электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 003.26.7 004.9

Подходы по защите от атак на сетевую карту

Воскресенский Г.А., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»

Чаплыгин А.М., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»

Научный руководитель: Алешин В.А., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»
bauman@bmstu.ru

В работе [1], продемонстрировано, как может злоумышленник получить полный контроль над компьютером, используя уязвимость в сетевом адаптере[1]. Злоумышленник может получить полный контроль над адаптером и добавить бэкдор в ядро ОС, используя DMA доступ.

В то время, как предотвращение от несанкционированного доступа к сетевой карте через операционную систему возможно используя имеющиеся механизмы, использование скомпрометированной сетевой карты остается серьезной проблемой, не только потому, что сетевая карта является важнейшим компонентом с точки зрения безопасности, но и потому, что зараженное устройство может быть использовано для заражения окружающих внешних устройств на компьютере.

Возможные контрмеры были рассмотрены в [3], но ни один из них не показался действительно убедительным. Лучший способ предотвратить сетевую карту от заражения, вероятно, состоит в формальной проверке, что код, работающий в прошивке корректен. Учитывая, что код прошивки сетевых адаптеров все чаще является сложным и, как правило, проприетарным, проблема предотвращения сводится к проблеме обнаружения. В работе [1] предлагается подход к обнаружению повреждений сетевой карты, где сканер находится внутри операционной системы. Такой вид атак и их обнаружение мало изучено, не в полной мере исследуется специалистами по обнаружению вторжений. Тем не менее, эти нападения представляют реальную угрозу, учитывая уровень привилегий, который

[1] См. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0104.

злоумышленник может получить путем успешного использования основных уязвимостей. Более того, изучение подхода обнаружения (в отличие от подхода предотвращения) является наиболее актуальным, так как уязвимости находятся в компоненте, который в свою очередь не находится под полным контролем пользователя.

Рассмотрение включает две части. Во-первых, рассматривается возможные угрозы, связанные с атакой на сетевое устройство. Во-вторых, рассматриваются 2 подхода защиты, такие как контроль целостности порядка исполнения в коде и удаленную аттестацию прошивки. Цель состоит в том, чтобы блокировать атаки, содержащие модификации потока управления встроенного устройства, сохраняя при этом хорошую производительность.

В последние несколько лет некоторые исследователи изучали безопасность прошивки и встроенного программного обеспечения в различных устройствах, таких как модемы, сетевые карты, контроллеры клавиатуры или наборы микросхем.

Эти атаки могут позволить злоумышленнику получить полный контроль над компонентом и использовать его как базу для запуска других атак на ОС (DMA атаки) или на другие внешние устройства. Даже без получения контроля над компонентом, сама атака может быть использования для подслушивания данных (кейлоггер на контроллере клавиатуры) или тайно выполнять атаку man in the middle (на сетевой карте).

Защищать систему против подобных атак сложно, так как прошивка карты работает без контроля ОС, а также имеет потенциальный доступ к другим системным ресурсам (например к PCI шине), реальную деятельность которых ОС не контролирует.

Внесение исправлений (в дальнейшем патчинг) - самая очевидная контрмера. Однако, пропатчить можно только известные уязвимости, а исправить непосредственно прошивку еще сложнее, чем компоненты ОС. Более того, адаптеры обычно запускают свою резидентную прошивку из своей энергонезависимой памяти перед тем как загрузить новую прошивку. Эту резидентную прошивку исправить нельзя, по этой причине существует промежуток времени перед тем как безопасно загрузится новая исправленная прошивка.

Как показал Sang с соавт. [4] блок управления памятью для ввода-вывода (IOMMU) может помочь защищать систему, но не на 100% эффективно, так как не может защитить остальные периферийные устройства,. Кроме того IOMMU не защищает затронутые подсистемы, чьи повреждения являются критическими, особенно если это сетевая карта (как уже упоминалось ранее, это может привести к пассивному прослушиванию).

Существуют несколько методов нейтрализации уязвимостей. Например, защита от исполнения произвольного кода, к этому типу атаки относится ASLR - изменение

случайным образом расположения в адресном пространстве процесса важных структур, т.н. «осведомители» (canaries, данные, которые используются для мониторинга возможного переполнения буфера), система NX bit (аппаратная защита от переполнения буфера), а также data-tainting (флагирование данных, которые запрещено возвращать по запросу стандартными способами). Тем не менее, атакующие могут обойти некоторые способы защиты.

NX bit обходится при помощи возвратно-ориентированного программирования (ROP), а canaries не работают против ROP без возвратов. Но что более важно, эти защитные методы не применяются в случае прошивок устройств, потому что у этих систем обычно отсутствуют определенные базовые функции, так как работают на аппаратных устройствах с ограниченными встроенными процессорами, например MIPS.

В [1] предложен подход, который состоит в проверке целостности прошивки сетевой карты в процессе выполнения с целью обнаружения вредоносных изменений в порядке исполнения, называемых CFI (контроль целостности порядка исполнения) и удаленная проверка прошивки.

Контроль целостности порядка исполнения

Классическая политика CFI требует, чтобы программное обеспечение следовало графу порядка исполнения (CFG), который заранее определен. CFG может быть определен путем анализа исходного кода, бинарника или исполняемого файла.

Подход к обнаружению преследует те же цели что и CFI, но по отношению к прошивке, как предложил Francillion и соавт. [2]. В этом случае контролируется доступ к участкам памяти при помощи программного модуля контроля доступа к оперативной памяти (SMAC), а также используем теневой стек вызовов для обнаружения. Система слежения использует профиль исполнения сетевой карты, который можно рассматривать как вариант очень грубой и примитивной формы политики контроля доступа. Профиль подстраивается заранее путем инспекций нескольких выполнений прошивки. Его использует система слежения чтобы обнаруживать аномальные процессы.

Подобно CFI, программные средства защиты переписывают и вставляют свои части кода в программу. Эти элементы могут выполнять определенные задачи в процессе выполнения, чтобы защитить программу от модификаций (проверка контрольных сумм и т.п.). Они изначально использовались чтобы реализовать защиту от взлома программ (cracking), но их можно также использовать для защиты в прошивках.

Удаленная проверка прошивки

Проверка целостности выполняемого потока может быть достигнута при помощи программных дистанционных аттестаций [2]. Верификация выполняется доверенным лицом в процессе выполнения на целевой системе. В нашем случае целью является сетевой адаптер, а доверенным лицом будет ОС.

Удаленная аттестация устройства основана на классическом протоколе запроса-ответа, где проверяющий отсылает несколько случайных чисел цели. Затем цель считает контрольную сумму всей своей памяти используя переданное число как начальное значение (seed) и возвращает контрольную сумму проверяющему. Затем проверяющий проверяет правильность результата.

Целевая информация и неиспользуемая память стираются с предсказуемым значением. Потом память псевдослучайно читается, таким образом предотвращая возможность заранее просчитать контрольную сумму. Все вмешательства отключены во время подсчета контрольной суммы. Устройство сбрасывается после возврата контрольной суммы.

У проверяющего есть копия ожидаемого содержимого оперативной памяти цели и он сравнивает возвращенную контрольную сумму с собственными подсчетами. Проверяющий также проверяет чтобы подсчет занимал строго определенное время.

Как обсуждалось в [5], удаленная аттестация прошивки крайне сложна. Во-первых, вредоносная программа может держать сжатую копию оригинального кода прошивки в памяти и перенаправлять попытки чтения памяти чтобы получить корректную контрольную сумму. Из-за этого время подсчета контрольной суммы должно быть предсказуемым и почти оптимальным, чтобы засечь подобные попытки. Также, проверяющий должен знать точную конфигурацию устройства цели. Во-вторых, память должна быть сброшена в предсказуемое состояние перед аттестацией псевдослучайными значениями, иначе память может быть непредсказуема и вероятно содержать вредоносный код.

В [4], дистанционная аттестация была проведена на прошивке клавиатуры Apple, которая является довольно простым устройством. Однако аттестация занимает вплоть до 2х секунд, в течение которых устройство не отвечает. Это все приводит к вопросу, подходит ли такая аттестация для сложных устройств, например для сетевых адаптеров? Действительно, функция подсчета контрольной суммы накладывает жесткие условия: требуется сбросить память устройства и блокировать все прерывания, что может занимать время устройства. Кроме того, не вполне ясно, что дистанционная аттестация в настоящее время подходит для устройств, работающих в реальном времени

Список литературы

1. Duflot L., Perez Y.-A., Morin B. What If You Can’t Trust Your Network Card? Available at: http://www.ssi.gouv.fr/IMG/pdf/paper.pdf , accessed 25.12.2014.
2. Castelluccia C., Francillon A., Perito D., Soriente C. On the difficulty of Software-based attestation of embedded devices. Available at: http://www.inrialpes.fr/planete/people/francill/Papers/CCS09_Software_based_attestation.pdf , accessed 25.12.2014.
3. Li, Y., McCune, J.M., Perrig, A. Software-Based Attestation for Peripherals. Available at http://users.ece.cmu.edu/~jmmccune/papers/li_mccune_perrig_SBAP_trust10.pdf, accessed 25.12.2014.
4. Sang, F.L., Lacombe, E., Nicomette, V., Deswarte, Y.: Exploiting an I/OMMU vulnerability. Available at: https://www.deepdyve.com/lp/institute-of-electrical-and-electronics-engineers/exploiting-an-i-ommu-vulnerability-0ECFj6D5Ub, accessed 25.12.2014.
5. Francillon A. Attacking and Protecting Constrained Embedded Systems from Control Flow Attacks. Available at: https://tel.archives-ouvertes.fr/tel-00540371/document, accessed 25.12.2014.