Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Валейко Михаил Сергеевич

Модуль анализа поведения пользователей в SIEM-системах с целью выявления вредоносного ПО

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Графеева Н. Г.

Рецензент:
ведущий инженер Степанов Д. С.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Chair of Analytical Information Systems

Mikhail Valeyko

Module for analysis of user behavior in SIEM-systems to identify malicious software

Bachelor's Thesis

Scientific supervisor:
associate professor Natalia Grafeeva

Reviewer:
senior developer Dmitriy Stepanov

Saint-Petersburg
2016

Оглавление

Введение

В информационной безопасности анализ поведения пользователей используется для обнаружения внутренних угроз или вредоносных атак, целью которых является причинение вреда или получение несанкционированного доступа к внутренним ресурсам.

Ежедневно создаются десятки тысяч вредоносных программ, а также совершенствуются уже существующие. Атаки становятся не случайными, а целенаправленными, организованными и мотивированными. Обнаружение и дальнейшее предотвращение таких атак является непростой и актуальной на сегодня задачей информационной безопасности.

SIEM-системы обеспечивают анализ в реальном времени событий безопасности, исходящих от сетевых устройств и приложений. Они используются для журналирования данных, их агрегации, а также хранения. Объемы хранимой информации очень велики, но ее большая часть описывает обычное поведение пользователей, которое не является вредоносным. Лишь малый процент этих данных - это информация о разного рода нарушениях безопасности: вторжениях, внедрении вредоносного ПО. Таким образом, проблему обнаружения угроз безопасности можно свести к задаче выявления *аномалий* или отклонений от *нормального* поведения пользователей и системы в целом.

Выявление аномалий в Data Mining - это процесс нахождения элементов, событий или наблюдений, которые не соответствуют ожидаемым паттернам или значительно отличаются от других элементов во множестве данных. Аномалии в разных источниках могут упоминаться как отклонения, выбросы, шум или исключения.

В машинном обучении существуют различные подходы к решению данной проблемы [7]. Выбор конкретного алгоритма зависит от специфичности данных, их размеров и других факторов.

В рамках данной работы было рассмотрено несколько методов машинного обучения без учителя, проведено сравнение этих методов на размеченном наборе данных и выбран лучший метод в качестве алго-

ритма выявления аномалий, используемого в модуле по умолчанию. Также были исследованы различные методы обработки данных и оценки точности алгоритмов, описан подход обогащения данных и реализован модуль анализа поведения пользователей.

1. Постановка задачи

Реальные исходные данные, на основе которых требовалось реализовать модуль, состоят из логов сетевых соединений NGFW Palo-Alto [5]. Логи сгенерированы внутри сети компании "PETER-SERVICE" и обезличены в целях безопасности. Основные атрибуты данных представлены в Таблице 1.

Однако, исходные данные не столь содержательны, и выявление вредоносных соединений на их основе, может оказаться затруднительным. Для этого было принято решение использовать внешнюю информацию из стороннего источника - Threat Intelligence системы VirusTotal [12]. С помощью публичного API, предоставляемого системой VirusTotal, можно получить информацию о связанных с IP, указанным в качестве параметра запроса, адресах и файлах, показатели их вредоносности.

Данные VirusTotal могут улучшить анализ исходных данных, но явных указаний о вредоносности того или иного соединения они не содержат. Иначе говоря, исходные данные неразмечены - неизвестно, какие сетевые соединения являются вредоносными, а какие относятся к нормальному поведению пользователей. В связи с этим отсутствовала возможность для оценки и сравнения результатов работы различных методов. С этой целью был найден dataset схожей структуры, который впоследствии использовался для сравнения алгоритмов.

Набор данных соревнования KDD Cup 1999 [1] - размеченный dataset, содержащий сетевые соединения с большим количеством симулированных внутри сети вторжений. Каждое соединение данного набора помечено либо как нормальное, либо как вредоносное.

Таким образом в работе использовалось два набора данных:

- исходные данные - логи реальных сетевых соединений NGFW Palo-Alto, сгенерированные внутри сети компании "PETER-SERVICE"
- размеченный KDD dataset, необходимый для оценки и сравнения алгоритмов выявления аномалий

Целью данной работы является реализация модуля анализа пове-

Наименование атрибута	**Описание**
Source IP	IP адрес отправителя
Destination IP	IP адрес получателя
Source Port	Порт отправителя, используемый в сессии
Destination Port	Порт получателя, используемый в сессии
Application	Приложение ассоциированное с сессией. Наиболее популярные значения среди исходных данных: • skype • google-base • ftp • vkontakte-base
Flags	32-битное поле, содержащее детали сессии. Некоторые возможные значения: • 0x80000000 - PCAP сессия • 0x02000000 - IPv6 сессия • 0x01000000 - SSL сессия
Category	Категория URL, ассоциированная с сессией
Protocol	IP протокол, ассоциированный с сессией
Bytes	Общее количество байтов (отправленных и полученных) за время сессии
Bytes Sent	Количество отправленных байт
Bytes Received	Количество полученных байт
Packets	Общее количество пакетов (отправленных и полученных) за время сессии
Packets Sent	Количество отправленных пакетов
Packets Received	Количество полученных пакетов
Destination Location	Страна получателя или внутренний регион для приватных адресов

Таблица 1: Основные атрибуты исходных данных

дения пользователей, выявляющего вредоносные соединения, среди реальных исходных данных. На вход модулю подаются логи сетевых соединений, представленные в Таблице 1. На выходе модуль должен сформировать отчет со списком подозрительных соединений для дальнейшего анализа специалистом.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Провести анализ существующих методов машинного обучения без учителя для выявления аномалий сетевых соединений.

2. Сравнить реализованные в популярной Python библиотеке для машинного обучения scikit-learn [13] методы на размеченном наборе данных KDD соревнования.

3. Обогатить реальные исходные данные с помощью внешней системы VirusTotal.

4. Реализовать модуль анализа поведения пользователей, позволяющий выявлять вредоносные соединения на основе реальных исходных данных.

2. Обзор существующих подходов

В данном разделе приводится обзор существующих подходов, которые использовались в реализации модуля: методы выявления аномалий сетевых соединений, метрики для оценки точности работы алгоритмов выявления аномалий, а также подходы для обработки различных типов данных.

2.1. Методы выявления аномалий

Согласно [7], методы машинного обучения для выявления аномалий имеют схожую функциональную архитектуру, представленную на Рис. 1.

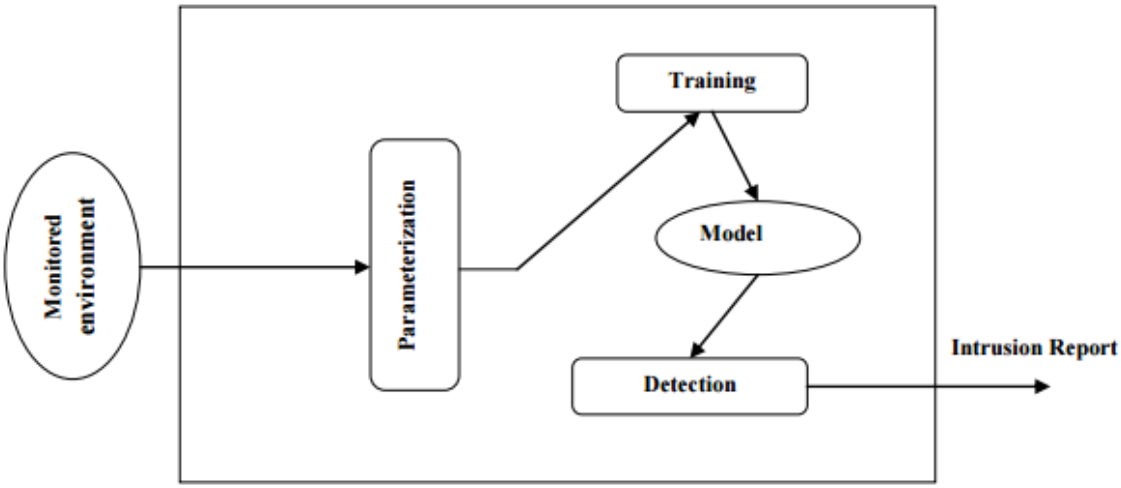

Рис. 1: Архитектура методов выявления аномалий

Можно выделить несколько стадий:

- параметризация
- обучение
- выявление

Параметризация включает в себя сбор и дальнейшую обработку данных из систем-мониторинга. Далее, на стадии обучения, строится модель системы с использованием ручных или автоматических методов.

На заключительном этапе с помощью построенной модели выделяются отклонения.

Алгоритмы машинного обучения строят требуемую модель автоматически, получая на вход тренировочные данные. Методы машинного обучения для выявления аномалий можно разделить на алгоритмы, требующие обучения с учителем и не требующие этого.

Обучение с учителем использует тренировочное множество, которое состоит из примеров входов и выходов алгоритма. Но реальные исходные данные неразмечены (отсутствуют выходы), поэтому обучение с учителем не рассматривалось.

Обучение без учителя не требует размеченных данных. При таком подходе к выявлению аномалий в сетевом трафике предполагается, что большая часть сетевых соединений - это нормальный трафик, и очень небольшой процент трафика - аномальные соединения. В работе [7] выполнен обзор методов машинного обучения для выявления аномалий. Среди методов обучения без учителя, применимых для выявления аномалий, в библиотеке scikit-learn релизованы следующие алгоритмы: метод k-средних, EM-алгоритм и метод опорных векторов с одним классом.

2.1.1. Метод k-средних

Метод k-средних является одним из самых простых алгоритмов в машинном обучении, решающих задачу кластеризации. Он разбивает множество элементов векторного пространства на заранее известное число кластеров k. Основная идея заключается в том, что на каждой итерации перевычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения центра масс кластеров.

В работе [4] используется улучшенная версия алгоритма k-means и демонстрируется его высокая эффективность для выявления сетевых аномалий.

2.1.2. Expectation–maximization алгоритм

EM-алгоритм в математической статистике используется для нахождения оценок максимального правдоподобия параметров вероятностных моделей. Алгоритм чередует два шага: вычисление ожидаемого значения функции правдоподобия (E-шаг), вычисление оценки максимального правдоподобия (M-шаг). Значение, полученное на шаге M, используется для шага E на следующей итерации.

В [8] предложен метод выявления аномалий в сетевом трафике, использующий EM-алгоритм.

2.1.3. Метод опорных векторов с одним классом

Метод опорных векторов с одним классом - специфичный пример метода опорных векторов, ориентированный на обнаружение отклонений. One-Class Support Vector Machines (OCSVM) можно рассматривать как SVM с двумя классами, для которого все точки из тренировочного множества лежат в одном классе, а начало координат - единственный представитель второго класса. Основная идея OCSVM состоит в том, чтобы отобразить входные данные во множество признаков большой размерности с помощью соотсвествующей функции ядра и построить функцию для наилучшего разделения одного класса данных от другого. Построенная функция принимает два значения: 1 (нормальные данные) и -1 (аномальные данные).

В работе [11] начало координат и точки, достаточно близкие к нему, относят ко второму классу и рассматривают как некоторые аномальные данные.

2.2. Оценка точности алгоритмов

Для сравнения методов между собой прежде необходимо получить оценку результатов их работы. Алгоритмы выявления аномалий на выходе могут принимать два значения: является входное соединение аномальным или не является таковым. Для оценки бинарной классифика-

ции используются следующие метрики:

- true positive (tp) - количество элементов, которые классификатор правильно отнес к первому классу; false negative (fn) - количество элементов, которые классификатор неправильно отнес ко второму классу. Аналогично определяются true negative (tn) и false positive (fp)
- $accuracy = \frac{tp+tn}{tp+fp+fn+tn}$
- $precision = \frac{tp}{tp+fp}$
- $recall = \frac{tp}{tp+fn}$
- $F_1\text{-}score = 2 \cdot \frac{precision \times recall}{precision+recall}$

Специфично для оценки алгоритмов выявления аномалий сетевых соединений также рассматривают *false alarms* (ложные срабатывания) - процент нормальных соединений, которые модель посчитала вредоносными, а также *detection rate* (полнота выявления) - процент вредоносных соединений, которые обнаружила модель.

2.3. Обработка данных

И исходные данные, и размеченный dataset имеют разные типы признаков: категориальные и численные. Поэтому для их обработки необходимы различные подходы.

Значение категориальных атрибутов имеет дискретное распределение, отличаясь этим от численных непрерывных данных. Исходные данные в большей части состоят из категориальных атрибутов, таких как протокол, флаги, страна получателя и другие. Однако, многие алгоритмы работают именно с численными данными. В связи с этим возникает необходимость в предварительной обработке категориальных атрибутов.

2.3.1. Обработка категориальных признаков

В работе [3] описано два подхода для обработки атрибутов с ограниченным числом значений: фиктивные переменные и анализ соответствий.

Фиктивные переменные Пусть имеется атрибут, который принимает K различных значений. Тогда для такого атрибута может быть сгенерировано K фиктивных переменных следующим образом:

$$x_1 = \begin{cases} 1, & \text{если значение атрибута категории 1} \\ 0, & \text{иначе} \end{cases}$$

$$x_2 = \begin{cases} 1, & \text{если значение атрибута категории 2} \\ 0, & \text{иначе} \end{cases}$$

$$\vdots$$

$$x_K = \begin{cases} 1, & \text{если значение атрибута категории } K \\ 0, & \text{иначе} \end{cases}$$

$K - 1$ фиктивных переменных однозначно определяют значение категориального атрибута.

Множественный анализ соответствий Множественный анализ соответствий (MAC) позволяет исследовать соотношения между строками и столбцами таблицы сопряженности большой размерности. Метод во многом похож на факторный анализ, который используется для непрерывных данных, но в отличие от него, в анализе соотвествий исследуются таблицы сопряженности. Подробно метод описан в работе [2].

При обработке исходных данных все номинальные признаки, имеющие более двух категорий, сначала преобразовывались MAC, а затем первые две главные оси использовались для представления таких номинальных признаков. Для номинальных признаков, имеющих только

две категории, использовались 0 и 1.

2.3.2. Обработка численных данных

Большинство алгоритмов чувствительно к скалированию данных. Поэтому перед запуском алгоритмов необходимо нормализовать численные атрибуты - преобразовать их так, чтобы каждый признак лежал в промежутке от 0 до 1, либо от -1 до 1.

Для масштабирования значений можно применить формулу:

$$\text{новое значение} = \frac{\text{старое значение}}{\text{норма вектора признака}}$$

В реализации модуля использовались следующие нормы:

- $L^1 : ||v||_1 = \sum_i |v_i|$
- $L^2 : ||v||_2 = \sqrt{\sum_i |v_i|^2}$
- $max : ||v||_\infty = \max_i |v_i|$

3. Экспериментальное сравнение методов

Алгоритмы сравнивались на размеченных данных соревнования KDD Cup 1999 [1]. Указанный dataset содержит набор сетевых соединений, среди которых большой спектр симулированных внутри сети вторжений.

3.1. Описание экспериментов

Эксперименты проводились на тренировочном множестве, в котором каждое соединение помечено либо как нормальное, либо как вредоносное (один из 22 типов атак). Набор данных содержит 494021 элемента, из которых 97278 - это нормальные соединения, остальные соединения - аномальные (вредоносные). Для тестирования использовался набор из всех нормальных соединений и случайно выбранная часть вредоносных соединений, составляющая 10% от числа нормальных. Категориальные атрибуты были преобразованы в фиктивные переменные, численные атрибуты нормировались с помощью нормы L^2.

Исследуемые алгоритмы относятся к числу недетерминированных, поэтому проводилось несколько экспериментов для получения более справедливой оценки. В Таблице 2 представлены средние результаты из 5 независимых экспериментов.

	k-means	EM-algorithm	One-Class SVM
Accuracy	0.80	0.81	**0.86**
F1-score	0.47	0.48	**0.56**
False alarms, %	20.17	19.32	**7.28**

Таблица 2: Сравнение методов на KDD dataset

3.2. Интерпретация результатов

Accuracy (точность) показывает соотношение правильно классифицированных примеров и их общего количества: 0 - ни одного верно классифицированного примера, 1 - все примеры классифицированы верно.

F_1-*score* - это взвешенное среднее *precision* и *recall*. Как и точность, F_1-*score* варьируется от 0 (худший случай) до 1 (лучший). False alarms (процент ложных срабатываний) указывает на процент примеров, которые по ошибке были классифицированы как вредоносные. Меньший процент ошибок соответствует лучшему результату.

Из Таблицы 2 видно, что лучшие результаты показал *метод опорных векторов с одним классом*. OCSVM превзошел остальные методы по всем описанным метрикам оценки, поэтому в модуле использовался как алгоритм выявления аномалий по умолчанию.

4. Обогащение исходных данных

Реальные исходные данные не столь содержательны, как размеченный KDD dataset, используемый для сравнения методов. Другими словами, выявление аномалий на их основе может оказаться затруднительным.

Однако, среди атрибутов логов NGFW Palo-Alto содержится IP адрес получателя, используя который, можно провести процесс обогащения данных - получить дополнительную информацию из Threat Intelligence систем. Такие системы хранят информацию об угрозах и нарушителях и непрерывно поддерживают ее актуальность.

В данной работе в качестве Threat Intelligence системы была выбрана платформа VirusTotal [12]. VirusTotal имеет богатую базу знаний, удобный интерфейс, хоть и существенно ограниченное, но бесплатное API. С помощью публичного API системы можно получить информацию о подозрительности IP адреса получателя, указав его в качестве параметра запроса.

В силу ограниченности публичного API VirusTotal, при реализации модуля было принято решение хранить результаты запросов в базе данных. Таким образом, информация, полученная один раз с помощью HTTP запроса, сохранялась в локальной базе данных. При следующих запросах данной информации результат извлекался уже из базы данных. Так как многие IP адреса среди внутрисетевого трафика неоднократно повторяются (приблизительно 1 уникальный IP адрес на 20 сетевых соединений), такой подход значительно уменьшил время получения данных VirusTotal, а также снизил расход Интернет трафика.

4.1. Работа с API VirusTotal

Для получения отчета по IP адресу необходимо выполнить HTTP GET запрос с параметрами:

- валидный IP адрес
- API ключ

В ответ на такой запрос VirusTotal выдает отчет в формате JSON:

Листинг 1: Пример отчета по IP адресу

```
{'response_code': 1,
 'verbose_msg': 'IP address found in dataset',
 'resolutions': [
    {'last_resolved': '2013-04-08 00:00:00', 'hostname
       ': '027.ru'},
    {'last_resolved': '2013-04-08 00:00:00', 'hostname
       ': 'auto.rema-tiptop.ru'},
    {'last_resolved': '2013-04-08 00:00:00', 'hostname
       ': 'catalog24de.ru'},
    {'last_resolved': '2013-04-08 00:00:00', 'hostname
       ': 'club.velhod.ru'},
    {'last_resolved': '2013-04-08 00:00:00', 'hostname
       ': 'danilova.pro'},
    ... continues ...
 ],
 'detected_urls': [
    {"url": "http://027.ru/", "positives": 2, "total":
        37, "scan_date": "2013-04-07 07:18:09"},
    ... continues ...
]}
```

4.2. Выделение атрибутов для анализа

Помимо *'detected urls'*, отчет также может содержать *'undetected urls'*, *'detected downloaded samples'*, *'undetected downloaded samples'* , *'detected referrer samples'*, *'undetected referrer samples'*, *'detected communicating samples'*, *'undetected communicating samples'* при наличии тех или иных связей с данным IP адресом.

Перечисленные элементы содержат описание URL адресов или файлов, связанных с запрашиваемым IP адресом. В описании: *'total'* - это

количество антивирусов, которыми был просканирован источник; *'positives'* - количество антивирусов, определивших источник вредоносным.

Эта информация была использована следующим образом: для каждого элемента вычислялась сумма $\frac{positives}{total}$ по каждому источнику. В результате для входных параметров алгоритмов использовался вектор из 8 значений.

5. Описание модуля

На основе проведенных исследований и экспериментов был реализован конфигурируемый модуль анализа.

5.1. Общее описание

Модуль полностью написан на языке Python с использованием следующих библиотек:

- sklearn - библиотека для машинного обучения [13]
- pandas - библиотека для анализа данных [9]
- numpy - библиотека для научных вычислений [6]
- requests - библиотека для работы с HTTP [10]

В общем работа модуля состоит из следующих шагов:

1. Парсинг логов сетевых соединений (*infile*).
2. Преобразование категориальных признаков (*categorical*).
3. Нормализация численных признаков (*norm*).
4. Получение информации о подозрительности IP адресов получателя из VirusTotal (*virsutotal*).
5. Построение модели (*model*).
6. Вывод подозрительных соединений в файл в формате HTML (*outfile*).

Параметры для отдельных шагов задаются в файле конфигурации перед запуском Python скрипта.

- *infile* - путь до входного файла с логами сетевых соединений NGFW Palo-Alto

- *categorical* - метод обработки категориальных атрибутов. Может принимать одно из двух значений: *dummy* - фиктивные переменные, *mca* - множественный анализ соответствий

- *norm* - норма для масштабирования численных признаков. Возможные значения: *l1, l2, max*

- *virustotal* - использование режима VirusTotal. Возможные значения: *True, False*

- *model* - модель, используемая для выявления аномалий. Возможные значения: *ocsvm, kmeans, ema*

- *outfile* - путь до выходного файла с отчетом о вредоносных соединениях

Обязательным, среди перечисленных параметров, является только *infile*. Остальные параметры могут быть не указаны - в таком случае используются значения по умолчанию.

5.2. Параметры по умолчанию

На основе проведенных исследований и экспериментов были выбраны параметры, которые используются в модуле по умолчанию.

Так для преобразования категориальных атрибутов по умолчанию используется значение *dummy*, для нормализации численных признаков - *l2*. Режим VirusTotal по умолчанию не используется, а для его использования необходимо задать дополнительные параметры, описанные далее.

Как упоминалось ранее, наилучшие результаты в выявлении аномалий показал метод опорных векторов с одним классом, поэтому параметр *model* по умолчанию принимает значение *ocsvm*. Если не задан путь до выходного файла, то используется директория входного файла, а имя принимает значение *anomaly_report_%datetime%.html*, где *%datetime%* - строковое представление даты и времени формирования отчета.

5.3. Параметры моделей

Отдельно в файле конфигурации можно задать параметры моделей. Описание всех параметров доступных моделей можно найти в официальной документации к sklearn [13]. По умолчанию установлены следующие параметры:

- **ocsvm**: *kernel=rbf, degree=3, gamma=auto, coef0=0.0, tol=0.001, nu=0.05, shrinking=True, cache_size=500, verbose=False, max_iter=-1, random_state=None*
- **kmeans**: *n_clusters=2, init=k-means++, n_init=10, max_iter=300, tol=0.0001, precompute_distances=auto, verbose=0, random_state=None, copy_x=True, n_jobs=1*
- **ema**: *n_components=2, covariance_type=diag, random_state=None, thresh=None, tol=0.001, min_covar=0.001, n_iter=200, n_init=2, params=wmc, init_params=wmc, verbose=0*

5.4. Режим VirusTotal

Для хранения информации из VirusTotal использовался MySQL Server. Поэтому для работы модуля в режиме VirsuTotal необходимо дополнительно указать параметры подключения к базе данных. Кроме того, для использования публичного API VirusTotal необходим API ключ, который указывается в отдельном параметре.

Таким образом, при установленном значении *True* для параметра *virustotal*, необходимо также задать:

- параметры подключения к базе данных: *user, password, host, database*
- API ключ: *apikey*

В случае использования режима VirusTotal все перечисленные параметры являются обязательными.

Заключение

В данной работе был реализован модуль анализа поведения пользователей, выявляющий вредоносные соединения. Для достижения поставленной цели были выполнены следующие задачи:

1. Проведен анализ методов машинного обучения без учителя для выявления аномалий сетевых соединений.

2. Проведено сравнение методов на размеченном наборе данных KDD соревнования.

3. Реализовано обогащение реальных исходных данных с помощью внешней системы VirusTotal.

4. На основе проведенных исследований и экспериментов реализован конфигурируемый модуль анализа поведения пользователей, позволяющий выявлять вредоносные соединения на основе логов сетевых соединений NGFW Palo-Alto.

В дальнейших планах рассматривается внедрение реализованного модуля в промышленную эксплуатацию.

Список литературы

[1] 99 KDD Cup. — URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[2] Abdi Hervé, Valentin Dominique. Multiple Correspondence Analysis. — URL: https://www.utdallas.edu/~herve/Abdi-MCA2007-pretty.pdf.

[3] Handling Nominal Features in Anomaly Intrusion Detection Problems / Mei-Ling Shyu, Kanoksri Sarinnapakorn, Indika Kuruppu-Appuhamilage et al. // Proceedings of the 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications. — RIDE '05. — Washington, DC, USA : IEEE Computer Society, 2005. — P. 55–62. — URL: http://dx.doi.org/10.1109/RIDE.2005.10.

[4] Li H. Research and Implementation of an Anomaly Detection Model Based on Clustering Analysis // Intelligence Information Processing and Trusted Computing (IPTC), 2010 International Symposium on. — 2010. — Oct. — P. 458–462.

[5] Networks Palo Alto. Next Generation Firewalls. — URL: https://www.paloaltonetworks.com.

[6] NumPy. — URL: http://www.numpy.org.

[7] Omar Salima, Ngadi Asri, Jebur Hamid H. Article: Machine Learning Techniques for Anomaly Detection: An Overview // International Journal of Computer Applications. — 2013. — October. — Vol. 79, no. 2. — P. 33–41. — Full text available.

[8] Patcha Animesh, Park Jung-Min. Network anomaly detection with incomplete audit data // Computer Networks. — 2007. — Vol. 51, no. 13. — P. 3935 – 3955. — URL: http://www.sciencedirect.com/science/article/pii/S1389128607001235.

[9] Python Data Analysis Library. — URL: http://pandas.pydata.org.

[10] Requests: HTTP for Humans. — URL: http://docs.python-requests.org.

[11] Rui Zhang Shaoyan Zhang Yang Lan Jianmin Jiang. Network Anomaly Detection Using One Class Support Vector Machine // Proceedings of the International MultiConference of Engineers and Computer Scientists 2008. — IMECS 2008. — 2008. — P. 452–456. — URL: http://www.iaeng.org/publication/IMECS2008/IMECS2008_pp452-456.pdf.

[12] VirusTotal. — URL: https://www.virustotal.com.

[13] scikit-learn. Machine Learning in Python. — URL: http://scikit-learn.org.