5)$, определились точно не искажённые остатки: $a_2=0$ и $a_3=0$. Ошибочными могут быть остатки по основаниям m_1, m_4 и m_5, т.е. остатки $\bar{a}_1=0$, $\bar{a}_4=0$ и $\bar{a}_5=5$. В этом случае необходимо провести исправление остатков $\bar{a}_1$, $\bar{a}_4$ и $\bar{a}_5$.

III. Исправление ошибок данных $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$. Используя известную [1] формулу

$$a_i = \left(\overline{a}_i + \left[\frac{m_i \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \overline{m}_i} - \frac{\tilde{A}}{B_i} \right] \right) \bmod m_i, \qquad (7)$$

проведём исправление $\overline{a}_1$, $\overline{a}_4$ и $\overline{a}_5$ искажённых остатков a_1, a_4 и a_5, где $r = 1, 2, 3, \ldots$.

В результате получаем, что:

$$a_1 = \left(\overline{a}_1 + \left[\frac{m_1 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \overline{m}_1} - \frac{\tilde{A}}{B_1} \right] \right) \bmod m_1 = \left(0 + \left[\frac{3 \cdot (1 + r \cdot 11)}{11 \cdot 1} - \frac{3360}{1540} \right] \right) \bmod 3 = (0 +$$

$$+ [3,27 - 2,18]) \bmod 3 = (0 + [1,09]) \bmod 3 = (0 + 1) \bmod 3 = 1;$$

$$a_4 = \left(\overline{a}_4 + \left[\frac{m_4 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \overline{m}_4} - \frac{\tilde{A}}{B_4} \right] \right) \bmod m_4 = \left(0 + \left[\frac{7 \cdot 12}{11 \cdot 4} - \frac{3360}{2640} \right] \right) \bmod 7 = (0 + [1,9 -$$

$$- 1,27]) \bmod 7 = (0 + [0,63]) \bmod 7 = (0 + 0) \bmod 7 = 0;$$

$$a_5 = \left(\overline{a}_5 + \left[\frac{m_{n+1} \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \overline{m}_{n+1}} - \frac{\tilde{A}}{B_5} \right] \right) \bmod m_{n+1} = \left(5 + \left[\frac{11 \cdot (1 + 11)}{11 \cdot 6} - \frac{3360}{2520} \right] \right) \bmod 11 =$$

$$= (5 + [2 - 1,3]) \bmod 11 = (5 + [0,7]) \bmod 11 = (5 + 0) \bmod 5 = 0.$$

По полученным остаткам $a_1 = 1$, $a_4 = 0$ и $a_5 = 0$ восстанавливаем (исправляем) искажённое число $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$, т.е. правильное число, будет иметь следующий вид: $\tilde{A}_{исп.} = (1 \| 0 \| 0 \| 0 \| 5)$.

Для проверки правильности исправления данных, по известной формуле, определим значения числа $\tilde{A}_{исп.} = (1 \| 0 \| 0 \| 0 \| 5)$ следующим образом:

$$\tilde{A}_{исп.ПСС} = \left(\sum_{i=1}^{5} a_i \cdot B_i \right) \bmod M_0 = (a_1 \cdot B_1 + a_2 \cdot B_2 + a_3 \cdot B_3 + a_4 \cdot B_4 + a_5 \cdot B_5) \bmod M_0 =$$

$$= (1 \cdot 1540 + 0 \cdot 3465 + 0 \cdot 3696 + 0 \cdot 2640 + 5 \cdot 2520) \bmod 4620 = 14140 (\bmod 4620) = 280.$$

Так как $280 < M = 420$, то число $\tilde{A}_{280} = (1 \| 0 \| 0 \| 0 \| 5)$ является правильным.

С целью уточнения верности процедуры коррекции числа $\tilde{A}_{3360}$, проведём расчёт и сравнение значений и правильных остатков $a_2 = 0$ и $a_3 = 0$. В этом случае:

$$a_2 = \left(0 + \left[\frac{4 \cdot (1 + 11)}{11 \cdot 3} - \frac{3360}{3465} \right] \right) \bmod 4 = 0 \text{ и } a_3 = \left(0 + \left[\frac{5 \cdot (1 + 11)}{11 \cdot 4} - \frac{3360}{3696} \right] \right) \bmod 5 = 0.$$

Полученные результаты $a_2 = 0$ и $a_3 = 0$ расчётов остатков по модулям m_2 и m_3 КВ, подтверждают правильность коррекции неправильного числа $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$.

Таким образом, исходное число $\tilde{A}_{KB} = (0 \| 0 \| 0 \| 0 \| 5)$ является неправильным $\tilde{A}_{3360}$, в котором однократная ошибка $\Delta a_1 = 1$ произошла по модулю m_1. Данная ошибка перевела правильное число A_{280} в неправильное $\tilde{A}_{3360}$.

Для того, чтобы выяснить является ли правильное число A_{280} истинным проведём дополнительные исследования процессов искажения и коррекции числа A_{280} по основанию $m_1 = 3$. Количество N_{HC} возможных неправильных (*искажённых*) $\tilde{A}_{KB}$ кодовых слов (*только при однократной ошибке*) для каждого правильного A_{KB} числа равно $N_{HC} = \sum_{i=1}^{n+1} m_i - (n+1)$.

Результаты анализа показали, что искажение остатка a_1 по модулю $m_1 = 3$ правильного числа A_{280} может привести только к двум неправильным числам $\tilde{A}_{3360} = (\tilde{0} \| 0 \| 0 \| 0 \| 5)$ и $\tilde{A}_{1820} = (\tilde{2} \| 0 \| 0 \| 0 \| 5)$.

Этот факт говорит о том, что исправленное число $A_{исп.} = A_{280} = (1 \| 0 \| 0 \| 0 \| 5)$, является не только правильным *(лежащем в интервале [0, 420))*, но и истинным.

Истинность полученного $A_{280} = (\hat{1} \| 0 \| 0 \| 0 \| 5)$ числа подтверждается тем, что только однократная ошибка $\Delta a_1 = 2$ по основанию $m_1 = 3$ переводит это число

$$(\tilde{A} = (A + \Delta A) \bmod M_0 = (1 \| 0 \| 0 \| 0 \| 5) + (2 \| 0 \| 0 \| 0 \| 0) = \left[(1+2) \bmod 3 \| 0 \| 0 \| 0 \| 5\right] =$$
$$= (\tilde{0} \| 0 \| 0 \| 0 \| 5)),$$

в единственно неправильное число $\tilde{A}_{3360} = (\tilde{0} \| 0 \| 0 \| 0 \| 5)$.

Пример 2. Пусть правильное число равно $A_{280} = (1 \| 0 \| 0 \| 0 \| 5)$ и пусть $\Delta a_1 = 1$.

Тогда $\tilde{A} = (A + \Delta A) \bmod M_0 = (1 \| 0 \| 0 \| 0 \| 5) + (1 \| 0 \| 0 \| 0 \| 0) = \left[(1+1) \bmod 3 \| 0 \| 0 \| 0 \| 5\right] =$ $= (\tilde{2} \| 0 \| 0 \| 0 \| 5)$. Данному числу в КВ соответствует число 1820 в ПСС, т.е. число $\tilde{A}_{1820}$ неправильное. Проведём исправление числа $\tilde{A}_{1820}$.

Перед исправлением числа $\tilde{A}_{1820}$ проведём диагностику данных. Для этого предварительно составим проекции A_j $(j = \overline{1, 5})$ числа $\tilde{A}_{1820} = (2 \| 0 \| 0 \| 0 \| 5)$. Это будут следующие кодовые структуры в КВ: $\tilde{A}_1 = (0 \| 0 \| 0 \| 5)$, $\tilde{A}_2 = (2 \| 0 \| 0 \| 5)$, $\tilde{A}_3 = (2 \| 0 \| 0 \| 5)$, $\tilde{A}_4 = (2 \| 0 \| 0 \| 5)$ и $\tilde{A}_5 = (2 \| 0 \| 0 \| 0)$.

Далее определим все значения проекций $\tilde{A}_{j\,ПСС}$:

$$\tilde{A}_{1ПСС} = (5 \cdot 980) \bmod 1540 = 280 < 420 = M;$$
$$\tilde{A}_{2ПСС} = (2 \cdot 385 + 5 \cdot 231) \bmod 1155 = 1925 \left(\bmod 1155\right) = 770 > 420 = M;$$
$$\tilde{A}_{3ПСС} = (2 \cdot 616 + 5 \cdot 672) \bmod 924 = 4592 \left(\bmod 924\right) = 896 > 420 = M;$$
$$\tilde{A}_{4ПСС} = (2 \cdot 220 + 5 \cdot 540) \bmod 660 = 3140 \left(\bmod 660\right) = 500 > 420 = M;$$
$$\tilde{A}_{5ПСС} = 2 \cdot 280 (\bmod 420) = 560 \left(\bmod 420\right) = 140 < 420 = M.$$

Так как $\tilde{A}_{2ПСС}$, $\tilde{A}_{3ПСС}$ и $\tilde{A}_{4ПСС} > 420$, тогда делается вывод о том, что остатки $a_2 = 0$, $a_3 = 0$ и $a_4 = 0$ числа $\tilde{A}_5 = (2 \| 0 \| 0 \| 0 \| 5)$ не искажены. Искаженными $\bar{a}_1 = 2$ и $\bar{a}_5 = 5$ могут быть только остатки a_1 и a_5. Вначале проведём исправление остатка $\bar{a}_1 = 2$.

Получаем, что

$$a_1 = \left(\bar{a}_1 + \left[\frac{m_1 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \bar{m}_1} - \frac{\tilde{A}}{B_1} \right] \right) \bmod m_1 = \left(2 + \left[\frac{3 \cdot (1+11)}{11 \cdot 1} - \frac{1820}{1540} \right] \right) \bmod 3 =$$
$$= \left(2 + [3,27 - 1,18]\right) \bmod 3 = \left(2 + [2,09]\right) \bmod 3 = (2+2) \bmod 3 = 4 (\bmod 3) = 1.$$

Таким образом, исправленный остаток по модулю m_1 равен $a_1 = 1$.

Аналогичным путём получим значение $a_5 = 5$. По полученным остаткам a_1, a_5 исправляем неправильное число $\tilde{A}_{1820} = (\tilde{2} \| 0 \| 0 \| 0 \| 5)$. В конечном итоге в процессе коррекции получим правильное $A_{280} = (1 \| 0 \| 0 \| 0 \| 5)$ число.

Пример 3. Осуществить контроль числа $A_{KB}=(0\|0\|0\|2\|1)$. В случае его искажения, провести диагностику и коррекцию данных.

I. Контроль данных $A_{KB}=(0\|0\|0\|2\|1)$. В соответствии с известной процедурой контроля определим $A_{ПСС}$ по формуле

$$A_{ПСС}=\left(\sum_{i=1}^{n+1}a_i\cdot B_i\right)\bmod M_0=(0\cdot 1540+0\cdot 3465+0\cdot 3696+2\cdot 2640+$$

$+1\cdot 2520)\bmod 4620=7800(\bmod 4620)=3180>420$. Данное число неправильное $\tilde{A}_{3180}$.

II. Диагностика данных $\tilde{A}_{3180}=(0\|0\|0\|2\|1)$. Составим все возможные проекции $\tilde{A}_j$ числа $\tilde{A}_{3180}$: $\tilde{A}_1=(0\|0\|2\|1)$, $\tilde{A}_2=(0\|0\|2\|1)$, $\tilde{A}_3=(0\|0\|2\|1)$, $\tilde{A}_4=(0\|0\|0\|1)$ и $\tilde{A}_5=(0\|0\|0\|2)$.

Определим значения величин всех пяти проекций $\tilde{A}_j$ в ПСС:

$$\tilde{A}_{1KB}=(0\|0\|2\|1)=\tilde{A}_{1ПСС}=(a_1\cdot B_{11}+a_2\cdot B_{21}+a_3\cdot B_{31}+a_4\cdot B_{41})\bmod M_1=$$
$$=(0\cdot 385+0\cdot 616+2\cdot 1100+1\cdot 980)\bmod 1540=100<M=420;$$
$$\tilde{A}_{2KB}=(0\|0\|2\|1)=\tilde{A}_{2ПСС}=(a_1\cdot B_{12}+a_2\cdot B_{22}+a_3\cdot B_{32}+a_4\cdot B_{42})\bmod M_2=$$
$$=(0\cdot 385+0\cdot 231+2\cdot 330+1\cdot 210)\bmod 1155=870>M=420;$$
$$\tilde{A}_{3KB}=(0\|0\|2\|1)=\tilde{A}_{3ПСС}=(a_1\cdot B_{13}+a_2\cdot B_{23}+a_3\cdot B_{33}+a_4\cdot B_{43})\bmod M_3=$$
$$=(0\cdot 616+0\cdot 693+2\cdot 792+1\cdot 672)\bmod 924=418<M=420;$$
$$\tilde{A}_{4KB}=(0\|0\|0\|1)=\tilde{A}_{4ПСС}=(a_1\cdot B_{14}+a_2\cdot B_{24}+a_3\cdot B_{34}+a_4\cdot B_{44})\bmod M_4=$$
$$=(0\cdot 220+0\cdot 165+2\cdot 396+1\cdot 540)\bmod 660=540>M=420;$$
$$\tilde{A}_{5KB}=(0\|0\|0\|2)=\tilde{A}_{5ПСС}=(a_1\cdot B_{15}+a_2\cdot B_{25}+a_3\cdot B_{35}+a_4\cdot B_{45})\bmod M_5=$$
$$=(0\cdot 280+0\cdot 105+2\cdot 336+1\cdot 120)\bmod 420=240<M=420.$$

В результате расчётов значений $\tilde{A}_{j\,ПСС}$ и сравнения их с величиной $M=420$ длины интервала $[0,420)$ обработки правильных чисел A_{KB} в КВ получим следующее. Совокупность остатков $a_2=0$, $a_4=0$ является правильной (*остатки не искажены*), а остатки $\bar{a}_1=0$, $\bar{a}_3=0$ и $\bar{a}_5=1$ неправильного числа $\tilde{A}_{3180}=(0\|0\|0\|2\|1)$ могут быть искажены (*могут быть неправильными*).

III. Исправление возможно искажённых $\bar{a}_1$, $\bar{a}_3$ и $\bar{a}_5$ остатков числа $\tilde{A}_{3180}$.

Необходимо исправить, возможно, искажённые остатки $\bar{a}_1=0$, $\bar{a}_3=0$ и $\bar{a}_5=1$ по формуле $a_i=\left(\tilde{a}_i+\left[\frac{m_i\cdot(1+r\cdot m_{n+1})}{m_{n+1}\cdot\bar{m}_i}-\frac{\tilde{A}}{B_i}\right]\right)\bmod m_i$. Тогда имеем, следующее:

$$a_1=\left(\bar{a}_1+\left[\frac{m_1\cdot(1+r\cdot m_{n+1})}{m_{n+1}\cdot\bar{m}_1}-\frac{\tilde{A}}{B_1}\right]\right)\bmod m_1=\left(0+\left[\frac{3\cdot(1+r\cdot 11)}{11\cdot 1}-\frac{3180}{1540}\right]\right)\bmod 3=$$
$$=\left(0+[3,27-2,06]\right)\bmod 3=\left(0+[1,21]\right)\bmod 3=(0+1)\bmod 3=1.$$

Таким образом $a_1=1$. Соответственно, для значения $\bar{a}_3$ имеем

$$a_3=\left(\tilde{a}_3+\left[\frac{m_3\cdot(1+r\cdot m_{n+1})}{m_{n+1}\cdot\bar{m}_3}-\frac{\tilde{A}}{B_3}\right]\right)\bmod m_3=\left(0+\left[\frac{5\cdot(1+r\cdot 11)}{11\cdot 4}-\frac{3180}{3696}\right]\right)\bmod 5=$$

$$= \left(0 + \left[1,36 - 0,86\right]\right) \bmod 5 = \left(0 + \left[0,5\right]\right) \bmod 5 = \left(0 + 0\right) \bmod 5 = 0.$$

В этом случае $a_3 = 0$.

Для значения остатка $\bar{a}_5$ получим

$$a_5 = \left(\tilde{a}_5 + \left[\frac{m_5 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot \bar{m}_5} - \frac{\tilde{A}}{B_5} \right] \right) \bmod m_5 = \left(1 + \left[\frac{11 \cdot (1 + r \cdot 11)}{11 \cdot 6} - \frac{3180}{2520} \right] \right) \bmod 11 =$$

$$= \left(1 + \left[2 - 1,26\right]\right) \bmod 11 = \left(1 + \left[0,74\right]\right) \bmod 11 = \left(1 + 0\right) \bmod 11 = 1.$$

В итоге имеем, что $a_5 = 1$.

Таким образом, по полученным значениям $a_1 = 1$, $a_3 = 0$ и $a_5 = 1$ восстановленных остатков исправляем искажённое число $\tilde{A}_{KB} = (0 \| 0 \| 0 \| 2 \| 1)$ на правильное $A_{KB} = (1 \| 0 \| 0 \| 2 \| 1)$ число. Проверка: - $100 < 420$.

3 Выводы

В представленной статье рассмотрены примеры коррекции ошибок данных в классе вычетов (КВ). Показано, что в некоторых случаях, непозиционное кодирования в КВ может обеспечить возможность исправления однократных ошибок при МКР равном $d_{\min}^{(KB)} = 2$.

Отмечено, что для исправления однократных ошибок требуется проведение дополнительных процедур обработки данных, то есть, применение, дополнительно к информационному резервированию, еще и временного резервирования.

Приведенные примеры конкретной реализации процедур коррекции однократных ошибок, показывают практическую реализуемость метода коррекции ошибок данных, представленных в классе вычетов.

Ссылки

[1] F. MacWilliams, N. Sloane, *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[2] J. Proakis, *Digital communications*, McGraw Hill, 2001.

[3] B. Sklar, *Digital Communications: Fundamentals and Applications. Prentice Hall Communications Engineering and Emerging Techno*, Pearson Education, 2016.

[4] F. Barsi and P. Maestrini, "Error Correcting Properties of Redundant Residue Number Systems," in *IEEE Transactions on Computers*, vol. C-22, no.3, pp. 307-315, March 1973.

[5] V.Krasnobayev, A.Kuznetsov, S.Koshman, S. Moroz, "Improved Method of Determining the Alternative Set of Numbers in Residue Number System", in *Recent Developments in Data Science and Intelligent Analysis of Information. ICDSIAI 2018. Advances in Intelligent Systems and Computing*, vol. 836. Springer, Cham, pp. 319-328, 05 August 2018. doi: 10.1007/978-3-319-97885-7_31.

[6] Y. N. Kocherov, D. V. Samoylenko and A. I. Koldaev, "Development of an Antinoise Method of Data Sharing Based on the Application of a Two-Step-Up System of Residual Classes", *2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, Vladivostok, pp. 1-5, 2018.

[7] V. Krasnobayev, A. Kuznetsov, S. Koshman, S. Moroz, "Improved Method of Determining the Alternative Set of Numbers in Residue Number System", in *Recent Developments in Data Science and Intelligent Analysis of Information. ICDSIAI 2018. Advances in Intelligent Systems and Computing*, vol. 836. Springer, Cham, pp. 319-328, 05 August 2018. doi: 10.1007/978-3-319-97885-7_31.

[8] M. Kasianchuk, I. Yakymenko, I. Pazdriy, A. Melnyk and S. Ivasiev, "Rabin's modified method of encryption using various forms of system of residual classes", *14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, Lviv, 2017, pp. 222-224. doi: 10.1109/CADSM.2017.7916120.

[9] V. Krasnobayev, A. Kuznetsov, A. Kononchenko, T. Kuznetsova, "Method of data control in the residue classes", in *Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, Ukraine, April 15-19, 2019, pp. 241-252.

[10] M. Karpinski, S. Ivasiev, I. Yakymenko, M. Kasianchuk and T. Gancarczyk, "Advanced method of factorization of multi-bit numbers based on Fermat's theorem in the system of residual classes", *16th International Conference on Control, Automation and Systems (ICCAS)*, Gyeongju, 2016, pp. 1484-1486.

[11] V. Krasnobayev, A. Kuznetsov, I. Lokotkova and A. Dyachenko, "The Method of Single Errors Correction in the Residue Class", *3rd International Conference on Advanced Information and Communications Technologies (AICT)*, Lviv, Ukraine, 2019, pp. 125-128. doi: 10.1109/AIACT.2019.8847845.

[12] V. Krasnobayev, A. Kuznetsov, M. Zub, K. Kuznetsova, "Methods for comparing numbers in non-positional notation of residual classes", in *Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, Ukraine, April 15-19, 2019., pp. 581-595.

[13] V. Krasnobaev, A. Kuznetsov, V. Babenko, M. Denysenko, M. Zub and V. Hryhorenko, "The Method of Raising Numbers, Represented in the System of Residual Classes to an Arbitrary Power of a Natural Number", *IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Lviv, Ukraine, 2019, pp. 1133-1138. doi: 10.1109/UKRCON.2019.8879793

[14] K. Tao, L. Peng, K. Liang and B. Zhuo, "Irregular repeat accumulate low-density parity-check codes based on residue class pair", *IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, Guangzhou, 2017, pp. 127-131. doi: 10.1109/ICCSN.2017.8230092.

[15] G. Harman and I. E. Shparlinski, "Products of Small Integers in Residue Classes and Additive Properties of Fermat Quotients", in *International Mathematics Research Notices*, vol. 2016, no. 5, pp. 1424-1446, Jan. 2016. doi: 10.1093/imrn/rnv182.

[16] C. Fan and G. Ge, "A Unified Approach to Whiteman's and Ding-Helleseth's Generalized Cyclotomy Over Residue Class Rings", in *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1326-1336, Feb. 2014. doi: 10.1109/TIT.2013.2290694.

[17] A.Wang, A. Zhang and Sh. Chen, "Study On simulation and design of data links human computer interface," *International Conference on Computer Science and Service System (CSSS)*, Nanjing, 2011, pp. 4066-4069. doi: 10.1109/CSSS.2011.5974893.

[18] S. Irfan and S. Ghosh, "Optimization of information retrieval using evolutionary computation: A survey", *International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, 2017, pp. 328-333. doi: 10.1109/CCAA.2017.8229837.

[19] S. Shu, Y. Wang and Y. Wang, "A research of architecture-based reliability with fault propagation for software-intensive systems", *Annual Reliability and Maintainability Symposium (RAMS)*, Tucson, AZ, 2016, pp. 1-6.

[20] S. S. Gokhale, M. R. Lyu and K. S. Trivedi, "Reliability simulation of component-based software systems", *Proceedings Ninth International Symposium on Software Reliability Engineering (Cat. No.98TB100257)*, Paderborn, Germany, 1998, pp. 192-201.

[21] Tiwari, K. Tomko, "Enhanced Reliability of Finite State Machines in FPGA Through Efficient Fault Detection and Correction", *IEEE Transaction on Reliability*, vol. 54, no. 3, pp. 459-467.

[22] Singh and A. Sprintson, "Reliability assurance of cyber-physical power systems", *IEEE PES General Meeting*, Providence, RI, 2010, pp. 1-6.

[23] M. Reddy and N. Nalini, "FT2R2Cloud: Fault tolerance using time-out and retransmission of requests for cloud applications", *International Conference on Advances in Electronics Computers and Communications*, Bangalore, 2014, pp. 1-4.

[24] Braun and H. Wunderlich, "Algorithm-based fault tolerance for many-core architectures", *15th IEEE European Test Symposium*, Praha, 2010, pp. 253-253.

[25] M. Radu, "Reliability and fault tolerance analysis of FPGA platforms", *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*, Farmingdale, NY, 2014, pp. 1-4.

Reviewer: Alexey Stakhov, Doctor of Sciences (Engineering), Full Prof., Academicians of the Academy of Engineering Sciences of Ukraine, International Club of the Golden Section, 6 McCreary Trail, Bolton, Ont., L7E 2C8, Canada.
E-mail: goldenmuseum@rogers.com

Received on November 2019.

Authors:
Serghii Koshman, Doctor of Sciences (Eng.), V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: s.koshman@karazin.ua

Victoria Popenko, Ocmputer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: vita.popenko@gmail.com

Anna Kononchenko, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: akononpro@gmail.com

Examples of usage of method of data errors correction which are presented by the residual classes.

Abstract. The article discusses a method for correcting single errors in the residue class. The results of the analysis of the corrective capabilities of the arithmetic code showed the high efficiency of using non-position code structures in the class of residues. To correct single errors, additional data processing procedures are required, i.e. the use, in addition to the information reservation, also temporary reservation. Examples of correction of one-time data errors represented by code in the residue class are given. The considered examples confirm the practical feasibility of this error correction method. Using a system of residual classes can be useful for implementing fast computer calculations with the possibility of parallelizing some processes. In addition, it can be used to create reliable and fault-tolerant components of computer systems.

Keywords: Residual class system; Data errors correction; Non-positional code structure; Computational process.

Рецензент: Алексей Стахов, д.т.н., проф., академик Академии инженерных наук Украины, Международный Клуб Золотого Сечения, Онтарио, Канада. E-mail: goldenmuseum@rogers.com

Поступила: Ноябрь 2019.

Автори:
Сергій Кошман, д.т.н., Харківський національний університет імені В. Н. Каразіна, Харків, Україна.
E-mail: s.koshman@karazin.ua

Вікторія Попенко, студентка факультету комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, Харків, Україна.
E-mail: vita.popenko@gmail.com

Анна Кононченко, студентка факультету комп’ютерних наук, Харківський національний університет імені В. Н. Каразіна, Харків, Україна.
E-mail: akononpro@gmail.com

Приклади використання методу корекції помилок даних, що представлені в класі лишків.

Анотація. У статті розглянуто метод виправлення однократних помилок у класі лишків (КЛ). Результати аналізу коректуючих можливостей арифметичного коду показали високу ефективність використання непозиційних кодових структур у КЛ. Для виправлення однократних помилок потрібно проводити додаткові процедури обробки даних, тобто застосування, додатково до інформаційного резервування, ще й часового резервування. Наведені приклади виправлення однократних помилок даних, що представлені кодом КЛ. Розглянути приклади підтверджують практичну реалізованість даного методу корекції помилок. Використання системи залишкових класів може бути корисно для реалізації швидких комп'ютерних обчислень з можливістю розпаралелювання деяких процесів. Крім того, її можна використовувати для створення надійних і стійких до відмов компонентів комп'ютерних систем.

Ключові слова: непозиційних система числення в залишковому класі; виправлення одноразових помилок даних; арифметичне непозиційне кодування інформації.

УДК 004.056.5: 004.738 DOI: 10.26565/2519-2310-2019-3-05

КЛАСИФІКАЦІЯ АТАК ПОДВІЙНИХ ВИТРАТ В БЛОКЧЕЙН СИСТЕМАХ

Євгеній Деменко, Олександр Онікійчук, Микита Гончаров,
Сергій Даценко, Микола Полуяненко

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
demenjay@gmail.com, onik4524a@gmail.com, wdpgames@yandex.ru, sergdacenko@gmail.com
nlfsr01@gmail.com

Рецензент: В'ячеслав Калашников, д.ф.-м.н., проф., Технологічний університет Монтеррея, Монтеррей, 64849, Мексика.
kalash@itesm.mx

Надійшло: Листопад 2019.

***Анотація**: У статті наведено короткий огляд та проведено систематизацію інформації за проблематикою подвійних витрат в блокчейн системах з ймовірнісними методами консенсусу і можливими шляхами її вирішення. Описано процедури, за допомогою яких реалізуються атаки подвійних витрат. Розкрито сутність маніпуляцій за допомогою яких зловмисник може намагатися провести подвійні витрати у децентралізованих платіжних системах. Наведено детальних опис дій атакуючого та шляхи запобігання атаки. Розгляд починається від простих атак заснованих на створені дублюючих транзакцій та закінчується більш складними атаками, такими як: атака-гонка; атака Фіннеєм; атака Vector76; атака «51%». Ці атаки вимагають від атакуючого значних ресурсів та можливості розгалуження блокчейн реєстру. Остання група атак проаналізована більш детальніше, з наведенням варіантів її застосування. В якості найбільш небезпечної виділено атаку «51 %», що, на думку авторів, є найбільшою загрозою для безпеки блокчейн систем з ймовірнісними алгоритмами консенсусу.*

***Ключові слова:** комп'ютерні мережі; децентралізовані системи; блокчейн; атака на блокчейн мережі; подвійні витрати.*

1 Вступ

Як правило, всі «класичні» платіжні системи є централізованими, що мають адміністративну ланку, яка забезпечує контроль легітимності будь-якої операції [1]. При цьому, підстава для прийняття рішень про легітимність платежу є інформація, яка надається адміністратором, а не інформація, яка представлена платником. Тому платник в змозі лише сформувати заявку на повторну витрату одних і тих же засобів, а адміністративна ланка підтвердить тільки першу заявку і відкине всі інші, що блокує можливість подвійної витрати одних і тих же цінностей.

У блокчейн системах передбачається відсутність адміністративного ресурсу, і отже, можливість проведення подвійний витрати одних і тих же цінностей стає можливим. Для захисту від атаки подвійної витрати продавці можуть приймати різні заходи захисту, найбільш ефективним з них, є очікування включення транзакції з оплатою в один з блоків блокчейн реєстру. При цьому вузол який формує блок не допустить включення в блок транзакції, які намагаються повторно витратити раніш витрачених коштів. І, якщо навіть такий блок буде сформовано вузлом зловмисника, його відкинуть вузли чесної мережі і блок не буде додано до блокчейн реєстр чесних користувачів.

Процес включення транзакції до складу нового блоку називається підтвердженням транзакції. Включення в один блок відповідає одному підтвердженню. Формування і додавання до реєстру блокчейн ланцюжка ще з *(N-1)* блоків, які посилаються на блок з транзакцією, відповідає N підтвердженням. Однак, якщо зловмисник має досить великі ресурси (володіє високопродуктивним обладнанням, здатним забезпечити високий гешрейт (*англ. – hashrate*) зловмисника) у нього все ще залишається досить висока ймовірність успішно провести подвійну витрату шляхом формування альтернативного ланцюжка блокчейн реєстру.

Успіх атаки подвійної витрати безпосередньо залежить від ресурсів (гешрейта) атакуючого і кількості підтверджень. Ймовірність формування альтернативного ланцюжка експоненціальна зменшується зі зростанням кількості підтверджень і зменшенням гешрейта атакую-

чого. Чим більше підтверджень має транзакція, тим менш імовірне скасування транзакції через заміну діючого ланцюжка альтернативним, що сформовано зловмисником. Однак, з іншого боку, чим більше продавець чекає підтверджень, тим довше затримується проведення самої угоди, що внаслідок призводить до значних затримок, дискомфорту використання системи та збиткам взаємодіючих сторін.

Тому угоди, з нульовим підтвердженням, потенційно мають великий ризик стати жертвою атаки подвійної витрати, а угоди, які очікують велику кількість підтверджень – зазнати збитків через затримки в їх укладанні. Тому, питання знаходження оптимальної кількості підтверджень, при яких ризик атаки подвійний витрати буде нижче деякого прийнятного рівня, а час очікування буде мінімально необхідним, є актуальним завданням. Наприклад, існує думка ([2-5] та ін.), якщо використовується механізм консенсусу на основі Доказу виконаної роботи (*англ. PoW – Proof-of-work*) на основі геш-функції і у атакуючого знаходиться 10 % обчислювальної потужності (гешрейт) від загальної мережі, і очікується 6 підтверджень – ймовірність успіху такої атаки, приблизно, складе 0,1 %. Наведена оцінка ґрунтується на моделі «розорення гравця» [6].

2 Опис та типи атак подвійної витрати

Подвійні витрати (*англ. Double-spending*) – ситуація в децентралізованих платіжних системах (*криптовалютах*), коли користувач пробує повторно використовувати раніше передане [15]. Зазвичай мережа не прийме таку транзакцію як дійсну. Але в паралельних розгалуженнях ланцюга блоків можуть перебувати транзакції, які по-різному розпоряджаються одним і тим же.

Коли здійснюється угода за криптовалюту, то передбачається, що після перерахування монет відправник отримує у відповідь продукт або послугу, яку він оплатив. Враховуючи ці процедурні особливості, сутність атаки подвійної витрати полягає в тому, що спочатку зловмисник переконує продавця в тому, що транзакція на оплату вже була проведена, після чого продавець передає свій товар, а покупець (зловмисник) отримує його. Після отримання товару зловмисник робить все можливе щоб блокчейн мережа прийняла та зберегла іншу транзакцію. Таким чином, в разі успіху зловмисника у продавця не залишається а ні товару, а ні плати за нього.

Атака подвійної витрати існує в багатьох формах [11]. Кожен з можливих методів, що реалізує ту чи іншу форму, повинен перевірятися і оброблятися відповідним програмним забезпеченням (ПЗ) повного вузла. Наведемо методи, що можуть бути застосовані для проведення повторної витрати одних і тих же коштів:

- одна транзакція в мемпулі (*англ. Bitcoin Mempool* [7]), що витрачає одні й ті ж вхідні значення (*UTXO – Unspent Transaction (TX) Output*) кілька разів;
- кілька транзакцій в мемпулі (*англ. Bitcoin Mempool*), які витрачають кошти, посилаючись на одні й ті ж вхідні значення (*UTXO*);
- транзакція в одному блоці, яка проводить одні й ті ж вхідні значення (*UTXO*) кілька разів;
- кілька транзакцій в різних блоках витрачають одні й ті ж вхідні значення (*UTXO*);
- проведення атаки за допомогою вдалого розгалуження блокчейн реєстру, де кожна з гілок містить різні транзакції, що змінюють діючий стан блокчейн системи.

Якщо вразливостям, що засновані на перших чотирьох методах, можна запобігти за допомогою відповідної реалізації ПЗ, то останній метод, який засновано на самому принципі реалізації консенсусу (*використанні ймовірнісних механізмів консенсусу*), не виключає вдалої реалізації відповідної загрози.

Опис виявлених у Bitcoin Core вразливостей, що засновані на перших 4-х методах, а також детальний аналіз причини їх появи можна знайти у роботах [8-9]. Щоб максимально унеможливити маніпулювання блокчейн системою на користь тільки однієї особи, процес майнин-

га був розроблений, як дуже ресурсномістка операція. Так, для формування нового блоку з транзакціями в блокчейн системі, майнери повинні надати дійсні докази виконаної роботи. Але, не зважаючи на це, у зловмисника, який намагається використати п'ятий метод, так само є кілька варіантів його реалізації [4].

2.1 Атака-гонка

Атака-гонка [10] відноситься до випадку, коли торговець приймає непідтверджену транзакцію (*транзакція знаходиться в пулі транзакцій і очікує додавання в блок блокчейн реєстру*) і відразу ж надає платнику продукт/послугу, перш ніж ця транзакція буде підтверджена. Зловмисник з наміром ввести в оману продавця створює дві транзакції: - (I) транзакцію, яка платить продавцю необхідну суму в обмін на продукт/послугу; - (II) шахрайську транзакцію, яка платить ту ж суму на гаманець зловмисника. Обидві транзакції використовують одні і ті ж вхідні дані та намагаються витратити одну й ту ж криптовалюту. Зловмисник одночасно випускає обидві транзакції в блокчейн мережу. Майнер додає їх до мемпулу (*англ. Bitcoin Mempool*) та вважає обидві транзакції дійсними до тих пір, поки одна з них не буде додана до блокчейн реєстру. Транзакція, яка зберігається в блокчейн реєстрі, називається підтвердженою транзакцією. У цей момент входи збереженої транзакції не можуть бути використані в якості вхідних даних для інших транзакцій. Отже, шахрайська транзакція може бути перевірена першою і додана в ланцюжок блоків, що робить платіжну транзакцію недійсною. Неприпустима транзакція відхиляється системою та видаляється з mempool-ів транзакцій майнером.

Щоб уникнути атаки-гонки, торговці повинні дочекатися завершення процесу майнингу і появи транзакції в блокчейн реєстрі, перш ніж надавати платнику продукт або послугу.

2.2 Атака Фіннеєм

Атака Фіннеєм була вперше запропонована на форумі присвяченому біткойну [11]. Як і у випадку з атакою-перегонів, атакуючий, що виконує цю атаку, доб'ється успіху тільки в тому випадку, якщо торговець приймає непідтверджену транзакцію. Для цього атакуючий створює дві транзакції, схожі на ті, що беруть участь в гонці, і утримує їх обидві. Потім зловмисник починає формувати блок, що містить шахрайську транзакцію. Якщо зловмисник успішно сформує блок, він використовує іншу транзакцію, щоб негайно сплатити продавцю в обмін на продукт/послугу. Як тільки продавець здійснює операцію, зловмисник публікує видобутий блок в блокчейн мережу, який містить шахрайську транзакцію. Зважаючи на те, що блок вже сформовано, він буде негайно доданий до блокчейну. В результаті платіжна транзакція стане недійсною. На додаток до цього, зловмисник отримує винагороду за видобутий блок, який несе шахрайську транзакцію. Однак здатність самостійно добувати блок малоймовірна, враховуючи ресурси, які необхідні для виконання завдання.

2.4 Атака Vector76

В порівнянні з атаками-гонки і Фінні, атака Vector76 [12,13] вимагає, щоб продавець чекав створення одного блоку, та додав його в ланцюжок блоків в якості підтвердження. Щоб скасувати транзакцію, зловмисникам необхідно створити розгалуження в блокчейні. Спочатку зловмисник створює платіжну транзакцію продавця, но не передає її в мережу. Потім зловмисник намагається самостійно і таємно сформувати блок з цією транзакцією. У разі успіху атакуючий утримує блок, поки чесні майнери не сформують ще один блок. На наступному кроці, атакуючий публікує блок в мережу одночасно з тим, як чесні майнери публікують свій блок, що призводить до розгалуження блокчейн реєстру.

Перед тим, як розгалуження вирішиться, зловмисник створює шахрайську транзакцію, що повторно витрачає той же вхід транзакції, який використовувався в транзакції, оплачуваної продавцем. Потім зловмисник передає шахрайську транзакцію чесним майнерам, які не мають розгалуження з блоком, що містить транзакцію перераховуючи кошти продавцю. Ці майнери вважають шахрайську транзакцію дійсною і починають формувати з нею новий

блок. В результаті кожна гілка такого блокчейна зберігає одну з транзакцій. Якщо гілка, яка містить шахрайську транзакцію, збільшується в порівнянні з іншого гілкою, то спроба подвійного витрачання буде успішною.

2.5 Атака «51%»

Атака «51%» є найбільшою загрозою для блокчейн систем з консенсусами, які мають ймовірнісний характер завершеності [14]. Ця атака безпосередньо пов'язана з ресурсами, які може використовувати зловмисник [15]. Ресурси вимірюються з точки зору фінансової та обчислювальної потужності. Як правило великі організації мають кошти для контролю більшої частки наявної обчислювальної потужності, та в разі необхідності, можуть зруйнувати або підштовхнути систему до свого бажаного статусу. Важливо відзначити, що навіть при обчислювальній потужності менш 50%, зловмисник все ще може маніпулювати системою. Ця атака також згадується, як атака більшості, згідно якій зловмисник (*зазвичай група майнерів*) контролює більше половини всієї обчислювальної потужності системи. Таким чином контролюючі велику частину потужностей, зловмисник може втручатися в процес майнінгу блоків і скасовувати будь-який блок транзакцій. Під час атаки на «51%» система втрачає цілісність, оскільки у інших майнерів більше немає стимулу брати участь в процесі майнінгу.

Щоб краще зрозуміти особливості цієї атаки, розглянемо випадок, коли зловмисник генерує платіжну транзакцію та «випускає» її в мережу. Ланцюжок блоків можна уявити у вигляді дерева, що починається з початкового (генезис) блоку і йде послідовно. Гілки цього дерева представляють собою історії транзакцій. Гілка не може містити двох конфліктних транзакцій, однак може бути інша гілка, яка містить конфліктуючу транзакцію. Це відповідає ситуації, коли в один момент часу сформовано два різних блока і частина вузлів мережі додала до ланцюга перший блок, а інша частина - другий. Зазвичай, така розбіжність дозволяється, як тільки знаходиться наступний блок. Вірною гілкою цього дерева вважається та, яка включає в себе більш довгий ланцюг наступних блоків.

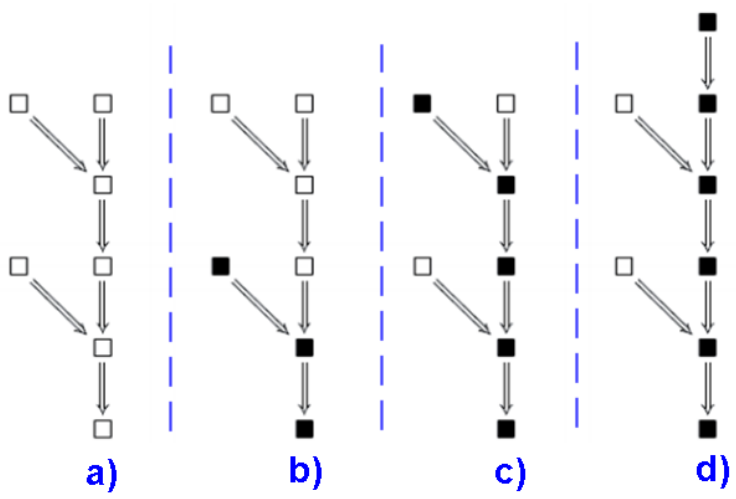

Рис. 1 – Приклади можливих ланцюжків блоків

Для прикладу розглянемо ланцюжок блоків, якій зображено на рис. 1 [16]. Дерево починається знизу, а стрілки вказують з якого блоку на який йде посилання в заголовку блоку. Використані на рис. 1 легенди мають наступне розшифрування: a) можливий варіант побудови дерева; b) позначена гілка недійсна, так як її довжина становить тільки три блоки, в той час як існує довша гілка за неї; c) темна гілка, що має, як і пряма, найбільшу довжину, з-за цього деякими вузлами вважається дійсною; d) випадок, коли знаходиться новий блок, що посилається тільки на один з попередніх блоків, то деяка гілка стає довшою і приймається усіма вузлами, як дійсна.

Узагальнюючи результати аналізу відомих форм реалізації атаки подвійної витрати можливо стверджувати, що для її успішного проведення потрібно виконати наступні кроки:

1. Провести транзакцію, яка атакує першу здійснену оплату.
2. Почати таємно майнити, використовуючи для цього той блок, який включає в себе цю останню транзакцію.
3. Дочекатися, поки транзакція, що відправляє гроші продавцеві, отримає достатню кількість підтверджуючих блоків, а продавець передасть свій товар, будучи впевненим, що гроші дійсно привласнені йому.

4. Продовжувати майнити таємну альтернативну гілку, поки вона не стане більшою, ніж публічна, після чого почати транслювати її в мережу. Оскільки нова гілка довша за всіх інших відомих, то вона буде хибно вважатися дійсною, а переказ одного біткоїна (*англ. BTC*) продавцеві буде замінений відправкою відповідних монет зловмиснику.

На рис. 2 приведено типовий алгоритм атаки подвійної витрати [16], де:

a) стан мережі до початку дій зловмисника;

b) створена гілка (*зліва*), яка включає в себе транзакцію відправки одного біткоїна продавця, що має два підтвердження. В результаті цього продавець передає свій товар. В цей час у зловмисника є згенерований блок, що включає атакуючу транзакцію;

c) якщо атакуючому вдасться створити більш довший ланцюжок, то він публікує його в мережу і біткоїни повертаються йому.

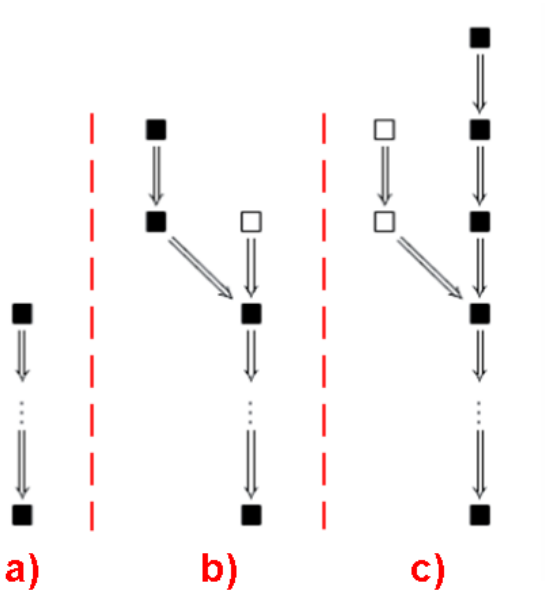

Рис. 2 – Здійснення атаки подвійної витрати

Технічно все це відбувається наступним чином. Зловмисник в блоці з номером, наприклад, 5, виконує деяку транзакцію, переказуючи гроші постачальнику послуг або товарів за покупку. Постачальник отримує ці гроші і, відповідно, поставляє покупцеві відповідну послугу/товар. Після отримання товару/послуги зловмисник швидко починає майнити інший блок з однаковим номером 5, тобто блок, який слідує за блоком номер 4, але в якому, або немає цієї фінансової транзакції, або він переводить ці гроші собі на інший гаманець. При цьому, для того щоб гарантувати прийняття чесними майнерами саме цього альтернативного ланцюжка, він на альтернативний блок з номером 5 намагається додати якомога більше блоків. Якщо у нього вийде зробити альтернативний ланцюжок більш довшим, то саме він, відповідно до протоколу консенсусу, буде вважатися вірним. Очевидно, що чим більше частка зловмисника (*не важливо, чи це обчислювальні потужності у разі Доказу виконаної роботи (англ. PoW – Proof-of-Work) або частка долі у разі підтвердження частки (англ. PoS – Proof-of-Stake*), тим більше у нього шансів успішно виконати цю атаку. Зокрема, якщо частка зловмисника більше половини, то ймовірність успіху цієї атаки прагне до 1.

4 Висновки

Надано огляд відомих схем проведення атак подвійної витрати. Розглянуті основні типи атак подвійної трати таки як: - атака-гонка; - атака Фіннеєм; - атака Vector76; - атака більшості. Визначені можливі методи захисту від наведених атак.

В якості найбільш значущої атаки, визначено атаку «51 %» (*атака більшості*). Підкреслено, що станом на сьогоднішній день, саме ця атака є найбільшою загрозою для безпеки блокчейн систем з ймовірнісними консенсусами. Це обумовлено тим, що вона базується на конструктивної особливості консенсусу, та не має гарантованого захисту від вдалого проведення.

За результатами аналізу доступних джерел за визначеною проблематикою, зроблено висновок, що успіх атаки подвійних витрат безпосередньо залежить від ресурсів (*гешрейта*) атакуючого і кількості підтверджень, тому ймовірність формування альтернативного (фіктивного) ланцюжка експоненціально зменшується зі зростанням кількості підтверджень та зменшенням гешрейту атакуючого. При цьому, зі збільшенням кількості підтверджень транзакції, зменшується ймовірність її скасування через заміну діючого ланцюжка альтернатив-

ним, який був сформований зловмисником. Однак, з іншого боку, чим більше продавець чекає підтверджень, тим більше затримується проведення самої угоди, що, в підсумку, призводить до певного дискомфорту від використання системи та збиткам взаємодіючих сторін.

В цілому, всі розглянуті атаки є дуже небезпечними саме для децентралізованих систем, тому вони потребують належної уваги зі сторони розробників блокчейн систем та фахівців з питань забезпечення інформаційної безпеки.

Незважаючи на те, що існує багато різних способів зменшення вірогідності успішного проведення відомих атак [16], питання щодо можливості їх повного запобігання все досі залишається відкритим, що обумовлює актуальність досліджень у даному напрямку.

Посилання

[1] Centralized, Decentralized, and Distributed Payment Mechanisms. [Online]. Available: https://www.aier.org/article/centralized-decentralized-and-distributed-payment-mechanisms/

[2] M. Rosenfeld, *Analysis of hashrate-based double-spending*, 2014. [Online]. Available: arXiv preprint arXiv:1402.2009

[3] A. Gervais, H. Ritzdorf, G. O. Karame, S. Čapkun, "Tampering with the delivery of blocks and transactions in Bitcoin", in *CCS 2015 - Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, vol. 2015-October, pp. 692-705), Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/2810103.2813655 https://eprint.iacr.org/2015/578.pdf

[4] E. Zaghloul, T. Li, M.W. Mutka, J. Ren, *Bitcoin and Blockchain: Security and Privacy*, 2019. [Online]. Available: ArXiv, abs/1904.11435

[5] BitcoinWiki: Double-spending. [Online]. Available: https://ru.bitcoinwiki.org/wiki/Double-spending

[6] А. Н. Ширяев, *Вероятность*: В 2-х кн. Кн. 1. Москва: МЦНМО, 2007.

[7] *The Bitcoin Mempool – A Beginner's Explanation*. [Online]. Available: https://99bitcoins.com/bitcoin/mempool/

[8] *Hackernoon: Two Ways to Double-Spend*. [Online]. Available: https://medium.com/hackernoon/bitcoin-core-bug-cve-2018-17144-an-analysis-f80d9d373362

[9] *BitcoinCore: CVE-2018-17144 Full Disclosure*. [Online]. Available: https://bitcoincore.org/en/2018/ 09/20/notice/

[10] *Blockchain Attack Vectors: Vulnerabilities of the Most Secure Technology*. [Online]. Available: https://www.apriorit.com/dev-blog/578-blockchain-attack-vectors

[11] H. Finney, *Best practice for fast transaction acceptance - how high is the risk?*. [Online]. Available: https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384, Feb. 2011

[12] *Bitcoin's Security Model Revisited*. [Online]. Available: https://arxiv.org/pdf/1605.09193.pdf

[13] Ch. Everett, *Blockchain Security*. [Online]. Available: https://www.simplexityanalysis.com/blog/2016/9/20/blockchain-security

[14] *The 51% Attack. What is it?* [Online]. Available: https://medium.com/swlh/the-51-attack-what-is-it-d295e70b9ac4

[15] *51% Attack Explained: The Attack on A Blockchain*. [Online]. Available: https://www.fxempire.com/education/article/51-attack-explained-the-attack-on-a-blockchain-513887

[16] П. Колесников, Ю. Бекетнова, Г. Крылов,*Технология Блокчейн. Анализ Атак, стратегии защиты*. [Online]. Available: https://www.mumcfm.ru/repository/7b9dcd8e4e51d467a0f8e1eff82157e504c569331681beb7e80117fd64e05d1a

Reviewer: Vyacheslav Kalashnikov, Dr. of Sciences (Physics and Mathematics), Full Prof., Department of Systems and Industrial Engineering, Campus Monterrey, Monterrey, Mexico.
E-mail: kalash@itesm.mx

Received on November 2019.

Authors:
Eugene Demenko, Computer Science Student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: demenjay@gmail.com

Alexander Onikiychuk, Computer Science Student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail:onik4524a@gmail.com

Nikita Goncharov, Computer Science Student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: wdpgames@yandex.ru

Sergey Datsenko, Computer Science Student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: sergdacenko@gmail.com

Nikolay Poluyanenko, Ph.D., Associate Professor, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: nlfsr01@gmail.com

Classification of double cost attack in blockchain system.

Abstract. The article provides a brief overview and systematization of information on the issue of double costs in blockchain systems with probabilistic consensus methods and possible ways to solve it. The procedures using which double-cost attacks are implemented are described. The essence of the manipulations with which an attacker can try to realize double costs in decentralized payment systems is disclosed. A detailed description of the attacker's actions and ways to prevent the attack is given. The review starts with simple attacks based on creating duplicate transactions and ends with more complex attacks such as: attack-Race; Phinea attack attack; Vector76 attack; «51 %» attack. These attacks require significant resources from the attacker and the possibility of branching

the registry blockchain. The last group of attacks is analyzed in more detail with an indication of their use cases. The most dangerous attack is highlighted. The attack «51 %» is highlighted as the most dangerous, which, according to the authors, poses the greatest threat to the safety of blockchain systems with probabilistic consensus algorithms.

Keywords: Computer Networks; Decentralization; Blockchain; Attack; Double Costs.

Рецензент: Вячеслав Калашников, д.ф.-м.н., проф., Технологический университет Монтеррея, Монтеррей, Мексика.
E-mail: kuznetsov@karazin.ua

Поступила: Ноябрь 2019.

Авторы:
Евгений Деменко, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: demenjay@gmail.com

Никита Гончаров, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: wdpgames@yandex.ru

Александр Оникийчук, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail:onik4524a@gmail.com

Сергей Даценко, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: sergdacenko@gmail.com

Николай Полуяненко, к.т.н., доцент кафедры безопасности информационных систем и технологий, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: nlfsr01@gmail.com

Классификация атак двойных трат в блокчейн системах.

Аннотация. В статье представлен краткий обзор и проведена систематизация информации по проблематике двойных расходов в блокчейн системах с вероятностными методами консенсуса и возможными путями ее решения. Описаны процедуры, с помощью которых реализуются атаки двойных трат. Раскрыта сущность манипуляций, с помощью которых злоумышленник может попытаться провести двойные расходы в децентрализованных платежных системах. Приведено подробные описание действий атакующего и пути предотвращения атаки. Рассмотрение начинается от простых атак, основанных на создании дублирующих транзакций и заканчивается более сложными атаками, такими как: атака-гонка; атака Финеем; атака Vector76; атака «51%». Эти атаки требуют от атакующего значительных ресурсов и возможности разветвления блокчейн реестра. Последняя группа атак проанализирована более детально с указанием вариантов их применения. В качестве наиболее опасной выделена атака «51 %», которая, по мнению авторов, представляет наибольшую угрозу для безопасности блокчейн систем с вероятностными алгоритмами консенсуса.

Ключевые слова: компьютерные сети; децентрализованные системы; блокчейн; атака; двойные траты.

ISSN 2519-2310 DOI: 10.26565/2519-2310-2019-3

KhNU

Наукове видання

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

Випуск 3(15) 2019

Міжнародний електронний науково-теоретичний журнал

Англійською, українською, російською мовами

Комп'ютерне верстання – Федоренко В.В., Єсіна М.В.

61022, Харків, майдан Свободи, 6
Харківський національний університет імені В.Н. Каразіна

V. N. Karazin Kharkiv National University Publishing

2019