ПРИМЕНЕНИЕ МЕТОДА МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА АТТРИБУТОВ ЗАПРОСА В ЗАДАЧАХ ОБНАРУЖЕНИЯ АТАК НА ВЕБ-ПРИЛОЖЕНИЯ

Уязвимости веб-приложения представляют собой значительную часть угроз безопасности компьютерных сетей. Для обнаружения известных сетевых атак системы обнаружения атак оснащены большим количеством подписей. К сожалению, трудно успеть за ростом уязвимостей, связанных с веб-сайтами, и, кроме того, уязвимости могут быть внедрены при установке конкретных веб-приложений. Атаки на Веб-ресурсы чаще всего происходят на прикладном уровне, в виде HTTP/HTTPS-запросов к сайту, где у традиционных межсетевых экранов крайне ограниченные возможности для анализа и обнаружения атак. Для защиты Веб-ресурсов от атак на прикладном уровне существуют специальные средства, в том числе, межсетевой экран для веб-приложения (WAF – Web Application Firewall). В этой статье представлены некоторые признаки обнаружения аномалий, и принципы работы межсетевого экрана для Веб-приложения с открытым текстом ModSecurity, в котором используется метод машинного обучения с тремя предлагаемыми признаками для обнаружения атак на веб-приложения.

Ключевые слова: веб-атака, сетевая безопасность, сигнатурный метод, метод обнаружения аномалий, метод машинного обучения.

Nguyen Manh Thang

THE APPLICATION OF MACHINE LEARNING METHOD AND ANALYSIS ATTRIBUTES OF QUERIES FOR DETECTING ATTACKS ON WEB APPLICATIONS

Web application Vulnerabilities represent a significant part of security threats to computer networks. To detect known network attacks, system detection attacks are equipped with a large number of signatures. Unfortunately, it is difficult to keep up with the daily disclosure of vulnerabilities associated with websites. In addition, vulnerabilities can be introduced when installing specific web applications. Attacks on Web resources most often occur at the application level, in the form of HTTP / HTTPS-requests to the site, where traditional firewalls have limited capabilities for analysis and detection attacks. To protect Web resources from attacks at application level, there are special tools - Web Application Firewall (WAF). This article presents some features of anomaly detection existed in the systems detection attacks and working principles of open source Web application firewall ModSecurity, in which uses machine learning method with three proposed features to detect attacks on Web applications.

Keywords: web attack, network security, signature method, anomaly detection method, machine learning method.

1. Введение

В современном обществе применение компьютерных технологий как для работы, так и для личного использования, быстро растет со временем. К сожалению, с ростом числа и размера компьютерных сетей и систем их уязвимость также увеличивается (Рис. 1)[1].

гут манипулировать этими запросами и создавать запросы, которые способны повредить сервер или собрать конфиденциальную информацию.

Важной причиной неуверенности веб-приложений является то, что многим программистам не хватает знаний о безопасном

Рис. 1. Глобальное количество веб-атак, заблокированных в день с 2015 по 2018 год (в тысячах секундах)

Одной из самых популярных целей атак являются веб-серверы и веб-приложения. Пользователи веб-серверов и веб-приложений запрашивают и отправляют информацию с помощью запросов. Все запросы и ответы в HTTP-трафике представляют собой строки, содержащие набор параметров, имеющих специальные значения [2, 3]. Противники мо-

программировании, поэтому они оставляют приложения с недостатками [4]. Однако механизмы безопасности веб-приложений ослабевают в двух крайних случаях. С одной стороны, есть методы, которые отвлекают программиста, например, межсетевой экран для веб-приложения и другие средства защиты [5]. С другой стороны, существуют методы, ко-

которые обнаруживают уязвимости, но программист не совсем понимает его внутреннюю структуру, например, метод тестирования черного ящика [6] и метод статического анализа [7].

Кроме того, некоторые типы атак, такие как атаки сканирования и зондирования, позволяют идентифицировать запущенные службы на веб-сервере с целью его компрометации. Наконец, атаки с использованием перехвата паролей нацелены на то, чтобы предоставить злоумышленникам привилегии администратора привилегии для веб-сервера, тогда атака типа "отказ в обслуживании" позволяет злоумышленникам прервать работу веб-серверов. Для защиты веб-серверов и веб-приложений могут использоваться множество подходов, такие как системы обнаружения атак (COA), межсетевой экран,....

Каждый подход имеет недостатки, поэтому с целью повышения точности обнаружения сетевых атак нам необходимо соединить несколько подходов и применить метод машинного обучения для обнаружения аномалий в системе защиты информации. В пункте 2 автор будет рассматривать положительные и отрицательные стороны существующих подходов защиты веб-приложений с целью преодоления недостатков в нашем подходе.

2. Исследование существующих способов защиты Веб-приложения

Используемые способы защиты веб-приложений могут быть распределены на три группы: защита при разработке веб-приложений (G1); оценка безопасности существующих веб-приложений (G2); использование решения защиты существующих веб-приложений во время выполнения (G3).

В этой статье автор выбирает конечное направление (G3). Самый распространенный подход этого направления – использование межсетевого экрана. В распространенных типах межсетевого экрана используется информация сетевого уровня и транспортного уровня (например, номер порта, протокол) для предотвращения несанкционированного доступа. *Iptables* является типичным примером межсетевого экрана сетевого уровня. Он работает по умолчанию в многих дистрибутивах операционных систем Linux (CentOS, Ubuntu ...) и на основе классификации пакетов ввода-вывода в соответствии с заранее установленными правилами. Чтобы получить доступ к веб-сайту, необходимо уста-

новить правила *Iptables* для открытия портов HTTP-80 и HTTPS - 443. А также для отправки почты, нам нужно открыть порты SMTP-22, SMTPS- 465/587 и т.д. Однако эти решения межсетевого экрана позволяют пакетам обращаться к серверам, которые широко доступны для многих пользователей, например, веб-серверам, FTP-серверам. Поэтому защита этих серверов переносится на создание приложений с защищенным исходным кодом. Создание программы без уязвимости невозможно. Другие способы защиты таких приложений, как шлюзы и веб-сайты, используют систему обнаружения атак.

Проблема с традиционной COA заключается в том, что они не могут определить правильность формата HTTP-пакетов. Это снова может быть методом, используемым для атаки на веб-серверы. Для решения этой проблемы возник межсетевой экран для веб-приложений (WAF). Примерами атак, которые могут обнаружить WAF, являются: SQL-инъекция, межсайтовое выполнение сценариев, захват сеанса, нарушение URL-адресов, переполнение буфера.

В настоящее время существует множество решений WAF с открытым исходным кодом, таких как ModSecurity [8, 9], AQTRONIX WebKnight [10]. Программное обеспечение с открытым исходным кодом имеет не только свои преимущества, но и недостатки.

Существует три распространенных типа ошибок веб-приложений: ошибки, связанные с введенными пользователем данными (T1); ошибки, связанные с управлением сеансом (T2); ошибки, связанные с логикой веб-приложения (T3).

Большинство типов ошибок, перечисленных OWASP, относятся к первому случаю (T1). Поэтому в этой статье автор фокусирует на решениях обнаружения атак, связанных с входными данными.

На этапе обучения нам надо моделировать атрибуты параметров (например, длина атрибутов, область значений параметров и т.д.) или атрибуты запросов GET и POST (запрос GET, показанный на рисунке 2). Затем эта информация используется для создания профилей для соответствующего приложения. Модель возвращает вероятность соответствующих компонентов (параметров, запроса). Если аномалия больше заданного порога, запрос считается атакой.

В пункте 3 автор будет рассматривать модель обнаружения сетевых атак с помощью

Рис. 2. Входящий данные запроса GET в файле регистрации веб-сервера

Рис. 3. Расположение WAF в системе ИБ для Веб-приложения

межсетевого экрана для веб-приложения и некоторые атрибуты, которые используются для обнаружения аномалий.

3. Модель обнаружения атак на веб-приложения

Наш подход к обнаружению аномалий заключается в том, что WAF анализирует HTTP-запросы, оправленные с веб-браузера на веб-сервер, и решает, выполнять или нет запросы на сервере (рис.3).

Работа WAF разделена на две фазы: фазу обучения и фазу обнаружения атаки.

3.1. Фаза обучения:

На фазе обучения WAF содержит 4 этапа:

Извлечение данных между веб-браузером и веб-сервером:

На первом этапе автор использует встроенные правила межсетевого экрана для веб-приложения (в этом случае Modsecurity).

SecRule ARGS,REQUEST_METHOD

"~(GET|PUT|HEAD|OPTIONS|DELETE|CONNECT|TRACE)\$"

"pass,setenv:input_ip=%{REMOTE_ADDR},setenv:input_raw=http://% {SERVER_NAME} % {REQUEST_URI}"

Вышеприведенное правило будет запущено, когда аргумент REQUEST_METHOD имеет значения, которые принадлежат к множеству значений {GET,PUT,HEAD,OPTIONS,DELETE,E,CONNECT,TRACE}.

Модуль машинного обучения запускает-

ся, когда клиент начинает отправлять запрос на сервер. Этот модуль выполняется с указанными выше входными параметрами, обрабатывается и после этого все результаты сохраняются в базе данных.

b. Вычисление необходимых значения:

На этом этапе автор рассчитывает значения полей / группу полей. Все необходимые значения, такие как математическое ожидание, дисперсия, стандартное отклонение, частота появления символов в запросах будут рассчитаны и сохранены в таблице базы данных. Эти значения считаются «стандартными» для будущих запросов. Входящие запросы будут основываться на этом стандарте для генерации восьми значений P для каждого запроса.

Автор рассматривает восемь атрибутов следующим образом: длина поля, отправленного из браузера (P1); распределение символов в запросе (P2); оценка значения поля в конечном множестве (P3); отсутствие поля (P4); порядок полей, отправленных из браузера (P5); **длина запроса, отправленного из браузера (P6); появление новых символов (P7); появление новых ключевых слов (P8).**

Пять атрибутов [P1–P5] можно читать в работе [11]. Автор предлагает три новых атрибута [P6–P8] для повышения точности обнаружения.

c. Вычисление значений восьми атрибутов:

Рис. 4. Процесс работы на фазе обучения WAF

На этом этапе все данные будут маркироваться марками «N» для чистых данных и «Y» для данных атаки. После прохождения процесса обучения и достижения заданного порога была создана таблица, содержащая значения P с меткой «Y/N».

d. Использование модели дерева принятия решений для установки порогов безопасности и атак:

Для построения дерева автор использует алгоритм J48 - популярная реализация алгоритма C4.5, написана на языке Java и входит в пакет Weka.

Пусть нам задано некоторое обучающее множество M, содержащее объекты (примеры), каждый из которых характеризуется 8 атрибутами, причем один из них указывает на принадлежность объекта к определенному классу. Пусть через {C1, C2} обозначены классы {атака, не атака}.

Для выбора наиболее подходящего атрибута, предлагается следующий критерий:

$$Gain(X) = Info(M) - Info_x(M) \quad (1)$$

где $Info(M)$ – энтропия множества M, а

$$Info_x(M) = \sum_{i=1}^n \frac{|M_i|}{|M|} * Info(M_i) \quad (2)$$

где $M_1, M_2, \dots, M_n$ получены при разбиении исходного множества M по проверке X. Выбирается атрибут, дающий максимальное значение по критерию (1).

3.2. Фаза обнаружение:

В обычной модели веб-клиент и веб-сервер обмениваются данными друг с другом посредством запросов и ответов. Для каждого запроса автор [11] предлагает использовать атрибуты на двух уровнях: по одному для каждого поля данных и по одному для запроса. Этим атрибутам присваивается оценка

Рис 5. Процесс работы на фазе обнаружения WAF

от 0 до 1, чтобы отразить тот же уровень, что и ранее известный профиль атрибута. С другой стороны, если значение переходит к 1, то атрибут говорит, что никакая атака не происходит с наблюдаемыми данными. Для каждого запроса на наблюдение авторы предлагают формулу для оценки аномалии:

$$\text{Оценка_аномалии} = \sum_{m \in \text{модель}} \omega_m * (1 - p_m) \quad (3)$$

В этом уравнении ω_m представляет собой вес, связанный с моделью m , а p_m – его возвращаемое значение вероятности. Вероятность p_m вычитается из 1, потому что значение, близкое к нулю, указывает на аномальное событие, которое должно давать высокую оценку аномалии.

Для построения дерева принятия решения в работе автор рассмотрит 3 новых атрибута [P6–P8].

а. Длина запроса, отправленного из браузера (P6)

Автор предполагает, что длина запроса, отправленного из браузера пользователя, является фиксированной. Изменение длины запроса пользователя в большинстве случаев невелико. Например, в случае SQL-инъекции, межсайтового выполнения сценариев. Поэтому автор предлагает использовать изменение длины запроса для выявления атак от пользователей.

• На фазе обучения: Пусть длина входных данных равна $l_1, l_2, \dots, l_n$. Математическое ожидание μ и дисперсия σ^2 набора данных.

• На фазе обнаружения: Заданы математическое ожидание μ и дисперсия σ^2 набора данных. Применение неравенства Чебышева даёт оценку вероятности того, что случайная величина примет значение, далёкое от своего среднего.

$$P(|x - \mu| > t) < \frac{\sigma^2}{t^2} \quad (4)$$

где, x : случайная величина, t : пороговое значение.

Соответственно, для любого распределения вероятностей со средним μ и дисперсией α^2 получается заданное значение x , то отклонение x от среднего μ превышает любой порог, заблокированный $\frac{\alpha^2}{l^2}$.

В этом случае автор выбирает $t = |1 - \mu|$. Чем больше значение, тем ближе значение l к среднему значению, тем больше вероятность того, что остальное может произойти.

$$P(|x - \mu| > |l - \mu|) < \frac{\alpha^2}{|l - \mu|^2} \quad (5)$$

b. Появление новых символов (P7)

- На фазе обучения: Автор сохранит существующий набор символов.

- На фазе обнаружения: Если появится новый символ, автор увидит его как атаку. Генерируем $p = 0$, если появится символ, не существующий в базе данных, то увеличивается $p = 1$.

c. Появление новых ключевых слов (P8)

Автор предлагает набор ключевых слов, которые используются при атаке в Интернете. В процессе обучения, без появления этих символов, которые появляются на этапе обнаружения атаки, также считается признаком атаки.

- На фазе обучения: Автор сохранит набор ключевых слов в интернете.

- На фазе обнаружения: Если появится новое ключевое слово, автор увидит его как атаку и p равно 1. В остальных случаях значения p равно 0.

4. Экспериментальная оценка точности разработанного алгоритма обнаружения атак на веб-ресурсы

В работе автор использовал набор данных CSIC 2010, специализирующийся на тестировании решений межсетевого экрана для веб-приложения, предложенного Институтом информационной безопасности Испанского национального исследовательского совета. CSIC 2010 включает в себя 36 000 безопасных запросов и 25 000 запросов с атаками (много типов: внедрение операторов SQL, выполнение команд ОС,...). Автор разделил набор данных на две части, таких как 80% запросов для обучения и 20% запросов для тестирования.

При рассмотрении атрибутов автор получит результат, представленный на Рис. 6.

Из исследования автор оценит, что метод дерева принятия решений является одним из наиболее популярных подходов к решению задач классификации из-за следующих преимуществ:

- простоты в понимании и интерпретации;
- возможности оценки модели при помощи статистических тестов;
- быстрой обучаемости.

Недостатки этого метода заключаются в следующем:

- алгоритмы не могут обеспечить оптимальность всего дерева в целом;

Рис. 6. Зависимость точности обнаружения от количества атрибутов

- высокая зависимость результата от качества обучающей выборки;
- неэффективность при решении задач классификации с большим числом классов.

Вывод

Веб-атаки должны решаться с помощью методов, которые составляют точность обнаружения на основе сигнатур с гибкостью системы обнаружения вторжений на основе аномалий.

В этой статье представлен новый подход к обнаружению аномалий, использующий в качестве входных HTTP-запросов, содержащих параметры.

Во-первых, анализируется обучающий набор HTTP-запросов, который не содержит никаких атак. После того, как вся необходимая информация была извлечена из журна-

лов, для описания модели поведения обычного пользователя применяются несколько схем обнаружения аномалий. Эта модель затем используется для обнаружения сетевых атак как отклонения от норм. Автор использует пять признаков обнаружения аномалий и добавляет три признака для повышения точности модели обнаружения сетевых атак.

Дальнейшая работа будет направлена на уменьшение количества ложных срабатываний путем уточнения разработанных алгоритмов и изучения дополнительных возможностей. Конечной целью является возможность обнаружения аномалий в реальном времени для веб-сайтов, которые обрабатывают миллионы запросов в день практически без ложных тревог.

Литература

1. Statista. Global number of web attacks blocked per day from 2015 to 2018 (in 1,000s). –2019. – [Online; accessed 2019]. <https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide/>.
2. Mukkamala Srinivas, Sung Andrew H. A comparative study of techniques for intrusion detection // null. – IEEE. 2003. – P. 570.
3. Medeiros Iberia, Neves Nuno F, Correia Miguel. Automatic detection and correction of web application vulnerabilities using data mining to predict false positives // Proceedings of the 23rd international conference on World wide web. – ACM. 2014. – P. 63-74.
4. Гуров Д.В., Гуров В.В., Иванов М.А., Шустова Л.И. Технология безопасного программирования и особенности ее преподавания в вузе // Дистанционное и виртуальное обучение. – 2010. – № 9. – С. 35–44.
5. Halfond William, Orso Alex, Manolios Pete. WASP: Protecting web applications using positive tainting and syntax-aware evaluation // IEEE Transactions on Software Engineering. – 2008. – Vol. 34, no. 1. – P. 65-81.
6. Antunes Joao, Neves Nuno, Correia Miguel, Verissimo Paulo, Neves Rui. Vulnerability discovery with attack injection // IEEE Transactions on Software Engineering. – 2010. – Vol. 36, no. 3. – P. 357-370.
7. Jovanovic Nenad, Kruegel Christopher, Kirda Engin. Precise alias analysis for static detection of web application vulnerabilities // Proceedings of the 2006 workshop on Programming languages and analysis for security. – ACM. 2006. – P. 27-36.
8. Akbar Memen, Ridha Muhammad Arif Fadhly, [et al.]. SQL Injection and Cross Site Scripting Prevention using OWASP ModSecurity Web Application Firewall // JOIV: International Journal on Informatics Visualization. – 2018. – Vol. 2, No. 4. – P. 286- 292.
9. Folini Christian, Ristic Ivan. Modsecurity handbook. – 2017.
10. Prandl Stefan, Lazarescu Mihai, Pham Duc Son. A study of web application firewall solutions // International Conference on Information Systems. – Springer. 2015. – P. 501-510.
11. Kruegel Christopher, Vigna Giovanni. Anomaly detection of web-based attacks // Proceedings of the 10th ACM conference on Computer and communications security. – ACM. 2003. – P. 251-261.

References

1. Statista. Global number of web attacks blocked per day from 2015 to 2018 (in 1,000s). –2019. – [Online; accessed 2019]. <https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide/>.
2. Mukkamala Srinivas, Sung Andrew H. A comparative study of techniques for intrusion detection // null. – IEEE. 2003. – P. 570.
3. Medeiros Iberia, Neves Nuno F, Correia Miguel. Automatic detection and correction of web

application vulnerabilities using data mining to predict false positives // Proceedings of the 23rd international conference on World wide web. – ACM. 2014. – P. 63-74.

4. Gurov D.V., Gurov V.V., Ivanov M.A., Shustova L.I. Tekhnologiya bezopasnogo programirovaniya i osobennosti yeye prepodavaniya v vuze // Distantionnoye i virtual'noye obucheniye. – 2010. – № 9. – S. 35–44.

5. Halfond William, Orso Alex, Manolios Pete. WASP: Protecting web applications using positive tainting and syntax-aware evaluation // IEEE Transactions on Software Engineering. – 2008. – Vol. 34, no. 1. – P. 65-81.

6. Antunes Joao, Neves Nuno, Correia Miguel, Verissimo Paulo, Neves Rui. Vulnerability discovery with attack injection // IEEE Transactions on Software Engineering. – 2010. – Vol. 36, no. 3. – P. 357-370.

7. Jovanovic Nenad, Kruegel Christopher, Kirda Engin. Precise alias analysis for static detection of web application vulnerabilities // Proceedings of the 2006 workshop on Programming languages and analysis for security. – ACM. 2006. – P. 27-36.

8. Akbar Memen, Ridha Muhammad Arif Fadhly, [et al.]. SQL Injection and Cross Site Scripting Prevention using OWASP ModSecurity Web Application Firewall // JOIV: International Journal on Informatics Visualization. – 2018. – Vol. 2, No. 4. – P. 286- 292.

9. Folini Christian, Ristic Ivan. Modsecurity handbook. – 2017.

10. Prandl Stefan, Lazarescu Mihai, Pham Duc Son. A study of web application firewall solutions // International Conference on Information Systems. – Springer. 2015. – P. 501-510.

11. Kruegel Christopher, Vigna Giovanni. Anomaly detection of web-based attacks // Proceedings of the 10th ACM conference on Computer and communications security. – ACM. 2003. – P. 251-261.

НГУЕН Мань Тханг, Сотрудник, Сотрудник, Академия криптографических техник Вьетнам. 134800, Вьетнам, г. Ханой, Тхань Чи, Тан Чиеу, ул. Тиен Тханг, д. 141, E-mail: chieumatxcova@gmail.com

NGUYEN Manh Thang, Contributor, Academy of Cryptography Techniques Viet Nam. 134800, Viet Nam, Ha noi City, Thanh tri, Tan trieu, Chien thang, house. 141. Email: chieumatxcova@gmail.com