Большинство алгоритмов АМАДВТ состоят из следующих этапов:

1) сбор данных — собираются данные о времени нажатия и отпускания клавиш во время ввода пользователем пароля;

2) создание модели пользователя — преобразование данных, полученных на первом этапе, в модель с параметрами HoldTime (время удержания клавиши), DownDownTime (время между последовательными нажатиями двух клавиш) и UpDownTime (время между отпусканием клавиши N и нажатием клавиши N + 1). В течение некоторого времени эти данные накапливаются, после чего их можно использовать;

3) расчёт схожести полученной выборки и идеальной — расчёт расстояния между моделью пользователя и полученной на этапе ввода выборкой. Чем меньшее значение получено, тем более похожими являются выборки;

4) анализ полученных результатов — расчёт вероятности того, что данные были введены истинным пользователем.

Для улучшения распознавания пользователя можно использовать несколько моделей пользователя, которые отличаются временем сбора данных (темп набора пользователя может изменяться в зависимости от времени суток).

У АМАДВТ есть серьёзный недостаток: его точность напрямую зависит от удобства клавиатуры (эргономика, ход клавиш, размеры), то есть при смене пользователем устройства ввода необходимо обучить систему аутентификации заново. Также стоит отметить тот факт, что модель пользователя может быть применена только для того устройства, на котором она создавалась, поэтому АМАДВТ желательно использовать в качестве дополнительного уровня идентификации.

Таким образом, аутентификация методом анализа динамики ввода текста является достаточно хорошим способом аутентификации, хоть и не лишённым недостатков.

Список использованных источников:

1.А.Г. Дьяконов, А.М. Головина — Выявление аномалий в работе механизмов методами машинного обучения — 2017. — С.389-396.

2.Dr.Abdulameer Khalaf Hussain, Mustafa Nouman Al-Hassan — Multifactor Strong Authentication Method Using Keystroke Dynamics — 2013. — Vol.2, No.2, — C.31-34.

TMOF УЯЗВИМОСТЬ

Трафимук М.П.

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

Селезнев И.Л. – к.т.н.,доцент

В современном мире все большее количество услуг и сервисов предоставляется посредством сети Интернет, поскольку Интернет и, соответственно, все его ресурсы доступны повсеместно. Помимо описанных положительных сторон, есть и отрицательные: доступность, стабильность и надежность зависят не от пользователя, а от владельца ресурса. Обеспечить максимальную стабильность – это первостепенная задача администратора ресурса, для выполнения которой необходимо знать возможные уязвимости и способы их устранения. Не менее важно помнить и про известные проблемы, которые не считаются уязвимостями, вроде TMOF – too many open files.

С течением времени все большее количество услуг и сервисов предоставляется посредством сети Интернет. Повсеместное распространение доступа к сети и ее ресурсам является основной причиной этого явления. Помимо положительных сторон, у этого явления есть и отрицательные моменты: доступность, стабильность и надежность зависят не от пользователя, а от владельца ресурса. Обеспечить максимальную стабильность – это первостепенная задача администратора ресурса, для выполнения которой необходимо знать возможные уязвимости и способы их устранения.

На сегодняшний день ключевые ресурсы сети Интернет являются достаточно устойчивыми к различного вида атакам, использующим уязвимости протоколов UDP и TCP, а также DNS-серверов и других широко используемых сущностей. Ранее использование недоработок в этих компонентах для реализации атак приводило к неприятным последствиям для владельцев и администраторов ресурсов сети Интернет. Ключевой идеей этих, устаревших на данный момент, атак является генерация и отправка больших потоков данных атакуемому ресурсу. Некоторые современные атаки используют схожие механизмы, но избегают условий, в которых воздействие считается атакой и нейтрализуется существующими методами противодействия вроде «blackholing».

Современные атаки используют узкопрофильные уязвимости, которые, тем не менее, могут покрывать достаточно большой процент ресурсов сети Интернет. Например, уязвимостям, направленным на подсистемы серверов на базе Unix, подвержены свыше 40% всех серверов сети

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