Д. М. Михайлов, М. А. Иванов

МЕХАНИЗМЫ ПРОВЕДЕНИЯ АТАК НА СОВРЕМЕННЫЕ СИСТЕМЫ

1. Обоснование реальности угрозы атак с использованием RFID-вирусов

Рассмотрим причины, по которым атаки на RFID-системы могут быть привлекательными для злоумышленников.

- Незаконное получение информации. Информация, которая помещается в RFID-метки, может являться частной и не подлежащей раскрытию. Однако информация с RFID-метки может быть считана в обход разрешения на то владельца объекта, снабженного меткой, или носителя метки. Этот недостаток технологии является одним из основных барьеров на пути широкого внедрения так называемых биометрических паспортов [1].

- Незаконное отслеживание местоположения. Во многих случаях RFID-метки применяются с целью определения местоположения объекта в пространстве. Это необходимо, в частности, когда речь идет о перемещении крайне важных лекарств или при отслеживании местоположения домашнего животного. Но существует опасность несанкционированного отслеживания местоположения человека или важного носителя информации. Такое отслеживание противоречит, например, праву свободы передвижения, закрепленному во многих конституциях мира.

- Подделка идентифицирующего тега. Еще одной опасной особенностью технологии RFID является тот факт, что RFID-метки могут быть подделаны. Это означает, что в областях применения RFID-технологии могут участиться случаи мошенничества. Фактически злоумышленники могут копировать существующие RFID-метки, создавая их полные клоны. Это означает, что реальные объекты в системе RFID могут быть заменены на дубликаты. Последствия подделки биометрического паспорта очевидны.

- Перехват сигнала. Еще одной опасностью использования технологии RFID является возможность перехвата сигнала от RFID-метки к приемнику RFID и последующий повтор данного сигнала. Такие перехваты могут, как следствие, привести к множественным случаям мошенничества. Таким образом могут быть произведены, например, хищения денежных средств за счет осуществления атаки на бесконтактные терминалы оплаты.

- Отказ в обслуживании. RFID-технологии также подвержены атаке, распространенной в настоящее время в компьютерном мире, — атаке отказа в доступе. Осуществить такую атаку можно не только стандартными — перегрузкой сети, но и специфичными для данной технологии способами, как, например, перехватом сигналов, наложением помех и т. д. Последствия такой атаки могут быть серьезными. К примеру, отказ в работе устройства мобильного военного медицинского центра, считывающего информацию из метки раненого, может привести к тому, что человеку не будет оказана первая медицинская помощь на поле боя.

2. Модель функционирования RFID-вируса

Рассмотрим модель типичного RFID-вируса, который может угрожать стандартной RFID-системе учета.

Пусть стандартная система RFID выглядит следующим образом. Существует некоторый поставщик сырья (уран, плутоний) и завод, где это сырье перерабатывается. Для транспортировки сырья от поставщика на завод используются многоразовые контейнеры, каждый из которых оборудован RFID-меткой, доступной для записи и для чтения. После того как на стороне поставщика сырье помещается в контейнер, в RFID-метку записывается информация о нем. При получении товара информация из RFID-метки считывается и поступает в базу данных ПО перерабатывающего завода [2].

Подобная RFID-система достаточно проста с точки зрения архитектуры. Система завода состоит из базы данных, нескольких считывающих/записывающих устройств, а также самого программного комплекса, осуществляющего обработку информации.

Стандартная таблица базы данных подобной системы выглядит следующим образом:

Тег	ContainerContents
123	Уран
234	Плутоний

Согласно данной таблице, в контейнере с номером 123 хранится уран, а в контейнере с номером 234 — плутоний.

Рассмотрим модель вируса, который осуществляет атаку на данную систему. Вирус распространяется на RFID-метку и использует механизм SQL-инъекции. Простейший вид SQL-инъекции заключается в добавлении ко всем данным базы данных поражаемого программного комплекса копии собственного кода SQL-инъекции. Таким образом, при попадании измененных вирусом данных на другие RFID-метки они также будут заражены. Такая перезапись в рассматриваемой системе происходит при обновлении информации о содержании контейнера. Если контейнерная RFID-метка будет поражена в системе поставщика, то этот же вирус может поразить и ПО системы завода.

Пример простейшего вида SQL-инъекции выглядит следующим образом:

```
uran;UPDATE NewContainerContents
SET ContainerContents = ContainerContents ||
``;[SQL Injection]'';
```

В директиве указывается, что во все строки таблицы NewContainerContents в поле ContainerContents дописывается текст самой SQL-инъекции, чтобы вирус мог распространяться.

У такого подхода имеется один недостаток. Дело в том, что данный вариант SQL-инъекции является рекурсивным, а это критичным.

Рекурсивность вытекает из самого вида SQL-инъекции:

```
[SQL Injection] = UPDATE NewContainerContents
SET ContainerContents = ContainerContents ||
``;[SQL Injection]'';
```

Для того чтобы обойти данный недостаток, создатели вируса могут использовать следующий подход. Большинство баз данных поддерживают механизм, согласно которому SQL-запрос может получить список выполняемых в данный момент SQL-запросов. Например, в Oracle такой запрос выглядит следующим образом:

```
SELECT SQL_TEXT FROM $sql WHERE INSTR(
SQL_TEXT,'``')>0;
```

Аналогичные команды существуют и в PostgreSQL, и в MySQL. Таким образом, конечная версия вируса будет выглядеть следующим образом:

```
Contents=Raspberries;
UPDATE NewContainerContents SET ContainerContents=
ContainerContents || ';' || CHR(10) || (SELECT
SQL_TEXT FROM v$sql WHERE INSTR(SQL_TEXT,'``')>0);
```

Тем не менее существует проблема и при использовании такого подхода.

Так, во многих базах данных в зависимости от нагрузки процессора и быстродействия архитектуры может иметь место существенная задержка в выполнении SQL-запросов.

В PostgreSQL такая задержка приведет к тому, что в списке текущих исполняемых запросов будет значиться '<IDLE>'. Исключение составляет лишь Oracle, где задержка на работе системы не сказывается.

Иначе говоря, запрос

SELECT SUBSTR(SQL TEXT,43,127)FROM $sql WHERE INSTR(SQL TEXT, ..payload...)>0)

отработает без особых проблем.

Можно предложить альтернативные варианты обращения к исполняемым в настоящий момент SQL-запросам. Выбирая каждый раз новый вариант, можно разнообразить вирусный код, а значит, сделать его обнаружение более сложным.

Новый подход основывается на существовании программ, которые могут вывести на экран свой собственный код.

Такие программы, как правило, имеют следующую структуру. Они состоят из двух частей: области кода и области данных. Область данных — это область, где представлен текст программного кода, выводимого на экран. Область кода обращается к области данных для вывода текста программы, а потом использует область данных уже для вывода самой области данных.

Обратимся к конкретному примеру. Для базы данных PostgreSQL SQL-запрос подобного рода можно реализовать следующим образом:

SELECT substr(source,1,93) || chr(39) || source ||
chr(39) || substr(source,94) FROM (SELECT 'SELECT
substr(source,1,93) || chr(39) || source || chr(39)
|| substr(source,94) FROM (SELECT ::text as source)
q;'::text as source) q;

Этот SQL-запрос выводит свой собственный текст и больше не делает ничего.

Конечно, реализация такого подхода требует гораздо большего объема памяти, и поэтому создаваемый вирус уже может не поместиться на простой RFID-метке. Тем не менее он вполне пригоден для того, чтобы быть использованным для создания вируса, который будет распространяться на прокси-картах, где объем памяти достаточно большой [2].

Еще один вариант вируса, который будет лишен рекурсивности и сможет заражать базы данных, распространяясь лишь на RFID-метках, может быть реализован следующим образом. Как уже говорилось выше, при считывании данных с RFID-метки системное ПО использует следующий запрос:

UPDATE ContainerContents SET OldContents='%contents%' WHERE TagID='%id%'

для того, чтобы обновить содержание базы данных. Если содержание тега не обрабатывается безопасным образом, то вставка символа верхней кавычки позволит злоумышленнику существенно изменить запрос.

Рассмотрим пример того, как это может быть реализовано злоумышленником.

%content%' WHERE TagId='%id%';
SET @a='UPDATE ContainerContents SET NewContents=concat(\'%content%
\\\' WHERE TagId=\\\'%id%\\\'; SET @a=\', QUOTE(@a), \'; \', @a); %pay-
load%; --';
UPDATE ContainerContents SET NewContents=concat('%content%\' WHERE
TagId=\'%id%\'; SET @a=', QUOTE(@a), '; ', @a); %payload%; --

Разберем подробно код данного вируса.

В идентификаторе **%content%**, очевидно, записывается безобидное содержимое RFID-метки, но сразу после него следует закрывающая верхняя кавычка, которая как раз и

позволяет после нее поместить основное тело вируса. Во второй строке приведенного выражения производится присвоение строки, содержащей тело вируса, переменной «**a**». Само тело вируса находится в третьей строке. Затем это же тело вируса присваивается переменной «**a**», но с соответствующими блокировками. В третьей строке, т. е. в самом теле вируса, приведен код, который обновляет поле NewContents для каждой записи. Начальный фрагмент этого кода — первая строка вируса, далее следует начальная часть второй строки вируса вплоть до того места, где приводится экранированное содержимое переменной «**a**». Это содержимое подставляется с помощью оператора MySql **QUOTE**, который как раз и отвечает за экранирование. Далее уже подставляется неэкранированное содержимое из переменной **«a»**. Таким образом, подобная техника позволяет вирусу распространяться, копируя свое тело.

RFID-вирус, заразивший систему, как уже было сказано, может распространяться через заражение других RFID-меток, обрабатываемых зараженной системой. Но такие вирусы также представляют опасность из-за того, что могут содержать условие, согласно которому какое-либо разрушительное действие будет осуществляться только лишь через заданное время (логическая бомба). Т. е. у вируса будет время на распространение, и только после того, как достаточное количество систем будет поражено, начнется непосредственная атака. Например, все базы данных, пораженные вирусом, будут одновременно уничтожены. Такой вариант осуществим ввиду того, что в SQL-синтаксисе предусмотрена возможность выполнения команд при наступлении определенных условий.

Одним недостатком приведенного вируса является его привязка к конкретной структуре базы данных. Т. е. в самой SQL-инъекции используется наименование таблиц базы данных. Это означает, что злоумышленник должен заранее хорошо знать атакуемую систему. Однако это не всегда означает универсальность вируса [3].

Для устранения этого недостатка, необходимо усовершенствовать текст SQL-инъекции таким образом, чтобы в ней отсутствовало указание на конкретную таблицу базы данных. Язык SQL позволяет это сделать. Тем не менее и у такого подхода есть свои минусы. Дело в том, что если запрос SQL-инъекции будет недостаточно интеллектуальным с точки зрения выбора поражаемых таблиц, то могут быть изменены данные тех таблиц, которые необходимы для распространения вируса. Поясним на примере. В рассмотренном случае с поставщиком и заводом к таким данным относятся номера меток. Вирус, исказив номера меток, сам же преградит себе путь к дальнейшему распространению, так как RFID-система уже не сможет определить, в метку какого контейнера какие данные помещать.

Также необходимо помнить, что SQL-язык может быть разным в зависимости от той технологии, на основе которой строится база данных. Например, запросы, написанные на языке SQL для MySQL, не могут быть исполнены в Oracle. Так, в базе данных PostgreSQL используется команда concat(), а в Oracle — команда ||. Еще один пример отличий — директива chr на PostgreSQL и char на Oracle.

В заключение необходимо отметить следующее. Вероятность появления в ближайшее время RFID-вирусов весьма велика. Причин тому несколько. Это высокая уязвимость RFID-систем, ценность информации, которую может получить злоумышленник в результате успешной атаки, а также критичность большинства систем с точки зрения возможных убытков в случае прекращения функционирования, что может быть привлекательным для вымогателей [4].

Приведенная в статье модель вируса наглядно демонстрирует реальность угрозы, и именно поэтому необходимо приложить максимум усилий для разработки системы RFID, которая была бы устойчивой к атакам подобных вирусов.

СПИСОК ЛИТЕРАТУРЫ:

1. *Иванов М. А., Ковалев А. В., Мацук Н. А., Михайлов Д. М., Чугунков И. В.* Стохастические методы и средства защиты информации в компьютерных системах и сетях. Под ред. И. Ю. Жукова. М.: КУДИЦ-ПРЕСС, 2009.
2. *Лахири С.* RFID, руководство по внедрению. М.: КУДИЦ-ПРЕСС, 2007.
3. *Моисеенков И.* Основы безопасности компьютерных систем // КомпьютерПресс. 1991. № 10. С. 19–24; № 11. С. 7–21.
4. *Белкин П. Ю., Михальский О. О., Першаков А. С. и др.* Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных: Учебное пособие для вузов. М.: Радио и связь, 2000.