УДК 004.62

І.В. Кобзев[1], К.Е. Петров[1], О.В. Орлов[2]

[1] *Харківський національний університет внутрішніх справ, Харків*
[2] *Харківський регіональний інститут Національної академії державного управління при Президентові України, Харків*

МОДЕЛІ ВИЗНАЧЕННЯ ТА ЗАПОБІГАННЯ АТАК НА WEB-СЕРВІСИ

В статті розглядається сервісно-орієнтована архітектура та Web-сервіси як її частина. Представлені дві моделі визначення і запобігання атак на Web-сервіси. Перша модель побудована для кожного з атрибутів прикладного програмного інтерфейсу і використовується для запобігання Web-атакам з використанням ін'єкцій. Друга модель описує поведінку тонкого клієнта.

Ключові слова: *атака, Web-сервіс, аналіз, модель, клієнт, атака, proxy.*

Вступ

Сервісно-орієнтована архітектура (Service-oriented architecture, SOA) – модульний підхід до розробки програмного забезпечення. Цей підхід заснований на використанні розподілених, слабко пов'язаних компонентів, що оснащені стандартизованими інтерфейсами для взаємодії за стандартизованими протоколам [1]. SOA забезпечує технологічні можливості для роботи з сервісами, тобто не просто з програмним або апаратним забезпеченням, а з певними завданнями бізнесу. SOA – це шаблон для розробки гнучких застосувань, що мінімізує зв'язки між компонентами і в той же час надає можливості повторного використання існуючих рішень. А підтримка єдиних стандартів в масштабі підприємства забезпечує простоту взаємодії між сервісами.

Одним із засобів реалізації SOA є використання Web-сервісів. У питанні інтеграції розподілених зовнішніх систем в теперішній час все частіше використовується технологія Web-сервісів, як засобів надання незалежного, уніфікованого інтерфейсу для видаленого доступу до інформаційних ресурсів. У такому контексті Web-сервіс виступає в ролі автономного застосування, яке надає засоби доступу до інформації зовнішнім клієнтам через набір послуг, що надаються ним. Технологія Web-сервісів базується на відкритих XML-стандартах, таких як:

- SOAP (Simple Object Access Protocol) – протокол обміну структурованими повідомленнями в розподілених обчислювальних системах, що базується на форматі XML. [2];
- WSDL (Web Services Description Language) – мова опису зовнішніх інтерфейсів Web-служби [3];
- UDDI (Universal Description, Discovery and Integration) – платформово-незалежний інструмент для розміщення описів Web-сервісів (WSDL) для забезпечення можливості їх пошуку та інтеграції у свої власні системи іншими організаціями. UDDI описує модель даних, що призначена для каталогізації і доступу до послуг, що надаються Web-сервісами [4].

Використання Web-сервісів надає широкі можливості щодо інтеграції різних інформаційних систем в рамках єдиного набору специфікацій.

Розвернувши платформу на базі SOA, організація отримує достатньо широкі можливості при визначенні того, які зі своїх служб можна зробити доступними для клієнтів, партнерів і постачальників. З іншого боку, клієнти, партнери і постачальники зможуть використовувати великий набір запропонованих сервісів. Технології Web-сервісів підвищують продуктивність і допомагають компаніям ефективніше реагувати на вимоги ринку. На відміну від інших технологій електронного бізнесу, архітектура Web Services Architecture (WSA) забезпечує повну незалежність від платформи реалізації за допомогою визначення набору Web-стандартів, в рамках яких здійснюється взаємодія, полегшуючи тим самим процес міжсистемної інтеграції і зменшуючи труднощі взаємодії для користувачів. Web-сервіси потенційно можуть забезпечити і вищий прибуток на інвестований капітал за рахунок скорочення термінів і витрат на впровадження нових застосувань, широкого впровадження Web-технологій і зростання доходів.

Зокрема, метою технології Web-сервісів є ліквідація багатьох обмежень, що накладаються сучасними інформаційними технологіями на взаємодію організацій, і, як наслідок цього, підвищення якості бізнес-процесів, що у результаті дозволяє досягти кращих кінцевих результатів. Постачальники технологій Web-сервісів в змозі запропонувати широкий асортимент програмних і апаратних засобів, а також партнерські послуги, що підтримують цю технологію.

Аналіз досліджень та публікацій. Протягом декількох останніх років було докладено багато зусиль, щоб розвинути інфраструктуру, підтримати

розгортання, відкриття і використання Web-сервісів. Головні постачальники комп'ютерної і програмної інфраструктури, включаючи IBM, Microsoft, Sun Microsystems, наполегливо працюють, розширюючи свою технологічну середу, щоб підтримати розвиток, розгортання, і обслуговування Web-сервісів. Їх рішення направлені на поліпшення безпеки, підтримку транзакцій, і покращення координації Web-сервісів.

Із-за високої складності розробки захищених систем в 90-і роки XX століття з'явилося і почало активно розвиватися направлення інформаційної безпеки, пов'язане з виявленням (і, можливо, подальшим реагуванням) порушень безпеки інформаційних систем, як ефективне тимчасове рішення, що дозволяє закриваи «проломи» в безпеці систем [5]. Даний напрям отримав назву «Виявлення атак» (intrusion detection); і за минулі роки в рамках академічних розробок були створені сотні систем виявлення атак для різних платформ: від систем класу mainframe до сучасних операційних систем загального призначення, СУБД і поширених застосувань [6, 7].

Методи виявлення атак в сучасних системах недостатньо опрацьовано в частині формальної моделі атаки, і, отже, для них досить складно строго оцінити такі властивості як обчислювальна складність, коректність, завершеність і так далі [7, 8]. Прийнято розділяти методи виявлення атак на методи виявлення аномалій і методи виявлення зловживань [8].

Існує багато академічних розробок для виявлення аномалій, але в промислових системах вони використовуються рідко і з великою обережністю, оскільки такі системи породжують велику кількість помилкових спрацьовувань.

До другого типу методів відносяться більшість сучасних комерційних систем (Cisco IPS, ISS RealSecure, NFR), які використовують сигнатурні (експертні) методи виявлення [8]. Для експертних систем основною проблемою є низька, близька до нуля, ефективність виявлення невідомих атак (адаптивність) [9]. Низька адаптивність до цього часу залишається великою проблемою, хоча такі якості як низька обчислювальна складність і мала вартість розгортання визначають домінування таких систем в даній предметній області.

Мета статті. На теперішній час існує безліч систем виявлення атак, як взагалі на комп'ютерні системи, так і на Web-сервіси зокрема. Існує досить обмежене число методик виявлення таких атак. Для виявлення атак на Web-сервіси найчастіше використовують системи виявлення сигнатур.

Головними недоліками систем виявлення сигнатур є: принципова неможливість виявлення нових атак і вторгнень; пропуски варіантів відомих вторгнень; необхідність розробки і постійного поповнення бази даних сигнатур [10].

Головними загрозами інформаційній безпеці Web-сервісів є несанкціоновані зміни повідомлень, втрата їх конфіденційності і автентичності, DoS-атаки, SQL-ін'єкції, переповнення буферу (BoF), завантаження довільних файлів (Unrestricted File Upload). Забезпечення інформаційної безпеки Web-сервісів передбачає використання таких загальноприйнятих технологій інформаційної безпеки як шифрування, цифровий підпис, парольний захист і т.ін.

В статті представлені дві моделі визначення і запобігання атак на Web-сервіси. Перша модель побудована для кожного з атрибутів прикладного програмного інтерфейсу (Application Programming Interface, API) Web-сервісу і використовується для запобігання Web-атакам з використанням ін'єкцій. Друга модель описує поведінку тонкого клієнту Web-сервісу і запобігає вторгненням в систему з цієї сторони.

Основні поняття та визначення

Введемо наступні визначення. Позначимо Web-сервіси як WS. $U = \{U_1, U_2, ..., U_n\}$ – набір користувачів Web-сервісу. Атрибути прикладного програмного інтерфейсу для WS визначимо як $API = \{API_1, API_2, ..., API_m\}$. Вхідний атрибут q для API_j позначимо $A_{j,q}$. Набір вхідних атрибутів будемо розглядати як рядок з Unicode символів. Такий рядок позначимо $S=\{s_i \,|\, s_i \in \text{Unicode characters}\}$. Ймовірність переходу до s_i позначимо $CT(s_i)$. Набір всіх вхідних атрибутів Web-сервісів (WS) для API_j позначимо $Q_j = \{A_{j,q} \,|\, A_{j,q}, A_{j,2}, ..., A_{j,k}, 1 \le q \le k\}$. $q_{i,j}$ визначає U_i клієнта Web-сервісу, який відправляє запит по протоколу SOAP з використанням API_j до WS. В загальному випадку запит виконується через протокол HTTP з використанням методу доступу POST. Атрибути необхідні для API визначаються специфікацією SOAP [11].

Наведемо приклад запиту згідно протоколу SOAP до віртуальної Інтернет-крамниці за API адресою http://virtual_shop.com/soap/.

```
POST /soap HTTP/1.1
Host: virtual_shop
Connection: Keep-Alive
User-Agent: PHP-SOAP/5.3.2
Content-Type: application/soap+xml;
charset=utf-8;
action="http://virtual_shop.com/soap#getProdInf"
Content-Length: 570
<soap:header>
```

```
<locale>ua</locale>
</soap:header>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/env
elope/">
<soap:Body>
<Inv:GetProductInformationByID>
<Inv:ID>1</Inv:ID>
<Inv:SecurityToken>12312s-dad23a4544sd-
343dff4</Inv:SecurityToken>
<Inv:GetProductInformationByID>
</soap:Body>
</soap:Envelope>
```

Клієнт системи U_i відправляє запит 'GetProductInformationByID' до API_j. Атрибутами API_j є 'ID' та 'SecurityToken'. Таким чином

$$Q_j = \{A_{j,1}, A_{j,2}\} = \{\text{ID, SecurityToken}\}.$$

Для ілюстрації вищеозначеного будемо використовувати віртуальний кошик користувача Інтернет-крамниці.

Для того, щоб отримати доступ до інформації про продукти крамниці користувач Web-сервісу повинен пройти аутентифікацію з використанням API протоколу.

Моделі запобігання атак на Web-сервіси

Для побудови першої моделі запобігання атакам дані представляються в вигляді набору Web-сервісів. Цей набір включає всі доступні Web-сервіси прикладного програмного інтерфейсу API_j та відповідні значення атрибутів $A_{j,q}$. Далі необхідно зібрати достатню кількість даних для кожного атрибуту $A_{j,q}$ використовуючи s_i та $CT(s_i)$.

Таким чином, перша модель показує всі можливі варіанти переходу для кожного вхідного атрибуту $A_{j,q}$. Якщо модель не може знайти шлях переходу для даного значення атрибуту, то відповідне значення атрибуту визначається як можливий рядок атаки. Ймовірність того, що $A_{j,q}$ є продуктом s_i та $CT(s_i)$ визначається як $P(A_{j,q}) = P(s_1, s_2, ..., s_n)$.

Таким чином $P(A_{j,q}) = P(s_1) \cdot \prod_{i=2}^{k} P(s_i) \cdot CT(s_i)$.

Модель для вхідного атрибуту 'username' та Web-сервісу 'Authenticate' за протоколом API представлена на рис. 1.

Припустимо, U_i має пройти аутентифікацію за протоколом API з username=guest. Використовуючи нашу модель, ми можемо перевірити ймовірність використання атрибуту username.

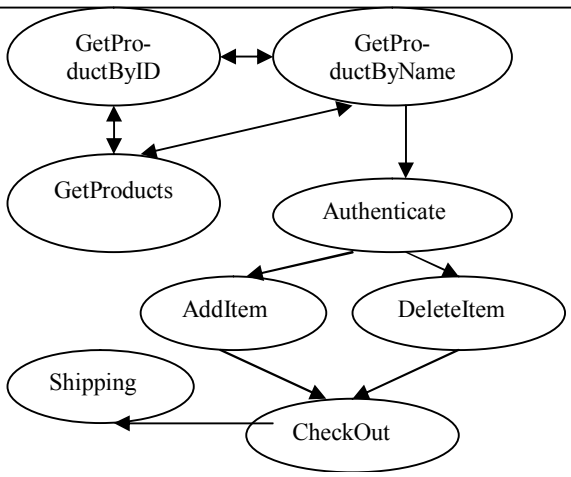

Рис. 1. Приклад переходів для запитів Web-сервісу 'Authenticate' за протоколом API

Друга модель повинна визначити невірні переходи до Web-сервісу згідно з протоколом API зі сторони тонкого клієнта. Правильні переходи до Web-сервісу можуть бути визначені за допомогою обрахування ймовірності Q_j для кожного API_j, тобто

$$P(Q_1, Q_2, Q_n, API_1, API_2, \ldots, API_n) =$$

$$= P(API_1) \cdot \prod_{m=2}^{n} P(API_m \,|\, API_{m-1}) \cdot P(Q_m \,|\, API_m),$$

де $Q_j = \left[\prod_{q=1}^{k} P(A_{j,k}) \right]^{1/k}$.

Використовуючи цю модель, ми отримуємо можливість визначити, чи є поведінка клієнта Web-сервісу втручанням у систему. Для цього треба обчислити ймовірність API переходу та порівняти отримане значення з нормальним граничним значенням. Якщо це значення менше ніж граничне, то робимо висновок про втручання у систему.

Всю система в цілому є можливість розділити на дві фази: фаза навчання і фазу реалізації.

У фазі навчання система вивчає модель для кожного вхідного атрибуту API_i Web-сервісу переходу для Web-служб (WS). Для цього набору даних в моделі відображається інформація, зібрана через SOAP запит. Перша модель будується шляхом обчислення ймовірності переходу, а друга будується на базі API Web-служб.

Пропонується використовувати Proxy-сервери (WS-проксі), які знаходяться між клієнтом Web-служби (WS-клієнт) і цільовою Web-службою (WS). На рис. 2 представлена архітектура подібної системи.

Якщо користувач виконує необґрунтовані API переходи до Web-сервісу, то запропонована модель оцінює ймовірність того, що така послідовність дорівнює нулю. WS-проксі виявляє такі нелогічні послідовності і захищає Web-сервіс.

Модель проаналізованої системи служить для оцінки результату атак і визначення реакції мережі на них. Вона складається з наступних основних модулів: розпізнавання дій зловмисника; обчислення результату проведення атак; генерації відгуків системи; бази знань про мережу (систему), що аналізується; бази даних сигнатур атак; мережевого інтерфейсу.

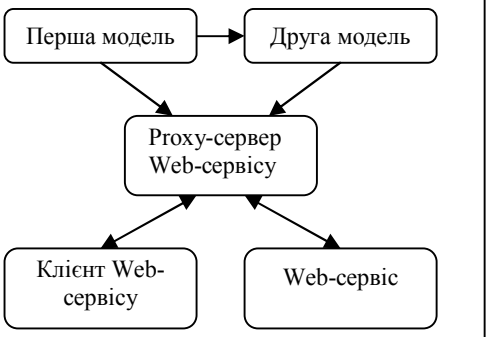

Рис. 2. Архітектура системи, яка використовує Web-сервіси

Одним з основних модулів є модуль обчислення результату атак, що використовує множину правил, які описують, який тип атаки, за яких умов і з якою ймовірністю успіху реалізується.

Вхідними даними для правил є ідентифікатор атаки і множина параметрів, що описують Web-сервіс. Вихідним значенням є ймовірність успішного виконання атаки.

Висновки

В статті розглянуто сервісно-орієнтована архітектура та Web-сервіси як її частина. Запропоновано дві моделі визначення і запобігання атак на Web-сервіси.

Подальші дослідження слід, в першу чергу, спрямувати на розробку моделей, інструментальних засобів і програмного забезпечення визначення вразливостей Web-сервісів.

Список літератури

1. Сервісно-орієнтована_архітектура / Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу URL: http://uk.wikipedia.org/wiki/Сервісно-орієнтована_архітектура.

2. SOAP / Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу URL: http://uk.wikipedia.org/wiki/SOAP.

3. W3C notes on Web Services Description Language (WSDL). [Електронний ресурс]. – Режим доступу URL: http://www.w3.org/TR/wsdl.

4. UDDI Spec TC. [Електронний ресурс]. – Режим доступу URL: https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm.

5. Anderson J.P. Computer Security Threat Monitoring and Surveillance / J.P. Anderson Co, Fort Washington, PA, April 1980.

6. Stefan Axelsson. Research in Intrusion-Detection Systems: A Survey / Stefan Axelsson // Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, 1999.

7. Stefan Axelsson. Intrusion detection systems: A survey and taxonomy / Stefan Axelsson // Technical Report 99-15, Chalmers Univ., March 2000.

8. Bace R. An Introduction to Intrusion Detection Assessment for System and Network Security Management / R. Bace. – 1999.

9. Lu W. A new unsupervised anomaly detection framework for detecting network attacks in real-time / W. Lu, I. Traore // Department of Electrical and Computer Engineering, University of Victoria, Lecture notes in computer science. – 2005.

10. Смыков Геннадий. Новый взгляд на обнаружение и предотвращение web-атак. [Електронний ресурс] / Геннадий Смыков. Режим доступу URL: http://www.securitylab.ru/contest/290792.php.

11. SOAP Version 1.2 Part 0: Primer. W3C Recommendation 24 June 2003. [Електронний ресурс]. – Режим доступу URL: http://www.w3.org/TR/2003/REC-soap12-part0-20030624.

Надійшла до редколегії 10.10.2013

Рецензент: д-р техн. наук, проф. С.Г. Удовенко, Харківський національний університет радіоелектроніки, Харків.

МОДЕЛИ ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ АТАК НА WEB-СЕРВИСЫ

И.В. Кобзев, К.Э. Петров, А.В. Орлов

В статье рассматривается сервисно-ориентированная архитектура и Web-сервисы как ее составляющая часть. Представлены две модели определения и предотвращения атак на Web-сервисы. Первая модель построена для каждого из атрибутов прикладного программного интерфейса и используется для предотвращения Web-атак с использованием инъекций. Вторая модель описывает поведение тонкого клиента.

Ключевые слова: *атака, Web-сервіс, анализ, модель, клиент, атака, proxy.*

DETECTION MODEL OF WEB SERVICES ATTACKS

I.V. Kobzev, K.E. Petrov, O.V. Orlov

In the article to be considered a service-oriented architecture model for Web-services attacks detection. These two models identify and prevent attack on Web-Services. First model using Web-service input attributes. Second model coupled with Web-service client behavior of the system.

Keywords: *attack, Web-сервис, analysis, model, client, attack, proxy.*