ОБНАРУЖЕНИЕ SYN-ACK FLOOD-АТАК ДЛЯ ПРЕДОТВРАЩЕНИЯ ПЕРЕГРУЗКИ СИСТЕМНЫХ РЕСУРСОВ ВЕБ-СЕРВЕРОВ

Борисов Р. С.[1]

Ключевые слова: *TCP-сессия, SYN-ACK Flood, теория массового обслуживания, DDoS-атака, распределение Пуассона.*

Аннотация

Цель: разработка математической модели раннего обнаружения SYN-ACK Flood для повышения эффективности использования системных ресурсов веб-сервера при реализации на него DDoS-атаки.

Методы: системно-прикладной анализ способов реализации DDoS-атак и протоколов передачи данных, математическое моделирование информационных потоков протокола TCP на основе теории систем массового обслуживания.

Результаты: выведены формальные отношения, позволяющие идентифицировать SYN-ACK-атаки для принятия превентивных мер от нехватки системных ресурсов (resource starvation), приводящих к отказу обслуживания запросов легитимных пользователей сети.

EDN: UNFQTJ

Введение

Возросшая популярность *DDoS*-атак (англ. *Distributed Denial of Service* — «распределённая атака „отказ в обслуживании“») в сфере киберпреступности связана с их прибыльностью и простотой реализации. «Атаки как услуга» (*Attacks-as-a-Service*) становятся всё более распространёнными благодаря возможностям анонимного взаимодействия заказчиков и исполнителей этих атак в части как общих, так и финансовых коммуникаций в глобальных сетях [4, 7, 9, 11].

Одним из наиболее популярных способов организации *DDoS*-атак является отправка на веб-сервер огромного количества ложных запросов на соединение, перегружающих систему и приводящих к её отказу. Эти атаки основаны на особенностях протокола *TCP* (*Transmission Control Protocol* — «протокол управления передачей»). При установлении соединения клиент посылает серверу запрос в виде пакета *TCP* с установленными флагом *SYN* (*sequence number* — «порядковый номер»). В ответ на этот запрос сервер посылает *TCP*-пакет с установленным флагом *ACK* (*acknowledgement number* — «номер подтверждения») и резервирует у себя ресурсы для продолжения информационного обмена. Соединение, для которого зарезервированы системные ресурсы сервера, но ещё не пришло подтверждение от пользователя, называется полуоткрытым. Если с сервером связывается легитимный пользователь, он продолжает информационный обмен до завершения соединения, а если злоумышленник — то он ничего не отправляет серверу, который может держать это соединение полуоткрытым довольно продолжительное время. При наличии большого числа запросов ресурсы сервера будут исчерпаны, и он перестанет нормально функционировать. Атаки такого вида получили название *SYN-ACK Flood*. Для этих атак киберпреступники используют *ботнеты* (*ботнет* — сеть из зараженных «зомби-компьютеров» и др.), состоящие из компьютеров, мобильных устройств обычных пользователей и даже устройств «умного дома», заражённых вредоносным программным обеспечением (вирусы, черви, трояны). После заражения достаточного числа сетевых устройств выдаётся команда на одновременную атаку целевого ресурса, выводя его из строя.

Такую атаку очень трудно обнаружить. Как правило, сам сервер не в состоянии отделить легитимный запрос от запроса злоумышленника, а учитывая огромный поток таких запросов, это не удается сделать и администратору. Таким образом, можно достаточно продолжительное время сохранять сервис недоступным. Для большого числа компаний сетевой ресурс играет ключевую роль в бизнесе [3], что позволяет недобро-

[1] ***Борисов Роман Сергеевич,*** *кандидат технических наук, доцент кафедры информационного права, информатики и математики Российского государственного университета правосудия, г. Москва, Российская Федерация.*

E-mail: bestseller@bk.ru

совестным конкурентам воспользоваться услугами киберпреступников, занимающимися *DDoS*-атаками [8].

На сегодняшний день существует ряд решений, которые способны покрыть риски отказа доступности сервиса, вызванные *DDoS*-атаками [10, 11], однако ввиду сложности и стоимости разработки, внедрения и обслуживания таких систем многие компании не могут их себе позволить. Это приводит к необходимости поиска и создания недорогих, простых и одновременно эффективных способов защиты от *DDoS*-атак. Реализованные в настоящее время способы защиты от популярных *ACK-SYN Flood*-атак основаны, как правило, на внесении корректировок в реализацию стека *TCP/IP* (сетевая модель взаимодействия *Transmission Control Protocol* / *Internet Protocol*) на сервере и обладают слабой продуктивностью распознавания из-за недостатков методик выбора показателей, позволяющих распознать атаку. Данная ситуация приводит к необходимости поиска новых подходов, позволяющих повысить качество идентификации вредоносных воздействий, уменьшающих вероятность ложных срабатываний для реализации процедур распознавания этого типа атак.

Общая постановка задачи

ACK-SYN Flood-атаки основаны на отправке большого количества пакетов *SYN* (*synchronize* — запрос клиента на подключение к серверу) с различных адресов, используя протокол *TCP*. Отправляемое количество пакетов превышает число максимально возможных открытых сокетов (программных интерфейсов обмена данными) клиента-жертвы, тем самым полностью парализуя его работу.

Большинство реальных потоков запросов к серверам можно отнести приближенно к пуассоновским[2], поскольку они обладают:

стационарностью — возможностью передачи k запросов в произвольный момент времени *t*;

отсутствием последствия — вероятность поступления k запросов после некоторого t_0 не связана с числом запросов, поступивших ранее, и временем их поступления, поэтому можно говорить о независимости числа запросов в непересекающиеся промежутки времени;

ординарностью — за незначительный период времени $\mathrm{d}T$ вероятность поступления более одного запроса крайне мала.

В этом случае представляется целесообразным описать взаимодействие пользователей с сервером в виде функционирования системы массового обслуживания с неограниченным числом обслуживающих приборов. На основе принятых допущений необходимо разработать *модель* достоверного распознавания атакующих запросов и их удаления из буфера веб-сервера.

Модель обнаружения атаки

Основными показателями обслуживающего прибора является время обработки одного запроса и, как следствие, его пропускная способность. Срок на обработку требований не имеет определенных значений, он является непостоянной величиной. Поэтому длительность обработки разных запросов одним прибором считается произвольной величиной с одинаковым законом распределения[3].

Определим основные параметры взаимодействия пользователей с сервером на основе модели системы массового обслуживания. К таким параметрам относится среднее число действующих устройств, возможные отказы, усредненное число заявок на обслуживание.

Среднее число действующих устройств можно описать следующим соотношением:

$$N = \sum_{k=1}^{n} k \cdot p_k = p_0 \sum_{k=1}^{n} \frac{\alpha^k}{(k-1)!} = \alpha(1 - p_n), \quad (1)$$

где N — число действующих устройств; $\alpha = \lambda / \mu$ — отношение интенсивности (λ) входного пуассоновского потока и интенсивности (μ) потока обслуживания; p_k — вероятность того, что системе в данный момент находится k заявок:

$$P_k = \frac{\frac{\alpha^n}{k!}}{\sum_{i=0}^{n} \frac{\alpha^i}{i!}} \quad (2)$$

В типовом режиме работы протокола серверы отвечают на пакет *SYN* путем отправки его же с добавлением *ACK*-флага и номера *ACK*. Поскольку входящие и исходящие пакеты фактически идентичны, можно говорить и об идентичности их потоков. Ресурсные мощности серверов, которые сохраняют параметры *TCP*-сессий, принимаем за множество приборов для обслуживания заявок. Обслуживание является способом зарезервировать определенные ресурсные мощности, пока устанавливается соединение либо пока не истечет заданный сервером таймаут.

Такая модель выявляет *SYN-ACK Flood*-атаки посредством резкого увеличения потока заявок. Когда сервер атакован, он использует ресурсы, занятые весь период заданного таймаута. В операционной системе этого временного периода хватает, чтобы заполнить все свободные серверные ресурсы с параметрическими данными *TCP*-сессии [1].

Рассмотрим детально ресурсные мощности серверов. Параметры TCP заложены в буфер, имеющий вид массива размера L, части которого служат хранилищем параметров. Их делят на три вида: установленные соединения, полуоткрытые соединения и свободные ячейки памяти (рис. 1). Допустим, что B — число открытых соединений. Тогда $n = L - B$ — число прочих

[2] *См., например*: Родионов В. В., Ловцов Д. А., Королев В. Т. Математика и информатика. Часть первая. Математика : учебное пособие / Под ред. Д.А. Ловцова. М. : РГУП, 2015. 246 с.

[3] Бегларян М.Е., Ващекин А.Н., Квачко В.Ю., Пичкуренко Е.А. Математика. Ч. 1. М. : РГУП, 2015. 184 с.

компонентов, общность которых примем за множество обслуживающих приборов.

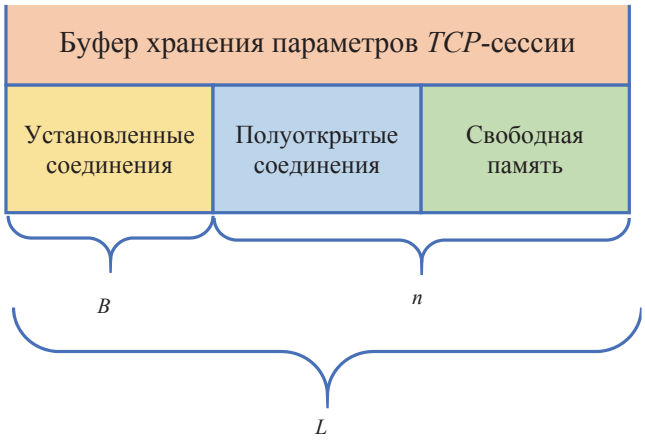

Рис. 1. Хранение параметров TCP-сессий в буфере

Если учесть, что соотношение показателя а интенсивности потоков и размеры L массивов зависимы, то выделяем *два* вида систем массового обслуживания. При интенсивности, намного меньшей мощности сервера, мы имеем дело с неограниченным количеством приборов. В противном случае данную систему следует рассматривать как систему с отказами. Поскольку в условиях штатного режима функционирования возможности сервера с большим запасом покроют все входящие запросы, целесообразно рассматривать *первый* вариант, т. е. систему с бесконечным числом обслуживающих приборов [14].

Для такой системы отношение интенсивности λ входного потока к среднему времени ($1/\mu$) обработки одной заявки будет определяться формулой $\alpha = \lambda / \mu$. Учитывая, что поток требований пуассоновский, вероятность количества (k) требований в системе можно выразить следующим образом:

$$p_k = \frac{\alpha^k e^{-\alpha}}{k!} \quad (3)$$

Объединив данное выражение с выражением (2), описывающим вероятность того, что в системе в данный момент времени находится *k* требований, получаем:

$$N = \sum_{k=1}^{\infty} k \cdot p_k = p_0 \sum_{k=1}^{\infty} \frac{\alpha^k}{(k-1)!} = \alpha(1 - p_\infty) \quad (4)$$

Следовательно,

$$p_\infty = \lim_{k\to\infty} \frac{\alpha^k e^{-\alpha}}{k!} = e^{-\alpha} \lim_{k\to\infty} \frac{\alpha^k}{k!} = 0 \quad (5)$$

Из выражений (4) и (5) для системы с бесконечным числом устройств получаем:

$$N = \alpha(1 - p_\infty) = \alpha(1 - 0) = \alpha \quad (6)$$

Данная модель может описать функционирование сервера в штатном порядке, максимальный уровень интенсивности входящих запросов и среднее время их обработки. Следует заметить, что в реальных условиях функционирования серверов необходимо учитывать вероятность потери пакетов в процессе передачи [2].

Чтобы улучшить модель, можно разделить ее на две подсистемы. *Первая* отвечает за те запросы, в которых полуоткрытые соединения установились, а *вторая* описывает ситуацию, когда запросы не получили соединения и были удалены.

Если считать, что протоколы TCP работают стабильно, то запросы, находящиеся в полуоткрытом состоянии, через время T_n должны быть удалены.

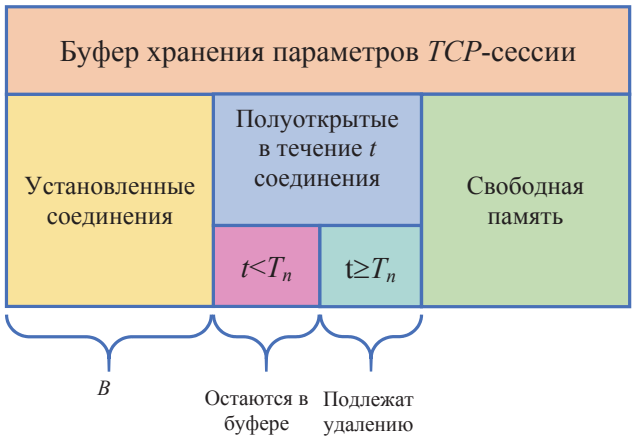

Рис. 2. Выявление атакующих соединений

Среднее значение числа полуоткрытых соединений можно определить, как:

$$N = s + l = \alpha_1 + \alpha_2 = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} \quad (7)$$

Из выражения (7) видно, что среднее число полуоткрытых соединений является случайным показателем, состоящим из двух слагаемых. *Первое* описывает среднее число полуоткрытых соединений, не являющихся сигналом начала кибератаки, а *второе* обозначает полуоткрытые неустановленные соединения, подлежащие удалению. Рост числа подобных заявок служит сигналом атаки на целевой ресурс.

Из дальнейшего рассмотрения уберём те пакеты *SYN+ACK*, для которых период ожидания не превышает некоторого порогового значения T_n. Будем считать эти пакеты легитимными до момента превышения порога.

Следует заметить, что в штатном режиме для каждой заявки может быть потерян один из пакетов *SYN+ACK* или *ACK*. Интенсивность потока этих заявок будет определяться как:

$$\lambda_2 = \lambda \cdot P_{no}, \quad (8)$$

где: λ — частота поступления *SYN*-пакетов на сетевой порт сервера; P_{no} — вероятность появления неустановленного полуоткрытого соединения с сервером.

Величина P_{no} в данном случае будет напрямую зависеть от пропускной способности и качества работы сети и будет определяться вероятностью потери пакета P_{nn}. Зависимость P_{no} от P_{nn} может быть определена следующим образом. Потерю *SYN+ACK* пакета обозна-

чим через событие A, а потерю *ACK*-пакета через B. Тогда вероятность события A будет равносильна вероятности потери пакета:

$$P(A) = P_{пп}, \quad (9)$$

Поскольку ответный *ACK*-пакет будет отправлен только после получения сервером *SYN+ACK* пакета (событие B наступает только в том случае, если не наступило событие A), то вероятность его наступления будет:

$$P(B) = P(\bar{A}) \cdot P_{пп} = (1 - P_{пп}) \cdot P_{пп}, \quad (10)$$

Обозначим через *C* событие появления полуоткрытого соединения. Наступление события *C* возможно по результату завершения событий *A* и *B*, следовательно:

$$P_{по} = P(C) = P(A + B) = P(A) + P(B) = \\ = P_{пп} + (1 - P_{пп}) \cdot P_{пп} = P_{пп} + P_{пп} - P_{пп}^2 = 2P_{пп} - P_{пп}^2. \quad (11)$$

Интенсивность потока требований второго порядка может быть получена из выражений (8) и (11):

$$\lambda_2 = \lambda \cdot P_{по} = \lambda \cdot (2P_{пп} - P_{пп}^2). \quad (12)$$

Поскольку современные операционные системы при установлении *TCP*-соединения формируют сразу несколько *SYN+ACK*-запросов до получения ответного *ACK*-пакета, примем за N_{SA} суммарное число отправленных *SYN+ACK* пакетов. Для целей обнаружения факта атаки значение имеет событие, когда ни на один из запросов *SYN+ACK* не придёт *ACK*-ответ от сервера. Следовательно, выражение (12) может быть записано следующим образом:

$$\lambda_2 = \lambda \cdot P_{по}^{N_{SA}} = \lambda \cdot (2P_{пп} - P_{пп}^2)^{N_{SA}}. \quad (13)$$

Поскольку интенсивность потока запросов прямо пропорциональна интенсивности исходного потока, то его также можно считать пуассоновским.

Второе слагаемое выражения (7) будет описывать среднее значение находящихся на обслуживании заявок:

$$l = \frac{\lambda_2}{\mu_2} = \frac{\lambda \cdot P_{пп}^{N_{SA}}}{\mu_2} = \frac{\lambda (2P_{пп} - P_{пп}^2)^{N_{SA}}}{\mu_2}, \quad (14)$$

где: μ_2 — заданный на сервере таймаут, необходимый для установления *TCP*-соединения; $P_{пп}$ — вероятность потери пакета в сети; N_{SA} — общее количество копий *SYN+ACK*-пакетов, отправленных операционной системой.

Выше было отмечено, что число занятых приборов подчиняется пуассоновскому распределению с параметром l. Вследствие пуассоновского распределения потока заявок математическое ожидание и дисперсия также будут равны величине l.

Примеры зависимости от числа n полуоткрытых соединений плотности $P(n)$ распределения случайной величины N по пуассоновскому закону (*рис.* 3) и закона $F(n)$ её распределения (*рис.* 4) приведены для $\lambda > 10$.

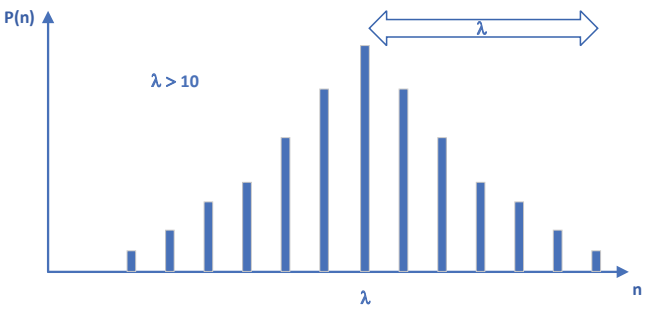

Рис. 3. Плотность распределения пуассоновской случайной величины при $\lambda > 10$

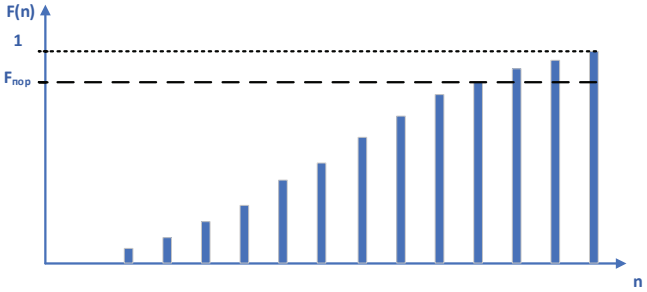

Рис. 4. Закон распределения пуассоновской случайной величины при $\lambda > 10$

Для выявления факта *SYN-ACK Flood*-атаки необходимо определение значения функции распределения:

$$F(n) = \sum_{i=1}^{n} p(i) \quad (15)$$

Определение факта SYN-ACK Flood-атаки можно зафиксировать при превышении значения функций распределения текущего состояния и числа полуоткрытых соединений определенного пикового значения $F_{\text{пор}}$ (см. *рис.* 4).

Фактические значения параметров межсетевого взаимодействия, необходимые для практической реализации предложенного подхода, достаточно просто получить средствами диагностики сети и в дальнейшем использовать в прикладных настраиваемых средствах защиты от *DDOS*-атак [13].

Получение исходных данных для модели

Эффективность использования предложенного подхода будет зависеть от качества определения фактических значений показателей сетевого взаимодействия. В число таких показателей входят: интенсивность заявок на обслуживание, вероятные потери пакетов и среднее время обработки заявок.

Рассмотрим возможные варианты оценки этих параметров.

Получение фактических значений интенсивности потока заявок на обслуживание является тривиальной

задачей[4] [12], которая может быть решена с помощью анализа лог-файлов веб-сервера за некоторый промежуток времени. Кроме того, для получения статистики можно использовать программное обеспечение для диагностики сети, работающее в режиме сетевого *сниффера* (от англ. *sniff* — нюхать), — аппаратное или программное средство для перехвата и анализа трафика. Для анализа может использоваться сам трафик или статистические подборки по нему [5]. Для получения данных непосредственно от маршрутизатора удобно использовать протокол *Netflow*, поддерживаемый рядом устройств. Протокол перенаправляет пакеты, удовлетворяющие определённым требованиям потока, что делает его пригодным для анализа текущей ситуации в сети и борьбы с *DDoS*-атаками. Протокол способен формировать таблицы, где в динамике показана вся статистика по потокам и пакетам: откуда поступили, куда отправляются, общее количество транзакций и др. Кроме того, информацию можно экспортировать на внешние ресурсы для обработки, что делает его весьма удобным.

Вероятные потери пакетов в сегменте сети могут быть оценены эмпирически с помощью утилиты ping, функционирующей на основе протокола *ICMP* (*Internet Control Message Protocol*). Утилита посылает *ICMP*-запросы, на которые получает ответы от узла, фиксируя необходимые нам параметры. Верный выбор набора тестируемых сетевых узлов позволит получить достоверные данные о фактических потерях пакетов определённого размера в исследуемых сегментах сети [12]. Значение вероятности потери пакетов может быть рассчитано как отношение числа запросов с просроченным таймаутом к общему числу сформированных запросов. При этом необходимо убрать из рассмотрения все узлы, от которых не пришло ни одного *ICMP*-ответа за время наблюдения [6].

Пороговое значение времени прохождения пакетов между узлами в стабильно работающих сетях можно найти либо с помощью эмпирических методов на основе наблюдения за реально функционирующей сетью, либо посредством формальной проверки гипотезы о законе распределения случайной величины, описывающей трафик. Эксперименты, проведенные с помощью утилиты ping с последующей проверкой гипотезы о соответствии полученного закона распределения времени обработки заявок пуассоновскому, показали, что в ряде случаев закон распределения не отличается от пуассоновского, а на некоторых интервалах предположительно представляет собой композицию нескольких распределений с различными параметрами. Это обстоятельство обусловливает необходимость дальнейших исследований по совершенствованию предложенного подхода.

Выводы

В работе рассмотрены вопросы построения математической модели для раннего обнаружения *SYN-ACK Flood*-атаки с ограничениями на число ложных срабатываний. Для решения использован математический аппарат теории массового обслуживания. В результате получены соотношения для определения пороговой интенсивности запросов на соединение, которую можно модифицировать в зависимости от параметров входящего трафика.

Практическая реализация предложенной модели предполагает режим тестирования и обучения системы при отсутствии *DDoS*-атак. Средства анализа исходных данных анализируют проходящий трафик и фиксируют сетевые параметры, соответствующие работе сервера при отсутствии вредоносных воздействий. При превышении числа полуоткрытых соединений порогового значения фиксируется факт начала *SYN-ACK Flood*-атаки. В этом случае сервер переходит в режим удаления всех полуоткрытых соединений, время существования которых превышает пороговое значение.

Это решение может использоваться для эффективного обнаружения *DDoS*-атак в составе систем обнаружения вторжений в организациях, деятельность которых зависит от качества функционирования веб-сервера (сайта) компании.

К числу достоинств такого подхода относятся возможность своевременного обнаружения атаки и способность подстраиваться к текущим показателям сети. Когда число потерянных пакетов превысит значение вычисленного значения вероятности этого события, это будет означать увеличение интенсивности подключений легитимных пользователей.

К недостаткам данного подхода можно отнести следующий. При выходе из строя сетевых устройств вероятность потери пакетов увеличится, что будет распознаваться как начало *SYN-ACK Flood*-атаки. Поэтому данный подход должен предусматривать дополнительные средства контроля исправности сетевого оборудования.

[4] Полтавский А.В., Федянина В.А., Скотченко А.С. Информационные и телекоммуникационные технологии в исследованиях: Компьютерный практикум. М. : Сам Полиграфист, 2020. 214 с.

Рецензент: ***Федосеев Сергей Витальевич,*** *кандидат технических наук, доцент, профессор кафедры информационного права, информатики и математики Российского государственного университета правосудия, г. Москва, Российская Федерация.*
E-mail: fedsergvit@mail.ru

Литература

1. Борисов Р.С., Черных А.М. Динамическая балансировка нагрузки гетерогенной вычислительной системы // Вестник компьютерных и информационных технологий. 2017. № 10 (160). С. 28—34.
2. Борисов Р.С., Ефименко А.А. Оптимизация размеров блоков элементарных заданий в задачах планирования параллельных вычислений // Прикладная информатика. 2018. Т. 13. № 3 (75). С. 77—82.
3. Ващекин А.Н., Ващекина И.В. Информационное право: прикладные задачи и математические методы // Информационное право. 2017. № 3. С. 17—21.
4. Ващекина И.В., Ващекин А.Н. Международные меры противодействия отмыванию нелегальных доходов пятого поколения — правовые условия укрепления безопасности финансового рынка // Вестник университета. 2021. № 1. С. 126—133. DOI: 10.26425/1816-4277-2021-1-126-133 .
5. Гончаров В.В., Гончаров А.В., Мишенина О.В. Моделирование обнаружения информационных атак на основе теории конечных автоматов // Правовая информатика. 2023. № 1. С. 41—51. DOI: 10.21681/1994-1404-2023-1-41-51 .
6. Ефименко А.А., Федосеев С.В. Организация инфраструктуры облачных вычислений на основе SDN сети // Экономика, статистика и информатика. Вестник УМО. 2013. № 5. С. 185—187.
7. Карпов Д.С., Ибрагимова З.А. Способы и средства обеспечения анонимности в глобальной сети Интернет // Правовая информатика. 2021. № 3. С. 60—67. DOI: 10.21681/1994-1404-2021-3-60-67 .
8. Ловцов Д.А. Информационная безопасность автоматизированных блокчейн-систем: угрозы и способы повышения // Труды II Межд. науч.-прак. конф. «Трансформация национальной социально-экономической системы России» (22 ноября 2019 г.) / РГУП. М. : РГУП, 2020. С. 464—473.
9. Ловцов Д.А. Информационная безопасность эргасистем: нетрадиционные угрозы, методы, модели // Информация и космос. 2009. № 4. С. 100—105.
10. Ловцов Д.А. Проблема гарантированного обеспечения информационной безопасности крупномасштабных автоматизированных систем // Правовая информатика. 2017. № 3. С. 66—74. DOI: 10.21681/1994-1404-2017-3-66-74 .
11. Ловцов Д.А. Теория защищенности информации в эргасистемах : монография. М. : РГУП, 2021. 276 с. ISBN 978-5-93916-896-0.
12. Свидетельство о государственной регистрации программы для ЭВМ № 2021665206 Российская Федерация. Infonetwork : № 2021664438: заявл. 15.09.2021: опубл. 21.09.2021 / Е.В. Царькова, Р.С. Борисов.
13. Свидетельство о государственной регистрации программы для ЭВМ № 2022661532 Российская Федерация. Модуль проверки полезной нагрузки пакета для обнаружения DoS-атак: № 2022660563: заявл. 09.06.2022: опубл. 22.06.2022 / Е.Р. Борисова, Р.С. Борисов, А.А. Ефименко.
14. Царькова Е.В. Оптимизационное моделирование // Менеджмент в России и за рубежом. 2020. № 5. С. 3—11.

DETECTING SYN-ACK FLOOD ATTACKS TO PREVENT AN OVERLOAD OF WEB SERVER RESOURCES

Roman Borisov, *Ph.D. (Technology), Associate Professor at the Department of Information Technology Law, Informatics and Mathematics of the Russian State University of Justice, Moscow, Russian Federation. E-mail: bestseller@bk.ru*

Keywords: *TCP session, SYN-ACK Flood, queueing theory, DDoS attack, Poisson distribution.*

Abstract

Purpose of the paper: developing a mathematical model for early detection of SYN-ACK Flood to improve the efficiency of using web server system resources when a DDoS attack is carried out against it.

Methods used: system and application analysis of ways of carrying out DDoS attacks and data transfer protocols, mathematical modelling of TCP protocol information flows based on the queueing systems theory.

Study findings: formal relationships are derived making it possible to identify SYN-ACK Flood attacks for taking preventive measures against a lack of system resources ('resource starvation') leading to service refusal for legitimate network users' requests.

References

1. Borisov R.S., Chernykh A.M. Dinamicheskaia balansirovka nagruzki geterogennoi vychislitel'noi sistemy. Vestnik komp'iuternykh i informatsionnykh tekhnologii, 2017, No. 10 (160), pp. 28–34.

2. Borisov R.S., Efimenko A.A. Optimizatsiia razmerov blokov elementarnykh zadanii v zadachakh planirovaniia parallel'nykh vychislenii. Prikladnaia informatika, 2018, t. 13, No. 3 (75), pp. 77–82.
3. Vashchekin A.N., Vashchekina I.V. Informatsionnoe pravo: prikladnye zadachi i matematicheskie metody. Informatsionnoe pravo, 2017, No. 3, pp. 17–21.
4. Vashchekina I.V., Vashchekin A.N. Mezhdunarodnye mery protivodeistviia otmyvaniiu nelegal'nykh dokhodov piatogo pokoleniia – pravovye usloviia ukrepleniia bezopasnosti finansovogo rynka. Vestnik universiteta, 2021, No. 1, pp. 126–133. DOI: 10.26425/1816-4277-2021-1-126-133 .
5. Goncharov V.V., Goncharov A.V., Mishenina O.V. Modelirovanie obnaruzheniia informatsionnykh atak na osnove teorii konechnykh avtomatov. Pravovaia informatika, 2023, No. 1, pp. 41–51. DOI: 10.21681/1994-1404-2023-1-41-51 .
6. Efimenko A.A., Fedoseev S.V. Organizatsiia infrastruktury oblachnykh vychislenii na osnove SDN seti. Ekonomika, statistika i informatika. Vestnik UMO, 2013, No. 5, pp. 185–187.
7. Karpov D.S., Ibragimova Z.A. Sposoby i sredstva obespecheniia anonimnosti v global'noi seti Internet. Pravovaia informatika, 2021, No. 3, pp. 60–67. DOI: 10.21681/1994-1404-2021-3-60-67 .
8. Lovtsov D.A. Informatsionnaia bezopasnost' avtomatizirovannykh blokchein-sistem: ugrozy i sposoby povysheniia. Trudy II Mezhd. nauch.-prak. konf. "Transformatsiia natsional'noi sotsial'no-ekonomicheskoi sistemy Rossii" (22 noiabria 2019 g.). RGUP. M. : RGUP, 2020, pp. 464–473.
9. Lovtsov D.A. Informatsionnaia bezopasnost' ergasistem: netraditsionnye ugrozy, metody, modeli. Informatsiia i kosmos, 2009, No. 4, pp. 100–105.
10. Lovtsov D.A. Problema garantirovannogo obespecheniia informatsionnoi bezopasnosti krupnomasshtabnykh avtomatizirovannykh sistem. Pravovaia informatika, 2017, No. 3, pp. 66–74. DOI: 10.21681/1994-1404-2017-3-66-74 .
11. Lovtsov D.A. Teoriia zashchishchennosti informatsii v ergasistemakh : monografiia. M. : RGUP, 2021. 276 pp. ISBN 978-5-93916-896-0.
12. Svidetel'stvo o gosudarstvennoi registratsii programmy dlia EVM No. 2021665206 Rossiiskaia Federatsiia. Infonetwork : No. 2021664438: zaiavl. 15.09.2021: opubl. 21.09.2021 / E.V. Tsar'kova, R.S. Borisov.
13. Svidetel'stvo o gosudarstvennoi registratsii programmy dlia EVM No. 2022661532 Rossiiskaia Federatsiia. Modul' proverki poleznoi nagruzki paketa dlia obnaruzheniia DoS-atak: No. 2022660563: zaiavl. 09.06.2022: opubl. 22.06.2022 / E.R. Borisova, R.S. Borisov, A.A. Efimenko.
14. Tsar'kova E.V. Optimizatsionnoe modelirovanie. Menedzhment v Rossii i za rubezhom, 2020, No. 5, pp. 3–11.