УДК 004.056.53

А.Д. Москвичев, М.В. Долгачев

АЛГОРИТМЫ КОРРЕЛЯЦИИ СОБЫТИЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Москвичев Антон Дмитриевич, аспирант кафедры «Вычислительная техника» Тихоокеанского государственного университета (ТОГУ), окончил ТОГУ по специальности «Компьютерная безопасность». Инженер центра мониторинга и управления информационной безопасностью ООО «МАСКОМ-Техлайн». Имеет статьи, свидетельства о регистрации программных средств. [e-mail: anton.moskvichev.1996@yandex.ru].

Долгачев Михаил Владимирович, кандидат технических наук, доцент кафедры «Промышленное и гражданское строительство» ТОГУ, окончил ТОГУ по специальности «Технология машиностроения». Преподаватель кафедры «Высшая математика» ТОГУ. Начальник отдела экспортного контроля и защиты информации ТОГУ. Имеет статьи, свидетельства о регистрации программных средств. [e-mail: 007428@pnu.edu.ru].

Аннотация

Система корреляции событий – это система, которая получает события от различных систем обнаружения вторжений и уменьшает количество ложных событий, обнаруживает высокоуровневые атаки, повышает значение произошедших инцидентов, прогнозирует будущие атаки и обнаруживает источники атак. Для достижения этих целей в мире было внедрено множество алгоритмов, имеющих свои преимущества и недостатки. В данной статье представлен обзор уже существующих алгоритмов корреляции событий. Материал, представленный в статье, сфокусирован на алгоритмах, используемых в корреляционных механизмах, которые могут работать в корпоративных сетях. Для сравнения эффективности алгоритмов вводятся функции, связанные с точностью, функциональностью и вычислительными возможностями, и все категории алгоритмов оцениваются с помощью этих функций. Результат этого обзора показывает, что каждая категория алгоритмов имеет свои сильные стороны, и идеальные системы корреляции событий должны обладать сильными сторонами каждой из категорий. В заключении авторы статьи делают вывод об эффективности данных алгоритмов и возможности их использования в качестве модуля коррелятора в системах класса SIEM. На основе полученных результатов авторы делают выбор в пользу алгоритмов на основе причинно-следственных связей в силу их высокой точности, что является обязательным условием для применения алгоритма в области информационной безопасности, и низкого потребления ресурсов.

Ключевые слова: информационная безопасность, SIEM, инцидент, база знаний, сценарий атаки, корреляция событий.

doi: 10.35752/1991-2927-2020-3-61-50-59

INFORMATION SECURITY EVENT CORRELATION ALGORITHMS

Anton Dmitrievich Moskvichev, Postgraduate Student of the Department of Computer Engineering at Pacific National University; graduated from PNU with a degree in Computer Security; an engineer of the Information Security Monitoring and Management Center in MASCOM-Techline, LLC; an author of articles; a holder of certificates of software registration. e-mail: anton.moskvichev.1996@yandex.ru.

Mikhail Vladimirovich Dolgachev, Candidate of Sciences in Engineering; Associate Professor of the Department of Industrial and Civil Engineering of Pacific National University; graduated from PNU with a degree in Mechanical Engineering Technology; Lecturer at the Department of Higher Mathematics at PNU; Head of the Department of Export Control and Information Protection of PNU; an author of articles, a holder of certificates of software registration. e-mail: 007428@pnu.edu.ru.

Abstract

An event correlation system is a system that receives events from various intrusion detection systems, reduces the number of false events, detects high-level attacks, raises the value of incidents, predicts future attacks, and detects sources of attacks. Many algorithms have their advantages and disadvantages. This article provides an overview of existing event correlation algorithms. The material presented in the article is focused on the algorithms used in correlation mechanisms.

The authors of the article introduce functions related to accuracy, functionality and computational capabilities, and compare the categories of algorithms using these functions. The result of this review shows that each category of algorithms has its own strengths, and ideal event correlation systems should have the strengths of each of the categories. In conclusion, the authors of the article conclude that these algorithms are effective and can be used as a correlator module in systems of the SIEM class. Based on the results, the authors make a choice in favor of knowledge base algorithms because of their high accuracy, which is a prerequisite for the application of the algorithm in the field of information security, and low resource consumption.

Key words: information security, SIEM, incident, knowledge base, attack scenario, event correlation.

ВВЕДЕНИЕ

Целью данной статьи является обзор существующих алгоритмов корреляции, их сравнение и выявление лучшего для дальнейшей реализации в качестве модуля собственной SIEM-системы. SIEM – это программная или программно-аппаратная платформа, используемая с целью выявления угроз информационной безопасности на основе анализа журналов событий, получаемых от приложений, сетевого оборудования, систем обнаружения и предотвращения вторжений.

Актуальность статьи объясняется тем, что в современных SIEM-системах используются только алгоритмы, основанные на базе знаний. Такой подход обеспечивает высокую точность, но имеет ряд недостатков, таких как невозможность выявления неизвестных ранее инцидентов, формирование базы знаний. В связи с этим, актуальной задачей является внедрение альтернативных решений, не обладающих недостатками алгоритмов, основанных на базе знаний.

Научная новизна заключается в выявлении алгоритмов корреляции для дальнейшей реализации в собственной SIEM-системе, являющихся альтернативными алгоритмам, основанным на базе знаний.

Для достижения заданной цели поставлены следующие задачи:

- 1) поиск существующих алгоритмов корреляции, их классификация;
- 2) выделение характеристик, используемых для сравнения найденных алгоритмов;
- 3) сравнительная оценка алгоритмов по выделенным характеристикам;
- 4) выявление оптимального решения для дальнейшей реализации в собственной SIEM-системе.

Статья содержит перечень классов алгоритмов корреляции, их особенности, преимущества и недостатки. На основании полученной информации производится сравнение алгоритмов и делается вывод об их эффективности.

Практическая полезность заключается в том, что полученные результаты можно применять при проектировании средств, обеспечивающих информационную безопасность.

1 ОБЩИЕ СВЕДЕНИЯ

Журнал событий – стандартный способ для приложений и операционной системы записи и централизованного хранения информации о важных программных

и аппаратных событиях. Анализ журналов событий является необходимым инструментом для выявления инцидентов в сфере информационной безопасности.

Одним из механизмов выявления угроз информационной безопасности является корреляция событий, получаемых из журналов событий. Корреляция событий – процесс выявления угроз информационной безопасности на основе существующих единичных событий. Результатом корреляции является скоррелированное событие.

Алгоритмы корреляции событий можно разделить на три категории на основе их характеристик [1]:

- 1) на основе подобия;
- 2) на основе знаний;
- 3) на основе статистики.

Эта классификация не является полностью точной и некоторые алгоритмы находятся на грани между двумя категориями. Назначение алгоритму категории основывается на наибольшем сходстве.

Все алгоритмы корреляции работают по единому общему принципу:

- 1) на вход алгоритма поступает событие, являющееся записью из журнала операционной системы, средства обнаружения вторжений или иного источника;
- 2) алгоритм агрегирует (собирает) получаемые события и анализирует их;
- 3) в результате анализа формируется новое (скоррелированное) событие, передающееся на выход.

Например, при получении 100 событий о неудачной попытке входа в систему по протоколу SSH за короткий промежуток времени, алгоритм должен выдать сообщение об атаке типа «Подбор пароля».

2 АЛГОРИТМЫ, ОСНОВАННЫЕ НА ПОДОБИИ

Основой для этой категории алгоритмов является сравнение подобия либо двух событий, либо события с шаблоном событий. Особенностью этих алгоритмов является кластеризация подобных событий во времени. Кластеризация – объединение событий в один шаблон. Важнейшим преимуществом этих алгоритмов является отсутствие необходимости точного определения типов атак.

Для этих типов алгоритмов существует три основных подкатегории. Первая подкатегория основана на определении очень простых правил для выражения отношений между событиями. Вторая подкатегория – иерархические правила – ориентирована на выявление

основных недостатков в структуре сети. Третья подкатегория включает в себя алгоритмы, которые создают коэффициенты сравнения с использованием моделей, основанных на машинном обучении [2].

Выделяются следующие подкатегории алгоритмов:

1) *Простые правила*. Работа алгоритмов основана на определении очень простых правил для выражения отношений между событиями. В этой подкатегории алгоритмы пытаются определить простые правила, чтобы вычислить сходство между атрибутами событий и найти связи [3].

Такой подход реализован в программном средстве «Simple Event Correlator» (SEC) – это легкий и независимый от платформы коррелятор событий. Пользователь может запустить его в качестве демона, работать с ним в интерактивном режиме в терминале, одновременно запускать множество процессов SEC для различных задач и использовать его для множества других целей. SEC считывает строки из файлов, именованных каналов или стандартного ввода, сопоставляет строки с шаблонами для распознавания входных событий и коррелирует события в соответствии с правилами в своих файлах конфигурации. SEC может производить вывод, выполняя внешние

программы, записывая в файлы, отправляя данные на серверы на основе TCP и UDP, вызывая предварительно скомпилированные подпрограммы Perl и так далее [4].

На рисунке 1 представлен пример правила корреляции SEC для идентификации инцидента «Перебор пароля» для SSH. В случае получения четырех событий о неудачной аутентификации от одного узла в течение минуты, SEC блокирует доступ по SSH для этого узла на 2 часа.

В данном примере результатом работы алгоритма является не скоррелированное событие, а команда для операционной системы.

2) *Иерархические правила*. Эта подкатегория включает в себя алгоритмы, которые иерархически формируют уровни абстракций. На основе этих уровней абстракций принимаются решения о выявлении скоррелированных событий. Этот набор алгоритмов используют факторы сходства в своей работе [5].

Иерархические правила формируются с помощью контекстно-свободных грамматик (ACF-грамматик) [6]. На рисунке 2 представлен пример сценария для многоступенчатой атаки, прописанный с использованием ACF-грамматик.

```
type=SingleWith2Thresholds
ptype=RegExp
pattern=sshd.*: authentication failure[ ;].* rhost=(\S+)
desc=Multiple failed ssh authentication attempts from $1
action=logonly ; shellcmd ( /usr/local/sbin/iptables -I FORWARD -s $1 -p tcp --dport 22 -j DROP ;
                             /usr/local/sbin/iptables -I INPUT -s $1 -p tcp --dport 22 -j DROP )

window=60
thresh=4
desc2=Pruning iptables firewall rule blocking ssh from $1
action2=logonly ; shellcmd ( /sbin/iptables -D FORWARD -s $1 -p tcp --dport 22 -j DROP ;
                             /sbin/iptables -D INPUT -s $1 -p tcp --dport 22 -j DROP )

window2=7200
thresh2=0
```

Рис. 1. Пример правила корреляции SEC

Productions	Semantic Rules
Scenario → HP SP SC ACR TA AV	Subattack _i .pre = Subattack _{i+1} .con Spoofed(AV.sourceIP) Spoofed(AV.targetIP)
HP → icmp_ping HP ε	HP.pre = icmp_ping.pre HP.con = icmp_ping.con
SP → sadmind_ping SP ε	SP.pre = sadmind_ping.pre SP.con = sadmind_ping.con
SC → sadmind_amslverify_overflow SC ε	SC.pre = sadmind_amslverify_overflow.pre SC.con = sadmind_amslverify_overflow.con
ACR → rsh ACR ε	ACR.pre = rsh.pre ACR.con = rsh.con
TA → mstream_zombie TA ε	TA.pre = mstream_zombie.pre TA.con = mstream_zombie.con
AV → stream_dos	AV.pre = stream_dos.pre AV.con = stream_dos.con

Рис. 2. Сценарий многоступенчатой атаки в ACF-грамматике

Сценарий включает в себя следующие шаги:

- идентификация открытого порта на удаленном узле (HS);
- идентификация сервиса, соответствующего открытому порту (SP);
- поиск уязвимостей, имеющихся у сервиса (SC);
- активное взаимодействие с удаленным узлом через сервис (ACR);
- эксплуатация уязвимостей (TA);
- нарушение доступности (AV).

3) *Машинное обучение*. Здесь коэффициенты сравнения генерируются автоматически. Необходимым условием для алгоритмов является наличие набора кластеризованных событий, на основе которых они могут устанавливать параметры своей модели принятия решений [7].

Одним из примеров является алгоритм машинного обучения, работающий в два этапа. На первом этапе события объединяются в группы в соответствии с динамически генерируемыми шаблонами. Шаблоны строятся методом кластеризации получаемых событий по заранее выделенным полям. На втором этапе полученные шаблоны анализируются на наличие заданных наборов признаков. Каждый набор признаков соответствует классу инцидента (например, идентификация открытых портов, сервисов и другие). На первом шаге используется неуправляемый алгоритм машинного обучения, на втором – EM-алгоритм – алгоритм, используемый в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей, в случае, когда модель зависит от некоторых скрытых переменных [8].

3 АЛГОРИТМЫ, ОСНОВАННЫЕ НА ЗНАНИЯХ

Эта категория основана на базе знаний об определенных атаках. Алгоритмы этой категории делятся на две основные подкатегории [9]:

1) Основанные на причинно-следственных связях.

Алгоритмы в этой подкатегории отслеживают и контролируют поля событий, а затем обнаруживают инциденты безопасности. Кроме того, возможно извлечение и формирование связей между различными этапами атаки с базой знаний, которая описывает все существующие причинно-следственные связи событий, и базы данных, описывающей конфигурации и структуру сети. С точки зрения степени зависимости от знаний о предметной области, эти алгоритмы предъявляют самые высокие требования и, напротив, генерируют выводы без каких-либо смещений и полностью основаны на реальных значениях событий. Учитывая, что эти алгоритмы не используют заранее сформированную информацию в дополнение к знаниям о предметной области, они являются очень гибкими и расширяемыми, а поведение алгоритмов изменяется в режиме реального времени [10].

В качестве примера можно привести алгоритм, использующий в качестве базы знаний граф причинно-следственных связей (Causal Relation Graph – CRG) [11]. Пример CRG-графа представлен на рисунке 3.

CRG включает в себя следующие элементы:

- Вершины, каждая из которых содержит сигнатуру источника событий (AS);
- Вершины состояний (C);
- Ребра, ведущие от AS к C – являются моделью причинной связи;
- Ребра, ведущие от C к AS – являются моделью следственной связи.

Все ребра содержат метки, представляющие собой условные переменные. Они позволяют отделить события от разных источников. Поэтому в качестве меток используются IP-адреса, сетевые порты или другая информация. Необходимо отметить, что меткой ребра является имя поля, а не информация, соответствующая полю внутри события.

Рис. 3. Пример CRG-графа

На рисунке 4 изображен пример CRG-графа, отвечающий за обнаружения инцидента типа «Отказ в обслуживании».

2) *Основанные на сценариях.* Основная цель работы этих алгоритмов – обнаружение многошаговых атак. Они зависят от существующих сценариев атак. Для описания этих сценариев представлены разные языки, но основная идея во всех из них – указать шаги атаки, предпосылки и ее цели. Таким образом, с точки зрения необходимого объема знаний о предметной области, этот набор алгоритмов требует более высокого уровня знаний, но эти знания могут иметь меньший объем. Но из-за очень широкого диапазона возможных случаев кластеризация и распараллеливание будут затруднены. В случае определения языка для выражения сценариев эти алгоритмы являются полностью гибкими и расширяемыми, потому что поведение системы должно изменяться в реальном времени в соответствии с любым изменением или расширением в правилах. Требуемая память для обнаружения сценариев увеличивается

```
use bsm, unix;
scenario ftp_write
{
  int user;
  int pid;
  int inode;

  initial state s0 { }

  transition create_file (s0 -> s1)
    nonconsuming
    {
      [WRITE w] : (w.euid != 0) && (w.owner != w.ruid)
      { inode = w.inode; }
    }

  state s1 { }

  transition login (s1 -> s2)
    nonconsuming
    {
      [EXECUTE e] : match_name(e.objname, "login")
      {
        user = e.ruid;
        pid = e.pid;
      }
    }

  state s2 { }

  transition read_rhosts (s2 -> s3)
    consuming
    {
      [READ r] : (r.pid == pid) && (r.inode == inode)
    }

  state s3
  {
    {
      string username;
      userid2name(user, username);
      log("remote user %s gained local access", username);
    }
  }
}
```

Рис. 5. Сценарий атаки типа «FTP-write» на языке STATL

Рис. 4. Пример CRG-графа, выявляющего инцидент типа «Отказ в обслуживании»

в соответствии с числом определенных сценариев и требуемым временным промежутком [12].

Для реализации алгоритмов корреляции с использованием сценариев был разработан STATL – расширяемый язык описания атак на основе состояний и переходов. Этот язык позволяет описывать проникновение в компьютер как последовательность действий, которые злоумышленник выполняет для взлома компьютерной системы. На языке STATL формируется база знаний, содержащая сценарии атак [13].

На рисунке 5 изображен сценарий атаки типа «FTP-write» на языке STATL. Злоумышленник использует службу FTP для создания фиктивного файла «.rhosts» в доступном для записи домашнем каталоге службы FTP. Используя созданный файл, злоумышленник может открыть удаленный сеанс с помощью службы «rlogin» без необходимости ввода пароля. Сценарий атаки можно обобщить, сказав, что злоумышленник создает фиктивный файл «.rhosts» в домашнем каталоге любого другого

пользователя, а затем использует его для входа в систему без предоставления пароля [14].

4 СТАТИСТИЧЕСКИЕ АЛГОРИТМЫ

Основная идея этих алгоритмов состоит в том, что атаки имеют схожие статистические атрибуты, обнаружив которые можно их идентифицировать. Эти типы алгоритмов сохраняют причинно-следственные связи между различными инцидентами и анализируют их возникшие частоты в период обучения системы с использованием предыдущего статистического анализа данных, а затем генерируют шаги атаки. Изучив эти отношения, полученные знания используются для корреляции различных стадий атаки. Статистические алгоритмы не требуют никаких предварительных знаний о сценариях атак. Но научные результаты указывают на то, что использование этих алгоритмов возможно только в очень специфических областях, в которых предметная область учитывается при проектировании алгоритмов, иначе возникает большое число ошибок.

Эта категория также разделена на две подкатегории. Целью первой подкатегории является оценка причинно-следственных связей между событиями, прогнозирование следующего появления события и обнаружение атак. Целью второй подкатегории является сочетание достоверности с абсолютно аналогичными событиями [15].

Выделяются следующие подкатегории алгоритмов:

1) *Оценка причинно-следственной связи.* Целью этих алгоритмов является создание возможной модели для определения корреляционных связей между событиями. Используя эту модель, события могут быть сопоставлены без знания предметной области, но для получения точной и надежной модели требуется огромное количество данных об атаках.

Алгоритм действует в два этапа: обучение и функционирование. Обучение проводится в автономном режиме и может выполняться на архивных данных, поэтому его относительно большая нагрузка по обработке не создает проблем для практического использования. На этапе обучения, благодаря использованию предыдущих данных, нагрузка на обработку модели не слишком велика, и ее можно значительно снизить с помощью некоторых оптимизаций. Из-за высокой зависимости алгоритма обучения от огромного объема данных адаптировать алгоритм к новым условиям не просто, но с прямым вмешательством пользователя на этапе обучения поведение алгоритма может быть быстро изменено. Расширение алгоритма также возможно с использованием большего количества данных.

Ярким примером алгоритма, использующего оценки причинно-следственных связей, является алгоритм на основе скрытой модели Маркова. Скрытая модель Маркова (СММ) – статистическая модель, имитирующая работу процесса, похожего на марковский процесс с неизвестными параметрами, и задачей ставится разгадывание неизвестных параметров на основе наблюдаемых.

СММ характеризуется конечным набором состояний. Переходы между состояниями регулируются набором вероятностей (вероятностей переходов), связанных с каждым состоянием. В конкретном состоянии результат или наблюдение могут быть сгенерированы в соответствии с отдельным распределением вероятностей, связанным с этим состоянием. Внешнему наблюдателю виден только результат, а не состояние. Состояния не видны снаружи; отсюда и название СММ. Марковская модель, используемая для скрытого слоя, является марковской моделью первого порядка. Это означает, что вероятность нахождения в определенном состоянии зависит только от предыдущего состояния. Говорят, что в определенном состоянии марковская модель «выделяет» наблюдаемое соответствие этому состоянию. Одна из целей использования СММ состоит в том, чтобы вывести из набора выделяемых наблюдений наиболее вероятный путь во множестве состояний.

Система на основе СММ работает следующим образом: один или несколько датчиков отслеживают сетевой трафик и генерируют инциденты при обнаружении потенциальных вторжений. Инциденты от датчиков сохраняются в базе данных для последующей обработки системой. Данные собираются в коллекции для анализа модулем классификации СММ. В случае если СММ неправильно классифицирует инцидент, сетевой аналитик исправляет классификацию и отправляет обратно в базу данных оповещения, чтобы СММ могла обновиться.

Стандартная СММ предназначена для обнаружения одной категории атак, а область обнаружения вторжений включает несколько категорий атак. Поэтому используется множество СММ для каждой категории. Необходимо отметить, что данный подход облегчает распараллеливание работы алгоритма [16].

2) *Степень достоверности комбинаций.* Целью алгоритмов этой подкатегории является выявление инцидентов информационной безопасности из событий, получаемых от различных средств обнаружения вторжений. Основная задача работы алгоритмов – принять правильное решение по объединению событий от нескольких детекторов.

Представленные алгоритмы требуют огромного количества данных для генерации вероятностных моделей. Основная идея может быть упрощена путем удаления всех возможных деталей обработки и принятия только количества повторений событий как фактора, независимого от важности события и истории ресурсов.

Алгоритмы действуют в два этапа: обучение и функционирование. Таким образом, относительно большая величина нагрузки при обработке не оказывает влияния на скорость. Из-за независимости процесса кластеризации событий входные данные могут быть полностью разделены, и благодаря высокой простоте обработки внутри каждого кластера нет необходимости в распараллеливании. Гибкость низкая из-за необходимости обучения с большим количеством данных, но надежность различных сведений о ресурсах может измениться из-за прямого вмешательства системного

администратора и регулирования поведения алгоритма в режиме реального времени. Алгоритм расширяется путем увеличения объема обучающей выборки.

Самым простым примером является проверка событий, получаемых от средств обнаружения вторжений, с помощью теста отношения правдоподобия – статистический тест, используемый для проверки ограничений на параметры статистических моделей, оценённых на основе выборочных данных [17].

5 СРАВНЕНИЕ АЛГОРИТМОВ

Выявление наиболее эффективного алгоритма корреляции событий производится в два этапа. Первый этап – выявление характеристик, присущих общим классам алгоритмов. К ним относятся:

- 1) Объединение событий от различных источников. На практике события поступают от различных источников. Отсутствие данного механизма делает невозможным применение алгоритма;
- 2) Необходимость в базе знаний. Формирование базы знаний – сложный процесс, так как требует ручного сбора информации о событиях и инцидентах. Поэтому отсутствие базы знаний является преимуществом в работе алгоритма;
- 3) Обнаружение ложных срабатываний;
- 4) Обнаружение многоступенчатых атак;
- 5) Выявление новых атак;
- 6) Вероятность ложных срабатываний. Невозможно точно вычислить данную величину, так как она зависит от входных данных. Можно лишь провести оценку относительно других классов алгоритмов.

Результаты сравнения представлены в таблице 1. Очевидно, что статистические алгоритмы неэффективны в силу отсутствия механизма объединения событий, их можно убрать из дальнейшего рассмотрения.

Вторым этапом является сравнение подкатегорий алгоритмов для оставшихся классов по характеристикам, неприменимым к классам. К таким характеристикам относятся:

- 1) Гибкость – насколько эффективно алгоритм подстраивается под информационную систему;

- 2) Расширяемость – возможность алгоритма справляться с увеличением числа типов источников;
- 3) Необходимый объем памяти;
- 4) Необходимые вычислительные мощности;
- 5) Возможность распараллеливания;
- 6) Точность – насколько точно алгоритм классифицирует инцидент;
- 7) Обнаружение многоступенчатых атак.

Оценки подкатегорий алгоритмов по перечисленным характеристикам представлены в таблице 2.

В таблице используются следующие сокращения:

- 1) В – высокий показатель;
- 2) С – средний показатель;
- 3) Н – низкий показатель.

Алгоритмы, основанные на машинном обучении, работают без участия пользователя, в связи с этим имеют гибкость и расширяемость ниже в сравнении с другими алгоритмами.

Как говорилось выше, алгоритмы, основанные на причинно-следственных связях, предъявляют самые высокие требования с точки зрения зависимости от предметной области. Следовательно, для их работы требуется база знаний большего объема, что свидетельствует о большем объеме необходимой для работы памяти в сравнении с другими алгоритмами.

Алгоритмы, основанные на причинно-следственных связях, требуют вычислений небольшого объема, так как в основе их работы лежит взаимодействие с базой знаний.

Работа алгоритмов, использующих базы знаний, основана на построении цепочек событий. Построение цепочек трудно поддается распараллеливанию. Алгоритмы, использующие простые и иерархические правила, легко распараллеливаются, так как каждый процесс может отвечать за отдельные правила.

Нельзя оценить уровень распараллеливания алгоритмов, использующих машинное обучение, так как это зависит от алгоритма обучения.

Самая высокая точность у алгоритмов, основанных на базах знаний в силу использования заранее прописанных правил корреляций. Самая низкая – у алгорит-

Таблица 1

Сравнение классов алгоритмов

Характеристика	На основе подобию	На основе базы знаний	Статистические алгоритмы
Объединение событий от различных источников	Да	Да	Нет
Необходимость в базе знаний	Да	Да	Нет
Обнаружение ложных событий	Да	Да	В зависимости от алгоритма
Обнаружение многоступенчатых атак	В зависимости от алгоритма	Да	В зависимости от алгоритма
Поиск новых атак	Да	Нет	Да
Вероятность ложных срабатываний	Средняя	Низкая	Высокая

мов, использующих машинное обучение, так как идентификация происходит автоматически.

Каждому критерию из таблицы 2 необходимо присвоить числовую оценку по следующему правилу:

1) точность и требуемые вычислительные мощности оцениваются по шкале от 0 до 2, так как эти критерии являются наиболее важными;

2) остальные оцениваются от 0 до 1.

Результаты оценки приведены в таблице 3. В результате, самым эффективным алгоритмом является алгоритм, использующий причинно-следственные связи.

Эмпирически вывод об эффективности алгоритмов на основе причинно-следственных связей обосновывается следующим образом:

1) Алгоритм, основанный на машинном обучении, не является эффективным при выявлении инцидентов информационной безопасности;

2) Алгоритм, использующий сценарии, не является эффективным, так как алгоритм, основанный на причинно-следственных связях, использует меньшие вычислительные мощности при большем объеме памяти, а увеличение объема памяти дешевле с экономической точки зрения, чем увеличение вычислительных мощностей.

3) Алгоритм, использующий простые правила, менее эффективен в сравнении с алгоритмом, использующим иерархические правила, так как не дает возможности выявлять многоступенчатые атаки. Следовательно, он не является эффективным.

4) Алгоритм, использующий причинно-следственные связи, является более эффективным, чем алгоритм, использующий иерархические правила, так как требует вычислительной мощности меньшего объема и более точно идентифицирует инциденты информационной безопасности.

Таблица 2

Сравнение подкатегорий алгоритмов

		Гибкость	Расширяемость	Необходимый объем памяти	Вычислительные мощности	Распараллеливание	Точность	Обнаружение многоступенчатых атак
На основе подобию	Простые правила	В	В	Н	В	В	С	Нет
	Иерархические правила	В	В	Н	В	В	С	Да
	Машинное обучение	Н	Н	Н	В	С	Н	Да
На основе базы знаний	Причинно-следственные связи	В	В	В	Н	Н	В	Да
	Сценарии	В	В	Н	В	Н	В	Да

Таблица 3

Численное сравнение подкатегорий алгоритмов

		Гибкость	Расширяемость	Необходимый объем памяти	Вычислительные мощности	Распараллеливание	Точность	Обнаружение многоступенчатых атак	Сумма
На основе подобию	Простые правила	1	1	1	0	1	1	0	5
	Иерархические правила	1	1	1	0	1	1	1	6
	Машинное обучение	0	0	1	0	0,5	0	1	2,5
На основе базы знаний	Причинно-следственные связи	1	1	0	2	0	2	1	7
	Сценарии	1	1	1	0	0	2	1	6

Таким образом, алгоритм, использующий причинно-следственные связи, является наиболее эффективным для корреляции событий информационной безопасности.

ЗАКЛЮЧЕНИЕ

Анализ алгоритмов корреляции событий, приведенных в данной статье, показывает, что у каждой категории есть свои преимущества и недостатки. Алгоритм корреляции необходимо выбирать с учетом предметной области, уровня абстракции, ресурсов, количества событий в единицу времени, выделяемых для программного средства, реализующего выбранный алгоритм.

Целью данной статьи являлся выбор алгоритма корреляции для SIEM-системы. Исходя из результатов, полученных в ходе исследования, оптимальным будет использование алгоритма на основе причинно-следственных связей.

Цель достигнута в полном объеме за счет решения следующих задач:

- 1) найдены существующие классы алгоритмов корреляции;
- 2) выделены сравнительные характеристики алгоритмов;
- 3) проведена сравнительная оценка алгоритмов;
- 4) выявлено оптимальное решение для дальнейшей реализации в собственной SIEM-системе.

СПИСОК ЛИТЕРАТУРЫ

1. Al-Mamory S.O., Zhang H. A survey on IDS alerts processing techniques // *Proceeding of the 6th WSEAS International Conference on Information Security and Privacy (ISP)*. 2007. pp. 69–78.
2. Valdes A., Skinner K. Probabilistic alert correlation // *Recent Advances in Intrusion Detection (RAID)*. 2001. pp. 54–68.
3. Debar H., Wespi A. Aggregation and correlation of intrusion-detection alerts // *Recent Advances in Intrusion Detection (RAID)*. 2001. pp. 85–103.
4. Rouillard John P. Real-time Log File Analysis Using the Simple Event Correlator (SEC) // *Proceedings of the 18th Conference on Systems Administration (LISA 2004)*. 2004. pp. 133–150.
5. Cuppens F. Managing alerts in a multi-intrusion detection environment // *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*. 2001. pp. 22–31.
6. Al-Mamory S.O., Zhang H. IDS alerts correlation using grammar-based approach // *Journal of Computer Virology*. 2009. Vol. 5, no. 4. pp. 271–282.
7. Comprehensive approach to intrusion detection alert correlation / F. Valeur, G. Vigna, C. Kruegel, R.A. Kemmerer // *IEEE Transactions on Dependable and Secure Computing*. 2004. Vol. 1, no. 3. pp. 146–169.

8. Using Unsupervised Learning for Network Alert Correlation / R. Smith, N. Japkowicz, M. Dondo, P. Mason // *Conference of the Canadian Society for Computational Studies of Intelligence*. 2008. pp. 308–319.

9. Templeton S.J., Levitt K.A. requires/provides model for computer attacks // *Proceedings of the Workshop on New Security Paradigms*. 2001. pp. 31–38.

10. Ning P., Cui Y. An intrusion alert correlator based on pre-requisites of intrusions. 2002.

11. Zali Z., Hashemi M.R., Saidi H. Real-Time Intrusion Detection Alert Correlation and Attack Scenario Extraction Based on the Prerequisite-Consequence Approach // *ISC International Journal of Information Security*. 2012. Vol. 4. pp. 125–136.

12. Al-Mamory S.O., Zhang H. Intrusion detection alarms reduction using root cause Analysis and clustering // *Computer Communications*. 2009. Vol. 32, no. 2. pp. 419–430.

13. A logic-based model to support alert correlation in intrusion detection / B. Morin, L. Me, H. Debar, M. Ducasse // *Information Fusion*. 2009. Vol. 10, no. 4. pp. 285–299.

14. Eckmann S., Vigna G., Kemmerer R. STATL: An Attack Language for State-based Intrusion Detection // *Journal of Computer Security*. 2002. Vol. 10, № 1–2. pp. 71–103

15. Lee W., Qin X. Statistical causality Analysis of INFOSEC alert data // *Managing Cyber Threats*. 2003. pp. 101–127.

16. Applications of hidden markov models to detecting multi-stage network attacks / D. Ourston, S. Matzner, W. Stump, B. Hopkins // *Proceedings of the 36th Annual IEEE Hawaii International Conference on System Sciences*. 2003.

17. Gu G., Cardenas A.A., Lee W. Principled reasoning and practical applications of alert fusion in intrusion detection systems // *Proceedings of ACM Symposium on Information, Computer and Communications Security*. 2008. pp. 136–147.

REFERENCES

1. Al-Mamory S.O., Zhang H. A Survey on IDS Alerts Processing Techniques. *Proc. of the 6th WSEAS International Conference on Information Security and Privacy (ISP)*. 2007, pp. 69–78.
2. Valdes A., Skinner K. Probabilistic Alert Correlation. *Recent Advances in Intrusion Detection (RAID)*. 2001, pp. 54–68.
3. Debar H., Wespi A. Aggregation and Correlation of Intrusion-Detection Alerts. *Recent Advances in Intrusion Detection (RAID)*. 2001, pp. 85–103.
4. Rouillard John P. Real-time Log File Analysis Using the Simple Event Correlator (SEC). *Proceedings of the 18th Conference on Systems Administration (LISA 2004)*. 2004, pp. 133–150.
5. Cuppens F. Managing Alerts in a Multi-Intrusion Detection Environment. *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*. 2001, pp. 22–31.

6. Al-Mamory S.O., Zhang H. IDS Alerts Correlation Using Grammar-Based Approach. *Journal of Computer Virology*, 2009, vol. 5, no. 4, pp. 271–282.
7. Valeur F., Vigna G., Kruegel C., Kemmerer R.A. Comprehensive Approach to Intrusion Detection Alert Correlation. *IEEE Transactions on Dependable and Secure Computing*, 2004, vol. 1, no. 3, pp. 146–169.
8. Smith R., Japkowicz N., Dondo M., Mason P. Using Unsupervised Learning for Network Alert Correlation. *Conference of the Canadian Society for Computational Studies of Intelligence*. 2008, pp. 308–319.
9. Templeton S.J., Levitt K. A Requires/Provides Model for Computer Attacks. *Proceedings of the Workshop on New Security Paradigms*. 2001, pp. 31–38.
10. Ning P., Cui Y. An Intrusion Alert Correlator Based on Pre-requisites of Intrusions. 2002.
11. Zali Z., Hashemi M.R., Saidi H. Real-Time Intrusion Detection Alert Correlation and Attack Scenario Extraction Based on the Prerequisite-Consequence Approach. *ISC International Journal of Information Security*, 2012, vol. 4, pp. 125–136.
12. Al-Mamory S.O., Zhang H. Intrusion Detection Alarms Reduction Using Root Cause Analysis and Clustering. *Computer Communications*, 2009, vol. 32, no. 2, pp. 419–430.
13. Morin B., Me L., Debar H., Ducasse M. A Logic-Based Model to Support Alert Correlation in Intrusion Detection. *Information Fusion*, 2009, vol. 10, no. 4, pp. 285–299.
14. Eckmann S., Vigna G., Kemmerer R. STATL: An Attack Language for State-Based Intrusion Detection. *Journal of Computer Security*, 2002, vol. 10, no. 1–2.
15. Lee W., Qin X. Statistical Causality Analysis of INFOSEC Alert Data. *Managing Cyber Threats*, 2003, pp. 101–127.
16. Ourston D., Matzner S., Stump W., Hopkins B. Applications of Hidden Markov Models to Detecting Multi-Stage Network Attacks. *Proceedings of the 36th Annual IEEE Hawaii International Conference on System Sciences*. 2003.
17. Gu G., Cardenas A.A., Lee W. Principled Reasoning and Practical Applications of Alert Fusion in Intrusion Detection Systems. *Proceedings of ACM Symposium on Information, Computer and Communications Security*. 2008, pp. 136–147.