Устройство сервера ловушки для анализа атак на устройства Интернета вещей

А.А. Гладкий[1], Д.А. Шкирдов[1]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. В последние годы наблюдается активный рост числа устройств и систем Интернета вещей. Устройства Интернета вещей собирают, хранят, обрабатывают и передают через сеть большие объемы данных, что делает их привлекательной целью для атак злоумышленников. Эти устройства используются во многих областях, где конфиденциальность и информационная безопасность критически важны. Это делает задачу обеспечения безопасности Интернета вещей одной из наиболее важных для эффективного использования этой технологии. Для разработки систем безопасности необходимо определить стратегии и методы, используемые злоумышленниками. При решении этой задачи может быть использована технология ловушек (honeypot). В работе представлено устройство ловушки, которая имитирует устройства Интернета вещей и записывает всю сетевую активность. Логи сетевого трафика затем могут быть проанализированы для установления векторов атак и создания средств противодействия. Предложенная конструкция ловушки является универсальной и может рассматриваться как методология построения ловушек для устройств Интернета вещей, работающих на базе протокола HTTP.

1. Введение

В последние годы происходит активное развитие Интернета вещей и его внедрение во все области жизни. Интернет вещей представляет собой совокупность физических объектов, оснащенных сенсорами, актуаторами и вычислительными модулями и объединенных сетью [1]. Технология Интернета вещей используется на производстве, в здравоохранении, энергетической отрасли, сельском хозяйстве, быту и многих других областях жизни человека. Возможности применения этой технологии ограничиваются лишь изобретательностью разработчиков.

Интернет вещей открывает целый спектр новых возможностей, но при этом имеет ряд проблем, связанных с безопасностью данных. Проблемы безопасности обусловлены особенностями самой технологии: открытость архитектуры, мобильность, разнородность и ограниченность ресурсов устройств Интернета вещей, беспроводная передача данных и масштабность систем [2]. Перечисленные особенности значительно усложняют задачу обеспечения информационной безопасности в среде Интернета вещей. Именно недостаточный уровень обеспечения безопасности, а также ценность и количество обрабатываемой и передаваемой информации делают системы и устройства Интернета вещей (которые также называют «умными» устройствами) привлекательной целью для атак злоумышленников. Атаки в сегменте Интернета вещей могут быть направлены на перехват или подмену данных, а также

на перехват управления устройствами Интернета вещей. Из-за возможности применения «умных» устройств в самых разных областях жизни атаки на них могут привести к серьезным негативным последствиям. К примеру, ботнет Mirai, состоящий из тысяч взломанных «умных» устройств Интернета вещей, осуществлял одни из самых мощных DOS атак. В результате такой атаки на американский DNS провайдер Dyn DNS большое количество веб-сайтов было недоступно, в том числе и такие ресурсы, как Twitter и GitHub [3]. Таким образом, задача обеспечения безопасности устройств и систем Интернета вещей является критически важной для реального применения этой технологии.

Целью данной работы является разработка конструкции сервера-ловушки для сбора статистики атак на устройства Интернета вещей. Метод ловушек предполагает замену реального устройства его аналогом, на котором настроены механизмы сбора и анализа атакующих запросов. Подобный сервер-ловушка устанавливается анонимно в сети Интернет, о факте его установки нигде не сообщается. После этого сервер-ловушка начинает коллекционировать запросы. С большой вероятностью это будут атакующие запросы, так как обнаружить данное устройство можно только при помощи сплошного сканирования адресного пространства глобальной сети.

Трудность этой задачи состоит в отсутствии стандартов для управления устройствами Интернета вещей, что приводит к разнообразному программному обеспечению для имеющихся технологий. Интернет вещей требует создания отдельных стандартов беспроводных сетей. И в этом направлении ведутся активные работы. Так в сентябре 2019 года компания Amazon представила новый сетевой протокол Amazon Sidewalk. Протокол был специально разработан для коммуникации с недорогими устройствами с низкой пропускной способностью и малой мощностью, которые пользователи устанавливают на краю домашней сети. Примерами таких устройств могут служить различные сенсоры или умные лампочки. Протокол Amazon Sidewalk будет работать в диапазоне 900 МГц и дает возможность установить соединение на дальности до 1,5 км. При этом пропускная способность снизится, однако это допустимо из-за низкой пропускной способности самих устройств. Протокол Amazon Sidewalk позволит использовать умные устройства, находясь далеко от маршрутизатора [4].

Работы по реализации ловушек для устройств Интернета вещей уже были опубликованы рядом исследователей: Telnet IoT honeypot [5], Diaonaea [6], ThingPot [7] и др. Однако целью данных работ являлась именно реализация самой ловушки, эмулирующей отдельное устройство или протокол, а также демонстрация результатов проведенных измерений.

Новизна данной работы заключается в том, что здесь предлагаются общие принципы построения ловушек для устройств Интернета вещей на основе анализа конструкции реальных устройств, используемого программного обеспечения и сетевых протоколов. Эти принципы могут служить методологией для построения серверов-ловушек для устройств Интернета вещей.

Следуя предложенной методологии, авторами также была имплементирована и инсталлирована сервер-ловушка, работающая на базе протокола HTTP. Для получения репрезентативного набора данных для анализа ловушка должна осуществлять запись трафика в течение как минимум нескольких месяцев [8], поэтому результаты анализа данных с ловушки не представлены в данной работе и будут предметом будущих исследований.

2. Анализ существующих устройств и ловушек

Для того чтобы определиться с устройством ловушки, авторами были проанализированы принцип работы и конструкция ряда наиболее популярных устройств Интернета вещей. Также авторами были изучены уже существующие ловушки. Здесь приведем принцип работы одного из таких устройств - системы управления освещением умного дома Philips Hue - и ловушки, имитирующей это устройство - ThingPot.

Система состоит из умных лампочек Hue Light-bulb и шлюза Hue Bridge и позволяет пользователю управлять освещением с помощью мобильного приложения или через веб-сайт. Пользователь имеет возможность задать яркость, цвет или комбинацию цветов, а также расписание. Все команды от приложения или веб-сайта направляются шлюзу по протоколу

HTTP, а шлюз в свою очередь посылает команды «умным» лампам через протокол ZigBee. На шлюзе установлен веб-сервер nginx и реализовано REST API.

Установка системы состоит из нескольких шагов. Сначала шлюз подключается к Интернету с помощью кабеля Ethernet. Затем необходимо установить и запустить мобильное приложение. При первом запуске приложение сгенерирует имя пользователя. После происходит соединение приложения и шлюза. Если телефон пользователя, на котором установлено мобильное приложение, подключен к той же сети, что и шлюз, тогда соединение происходит без дополнительного конфигурирования, если же шлюз и телефон находятся в разных сетях, тогда в настройках приложения необходимо указать IP адрес и номер порта шлюза. Затем мобильное приложение отправляет GET запрос к шлюзу по пути */api/${username}*. Если в памяти шлюза установлено, что пользователь с этим именем аутентифицирован, то соединение успешно устанавливается, иначе пользователю требуется нажать на физическую кнопку на шлюзе. После нажатия на кнопку имя пользователя будет занесено в «белый список» шлюза.

Для начальных попыток исследования атак на устройства Интернета вещей была использована система управления освещением умного дома Philips Hue. Система состоит из умных лампочек Hue Light-bulb и шлюза Hue Bridge и позволяет пользователю управлять освещением с помощью мобильного приложения или через веб-сайт. Пользователь имеет возможность задать яркость, цвет или комбинацию цветов, а также расписание. Все команды от приложения или веб-сайта направляются шлюзу по протоколу HTTP, а шлюз в свою очередь посылает команды «умным» лампам через протокол ZigBee. На шлюзе установлен веб-сервер nginx и реализовано REST API.

После установления соединения приложение может отправлять разнообразные команды в виде HTTP запросов шлюзу для управления лампами. Например, для изменения состояния лампы приложение отправляет PUT запрос по пути */api/${username}/lights/${light-bulb-number}/state*, в теле запроса отправляется JSON с соответствующими параметрами.

Для эмуляции этой системы исследователями из Делфтского технического университета была разработана сервер-ловушка, которую назвали ThingPot. Особенностью этой ловушки является то, что она эмулирует целую платформу Интернета вещей. Платформа данной ловушки имитирует систему управления освещением умного дома Philips Hue и реализует коммуникацию сразу по двум протоколам – XMPP и HTTP. ThingPot состоит из трех компонентов: XMPP узлов, REST узлов и контроллера. Контроллер используется для логгирования, хранения информации, обновления кода и т.д. [7]

Сервисы работают в изолированной (виртуальной) среде для снижения вероятности того, что злоумышленник сможет добраться до самой системы, которая находится за эмуляцией. REST API реализован с помощью фреймворка DjangoREST и представляет собой полную копию интерфейса реального устройства. [7]

Описанный путь эмуляции устройства очень сложен и требует огромных усилий для использования, при этом осуществляется эмуляция лишь отдельных устройств. Но в настоящий момент это единственный путь для анализа атак методом ловушек.

3. Устройство ловушки

В настоящий момент мы можем сформулировать только общие моменты конструкции сервера-ловушки для среды Интернета вещей. В работе предлагается конструкция ловушки, имитирующей устройства, работающие на базе протокола HTTP. Протокол HTTP был выбран как один из самых широко распространённых коммуникационных протоколов в среде Интернета вещей.

Протокол передачи гипертекста (Hypertext Transfer Protocol – HTTP) — это прикладной протокол для передачи гипертекстовых документов, таких как HTML. Он создан для связи между веб-браузерами и веб-серверами, хотя в принципе HTTP может использоваться и для других целей. Протокол следует классической клиент-серверной модели, когда клиент открывает соединение для создания запроса, а затем ждет ответа. HTTP — это протокол без сохранения состояния, то есть сервер не сохраняет никаких данных (состояние) между двумя

парами "запрос-ответ". Несмотря на то, что HTTP основан на TCP/IP, он также может использовать любой другой протокол транспортного уровня с гарантированной доставкой [9].

Чаще всего взаимодействие по протоколу HTTP между элементами распределенной системы Интернета вещей построено на архитектуре REST (Representational State Transfer). REST представляет собой вид архитектуры, задающий набор из 6 ограничений, которые позволяют разрабатывать эффективные распределенные приложения. Говорят, что система является RESTful, если она удовлетворяет этим требованиям. Иногда несложные HTTP API для простоты называют RESTful API, хотя они могут и не соответствовать всем ограничениям. На устройстве Интернета вещей реализуется REST API, взаимодействие с которым осуществляется путем отправки HTTP запросов на определенные шаблоны URL [10]. В заголовке Request Method HTTP запроса устанавливается метод, который определяет тип операции над ресурсом: GET используется для получения данных от устройства, а методы PUT, DELETE и POST – для модификации данных или состояния самого устройства.

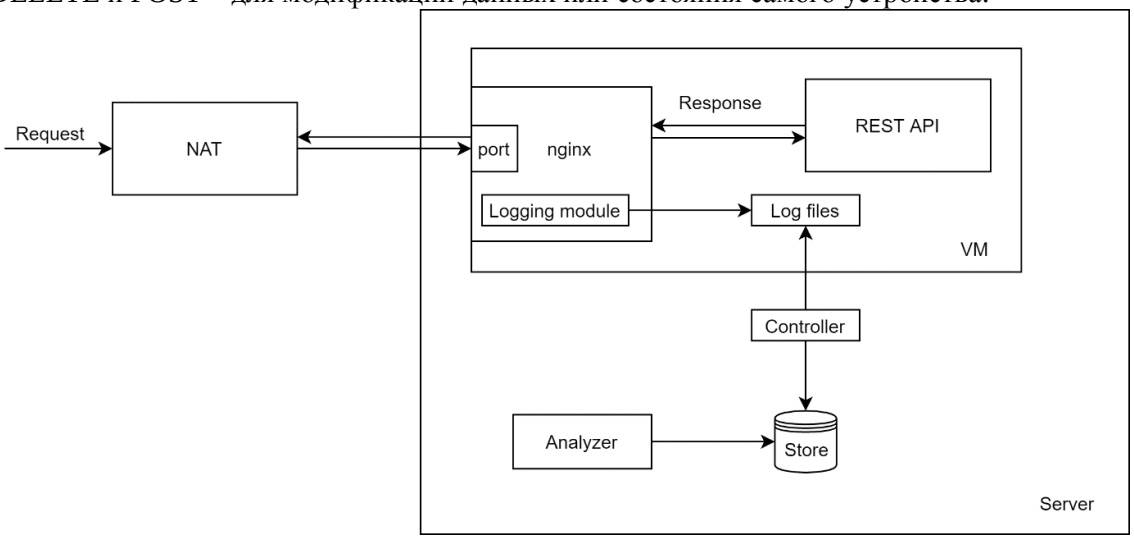

Рисунок 1. Концептуальная схема сервера-ловушки для устройств Интернета вещей.

На рисунке 1 представлена концептуальная схема ловушки. Ловушка состоит из набора связанных между собой элементов, функциональное назначение которых описано ниже.

NAT-устройство, за которым находится система, позволяет узлам из частной сети прозрачным для пользователей образом получать доступ к узлам внешних сетей. На маршрутизаторе, связывающем частную сеть с внешней сетью, устанавливается программное обеспечение NAT, что превращает этот маршрутизатор в NAT-устройство. Это NAT-устройство динамически отображает набор частных адресов на набор глобальных адресов, присвоенных внешнему интерфейсу маршрутизатора. В ловушке технология NAT используется по причине того, что все реальные системы устанавливаются за NAT-устройством, а также в целях безопасности, так как технология NAT позволяет скрыть адреса частной сети, чтобы не дать возможности злоумышленникам составить представление о структуре и масштабах частной сети, а также о структуре и интенсивности исходящего и входящего трафиков [11].

На выделенном для ловушки сервере необходимо запустить виртуальную машину, работающую на базе операционной системы Linux. Виртуальная машина используется в целях создания безопасной изолированной среды, которая позволит предотвратить заражение всей системы в случае внедрения вредоносного программного обеспечения злоумышленниками. Операционная система Linux нужна, во-первых, из-за того, что подавляющее большинство устройств Интернета вещей работают на базе Linux, и это снизит вероятность раскрытия ловушки, и, во-вторых, операционная система Linux имеет широкие сетевые возможности, позволяющие быстро развернуть модули ловушки.

nginx – это HTTP-сервер и обратный прокси-сервер, почтовый прокси-сервер, а также TCP/UDP прокси-сервер общего назначения [12]. nginx может быть использован как для

небольших, так и для высоконагруженных сайтов. nginx - быстрый, функциональный и простой в использовании сервер, именно поэтому предлагается использовать его для выполнения проксирования запросов к REST API. Также nginx имеет встроенные возможности по логгированию сетевых взаимодействий и позволяет тонко настроить средства записи с помощью механизмов условного логгирования. Обычно управление устройствами Интернета вещей осуществляется через порт 80, но номер может быть другим, поэтому при анализе эмулируемого устройства требуется определить номер порта, через который происходит управление устройством, и затем настроить nginx на прослушивание этого порта.

Модуль REST API эмулирует интерфейс взаимодействия с устройством. Разрабатывать полную эмуляцию нет необходимости, так как обычно разведка устройств осуществляется путем автоматического сканирования сетевых ответов. Если первый ответ показал, что устройство является устройством Интернета вещей, то затем с машин злоумышленника автоматически начинают отправляться атакующие запросы. Таким образом, достаточно реализовать API так, чтобы ответ на первый разведывательный запрос был неотличим от ответа реального устройства. На данный момент существует множество инструментов для разработки REST API на разных язык программирования: JavaScript – Express.js, Python – DjangoREST и т.д.

Контроллер представляет собой программный модуль, который извлекает записи логов, осуществляет их предобработку (если это необходимо) и сохраняет данные в *хранилище* вне виртуальной машины. Это делается для обеспечения сохранности собранных данных в случае проблем с виртуальной машиной или в случае заражения виртуальной среды злоумышленниками. В качестве хранилища может быть использована реляционная база данных или другое удобное средство хранения данных.

Анализатор – это программный модуль, осуществляющий аналитическую обработку записей в хранилище. Анализатор может использовать различные алгоритмы классификации, машинного обучения или другие методы обработки и анализа данных.

Предложенное устройство ловушки имеет универсальный характер и может быть использовано для эмуляции различных устройств Интернета вещей, работающих на базе протокола HTTP.

4. Выводы

В работе был проведен анализ сетевого подключения устройств Интернета вещей. На основании этого анализа сделаны выводы о конструкции сервера-ловушки, и представлена обобщенная схема построения ловушек для устройств Интернета вещей. Предложенная схема может использоваться в качестве методологии построения ловушек для устройств Интернета вещей, работающих на базе протокола HTTP.

Также на основе предложенной методологии был собран, запущен и приступил к работе honeypot, эмулирующий систему управления освещением Philips Hue. После сбора и анализа данных с ловушки появляется возможность разработки средств обеспечения информационной безопасности для устройств Интернета вещей. Такое средство или совокупность средств являются конечной целью будущих исследований авторов.

5. Литература

[1] Sethi, P. Internet of things: architectures, protocols, and applications / P. Sethi, S.R. Sarangi // Journal of Electrical and Computer Engineering. – 2017. – Т. 2017.

[2] Iqbal, A. Internet Of Things (Iot): On-Going Security Challenges And Risks // International Journal of Computer Science and Information Security. – 2016. – Т. 14, №. 11. – С. 671.

[3] Rizvi, S. Securing the internet of things (IoT): a security taxonomy for IoT // 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications. 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018. – P. 163-168.

[4] Introducing Amazon Sidewalk // The Amazon Blog [Electronic resource]. – Access mode: https://blog.aboutamazon.com/devices/introducing-amazon-sidewalk (10.02.2020).

[5] Phype. Telnet IoT honeypot // GitHub [Electronic resource]. – Access mode: https:// github.com/Phype/telnet-iot-honeypot (10.02.2020).
[6] DinoTools. Dionaea - catches bugs // GitHub [Electronic resource]. – Access mode: https://github.com/DinoTools/dionaea (10.02.2020).
[7] Wang, M. ThingPot: an interactive Internet-of-Things honeypot / M. Wang, J. Santillan, F. Kuipers //arXiv preprint arXiv: 1807.04114, 2018.
[8] Sagatov, E.S. Analysis of Network Threats Based on Data from Server-Traps / E.S. Sagatov, D.A. Shkirdov, A.M. Sukhov // 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019. – P. 1-5.
[9] Mozilla Developer Network [Electronic resource]. – Access mode: https://developer. mozilla.org/ru/docs/ (8.05.2019).
[10] REST // Mozilla Developer Network [Electronic resource]. – Access mode: https://developer. mozilla.org/en-US/docs/Glossary/REST (23.03.2019).
[11] Олифер, В.Г. Основы компьютерных сетей / В.Г. Олифер, Н.А. Олифер, 2009.
[12] nginx [Electronic resource]. – Access mode: https://nginx.org/ru/ (10.02.2020).

IoT honeypot design for attack strategies analysis

A.A. Gladkii[1], D.A. Shkirdov[1]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. In the recent years there has been a dynamic growth of the Internet of Things (IoT). IoT devices store, collect, process and transfer big amounts of data thus becoming a target for malicious attacks. These devices are used in many different fields where privacy and information security are crucial. It makes securing IoT one of the most important challenges for effective usage of this technology. In order to develop security systems, it is necessary to define strategies and techniques used by attackers. Honeypot technology can be used for this purpose. In this paper we present a design of an IoT honeypot that imitates real IoT device and logs all network activity. Logs of network traffic then can be used for analysis to determine threat vectors and for development of security systems. Design of a honeypot presented in this article is universal and can be considered as a methodology for developing IoT honeypots that use HTTP for communication.