УДК 004.492

Сильнов Д.С., Титов К.Е.

Национальный исследовательский ядерный университет МИФИ, г. Москва, Россия

РАЗРАБОТКА И РЕАЛИЗАЦИЯ HONEYPOT-ЛОВУШКИ СЕТЕВЫХ СЛУЖБ, ИСПОЛЬЗУЮЩИХ ПРОТОКОЛ SIP

АННОТАЦИЯ

Ни для кого не секрет, что в современном мире киберпреступления активно развиваются. Одной из IT-сфер, которые подвержены атакам, является сфера интернет телефонии. Сетевые ловушки являются относительно новым способом борьбы с постоянно развивающимися сценариями атак. Сетевая ловушка (она же honeypot) – это система, представляющая из себя приманку для злоумышленника. Она обычно состоит из компьютера, программ и информации, которые вместе симулируют поведение реальной системы, являющейся частью сети. Добросовестным пользователям нет смысла подключаться к такой системе, поэтому наблюдение за попытками получить доступ к ловушке и активностью в ней позволяет получить сведения об уровне угроз реальной системе. Информация, полученная в результате работы ловушки сетевых служб, систематизируется и анализируется. В результате анализа можно сделать вывод о стадии развития алгоритмов атак на тот протокол или систему, которая была подменена ловушкой. В статье обсуждается опасность сетевых атак на службы, использующие SIP протокол. Этот протокол является протоколом установления сессии в системах интернет телефонии. Рассматриваются популярные honeypot-ловушки, работающие по этому протоколу, а также описывается алгоритм и реализация собственной ловушки. В качестве сервера интернет телефонии выбрано программное обеспечение Asterisk, работающее на ОС Linux.

КЛЮЧЕВЫЕ СЛОВА

Honeypot, АТС, dialplan, конфигурационный файл, маршрутизация вызова, контекст, добавочный номер extention, приложение.

Silnov D.S., Titov K.E.

National research nuclear university MEPhI, Moscow, Russia

DEVELOMPENT AND IMPLEMENTATION OF HONEYPOT WORKING ON SIP PROTOCOL

ABSTRACT

Everybody knows that there are a lot of cybercrimes that actively develop in modern world. One of the IT sphere that prone to cyberattacks is internet telephony sphere. Honeypot is a new way of fighting against cybercrimes in this sphere. Honeypot is the system that attracts malefactors. It usually consists of computer, programs and data which simulate real system behavior. There is no need for honest people to contact with such system, that's why monitoring of attempts of getting an access to a honeypot and the activity in the system help to get an information about the level of the danger in the real system. As a result of honeypot activity the information is being systemized and analyzed. So you can make a conclusion about the level of attack algorithms on any protocol or system, which has been replaced by honeypot. This protocol describes the algorithm of session initiation in internet telephony systems. In the article the danger of SIP service attacks is discussed. The most popular SIP honeypots are reviewed, the algorithm and its realization are represented. Software Asterisk compatible with operation system Linux works as the server of internet telephony.

KEYWORDS

Honeypot, telephone exchange, dialplan, configuration file, call routing, context, extention number, extention, application.

В последнее время honeypot'ы получили широкое распространение в it-сфере. Honeypot (ловушка) – это система, представляющая из себя приманку для злоумышленников. Она обычно

состоит из компьютера, программ и информации, которые вместе симулируют поведение реальной системы, являющейся частью сети.

Добросовестным пользователям не имеет смысла подключаться к такой системе, поэтому наблюдение за попытками получить доступ к ловушке и активностью в ней позволяет получать сведения об уровне угроз реальной системы.

После анализа собранной информации по подключениям к ловушке можно получить примерные методы, которые используют злоумышленники при попытке взлома системы, а также разработать свои решения противодействия данным методам.

Существует два возможных метода создания ловушки: создание honeypot на выделенном сервере и эмулирование ловушки при помощи виртуальной машины. Honeypot, созданный на выделенном сервере максимально приближен к системе, однако в некоторых случаях целесообразнее провести эмуляцию honeypot'а. В компаниях с обширной сетевой архитектурой как правило используются выделенные серверы с преднамеренно допущенными ошибками в конфигурации системы. Эти ошибки позволяют заманить взломщиков в ловушку. [1]

В рамках данной статьи проводится описание процесса разработки ловушки на базе программного обеспечения Asterisk, поддерживающего протокол установления сеанса SIP и представляющего из себя программное решение автоматической телефонной станции.

Киберпреступления в области компьютерной телефонии все набирают популярность, и наиболее распространенным сценарием атаки в данной сфере являются звонки на платные номера.

Описываемая ловушка представляет из себя сервер с установленным на нем программным обеспечением Asterisk и выставленным в интернет IP-адресом. Злоумышленники, использующие выше обговоренный сценарий атаки, сканируют интернет на наличие IP-адресов подобных серверов. После нахождения такого адреса, злоумышленник подключается к серверу и через него совершает звонок на платный номер.

Разработанная ловушка не перенаправляет такой вызов на запрашиваемый адрес, а лишь собирает статистику по звонку, записывая ее в базу данных. По результатам работы такой ловушки будет собрана статистика звонков, на основе которой можно будет определить новые способы атак на системы компьютерной телефонии, а также модернизировать свою систему для защиты от них.

SIP - протокол установления сеанса, содержащий алгоритмы установления и завершения сеанса общения между двумя пользователями. Протокол описывает алгоритмы, за счет которых клиентское приложение запрашивает разрешение у удаленного клиента на установление с ним соединения по уникальному имени удаленного клиента.

SIP спроектирован для того, чтобы обеспечить контроль сессий для голосовых и мультимедиа соединений в сетях с пакетной передачей данных. Пример установления соединения между двумя клиентами посредством SIP протокола представлен на рисунке 1.

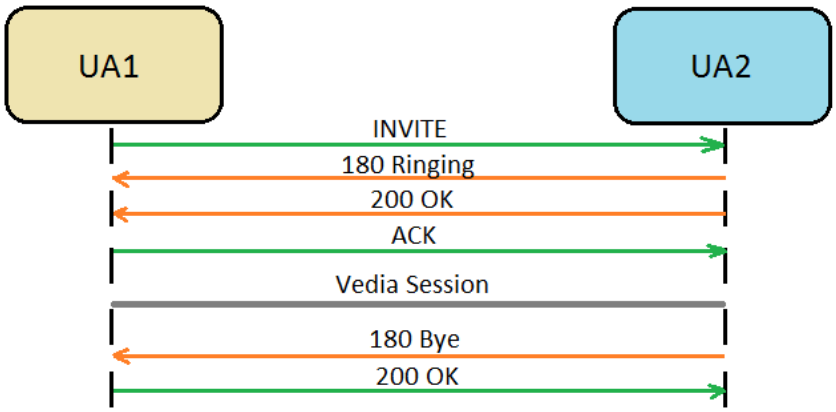

Рис. 1 – Взаимодействие между двумя клиентами с установлением и окончанием сессии при помощи SIP протокола

UA1 (User Agent 1) хочет создать сессию с UA2. Для этого он отсылает второму INVITE-сообщение, предложение присоединиться к сессии.

Далее у UA2 начинает звонить телефон, информация об этом присылается UA1 в пакете 180 Ringing. Если UA2 отвечает на вызов, то пакет с информацией об успешном соединении отсылается первому пользователю. UA1 отправляет ACK пакет, которым оповещает второго о том, что он успешно принял предыдущее сообщение. Так же в этом пакете отсылаются окончательные параметры соединения.

Когда один из пользователей кладет трубку, второму отсылается пакет 180 Bye. Ответом на это служит сообщение 200 OK. Сессия окончена. [2]

Asterisk – бесплатное программное обеспечение, позволяющее организовать на сервере цифровую автоматическую телефонную сеть с широким спектром возможностей.

Asterisk в комплекте с необходимым оборудованием обладает всеми возможностям классической автоматической телефонной сети и предоставляет пользователю богатые функции управления звонками.

Asterisk сильно отличается от классических АТС. В телефонной сети, построенной посредством Asterisk'а, все входящие и исходящие вызовы обрабатываются при помощи так называемого dialplan'а (план набора).

План набора – сердце Asterisk'а, скрипт, который подробно описывает управление потоком звонков, приходящих на сервер. При помощи плана набора можно заставить сервер пересылать звонки с одного абонента на другого, организовать голосовое меню, предлагающее пользователю нажимать кнопки на своем телефоне в зависимости от его выбора, а также оптимизировать работу автоматической телефонной станции в соответствии с требованиями организации, использующей сервер Asterisk, а также многое другое.

ПО поддерживает следующие протоколы:

- SIP (протокол установления сеанса, описывает алгоритмы установления и завершения сеанса общения между двумя пользователями);
- H.323 (содержит набор стандартов, необходимых для передачи мультимедиа-данных);
- MGCP (протокол контроля медиашлюзов);
- IAX-2 (протокол обмена Voice over IP данными между несколькими виртуальными АТС Asterisk);
- OSP (протокол, используемый провайдерами для авторизации при применении SIP-телефонии).

Asterisk может работать практически на любой ОС Linux, а также на некоторых других операционных системах. Кроме того, существуют готовые дистрибутивы, содержащие операционную систему, скомпилированный Астериск и стандартную конфигурацию.

Например, ОС AsteriskNow – операционная система на базе Linux CentOS, включающая в себя предустановленное программное обеспечение Asterisk и предназначенная для быстрого развертывания виртуальной автоматической телефонной станции на персональном компьютере.

Каналы в Asterisk представляют собой внешние или внутренние соединения, по которым вызовы доставляются до сервера с программным обеспечением. Каналом может являться как виртуальное соединение, предназначенное для общения посредством интернета, так и физическое соединение с телефоном или телефонной сетью.

Каналы настраиваются при помощи добавления новых записей в конфигурационный файл sip.conf.

Прежде чем мы перейти к обсуждению плана набора Asterisk'а в полном объеме, будет полезно привести пример взаимодействия между файлом конфигурации каналов (sip.conf) и dialplan'ом (extentions.conf).

Dialplan является сердцем системы Asterisk: он контролирует всю логику подключений по любым каналам. То есть именно он описывает порядок действий системы при входящем звонке от внешнего провайдера, либо при звонке на международный номер и т.д.

Взаимодействие между файлами sip.conf и extentions.conf демонстрируется на рисунке 2. Когда вызов приходит на Asterisk, sip.conf идентифицирует его к определенному каналу, проводит аутентификацию вызова и определяет точку входа в dialplan. Дальнейшая маршрутизация вызова проходит в соответствии с файлом extentions.conf.

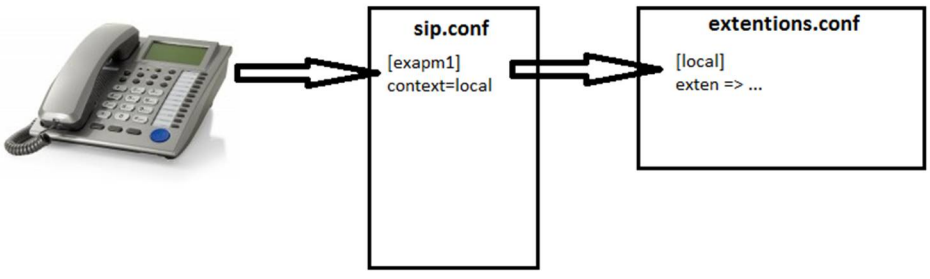

Рис. 2 – Взаимодействие между sip.conf и extentions.conf [3]

Dialplan (план набора) – описание логики маршрутизации телефонных звонков. Номерной план содержит подробное изложение действий системы при прохождении входящих и исходящих звонков через сервер с установленным ПО Asterisk на нем.

План набора разбит на секции, называемые контекстами (contexts). Контексты предотвращают взаимодействие различных частей dialplan'а друг с другом.

Например, для обработки звонков с городского номера и с внутреннего номера компании нужно придерживаться различных логик. Логичнее разнести эти алгоритмы по разным контекстам внутри одного плана набора.

Строго говоря, контексты нужны для выполнения основных функций автоматических телефонных станций:

- Безопасность (возможность настроить dialplan так, чтобы лишь ограниченный круг лиц имел доступ к определенному списку функций);
- Списки доступа (возможность заносить в черные списки неугодных вам абонентов);
- Маршрутизация вызовов (определение маршрута для вызова в зависимости от вызываемого абонента);
- Дневной/ночной режимы работы (возможность изменять режим работы АТС в зависимости от времени суток).

Каждый контекст может включать в себя неограниченное количество так называемых экстеншенов (extentions).

Экстеншен определяет уникальную серию шагов, которых будет придерживаться Asterisk, обрабатывая вызов, принадлежащий данному контексту.

Синтаксис экстеншена

Экстеншен в плане набора начинается со следующей комбинации символов:

exten =>

За ней идет имя или номер экстеншена. В традиционных АТС аналогом этого имени может послужить телефонный номер. В Asterisk'е роль имени может играть комбинация из чисел и букв.

Каждый шаг в экстеншене включает в себя три компонента:

- Имя или номер экстеншена;
- Приоритет (каждый экстеншен может содержать несколько шагов, порядковый номер шага называется его приоритетом);
- Приложение или команда, которая выполняется на этом шаге. В скобках указываются параметры, передаваемые команде, если они необходимы.

Конструкция шага в экстеншене:

exten => имя, приоритет, приложение()

Каждое приложение (команда) представляет из себя какое-либо действие над каналом, по которому проходит вызов. Например: проигрывание музыкальной дорожки, поиск информации в базе данных, установление канала, сброс вызова и так далее.

Простейший пример dialplan'а:

```
EXTEN => 20, 1, ANSWER()
EXTEN => 20, 2, WAIT(5)
EXTEN => 20, 3, PLAYBACK(GREETING)
EXTEN => 20, 4, HANGUP()
```

Следуя первому приоритету экстеншена, мы отвечаем на входящий вызов, далее мы ждем 5 секунд согласно команде Wait с аргументом 5. Затем проигрываем мелодию с названием greeting и разрываем соединение.

Построенный выше план набора всегда выполняет одни и те же действия при вызове экстеншена под номером 20. В большинстве планов набора требуется наличие более сложной логики, опирающейся на вводе символов абонентом.

Команда Goto() используется для отправки звонка в другую часть диалпална. Команде передаются аргументы: **контекст, экстеншен, приоритет**, которые определяют точное место следующего шага.

При вызове экстеншена 21 звонок будет перенаправляться на начальный экстеншен в контексте Menu.

```
EXTEN => 21, 1, GOTO(MENU, MN, 1)
…
[MENU]
        EXTEN => MN, 1, ANSWER()
```

Анализ существующих ловушек для протокола SIP

Целью создания ловушек в VoIP сетях является мониторинг и анализ зловредного трафика. Трафик наблюдается при помощи набора специальных приложений, которые могут следить за различными аспектами инфраструктуры IP-телефонии. Подобные приложения называются honeypot'ами. Использование honeypot'ов дает много информации об атаках на реальные системы без угрозы для них. Анализ информации, полученной при помощи ловушек, является ключевым моментом для улучшения существующих механизмов в инфраструктуре IP-телефонии.

Ловушка **Artemisa** может быть встроена в любую VoIP систему, которая использует SIP протокол. В этой системе она играет роль обычного SIP телефона. Экстеншен, обрабатывающий звонок на телефон-ловушку, должен находиться в диапазоне экстеншенов, используемых реальными оконечными устройствами, это нужно для того, чтобы замаскировать ловушку.

При вызове одного из экстеншенов, ассоциируемых с Artemisa, ловушка принимает звонок и в то же время начинает анализировать входящее SIP сообщение. Далее Artemisa классифицирует звонок и сохраняет результат для дальнейших исследований системным администратором.

Сообщение классифицируется следующим путем. В начале ловушка ищет совпадения с хорошо известными атаками. Затем она проверяет имена доменов и SIP порты на атакующей стороне. Наконец, программа проверяет полученный RTP поток. [4]

Данная последовательность действий позволяет программе классифицировать звонок. [4]

Следующая ловушка, **Dionaea**, не нуждается во внешнем VoIP сервере. Она просто ожидает любое SIP сообщение и пытается ответить на него. Ловушка поддерживает все SIP запросы (REGISTER, INVITE, ACK, CANCEL, BYE, OPTIONS), а также протокол RTP. Весь трафик отслеживается, логи сохраняются в базу данных. [5]

Разработка архитектуры логической ловушки

Примерная схема архитектуры будущей логической ловушки для протокола SIP изображена на рисунке 3.

При сканировании интернета злоумышленник обнаружит открытый порт нашего сервера, подключится к нему и совершит звонок на платный номер по SIP протоколу, который будет проходить через PSTN (телефонную сеть общего пользования).

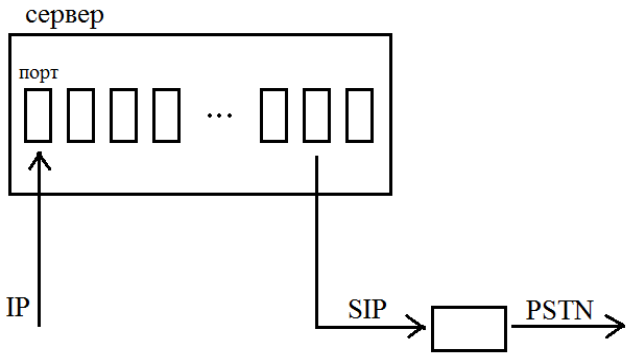

Рис. 3 – Схема архитектуры логической ловушки

Разработка алгоритмов

Разработаем алгоритм ловушки с использованием протокола SIP и сервера Asterisk.

Ловушка предназначена для того, чтобы собирать статистику по звонкам на международные платные номера, совершенным при помощи нашего сервера. Данный звонок будет совершен при участии российского оператора связи и оператора связи той страны, в которой находится платный номер. Счет за звонок на платный номер будет предоставлен зарубежным оператором российскому, который в свою очередь направит его владельцу сервера Asterisk. Для наглядности схема подобного звонка изображена на рисунке №3.

Рис. 4 – Звонок на платный номер направление вызова
счет за звонок

Чтобы получить доступ к серверу ip-телефонии и реализовать свой сценарий атаки, злоумышленник сканирует белые ip-адреса в интернете. Наш сервер выставлен в интернет, значит рано или поздно его ip-адрес будет обнаружен.

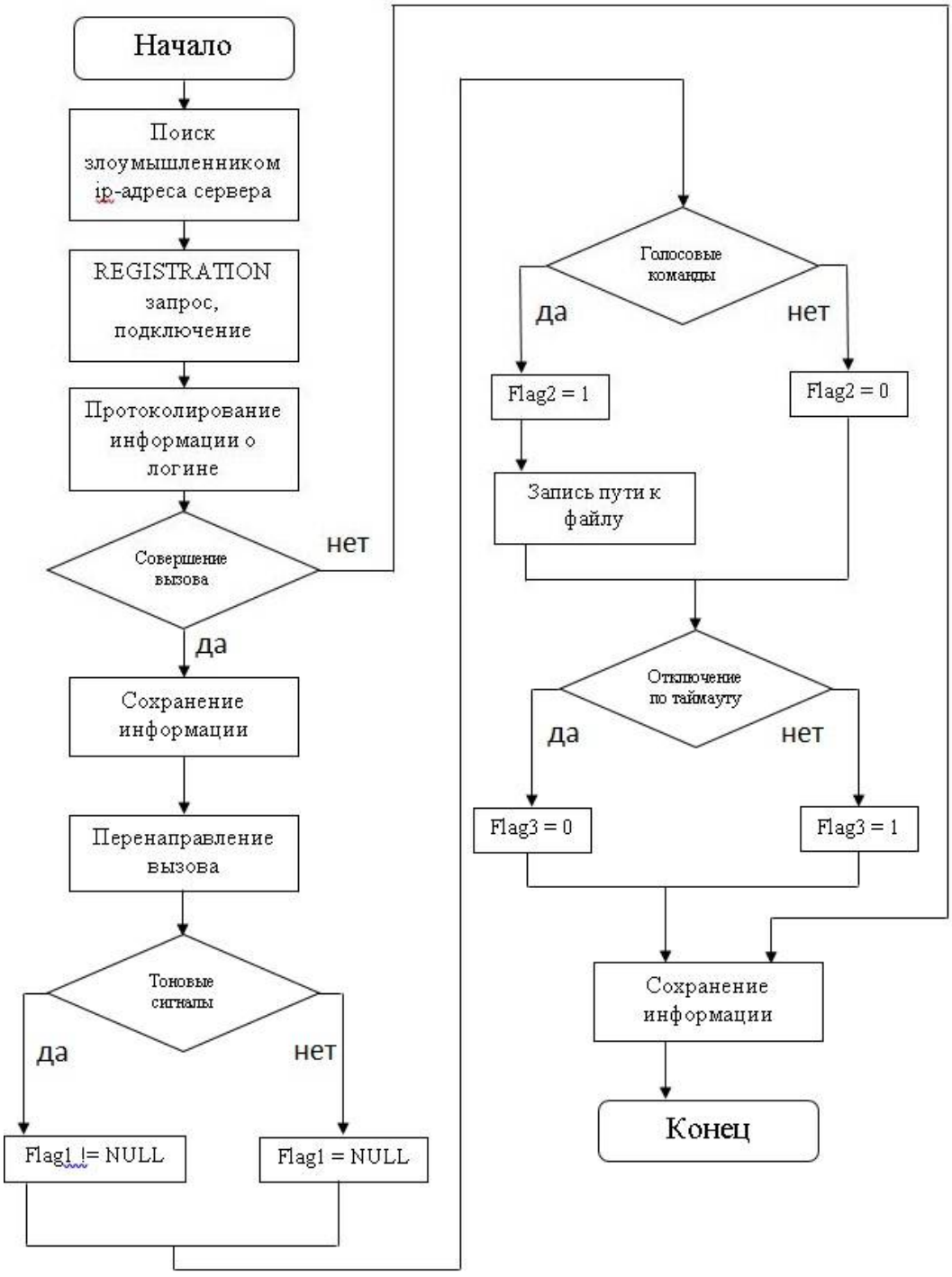

Рис. 4 – Алгоритм ловушки

В рамках реализации сценария атаки злоумышленник подключится к серверу и отправит REGISTRATION запрос нашему серверу с целью подключиться к нему.

После успешного подключения злоумышленника к серверу существует 2 варианта развития событий. Если вызов не будет совершен в течение определенного количества времени, мы записываем в базу данных информацию о потенциальном злоумышленнике, подключавшемся к серверу Asterisk. Если же совершается вызов на зарубежный платный номер (владельцем которого является сам злоумышленник, либо лицо, находящееся в сговоре с ним) мы заносим в базу данных номер телефона, куда идет звонок, и перенаправляем его на наш внутренний номер.

Далее собирается информация о характеристиках звонка и заносится в базу данных для

дальнейшего анализа возможных угроз и атак в сфере ip-телефонии.

В базу данных заносятся такие сведения, как номер, на который совершался звонок, ip-адрес злоумышленника, порт, на который пришло соединение, факт передачи дополнительных тоновых сигналов, голосовых команд, характер отключения злоумышленника от сервера: по таймауту или самостоятельно. Весь алгоритм представлен на рисунке 5.

Реализация модуля Asterisk

Итак, теперь перейдем непосредственно к реализации ловушки на сервере Asterisk.

После того, как сервер Asterisk будет обнаружен злоумышленником в результате сканирования белых ip-адресов, он может попытаться совершить вызов на зарубежный платный номер.

Для обработки данной ситуации нужно описать экстеншен в файле extentions.conf.

```
EXTEN => _X., 1, ANSWER() ;ОТВЕЧАЕМ, НЕ СОЕДИНЯЯ С НУЖНЫМ НОМЕРОМ
EXTEN => _X., N, PLAYBACK(WELCOME) ;ПРИВЕТСТВИЕ
EXTEN => _X., N, SET(CALLER = ${CALLERID(NUM)})
EXTEN =>_X., N, MIXMONITOR(DATA/${STRFTIME(,,%Y/%M/%D/%H_%M_%S)}_${CALLER}.GSM) ;ЗАПИСЬ
В ФАЙЛ
EXTEN =>_X., N, SET(CALLD = ${EXTEN}) ;ЗАНОСИМ В ПЕРЕМЕННУЮ НОМЕР
EXTEN =>_X., N, SET(F1 = 'NULL')
EXTEN =>_X., N, SET(F2 = 0)
EXTEN =>_X., N, SET(ADDRESS =
/VAR/SPOOL/ASTERISK/MONITOR/DATA/${STRFTIME(,,%Y/%M/%D/%H_%M_%S)}_${CALLER}.GSM)
EXTEN => _X., N, SET(CHAN = ${CHANNEL}) ;ИМЯ КАНАЛА
EXTEN => _X., N, SET(F3 = 1) ;
EXTEN => _X., N, SET(IP = ${SIPCHANINFO(RECVIP)}) ;IP-АДРЕС ЗВОНЯЩЕГО
EXTEN => _X., N, WAIT(60) ;ОЖИДАНИЕ
EXTEN => _X., N, HANGUP() ;ОТКЛЮЧЕНИЕ
EXTEN => _X., N, SET(F3 = 0) ;ОТКЛЮЧЕНИЕ ПО ТАЙМ-АУТУ
EXTEN => _X., N, MYSQL(CONNECT CONNID LOCALHOST USER PASS DBHONEYPOT) ;ПОДКЛЮЧАЕМСЯ К БАЗЕ
ДАННЫХ
EXTEN => _X., N, MYSQL(QUERY RESULTID ${CONNID} INSERT INTO INFO SET CALLD=${CALLD}, FLAG1=${F1},
FLAG2=${F2}, ADDR=${ADDRESS}, FLAG3=${F3}, IPADDR=${IP}, PORT=${CHAN})
```

В данной статье была поставлена цель объяснить целесообразность и разработать алгоритм ловушки, имитирующей работу сетевых служб SIP на базе программного обеспечения Asterisk.

После анализа поставленных задач алгоритм был разработан, и ловушка была внедрена, что позволило получить некоторою статистику, которая нуждается в систематизации и статистики. Одно можно сказать точно: на ловушку приходят запросы на подключение к серверу, а значит данное направление атак существует и его нужно исследовать.

Результаты проведенных в рамках статьи исследований можно использовать для дальнейших научных исследований тенденций в сфере кибернетических атак на системы компьютерной телефонии, а также для сбора статистики.

Литература

1. M. M. Rehman H., Honeypots and Routers Collecting Internet Attacks, 2015.
2. Гольдштейн Б.С., Справочник по телекоммуникационным протоколам. Протокол SIP.
3. M. L. Bryant R., Asterisk: The Definitive Guide, O'Reilly Meadia, 2013.
4. Jakub Šafařík, «Monitoring of Malicious Traffic in IP Telephony,» CESNET, Praha, 2012.
5. Dionaea URL http://www.edgis-security.org/honeypot/dionaea/.

References

1. M. M. Rehman H., Honeypots and Routers Collecting Internet Attacks, 2015.
2. Goldstein B. S., Spravochnik po telekommunikacionnim protokolam. Protocol SIP.
3. M. L. Bryant R., Asterisk: The Definitive Guide, O'Reilly Meadia, 2013.
4. Jakub Šafařík, «Monitoring of Malicious Traffic in IP Telephony,» CESNET, Praha, 2012.
5. Dionaea URL http://www.edgis-security.org/honeypot/dionaea/ .

Поступила: 13.10.2016

Об авторах:

Сильнов Дмитрий Сергеевич, кандидат технических наук, доцент кафедры компьютерных систем и технологий Национального Исследовательского Ядерного университета МИФИ, ds@silnov.pro;

Титов Кирилл Евгеньевич, студент факультета кибернетики и информационной безопасности Национального Исследовательского Ядерного университета МИФИ.