АНАЛИЗ И МЕТОДЫ ЗАЩИТЫ ВЕБ-ПРИЛОЖЕНИЙ ОТ АТАК ТИПА LDAP-ИНЪЕКЦИЯ

Бодров В.А, Белоусова Е.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Белоусова Е.С. – к.т.н., доцент

В работе приводятся результаты анализа LDAP-инъекций и методов защиты веб-приложений от данного типа атак. Данный метод можно использовать для защиты любых веб-приложений.

В настоящее время огромную популярность получили веб-приложения, что обусловлено доступностью из любой точки мира, кроссплатформенностью и простотой в обновлении. Разработчики данных приложений в первую очередь уделяют внимание функциональности приложения, а не обеспечению безопасности своих разработок.

Lightweight Directory Access Protocol (LDAP) – это сетевой протокол прикладного уровня модели TCP/IP, используемый для доступа к службам каталогов. Наиболее широко используемыми реализациями служб LDAP являются Microsoft ADAM и OpenLDAP [1]. Служба каталогов LDAP представляет собой древовидную структуру, которая хранит и систематизирует информацию по общим атрибутам. Операции над каталогом, в частности, запись, чтение, сравнение осуществляется с помощью фильтров, которые определены в RFC 4515. При работе с веб-приложениями, использующих LDAP используются три основных вида запросов:

1) запрос без логических операторов;
2) запрос с использованием логического оператора «И»;
3) запрос с использование логического оператора «ИЛИ».

Ниже приведен пример простейшего LDAP запроса с использованием логического оператора «ИЛИ»:

$$(|\,(attribute1 = parameter1)(attribute2 = parameter2)).$$

Недостаточное внимание к фильтрации вводимых данных пользователя веб-приложения ведет к возможности использования LDAP-инъекции. Данные атаки основаны на тех же методах, что и атаки с использованием SQL. Следовательно, основная концепция атак заключается в использовании параметров, веденных пользователем, для модификации исходного LDAP запроса [2].

Предположим, что приложение создает запрос, с использованием логического оператора «И», для поиска в каталоге LDAP имени пользователя и соответствующего ему пароля с целью проведения процесса аутентификации. В качестве вводимых данных выступает имя пользователя (uname) и пароль пользователя (pwd). В данном случае запрос будет выглядеть таким образом:

$$(\&(username = uname)(password = pwd)).$$

Если злоумышленник вводит правильное имя пользователя, например, «alexander», а затем вводит определенную последовательность, то проверку пароля можно обойти. Представленный ниже запрос основан на изменение значения «uname» на значение «alexander)(&))» и добавлении любой строчки в качестве значения «pwd»:

$$(\&(username = alexander)(\&))(password = pwd))$$

Приложение обрабатывает только первую часть запроса, то есть

$$(\&(username = alexander)(\&)),$$

которое всегда является истинным выражением. Таким образом, злоумышленник получит несанкционированный доступ к системе без ввода пароля.

Предположим, существует некая поисковая система, позволяющая получить информацию о пользователях (имя пользователя, время регистрации в системе, время последнего входа и т. п.). Данный поисковый запрос к каталогу LDAP может быть выполнен с использованием логического оператора «ИЛИ»:

$$(|\,(param = username)(param = registration\_time)).$$

Используя LDAP инъекцию, злоумышленник может манипулировать данными, возвращаемыми от приложения. Если злоумышленник в качестве значения поля «param» отправит выражение «alexander)(uid=*)», то запрос будет иметь следующий вид:

$$(|\,(param = alexander)(uid = *))(param = registration\_time)).$$

Специальный символ «*» можно использовать для замены одного или нескольких символов в конструкции фильтра. В результате запроса приложение вернет злоумышленнику всех пользователей системы.

Как и в случае с SQL инъекцией, LDAP инъекция может быть «слепой». Под «слепой» понимается инъекция, при которой приложение не будет отображать все поля записей или сообщения об ошибках. В этом случае злоумышленник может использовать данное поведение для проведения успешных LDAP инъекций.

Предположим, что у нас есть приложение, которое в результате ввода имени пользователя работника организации «uname», отображает адрес его электронной почты. Наш запрос к приложению примет следующий вид:

$$(\&(username = uname)(objectClass = employee))$$

В случае отправки имени пользователя, которого не существует, мы получим соответствующее сообщение. Данное поведение может быть использовано для получения различных данных о пользователе, например, его пароля. Может использоваться метод перебора символов с использованием специального символа «*». При использовании инъекции в поле «username» равной «alexander)(userPassword=a*». Конечный запрос с инъекцией будет выглядеть следующим образом:

$$(\&(username = alexander)(userPassword = a*)(objectClass = employee))$$

В случае если пароль пользователя «alexander» начинается с буквы «a», мы получим ожидаемый ответ на правильный запрос, а именно, адрес электронной почты пользователя. Таким образом мы можем перебирать все символы до тех пор, пока не получим полноценный пароль пользователя «alexander».

Поскольку данный тип атаки выполняется на уровне приложений, межсетевые экраны и механизмы обнаружения вторжений, работающие на сетевом уровне, не могут предотвратить атаку. Однако общие рекомендации безопасности для служб каталогов LDAP могут уменьшить вероятность возникновение данной уязвимости: отключение индексирования полей, использование принципа минимальных привилегий и. т. д.

Разработчики не осведомлены о подобных инъекциях, поскольку на сегодняшний день существует довольно малое число информационных статей, посвященных данному типу атак. Быстрый поиск «LDAP» на ресурсах с открытым исходным кодом приложений, позволяет обнаружить большое количество приложений, уязвимых к атакам типа LDAP-инъекция.

Сегодня, инструменты статического анализа кода не готовы обнаруживать в коде данные уязвимости. Таким образом, разработчик, не разбирающийся в методах обеспечения безопасности, легко создаст уязвимый код.

Однако для предотвращения LDAP инъекций достаточно использовать фильтрацию вводимых данных пользователя. Для правильной фильтрации данных, поступающих в веб-приложение, разработчики должны обращать внимание только на десять специальных символов «|, &, (,), *, <,>, =, ~,!». Если разработчик отфильтровывает эти символы, атаки типа LDAP-инъекции работать не будут [3].

Таким образом, в работе рассмотрены основные принципы атаки на веб-приложения типа LDAP-инъекция и основные методы защиты от нее. В дальнейшей работе планируется разработать программное обеспечение, позволяющее обнаруживать и эксплуатировать данный тип уязвимости.

Список использованных источников:

1.Об Интернете, информационных технологиях и не только. [Электронный ресурс]. – Режим доступа: https://www.styler.ru/styler/ldap-injection/.

2.E. Guillardoy, F. Guzman, H. Abbamonte. LDAP Injection. Attack and Defence Techniques, журнал «HITB Magazine», 2010, №1, с. 9 – 17.

3.Vulners – Vulnerability Data Base. [Электронный ресурс]. – Режим доступа: https://vulners.com/static/appercut/ru/Java/InjectionLdap.html.