ПРАВИЛА ДЕКОДИРОВАНИЯ СОБЫТИЙ И РЕАГИРОВАНИЯ ДЛЯ OSSEC**А.С. Гузарев***ПАО «Информационные телекоммуникационные технологии»,
Санкт-Петербург, Российская Федерация*

Рассматривается создание правил реагирования для многоагентной системы обнаружения вторжений OSSEC, применяемой для защиты автоматизированных комплексов связи от несанкционированного доступа. Целью работы является описание принципа создания правил для систем обнаружения вторжений, обрабатывающих регистрируемые события в соответствии с классом защищенности автоматизированных систем 1Б, обеспечивающего возможность внесения изменений в отдельные группы правил без необходимости изменения остальных. Применение такого принципа позволяет сократить время обработки или необходимые вычислительные ресурсы.

Ключевые слова: система обнаружения вторжений, система защиты, OSSEC, информационная безопасность, несанкционированный доступ.

RULES OF DECODING EVENTS AND RESPONSE TO OSSEC**A.S. Guzarev***PJSC "Information Telecommunication Technologies", Saint-Petersburg, Russian Federation*

Article consider the creation of response rules for multi-agent system for detection of intrusions OSSEC, used to protect automated communication systems from unauthorized access. The purpose of this work is to describe the principle of creating rules for intrusion detection systems that process recorded events in accordance with the security class of automated systems 1B, which provides the ability to make changes to individual groups of rules without the need to change the rest. The application of this principle allows to reduce the processing time or necessary computing resources.

Keywords: intrusion detection system, protective system, OSSEC, information security, unauthorized access.

Введение

Предприятие в процессе разработки автоматизированных комплексов связи (АКС) подводных лодок [1], надводных кораблей и различных проектов [2], наземных и корабельных средств авиационной связи провело разработку подсистемы обеспечения безопасности информации (ОБИ). Подсистема была испытана в 2017–2018 гг. в составе многофункционального интегрированного комплекса корабельной связи (МИКС), которому был присвоен индекс Р-772.

Разработка комплекса Р-772 проводилась в инициативном порядке по ТТЗ, утвержденному заказчиком. Она позволила подвести промежуточные итоги [3] по направлению автоматизированного рабочего места (АРМ) администратора ОБИ. АКС является автоматизированной системой (АС), подлежащий защите от несанкционированного доступа (НСД) к информации по классу защищенности АС 1Б [4]. В перечне регистрируемых действий присутствуют аутентификации пользователей, печати документов, запуска и завершения программных средств, операции доступа к защищаемым файлам и узлам сети, нарушения целостности, изменения полномочий субъектов доступа. Все попытки нарушения защиты должны сигнализироваться на АРМ ОБИ. Реализация данного требования была выполнена путем централизованного сбора и анализа журналов протоколирования (журналов аудита) операционных систем (ОС), служб и специального программного обеспечения (СПО).

Технические решения подобного рода принято называть многоагентами системами обнаружения вторжений (МСОВ) [5]. Была организована трехуровневая архитектура МСОВ, содержащая агенты регистрации, систему сбора и реагирования, интерфейс администратора ОБИ. Проблеме выбора программной реализации МСОВ посвящена работа [6]. По совокупности показателей для решения задачи по защите АКС от НСД был выбран OSSEC, позволяющий анализировать журналы, проверять целостность, обнаруживать и активно реагировать на попытки нарушения НСД. В функциональность заложена возможность анализа журналов как уровня ОС, так и отдельных приложений.

МСОВ OSSEC входит в набор модулей ОС специального назначения Astra Linux® Special Edition, применяемой на серверных устройствах МИКС. Программное средство состоит из трех компонентов, соответствующих трехуровневой архитектуре. МСОВ была возвращена в сети АКС. Поскольку базовые правила OSSEC не удовлетворяли нормативным требованиям класса 1Б, были ориентированы на небольшой набор параметров и ограниченное количество событий, трудно изменяемы, было принято решение о создании собственного набора правил, разрабатываемых по новому принципу. В работе приведено формализованное описание алгоритма работы МСОВ, изложены примененный принцип и логика создания правил реагирования на НСД.

Формализация работы МСОВ

От агентов в систему сбора и реагирования поступают сообщения о произошедших событиях с различными форматами и родом информации. Образующее множество сообщений E_i будем считать бесконечным:

$$E = \{E_1, E_2, \dots, E_i, \dots\}. \quad (1)$$

Информация в сообщениях E_i , как правило, передается в виде единой текстовой строки без дополнительных параметров. Дальнейшая работа с подобными сообщениями требует проведения синтаксического анализа.

В качестве примера сообщения E_i будем использовать сообщение, возникающее в результате удаленного подключения к устройству по протоколу ssh, определенному в RFC 4252.

Jul 4 08:40:57 arm1 sshd[5990]: Accepted password for user19 from 192.168.254.112 port 55525 ssh2

Сообщение автоматически регистрируется средствами операционной системы и имеет формат syslog. В приведенном примере подключение было выполнено к устройству arm1 с устройства, имеющего ip-адрес 192.168.254.112, для авторизации была использована учетная запись user19.

Первичная обработка сообщений строится по следующему принципу: поступающее сообщение декодируется, нормализуется, затем применяются правила реагирования и исключения. Данные правила применяются последовательно, если сообщение не соответствует ни одному правилу, оно отбрасывается.

Процесс декодирования заключается в разделении исходной строки на статические St и переменные Vl части в строгом порядке, соответствующем правилу декодирования Rd :

$$Rd_j = \{St_1, Vl_1, St_2, Vl_2, \dots, St_j, Vl_k, \dots\} \quad (2)$$

Правила декодирования Rd_j применяются по порядку до первого полного совпадения статических частей строки. Если совпадений не найдено, то сообщение будет пропущено. Множеству переменных частей $\{Vl_k\}$ ставится в соответствие множество параметров, такое что $Pr_k \longrightarrow Vl_k$. Образованный кортеж $\langle \{Pr_k\}, \{Vl_k\} \rangle$ является декодированным сообщением. Функцию декодирования F_{decode} сообщения можно записать в следующем формализованном виде:

$$F_{decode}(E_i, \{Rd_j\}) = \langle \{Pr_k\}, \{Vl_k\} \rangle. \quad (3)$$

Результат декодирования F_{decode} представленного ранее сообщения E будет выглядеть следующим образом:

```

**Phase 1: Completed pre-decoding.
  full event: 'Jul 4 08:40:57 arm1 sshd[5990]: Accepted password for user19 from
192.168.254.112 port 55525 ssh2'
  hostname: 'arm1'
  program_name: 'sshd'
  log: 'Accepted password for user19 from 192.168.254.112 port 55525 ssh2'
**Phase 2: Completed decoding.
  decoder: 'D_authentication_success'
  dstuser: 'user19'
  srcip: '192.168.254.112'
  srcport: '55525'
  proto: 'ssh2'

```

Как видно из листинга работы OSSEC, декодирование имеет две фазы: предварительное декодирование (pre-decoding) и декодирование по правилу «D_authentication_success».

Предварительное декодирование позволяет определить пары $\{Pr_k\}, \{Vl_k\}$, общие для формата syslog, включая наименование устройства, зарегистрировавшего событие и программное средство. Правила предварительного декодирования находятся в коде программы и применяются всегда. Правила декодирования разрабатываются в зависимости от требований проекта.

Декодирование по правилу позволяет определить пары $\{Pr_k\}, \{Vl_k\}$ специфичные для программного средства, действия которого были зарегистрированы. Для sshd декодированы следующие параметры:

- используемая учетная запись (user19);
- ip-адрес устройства с которого было выполнено подключение (192.168.254.112);
- порт удаленного устройства, используемый для подключения (55525);
- протокол использованный для удаленного подключения (ssh2).

После успешного декодирования сообщения выполняется поиск правила реагирования. Правила реагирования Rr содержат пары параметр Pr_{Rr} и его значение Vl_{Rr} . Соответствием правилу Rr_i сообщения E_i , имеющего параметры $\{Pr_k\}$ и значения $\{Vl_k\}$, является выполнение следующего условия:

$$\langle \{Pr_{Rr}\}, \{Vl_{Rr}\} \rangle \supseteq \langle \{Pr_k\}, \{Vl_k\} \rangle. \quad (4)$$

Результат работы OSSEC по правилу реагирования Rr_i на сообщение E_i будет выглядеть следующим образом:

```

**Phase 3: Completed filtering (rules).
  Rule id: '5501'
  Level: '2'
  Description: 'Аутентификация пользователя'
**Alert to be generated.

```

Реакция МСОВ на сообщение E_i является генерацией сообщения об успешной аутентификации пользователя с уровнем критичности. Кроме генерации сообщения, правило может содержать автоматические действия по блокировке ip-адресов или портов. Сообщение МСОВ включает в себя:

- сообщение E_i в изначальном виде;
- пары $\{Pr_k\}, \{Vl_k\}$ полученные в результате декодирования;
- номер сработавшего правила Rr_i ;

- уровень критичности (Level);
- текст описания события (Description).

Таким образом, функция MCOB F_{ids} заключается в преобразовании событий E_i в виде строки в набор формализованных параметров $\langle \{Pr_k\}, \{V_k\} \rangle$, который может быть использован для дальнейшей обработки, анализа или визуализации.

$$F_{ids}(E_i) = \langle \{Pr_{ids}\}, \{V_{ids}\} \rangle \quad (5)$$

Сообщения MCOB сохраняются в журнале работы OSSEC, в зависимости от настройки, могут дополнительно отправляться в базу данных или удаленный сервер.

Принцип разработки правил декодирования и реагирования

Работа MCOB OSSEC с базовыми правилами правил представляет собой последовательный перебор правил декодирования (декодеров), потом правил реагирования, что соответствует линейной алгоритмической сложности $O(n)$.

Сокращение времени операций поиска правил декодирования Rd и реагирования Rr предлагает объединение правил в группы, подгруппы и цепочки штатными возможностями OSSEC. При этом декодеры предлагается разделять на родительские и дочерние, а правила реагирования на группы примитивов и группы событий. Соответствующий порядок обработки сообщений приведен на рисунке. Пример следования сообщения по правилам обозначен пунктирными линиями.

Время декодирования напрямую зависит от длины строки, количества статических St и переменных Vl частей и количества правил декодирования Rd_j . Сокращение затрачиваемого времени может быть достигнуто путем перераспределения порядка правил и разделением на родительские и дочерние декодеры.

Порядок включения декодеров должен содержать наиболее часто срабатываемые правила вначале и наименее востребованные – в конце. По возможности декодеры должны быть составлены так, чтобы строго советовать форматам сообщений.

Каждое программное средство или их группа, если они разработаны по одинаковому принципу, при создании сообщений придерживаются общего формата, подставляя контекст в зависимости от ситуации. Родительские декодеры Rd_{parent} должны определять программное средство и основной тип сообщений. Дочерние декодеры Rd_{child} должны непосредственно содержать статические St и переменные Vl части для декодирования. Первым шагом будет найден родительский декодер, вторым шагом – дочерний декодер из ограниченного множества. Функцию декодирования F_{decode} сообщения можно переписать в виде композиции:

$$F_{decode} = Rd_{parent} \circ Rd_{child} \quad (6)$$

Пример родительского и дочерних декодеров представлен ниже:

```
<decoder name="D_authentication_success">
  <program_name>login|sshd</program_name>
  <prematch>^LOGIN ON|^Accepted password for</prematch>
</decoder>
<decoder name="D_authentication_success_ch1">
  <parent>D_authentication_success</parent>
  <program_name>login</program_name>
  <prematch>^LOGIN ON</prematch>
  <regex>LOGIN ON \S+ BY (\S+)</regex>
  <order>dstuser</order>
</decoder>
<decoder name="D_authentication_success_ch3">
  <parent>D_authentication_success</parent>
  <program_name>sshd</program_name>
  <prematch>^Accepted password for</prematch>
  <regex>^ (\S+) from (\S+) port (\d+) (\S+)</regex>
  <order>dstuser,srcip,srcport,proto</order>
</decoder>
```

В приведенном примере родительский декодер «D_authentication_success» будет срабатывать только на успешные операции удаленной и локальной аутентификации пользователя. При этом, если программное средство не соответствует «login» или «sshd», алгоритм сразу перейдет к следующему родительскому декодеру без проверки соответствия регулярному выражению из тега «prematch». Дочерние декодеры «D_authentication_success_ch1» и «D_authentication_success_ch3» определяются по тегу «parent». Поиск дочернего декодера выполняется по названию программного средства и по регулярному выражению сообщения. После того как найден первый подходящий дочерний декодер, выполняется операция декодирования F_{decode} .

Такой принцип составления правил позволяет избежать ситуации полного перебора и сократить количество регулярных выражений, по которым пройдет алгоритм.

OSSEC обеспечивает возможность определения порядка поиска правил реагирования за счет групп и порядковых номеров правил. Предлагаемый принцип строится на идее формирования групп правил на основе декодеров и дальнейшей работе с правилами в пределах заданной группы в порядке следования.

Поскольку целью сбора сообщений является выявление несанкционированных действий, то группирование правил можно по типу регистрируемых событий. В таком случае первая группа правил, называемая примитивами Rr_{PMTV} , должна соответствовать одному или нескольким декодерам и определять тип события, например, «успешная авторизация», «неправильный пароль или пользователь» и другие. При этом одни и те же декодеры могут входить в разные примитивы. Вторая группа правил, называемая событиями Rr_{EVNT} , служит для конкретизации зарегистрированных действий и реагирования, например, «неправильно введенный пароль при авторизации через ssh», «вход заблокированным пользователем» и другие. При этом правила группы событий делятся на базовые, которые должны отработать по умолчанию, и уточняющие, обрабатываемые при определенных значениях Vl_r параметров Pr_r .

Пример групп правил примитивов и событий представлен ниже:

```
<group name="PMTV_auth_ok">
  <rule id="5501" level="2">
    <decoded_as>D_authentication_success</decoded_as>
    <description>Аутентификация пользователя</description>
  </rule>
</group>
<group name="EVNT_auth_session">
  <rule id="5511" level="2">
    <if_group>PMTV_auth_session</if_group>
    <program_name>login</program_name>
    <description>Успешный вход в систему в текстовом режиме</description>
  </rule>
  ...
</group>
```

Приведенный выше пример содержит примитив «PMTV_auth_ok», соответствующий одному декодеру «D_authentication_success». Примитив может быть использован для генерации события «по умолчанию», если в группе событий не оказалось подходящего правила. Правила группы событий «EVNT_auth_session» обрабатываются только при срабатывании примитива, указанного в теге «if_group», и выполняют роль классификации события и уточнения уровня критичности. Порядок применения правил определяется идентификатором правила id. Правила могут объединяться в цепочки посредством применения тега «if_sid» и содержать исключаяющие правила. Таким образом, формула (5) может быть переписана в следующем виде:

$$F_{ids} = F_{decode} \circ Rr_{PMTV} \circ Rr_{EVNT}. \quad (7)$$

Такой подход позволяет объединять в примитивы сообщения от разных программных средств и создавать правила событий независимо от типов и форматов. Вместе с этим, обеспечивается классификация событий, позволяющая более точно определить характер действий в системе.

Выводы

1. Обеспечение требований по защите АКС от НСД по классу защищенности АС 1Б может быть выполнено за счет использования МСОВ OSSEC, входящей в состав ОС специального назначения Astra Linux® Special Edition. Недостатком данной системы является трудность расширения базы правил реагирования. Регистрируемые события имеют различные типы и форматы, что может потребовать переработку как правил декодирования, так и правил реагирования.

2. Предложенный новый принцип создания правил решает две проблемы. Во-первых, за счет введения объектно-ориентированного подхода правила разбиваются на уровни, за-

дача по изменению правил сводится к редактированию отдельных сущностей. Во-вторых, разделение правил на группы и подгруппы сокращает множество, в котором ищутся правила, что снижает алгоритмическую сложность с линейной $O(n)$ до логарифмической $O(\log n)$, сокращает время поиска и объемы вычислительных ресурсов.

3. Алгоритм работы OSSEC и принцип создания правил были описаны в формализованном виде и могут быть использованы при работе с другими МСОВ аналогичного типа.

4. Подсистема ОБИ апробирована в рамках работ по созданию МИКС (изделия Р-772), прошедшего государственные испытания на объекте заказчика.

5. Предложенный принцип создания правил обеспечивает работу с различными типами и форматами событий, разработанная подсистема ОБИ применима не только для АКС, но и для других распределенных автоматизированных систем, включая гражданские.

СПИСОК ЛИТЕРАТУРЫ

1. Интегрированный комплекс связи надводного корабля: патент РФ на изобретение / Николашин Ю.Л. [и др.]. № 2548023. Заявл. 18.03.2015.
2. Телекоммуникационный комплекс корабельной связи: патент на полезную модель / Николашин Ю.Л. [и др.]. № 133376. Заявл. 10.10.2013.
3. *Гузарев А.С.* Применение виртуализации для безопасной разработки элементов автоматизированных систем // Труды НИИ ОСИС ВМФ, ВМА: науч.-техн. сб. ст. и докл. «Интеллектуальные разработки в интересах строительства и развития ВМФ». СПб., 2018. С. 447–453.
4. Руководящий документ Гостехкомиссии России «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации». М.: ГТК РФ, 1992. 20 с.
5. *Волошин Б.В., Жуков В.Г.* Применение многоагентных систем в задаче построения системы обнаружения инцидентов информационной безопасности // Актуальные проблемы авиации и космонавтики. 2012. № 8.
6. *Астаулов Р.А., Тимохович А.С.* Повышение защищенности информационной системы путем реализации мер по обнаружению вторжений уровня узла // Наука и образование сегодня. 2017. № 6 (17).