О. В. Полубелова, И. В. Котенко

ПОСТРОЕНИЕ ОНТОЛОГИЙ УЯЗВИМОСТЕЙ И ПРИМЕНЕНИЕ ЛОГИЧЕСКОГО ВЫВОДА ДЛЯ УПРАВЛЕНИЯ ИНФОРМАЦИЕЙ И СОБЫТИЯМИ БЕЗОПАСНОСТИ[1]

Введение

В настоящее время одним из наиболее важных направлений исследований в области информационной безопасности компьютерных систем является технология управления информацией и событиями безопасности (Security Information and Event Management, SIEM) [1, 2]. Суть этой технологии заключается в обеспечении согласованной загрузки журналов безопасности из различных источников, их долго- или кратковременного хранения, моделирования и анализа для обнаружения атак, построения прогнозов.

В системах управления информацией и событиями безопасности (SIEM) важным аспектом, влияющим на скорость обработки и качество анализа данных, является способ их представления и хранения. Для этой цели в работе предлагается использовать один из популярных в настоящее время стандартов в области безопасности SCAP. Как правило, при построении модели данных на основе этого стандарта для SIEM-систем применяется реляционный поход. Он является не лучшим решением для сложной модели с большим количеством связей, которая в процессе управления безопасностью должна подвергаться разностороннему и глубокому анализу. В качестве альтернативы в работе для представления информации и событий безопасности предлагается использовать онтологический подход. Приводится пример построения онтологии для модели уязвимостей и атак на основе стандарта CVE (MITRE). Для хранения данных используется хранилище триплетов на базе сервера Virtuoso компании OpenLink software.

1. Стандарты представления данных в области информационной безопасности

В настоящий момент существует множество различных стандартов представления данных безопасности. Одним из популярных является протокол SCAP [3], который разрабатывается компанией Mitre и американским Национальным институтом по стандартизации и технологиям (National Institute of Standards and Technology) [4]. SCAP – это спецификация, которая объединяет ряд стандартов для унифицированного управления данными по безопасности. SCAP позволяет составить список используемых в системе платформ и приложений, задать особенности их конфигурации, неблагоприятно влияющие на защищенность, специфицировать список уязвимостей, оценить неблагоприятное влияние конфигураций и уязвимостей, выявить наиболее критичные уязвимости (обнаружить присутствие уязвимостей и присвоить им оценки критичности) [5]. SCAP включает в себя следующие стандарты: «Общее перечисление платформ» (Common Platform Enumeration – CPE) для задания программно-аппаратного обеспечения; «Общее перечисление конфигураций» (Common Configuration Enumeration – CCE) для представления особенностей программно-аппаратной конфигурации, неблагоприятно влияющих на защищенность; «Общие уязвимости и дефекты» (Common Vulnerabilities and Exposures – CVE) для описания списка уязвимостей данных продуктов; «Система оценки уязвимостей» (Common Vulnerabilities Scoring System – CVSS), используемая для определения неблагоприятного влияния конфигураций и уязвимостей, выявления наиболее критичных уязвимостей, на основе чего потом проводится исправление ошибок.

[1] Работа выполняется при финансовой поддержке РФФИ, программы фундаментальных исследований ОНИТ РАН (проект № 2.2), государственного контракта 11.519.11.4008 и при частичной финансовой поддержке, осуществляемой в рамках проектов Евросоюза SecFutur и MASSIF.

2. Онтологический подход для построения модели данных

Зачастую логика отношений между сущностями предметной области трудновыразима в реляционном представлении. При разработке модели вводится большое количество таблиц, и запросы на выборку таких данных получаются очень ресурсоемкими. Некоторые данные вообще невозможно представить с помощью реляционной алгебры, и они хранятся в базе в таком виде, что в дальнейшем требуется дополнительная программная обработка. А выборки из базы данных получаются очень большого размера. Это значительно увеличивает время работы программы.

Например, при описании уязвимости список продуктов, совокупность которых может привести к возникновению уязвимости, формируется с использованием логических операторов OR и AND. Список продуктов может быть любой длины. Пример такой уязвимости показан на рис. 1.

```
<vuln:vulnerable-configuration id="http://nvd.nist.gov/">
        <cpe-lang:logical-test negate="false" operator="OR">
                <cpe-lang:fact-ref name="cpe:/a:advantech:adam_opc_server:3.01.011"/>
        </cpe-lang:logical-test>
</vuln:vulnerable-configuration>
<vuln:vulnerable-configuration id="http://nvd.nist.gov/">
        <cpe-lang:logical-test negate="false" operator="OR">
                <cpe-lang:fact-ref
        name="cpe:/a:advantech:modbus_rtu_opc_server:3.01.010"/>
        </cpe-lang:logical-test>
</vuln:vulnerable-configuration>
<vuln:vulnerable-configuration id="http://nvd.nist.gov/">
        <cpe-lang:logical-test negate="false" operator="OR">
                <cpe-lang:fact-ref
        name="cpe:/a:advantech:modbus_tcp_opc_server:3.01.09"/>
</cpe-lang:logical-test>
2.
<vuln:vulnerable-configuration id="http://nvd.nist.gov/">
  <cpe-lang:logical-test negate="false" operator="AND">
      <cpe-lang:logical-test negate="false" operator="OR">
          <cpe-lang:fact-ref name="cpe:/a:microsoft:ie:9"/>
      </cpe-lang:logical-test>
      <cpe-lang:logical-test negate="false" operator="OR">
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_vista::sp2"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_vista::sp2:x64"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008::sp2:x86"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008::sp2:x64"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7:::x86"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::sp1:x86"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7:::x64"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_7::sp1:x64"/>
          <cpe-lang:fact-ref name="cpe:/o:microsoft:windows_server_2008:r2::x64"/>
          <cpe-lang:fact-ref
      name="cpe:/o:microsoft:windows_server_2008:r2::sp1:x64"/>
      </cpe-lang:logical-test>
  </cpe-lang:logical-test>
</vuln:vulnerable-configuration>
```

Рис.1. Пример представления уязвимости по стандарту CVE

В реляционной модели данных весь список продуктов, описывающих уязвимость, вместе с логическими операторами хранится в виде строки в таблице. Такая форма представления не позволяет задать параметризованный запрос с указанием названий продуктов, версий и т. д. Для анализа приходится использовать все уязвимости и обрабатывать их в программе. Этот процесс занимает очень много времени.

Одним из альтернативных решений по представлению данных является онтологический подход [6–8]. Суть этого подхода заключается в следующем. Сначала выделяется набор концептов (базовых понятий данной предметной области). Затем строятся связи между концептами, т. е. определяются отношения и взаимодействия базовых понятий. Концепты и отношения могут формулироваться с использованием дескрипционной логики, где термины словаря являются именами унарных и бинарных предикатов (соответственно концепты и отношения).

В рамках решения задачи представления моделей данных и реализации репозитория была разработана онтология для представления модели данных компонента Attack Modeling and Security Evaluation Component (AMSEC) [9]. На основе модели сети и вероятностей реализации уязвимостей (определяемых в виде весовых коэффициентов) AMSEC формирует граф атак. Далее AMSEC на основе построенного графа оценивает общий уровень защищенности сети, выявляет слабые места и оценивает возможные контрмеры, направленные на повышение уровня защищенности сети [10, 11]. Также AMSEC позволяет после обнаружения атаки вычислить вероятные характеристики нарушителя

(такие как уровень знаний, технические возможности, цели и т. д.), предсказать возможные направления развития атаки и возможные действия нарушителя, которые предшествовали проведению основной атаки (захват управления над сетевым оборудованием, кража паролей и т. д.) [9].

В основу разработанной для AMSEC модели данных был взят протокол SCAP. На рис. 2 представлена онтология, описывающая концепты уязвимостей, атак, программно-аппаратного обеспечения, производителей и другие понятия.

Рис. 2. Онтологическая модель для представления уязвимостей

В представленной модели отношения между программными продуктами и аппаратными компонентами, совокупность которых приводит к возникновению уязвимости, задаются с помощью дескрипционной логики. Связи между концептами представлены, в основном, через подклассы, а не через свойства объектов. Таким образом, логический вывод здесь сводится к задаче классификации, что повышает скорость обработки. На рис. 3 показан пример представления самой простой уязвимости, затрагивающей только один программный продукт Cache database компании Intersystem версии 5.

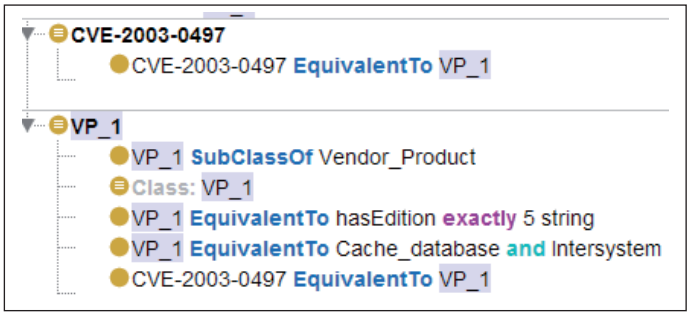

Рис. 3. Пример онтологического представления уязвимости

Такой подход представления уязвимостей для модуля AMSEC позволил загружать из базы намного меньший объем данных и избавиться от необходимости их программной обработки, переложив задачу анализа на систему логического вывода.

3. Хранение данных

Для хранения онтологий был разработан репозиторий. В качестве хранилища использовался сервер Virtuoso компании OpenLink software, который поддерживает как реляционную СУБД, так и хранилище триплетов.

В качестве основы для манипулирования данными используется сервис-ориентированная архитектура (SOA). Она реализует множество веб-сервисов для доступа к данным в репозитории. Преимуществами этой архитектуры являются гибкость и слабая связанность компонентов, что обеспечивает высокую масштабируемость и расширяемость системы.

Архитектура SOA представляет собой концепт распределенной информационной среды, которая объединяет воедино различные программные модули и приложения, основанные на хорошо определенных интерфейсах, и обеспечивает их взаимодействие.

Основным принципом архитектуры SOA является то, что элементы бизнес-процессов и элементы информационной инфраструктуры, их обеспечивающей, рассматриваются как компоненты, которые объединены и попеременно используются как «строительные блоки» для реализации корпоративных процессов.

Анализ данных в хранилище триплетов осуществляется с помощью системы логического вывода.

Заключение

Предложенный в работе онтологический подход к построению модели данных позволяет существенно уменьшить выборку данных из хранилища за счет повышения точности запросов. Использование систем логического вывода, являющихся надстройками для хранилищ триплетов (СУБД для онтологий), позволяет проводить глубокий и разносторонний анализ данных, выдавая для программных систем более качественный результат.

В дальнейшей работе планируется расширить предложенную онтологию уязвимостей, а также добавить различные задачи по анализу данных.

СПИСОК ЛИТЕРАТУРЫ:

1. *Котенко И. В., Саенко И. Б., Полубелова О. В., Чечулин А. А.* Применение технологии управления информацией и событиями безопасности для защиты информации в критически важных инфраструктурах // Труды СПИИРАН. Вып. 1 (20). СПб.: Наука, 2012. С. 27–56.

2. *Котенко И. В., Саенко И. Б., Полубелова О. В., Чечулин А. А.* Технологии управления информацией и событиями безопасности для защиты компьютерных сетей // Проблемы информационной безопасности. Компьютерные системы. 2012. № 2. С. 57–68.

3. SCAP. URL: http://scap.nist.gov/ (дата обращения: 05.04.2013)

4. NIST. URL: http://www.nist.gov/index.html (дата обращения: 05.04.2013)

5. *Котенко И. В., Дойникова Е. В.* Анализ протокола автоматизации управления данными безопасности SCAP // Защита информации. Инсайд. 2012. № 2. С. 56–63.

6. *Guo M., Wang J. A.* An Ontology-based Approach to Model Common Vulnerabilities and Exposures in Information Security // ASEE SE Section Conference. Marietta, GA. 2009.

7. *Parmelee M. C.* Toward an Ontology Architecture for Cyber-Security Standards // STIDS 2010 Proceedings. 2010.

8. *Elahi G., Yu E., Zannone N.* A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations // Proceedings of the 28th International Conference on Conceptual Modeling (ER 2009). Auckland, New Zealand, 2009. P. 99–114.

9. *Kotenko I., Chechulin A., Novikova E.* Attack Modelling and Security Evaluation for Security Information and Event Management // SECRYPT 2012. International Conference on Security and Cryptography. Proceedings. Rome, Italy. 24–27 July 2012. P. 391–394.

10. *Котенко И. В., Степашкин М. В.* Оценка защищенности компьютерных сетей на основе анализа графов атак // Проблемы управления рисками и безопасностью. Труды Института системного анализа Российской академии наук. Т. 31. М.: URSS, 2007. С. 30–49.

11. *Kotenko I., Stepashkin M., Doynikova E.* Security Analysis of Computer-aided Systems taking into account Social Engineering Attacks // Proceedings of the 19th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2011). IEEE Computer Society. 2011. P. 611–618.