Интернет (суммарное количество серверов на базе Apache и Nginx по данным декабря 2018) [1]. Одна из таких уязвимостей использует Pluggable Authentication Modules (PAM, подключаемые модули аутентификации), поставляемый практически во всех версиях операционных систем на базе Unix, и имеет название Too Many Open Files exploit (TMOF, «*слишком много открытых файлов*»).

Уязвимость заключается в следующем. Каждое подключение создает в PAM 2 файловых дескриптора: один – для передачи данных, второй – для чтения передаваемого файла. Максимальное количество существующих дескрипторов по умолчанию является 1024, как правило, при повышении нагрузки вручную увеличивается до 8000-16000, что достаточно для обслуживания крупного потока пользователей. При достижении этого лимита сервер перестает обслуживать новые подключения до освобождения ресурсов, добавляя в системный журнал сервера ошибку “Too Many Open Files (24)”.

Алгоритм атаки следующий:

1. создать сокет,
2. подключиться к серверу,
3. закрыть сокет на прием данных,
4. отправить пакет серверу,
5. уничтожить сокет.

Таким образом, на сервере создается как минимум 1 файловый дескриптор, который не уничтожится до истечения таймаута подключения. Средний современный компьютер способен сгенерировать и отправить 23000-45000 пакетов в секунду, в результате перегрузив целевой ресурс за несколько секунд. Как правило, сервер настроен таким образом, что при определенных критических ошибках или критическом количестве ошибок он останавливает свою работу и требует перезагрузки и обслуживания администратором, что и происходит в данном случае. Например, для внешнего наблюдателя сервер на любой HTTP запрос будет отправлять код ответа из 500 серии, как правило, код 503 или, в редких случаях, 500. На момент написания данной статьи ошибка Too Many Open Files является достаточно известной, однако использование ее в качестве уязвимости не имеет упоминаний на тематический форумах и новостных лентах, что делает ее достаточно опасной. Нейтрализацией данной уязвимости будет настройка роутера таким образом, чтобы более 2-3 подключений с одного IP-адреса блокировались, что приемлемо далеко не для всех.

Список использованных источников:

1. December 2018 Web Server Survey [Электронный ресурс]. – Режим доступа: https://news.netcraft.com/archives/2018/12/17/

ЗАЩИТА ПЕРЕДАВАЕМОЙ ИНФОРМАЦИИ В АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ

Турок М.А.

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

Сечко Г.В. – к.т.н., доцент

Решается задача защиты информации в микроконтроллерах автосигнализации мобильных объектов, причём под защитой понимается противодействие нарушению целостности (несанкционированному изменению, искажению, уничтожению информации) и противодействие считыванию.

Современная автосигнализация мобильных объектов, и в первую очередь транспортных средств, реализуется на микроконтроллерах. В каждом микроконтроллере заложена своя программа, в соответствии с которой контроллер управляет каким-либо устройством, выдавая управляющие сигналы. На разработку данного программного обеспечения производители тратят большие средства и время. В этих условиях актуальной является задача защиты информации в микроконтроллерах автосигнализации мобильных объектов, причем под защитой понимается противодействие нарушению целостности (несанкционированному изменению, искажению, уничтожению информации) [1] и противодействие считыванию. Поскольку устройства автосигнализации используют радиоканал передачи данных для взаимодействия между центральным блоком и пультом дистанционного управления, то параллельно с защитой необходимо повысить помехоустойчивости канала связи между данными устройствами.

На современном рынке существует три вида автосигнализаций: статические, динамические и диалоговые. Так как основную долю рынка занимают динамические автосигнализации, то было принято решение защищать информацию именно в них. Для понимания сути работы алгоритма введем следующие понятия: протокол передачи данных, посылка, сообщение. Сигнал в радиоэфир

передается по определенному правилу. В общем случае протокол передачи данных состоит из преамбулы (специфический набор данных дающий понять принимаемому устройству, что принимаемая информация есть искомая), и информационной части (информация, которая представляет команду для устройства управления). Посылка представляет собой команду, передаваемую устройству управления автосигнализацией (рисунок 1).

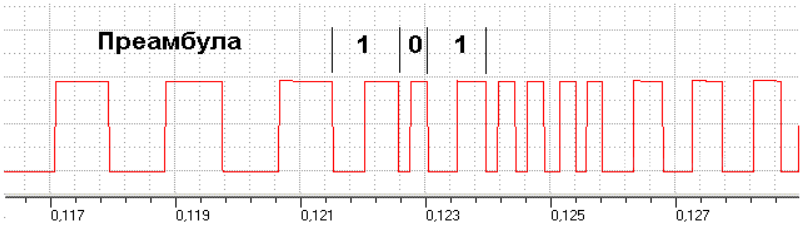

Рисунок 1 – Пример посылки сигнализации StarLine

Сообщение – это совокупность посылок (как минимум одна), которые идентичны друг другу. Суть алгоритма перехвата такова, что после подачи в радиоэфир сигнала, предназначенного для управления автосигнализацией, устройство перехвата, анализируя передаваемый сигнал, выдает в эфир радиопомеху, которая ставится в определенном месте информационной части посылки, и искажает данную команду [2]. Само устройство перехвата в данный момент времени запоминает неискаженную часть посылки. Данное действие не позволяет устройству управления автосигнализацией воспринимать передаваемую команду адекватно. Так как команда (посылка) в радиоэфир передается неоднократно, то это позволяет устройству перехвата одновременно принимать передающуюся посылку, выставлять помеху и запоминать искаженную команду, однако помеха ставится в месте, отличном от предыдущего, что дает возможность устройству перехвата восстановить истинную команду путем сложения неискаженных частей.

Предложенный алгоритм учитывает очистку передаваемых автосигнализацией сигналов от случайных помех в радиоэфире, возникающих вблизи крупных промышленных объектов, которых в густонаселенных городах достаточно много.

Список использованных источников:

1. Закон Республики Беларусь от 10 ноября 2008 г. No 455-З «Об информации, информатизации и защите информации» / Нац. реестр правовых актов Респ. Беларусь. – No 2/1552 (зарегистрировано 17 ноября 2008 г.).

2. Карпушкин, Э. М. Радиосистемы передачи информации / Уч. метод. пособие для студентов учреждений, обеспечивающих получение высшего образования по спец. "Радиоэлектронные системы". – Минск: БГУИР, 2008. – 62 с.

СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССАМИ ПРОЕКТИРОВАНИЯ ПРОГРАММ

Ульянко В.Г.

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

Сечко Г.В. – к.т.н.,доцент

В настоящее время системы управления процессами проектирования программ актуальны, востребованы и должны решать широкий диапазон задач. Уже существует ряд систем такого рода, однако у всех есть свои недостатки: от недостаточной глубины функциональности до чрезмерно высокой цены. Для того, чтобы превзойти аналоги, необходимо реализовать функциональность автоматического анализа проекта, с возможным автоматическим решением проблем.

В первую очередь стоит определить, что из себя представляет управление проектами. Основой успеха проектного управления есть наличие конкретного, заранее подготовленного набора действий для уменьшения рисков и ответвлений от начального плана [1].

Исходя из этого, можно выделить определение для систем управления процессами проектирования программ – это набор программных продуктов, состоящий из инструментов планирования задач, подготовки расписания, управления ценой и денежными средствами, распределения ресурсов, организации работы, а также из инструментов управления администрированием системы и документированием разрабатываемого продукта [2].

Для того, чтобы лучше понять значение и функции систем данного типа, стоит обратить внимание на решаемые задачи, которые можно разделить на 3 типа.