Т.И. Паюсова, Е.И. Кудрявцев

Тюменский государственный университет, г. Тюмень

УДК 004.056

РАЗРАБОТКА SIEM-СИСТЕМЫ

Аннотация: В статье представлен начальный этап работы по реализации SIEM-системы с открытым исходным кодом.

Ключевые слова: SIEM-система, корреляция, статистическая обработка данных, разработка на c#, инциденты информационной безопасности, система управления информационной безопасностью

Одной из важнейших задач ИБ является своевременное реагирование на инциденты ИБ. К событиям ИБ, например, относятся: сетевые атаки; вирусные эпидемии или отдельные вирусные заражения; уязвимости; фрод, бэкдоры и трояны; попытки несанкционированного доступа к конфиденциальной информации и пр.

Возможны ситуации, когда инцидент ИБ обнаружен несвоевременно, а события в журнале либо затёрты, либо журнал недоступен, либо соединение с каждым источником событий ИБ и отдельный анализ соответствующих событий займёт слишком много времени. SIEM система позволяет объединить и хранить данные о событиях ИБ, полученных с разнообразных источников, например, систем контроля доступа и аутентификации пользователей, журналов событий серверов и рабочих станций, сетевых устройств, IDS/IPS, антивирусов, сканеров уязвимостей, GRC-систем, DLP, антифрод систем.

В крупных компаниях все аппаратные и программные средства защиты информации генерируют огромное количество логов, обычному человеку невозможно просмотреть их все, да еще и выявить зависимости. Именно с целью нахождения зависимостей и отслеживания логов были придуманы SIEM-системы.

SIEM системы приводят записи о событиях к одному формату, позволяют анализировать инциденты, SIEM визуализируют наиболее важные моменты, отфильтровывают некритичные события, SIEM система не должна самостоятельно справляться с инцидентами ИБ. SIEM система должна своевременно автоматически обнаруживать инциденты и оповещать о событиях, вести записи в собственных лог-файлах.

Среди SIEM-систем, представленных на рынке, можно выделить ArcSight, Cisco MARS, RSA Envision, NetForensics, NetIQ, Symantec, и др. Они предлагают огромное количество различного функционала, но, как правило, работают по принципу чёрного ящика, а также они очень дорогие, и не каждое предприятие может себе их позволить. Таким предприятиям необходима бесплатная SIEM-система с открытым исходным кодом, которую они смогут настроить под свои нужды.

Таким образом, цель моей работы: - разработка SIEM-системы, с открытым исходным кодом, осуществляющей анализ данных, в частности поиск статистически достоверных зависимостей.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить принципы построения SIEM-систем
2. Изучить методы статистического анализа данных
3. Разработать SIEM-систему, осуществляющую анализ лог-файлов из определённых источников
4. Провести апробацию SIEM-системы

SIEM –системы состоят из следующих элементов (Рис.1):

1. агенты, устанавливаемые на инспектируемую информационную систему (актуально для операционных систем; агент представляет собой резидентную программу (сервис, демон), которая локально собирает журналы событий и по возможности передает их на сервер);
2. сервера-коллекторы, предназначенные для предварительной аккумуляции событий от множества источников;

3. сервер-коррелятор, отвечающий за сбор информации от коллекторов и агентов и обработку по правилам и алгоритмам корреляции;

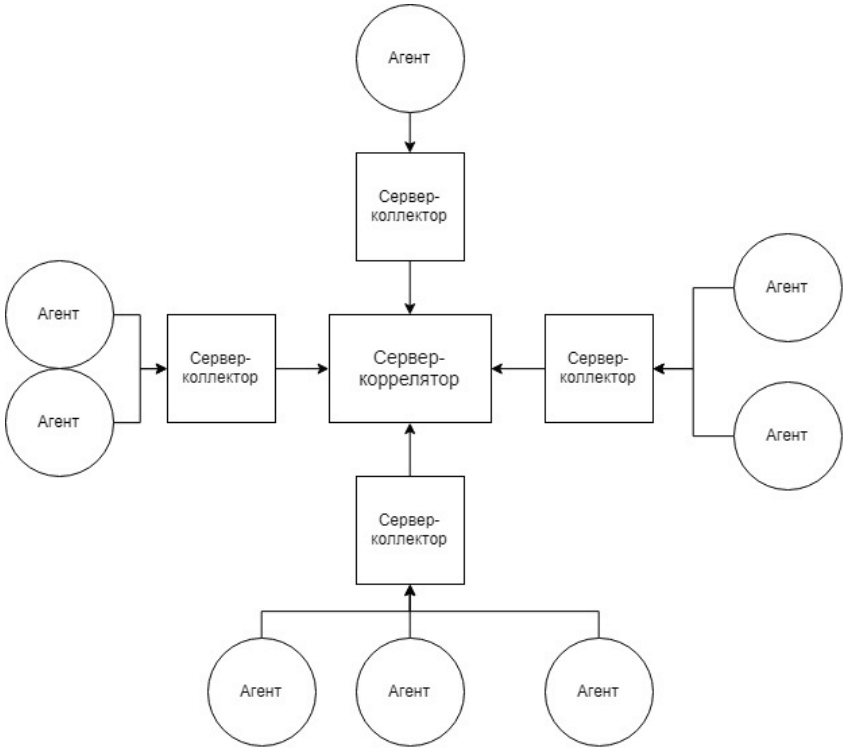

Рис. 1 Схема SIEM-системы

Существуют разные способы приведения файлов к одному виду, например, разные виды сериализаций, формирование простого текстового файла и др. Но ни один способ на самом деле не является универсальным, у каждого из перечисленных способов множество недостатков. Поэтому для каждого вида логов необходимо написание отдельного класса, который наследуется от абстрактного класса. Внутри него находиться абстрактный метод, который отвечает за парсинг логов. При выборе на клиенте файла логов, выбирается и класс, который будет его парсить.

Для разработки SIEM-системы выбран следующий инструментарий:

- Язык программирования C# в среде разработке Visual Studio 2013;
- Текстовый формат данных JSON, для хранения данных настроек программного обеспечения. Для работы с ним будет использована библиотека Newtonsoft.json;
- Для передачи данных между узлами будет использован протокол TCP. Будет использован встроенный в C# класс TcpListener;
- Ассиметричный шифр RSA для передачи данных по открытому каналу. Для шифрования будет использована встроенная реализация класса RSACryptoServiceProvider;
- Алгоритм хеширования SHA-512, для проверки на изменение файла.
- Для хранения данных будет использована СУБД MSSQL 2014.
- Так же для отслеживания изменения в файлах логах будет использован класс FileSystemWatcher, который при изменении файла сразу сообщит об это клиенту.

На данный момент реализованы агент, который будет устанавливаться на ПК или сервер, и пересылать выбранные лог файлы. Так же реализован сервер-коллектор, который будет получать лог файлы от клиента, приводить их к общему виду и пересылать их дальше.

Ниже приведен скриншот агента (Рис.2). При нажатии кнопки «Start» происходит запуск клиента, сначала проверяется, что произведенные все необходимые настройки: добавлен ip-адрес коллектора и выбраны файлы, которые необходимо передавать. Если все настройки корректные, то происходит попытка связаться с сервером, при неудаче выводится сообщение, если же все успешно, то клиент начинает следить за изменение файлов и передавать только изменения. При нажатии на кнопку «Stop», происходит остановка работы клиента. При нажатии на кнопку в виде шестеренки, открывается форма настройки (Рис.3), там есть возможность выбрать ip-адрес коллектора, файлы, которые необходимо передавать и класс

парсер, который будет приводить лог файл к общему виду. При нажатии на кнопку «Reload», происходит обновление списка классов парсеров, которые выбираются в форме настройки. При нажатии на кнопку «Exit», происходит закрытие программы.

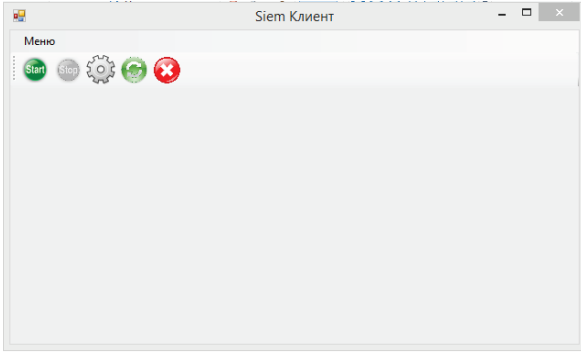

Рис. 2 Агент

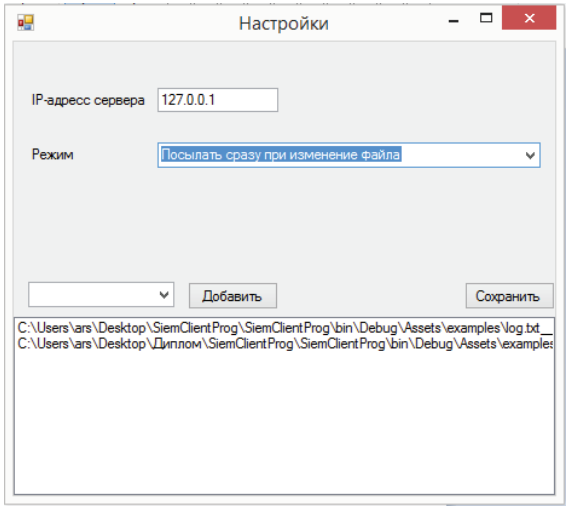

Рис. 3 Агент, настройка

Так же приведен скриншот сервера-коллектора (Рис.4). При нажатие на кнопку «Старт» сервер начинает слушать входящие соединения, при попытке соединения, он сначала посылает свой открытый ключ для асимметричного

шифрования, дальше получает зашифрованные данные, расшифровывает их, если получается найти нужный класс парсер, то парсит, если нет, то сообщает об этом клиенту, который уже удаляет файл из отслеживаемых источников. При нажатии кнопки «Stop», происходит остановка работы сервера. При нажатии кнопки «Settings» открывается форма настройки сервера, где можно добавить класс парсер и удобное имя, которое будет отображаться на клиенте. Для того чтобы убедиться, что вводимый класс существует, используется следующий код (Рис.5). Этот же код используется для динамического создания объекта необходимого класса парсера.

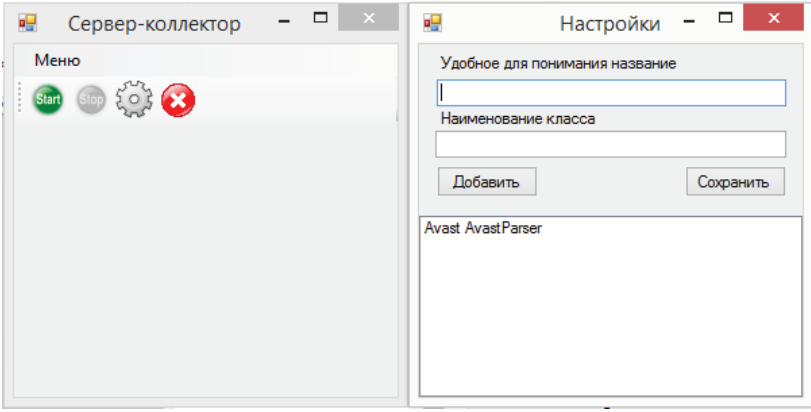

Рис. 4 Сервер-коллектор

```
var classExists = (from assembly in AppDomain.CurrentDomain.GetAssemblies()
                   from type in assembly.GetTypes()
                   where type.Name == SiemSettings.ParserClasses[Name]
                   select type).FirstOrDefault();
if (classExists == null)
{
    throw new Exception("Некоректная входная строка. Не найден класс Парсер для неё");
}
return Activator.CreateInstance(classExists);
```

Рис. 5 Проверка существования класса

Следующий этап разработки заключается в создании сервера-коррелятора. Сервер-коррелятор на вход будет получать приведенные к определенному формату лог файлы, дальше будет добавлять их в общую базу

и на основание выбранных правил искать зависимости, при нахождении подобных зависимостей будут оповещать об этом владельца.

Таким образом, на данном этапе проведён сравнительный анализ существующих siem-систем, обоснована целесообразность разработки нового продукта с открытым исходным кодом, реализован агент, сервер-коллектор.

Теперь планируется завершить разработку сервера – коррелятора и апробировать продукт в рамках реальной информационной системы.

СПИСОК ЛИТЕРАТУРЫ

1. Anton A. Chuvakin and Kevin J. Schmidt Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management 2012 г
2. Mark Talabis, Robert Mcpherson, I. Miyamoto, and Jason Martin Information Security Analytics: Finding Security Insights, Patterns, and Anomalies in Big Data 2014 г.
3. Новикова У.С., Котенко И.В. // Механизмы визуализации в siem-системах // Системы Высокой Доступности // 2012 г – С. 91-99
4. Шабуров А.С., Борисов В.И.// Разработка модели защиты информации корпоративной сети на основе внедрения siem-системы // Вестник пермского национального исследовательского политехнического университета. 2016 г. – С. 111-124
5. А.В. Федорченко, Д.С. Левшун, А.А. Чечулин, И.В. Котенко// 2016 г //Анализ методов корреляции событий безопасности в siem-системах. [On-line] http://proceedings.spiiras.nw.ru/ojs/index.php/sp/article/view/3366/0
6. HPE ArcSight ESM [On-line]:http://itgrd.ru/hp-arcsight-esm/
7. IBM QRadar Security Intelligence [On-line]: https://www.ibm.com/security/security-intelligence/qradar