УДК 004.7

Л.С. Крамаров, Л.К. Бабенко

ОБНАРУЖЕНИЕ СЕТЕВЫХ АТАК И ВЫБОР КОНТРМЕР В ОБЛАЧНЫХ СИСТЕМАХ*

В связи с массовым переходом на облачные вычисления, безопасность в облаке является одним из наиболее важных вопросов, которые привлекли много научных исследований и разработок в последние несколько лет. В частности, злоумышленники могут изучить уязвимые места облачной системы и скомпрометировать виртуальные машины (VM) для развертывания дальнейших крупномасштабных «Распределенных атак отказ в обслуживании» (Distributed Denial of Service, DDoS). DDoS - атаки обычно включают действия на ранней стадии, такие как многоступенчатая эксплуатация уязвимостей, низкочастотное сканирование, использование VM в качестве зомби, и, наконец, DDoS- атаки через зомби VM. В облачных системах, особенно, таких как «Инфраструктура как услуга» (Infrastructure as a Service, IaaS), обнаружить предварительные действия для зомби атаки крайне сложно. Это связано с тем, что пользователи облака могут сами устанавливать уязвимые приложения. Для предотвращения компрометации VM, предлагаются многофазные распределенные детекторы уязвимостей, система мер, а также механизм для выбора контрмер, который строит граф атак на основе аналитических моделей и реконфигурируемой виртуальной сети. Предлагаемая структура использует протокол управления процессом обработки данных (OpenFlow) для программного конфигурирования сети. С его помощью реализуется мониторинг и контролируемая область поверх распределенных программируемых виртуальных коммутаторов, для улучшения обнаружения атак и смягчения их последствий.

Сетевая безопасность; облачные вычисления; граф атак; обнаружение атак.

L.S. Kramarov, L.K. Babenko

DETECTION OF NETWORK ATTACKS AND COUNTERMEASURE SELECTION IN CLOUD SYSTEMS

Cloud security is one of most important issues that has attracted a lot of research and development effort in past few years. Particularly, attackers can explore vulnerabilities of a cloud system and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS). DDoS attacks usually involve early stage actions such as multistep exploitation, low-frequency vulnerability scanning, and compromising identified vulnerable virtual machines as zombies, and finally DDoS attacks through the compromised zombies. Within the cloud system, especially the Infrastructure-as-a-Service (IaaS) clouds, the detection of zombie exploration attacks is extremely difficult. This is because cloud users may install vulnerable applications on their virtual machines. To prevent vulnerable virtual machines from being compromised in the cloud, we propose a multiphase distributed vulnerability detection, measurement, and countermeasure selection mechanism called NICE, which is built on attack graph-based analytical models and reconfigurable virtual network-based countermeasures. The proposed framework leverages OpenFlow network programming APIs to build a monitor and control plane over distributed programmable virtual switches to significantly improve attack detection and mitigate attack consequences. The system and security evaluations demonstrate the efficiency and effectiveness of the proposed solution.

Network security; cloud computing; intrusion detection; attack graph.

Цели, проблемы, задачи. Недавние исследования показали, что при переходе на облачные системы безопасность для пользователей является наиболее важным фактором. Исследования «Ассоциации по безопасности облачных вычисле-

* Работа выполнена при поддержке гранта РФФИ № 12-07-00037-а.

ний» (Cloud Security Alliance, CSA) показывают, что злоупотребление и ненадлежащее использование облачных ресурсов рассматриваются в качестве важнейших угроз безопасности, при которых злоумышленники могут использовать уязвимости в облаках, а так же ресурсы облачной системы для развертывания других атак. В обычных центрах обработки данных, системные администраторы имеют полный контроль над серверами, поэтому уязвимости могут быть обнаружены и исправлены централизованно. Тем не менее, исправления, устраняющие уязвимости в облачных центрах обработки данных, могут работать неэффективно и нарушать соглашение об уровне обслуживания (например, где пользователи имеют привилегии для управления программным обеспечением). Кроме того, пользователи облака могут устанавливать уязвимое программное обеспечение на своих VM, что существенно снижает безопасность. Задача состоит в том, чтобы эффективно обнаруживать уязвимости и атаки, создать систему реагирования при точном определении атак и свести к минимуму последствия нарушения безопасности в облаке.

В [1] Армбрус говорит, что защита от перебоев в обслуживании является одной из важнейших проблем в облачных системах. Здесь инфраструктура является общей, поэтому злоумышленники так же могут использовать ее для разворачивания более масштабных атак. В облачной среде атаки являются более эффективными, потому что пользователи обычно разделяют вычислительные ресурсы, например, будучи подключенными через некие коммутирующие устройства, они делят оперативную память и файловую систему, с потенциальным злоумышленником. Это справедливо и для методов виртуализации, виртуальных ОС, программного обеспечения, виртуальной сети и так далее. Все это привлекает злоумышленников атаковать и компрометировать множество VM.

В этой статье предлагается многоуровневая система обнаружения вторжений. Для лучшего обнаружения атак, используется сценарий графа атак (Scenario Attack Graph, SAG), включенный в аналитические процедуры обнаружения вторжений. Нужно отметить, что мы не намерены улучшить существующие алгоритмы обнаружения вторжений. Мы используем реконфигурирование виртуальной сети, позволяющее обнаруживать и противодействовать использованию зомби VM.

Предлагается использовать два основных этапа:

1. Развернуть легковесные сетевые системы обнаружения вторжений (Network Intrusion Detection System, NIDS) на каждом облачном сервере для анализа трафика. Периодически сканировать уязвимости виртуальной системы для построения сценария графа атак, а затем в зависимости от серьезности выявленных уязвимостей, решать ставить VM на полный анализ или нет.
2. После того как VM проходит полный анализ, можно провести реконфигурацию сети, чтобы сделать потенциальные атаки более заметными.

Предложенное решение значительно совершенствует имеющиеся NIDS и системы предотвращения вторжений (Intrusion Prevention System, IPS). Используя программируемые виртуальные сети, мы можем построить динамически реконфигурируемые IPS. Зеркалирование трафика методом программной коммутации позволяет минимизировать воздействие на трафик пользователей по сравнению с IPS/IDS, работающими на шлюзе или прокси. Программируемая виртуальная сетевая архитектура так-же позволяет создать облако инспекции и карантина для подозрительных VM в соответствии с текущим состоянием уязвимостей в SAG. На основе коллективного поведения VM в облаке можно определить действия для подозрительных VM, например, глубокий анализ или фильтрация трафика. Используя этот подход, не нужно блокировать транспортные потоки подозрительной VM на ранней стадии атаки. На текущем этапе разработано следующее:

- система представляет новую многофазовую распределенную сеть для обнаружения и предотвращения вторжений в рамках виртуальной сетевой среды;
- ПО для помещения в карантин и проверки подозрительных VM, чтобы можно было провести дальнейший анализ, не прерывая работу облака;
- используется новый подход с графом для обнаружения и предотвращения атак методом сопоставления поведения атаки, а так-же производится расчет и выбор эффективных контрмер.

Модель угроз. В модели атак предполагается, что злоумышленник может находиться как снаружи так и внутри виртуальной сети. Основной целью атакующего является компрометация VM и ее использование в качестве зомби. Модель защиты VM основана на обнаружении атак и реконфигурировании сети для улучшения устойчивости к атакам, и исследования зомби VM. Нужно отметить, что работа не связана с IPS, которые необходимо устанавливать на отдельных VM, и не обрабатывает зашифрованный трафик.

Данное решение можно применить к облачной инфраструктуре и предполагается, что провайдером сервисных услуг можно пренебречь. Также считается, что пользователи облачных сервисов могут устанавливать любые операционные системы и приложения, даже если такие действия могут привнести уязвимости в VM. Физическая безопасность облачных серверов выходит за рамки данной статьи. Предполагается, что гипервизор безопасен и без уязвимостей. Вопрос о пользователях, которые могут выходить из виртуального окружения и получать доступ к физическому серверу был изучен в работе [2] и выходит за рамки данной статьи.

Модель графа атак. Граф атак является инструментом моделирования, иллюстрирует все возможные многоуровневые, мультинаправленные пути атак, является ключевым моментом к пониманию угроз, а так-же определяет соответствующие контрмеры. В графе атак каждый узел представляет либо предварительное условие, либо последовательность эксплуатаций уязвимостей. Подобные действия не всегда представляют атаки, потому, что взаимодействия в нормальной работе протокола могут быть использованы для атак. Граф атак является полезным для выявления потенциальных и известных угроз в облачной системе.

Поскольку граф атак содержит полную информацию о всех известных уязвимостях в системе и информацию о соединениях, мы получаем полное представление текущей безопасности системы, где можно прогнозировать возможные угрозы и нападения, обнаруженные путем сопоставления событий и активности в сети. Если событие признано потенциальной атакой, мы можем применить конкретные меры, чтобы смягчить ее последствия или принять меры чтобы предотвратить заражение облачной системы. В результате таких действий мы расширяем обозначения логики графа атак MulVAL, как представлено в [3] и определяем сценарий графа атак.

Определние 1. SAG = (V, E), где

1. $V = Nc \cup Nd \cup Nr$ обозначают множество вершин, включающих три типа, а именно Nc – эксплуатируемая уязвимость, Nd – результат эксплуатации уязвимости и Nr – начальный шаг сценария атак.

2. $E = Epre \cup Epost$ обозначает множество ориентированных ребер. Ребро $e \in Epre \subseteq Nd \times Nc$ предполагает, что Nd удовлетворяет достижимости Nc. Ребро $e \in Epost \subseteq Nc \times Nd$ означает, что последовательность Nd может быть получена, если Nc будет выполнено.

Узел $vc \in Nc$ определен как три последовательности (Hosts; vul; alert) представляющие набор IP адресов, информацию об уязвимости в общей базе уязвимостей (Common Vulnerabilities and Exposures, CVE), и предупреждения связанные с Vc. Nd действует как логика ИЛИ и включает результаты действий. Nr представляет корневой узел SAG.

Для соотношения оповещений мы ссылаемся на метод, описанный в [3] и определяем новый граф взаимосвязанных оповещений (ACG) для отображения оповещений ACG в соответствующие узлы SAG. Чтобы сохранять прогресс атаки, мы отслеживаем IP адреса источника и назначения.

Определение 2. ACG = (A, E, P), где

1. A – набор агрегированных оповещений. Оповещение a ϵ A есть данные со структурой (src, dst, cls, ts) представляющее IP адреса источника, назначения, тип оповещения и его временную метку.
2. Каждое оповещение отображается на пару вершин (Vc, Vd) в SAG, используя функцию map(a), т.е. map(a):

$$a \rightarrow \{(Vc, Vd) \mid (a.\mathrm{src} \in Vc.\mathrm{Hosts}) \wedge (a.\mathrm{dst} \in Vd.\mathrm{Hosts}) \wedge (a.\mathrm{cls} = Vc.\mathrm{vul})\}$$

3. E – набор ориентированных ребер представляющий соотношение между двумя оповещениями (a, a') если выполняется условие:
   a. $(a.\mathrm{ts} < a'.\mathrm{ts}) \wedge (a'.\mathrm{ts} - a.\mathrm{ts} < \text{порог})$
   b. $\exists (Vd, Vc) \in \mathrm{Epre}: (a.\mathrm{dst} \in Vd.\mathrm{Hosts} \wedge a'.\mathrm{src} \in Vc.\mathrm{Hosts})$.

3. P – набор путей в ACG. Путь Si ⊃ P представляет набор связанных оповещений в хронологическом порядке.

Мы предполагаем, что A содержит обработанные оповещения. Необработанные оповещения, имеющие те же адреса источника, назначения, типа атаки и метки времени в пределах указанной области считаются как мета оповещения. Каждая упорядоченная пара (a, a') в ACG отображается на две соседних в SAG с отметкой времени отличной от пороговой. ACG представляет оповещения в хронологическом порядке, поэтому для него в сценариях атак есть соответствующие оповещения. Множество P используется для хранения всех путей от корневых до целевых оповещений в SAG и каждый путь Si ⊃ P представляет предупреждения, принадлежащие к тому же сценарию атак.

SAG и ACG используются вместе, чтобы прогнозировать поведение злоумышленника. Предупреждение взаимосвязанного алгоритма следует для каждого обнаруженного оповещения и возвращает один или несколько путей Si. Каждое оповещение ac, которое было получено с IDS добавляется в ACG, если его там не было. Для этого новые оповещения ac, соответствующие вершинам в SAG обнаруживаются функцией map(ac). Для этой вершины в SAG, оповещения связанные с его родительской вершиной типа Nc, объединяются с текущим оповещением ac. Это создает новый набор оповещений, которые принадлежат к пути Si в ACG или новый разделенный путь Si + 1 из подмножества Si перед оповещением с добавленным ac к Si + 1. В конце этого алгоритма идентификатор ac будет добавлен к атрибутам вершины оповещения в SAG. Первый алгоритм возвращает набор путей атаки S в ACG.

Модель защиты виртуальных машин. Модель защиты VM состоит из индексатора безопасности и монитора состояний. Мы указываем индекс безопасности для всех VM сети в зависимости от различных факторов, таких как соединения, количество предоставленных уязвимостей и их оценка воздействия. Оценка уязвимости, определенная в общей системе оценки уязвимостей [4] (Common Vulnerability Scoring System, CVSS) позволяет судить о конфиденциальности, целостности и доступности после эксплуатации этой уязвимости. Подключение метрики VM решается путем оценки входящих и исходящих соединений.

Определение 3. Состояние виртуальной машины основано на информации, собранной с контроллера сети, и может принимать следующие значения:

1. Стабильное. На этих виртуальных машинах нет известных уязвимостей.
2. Уязвимое. Наличие одной или нескольких уязвимостей, которые не использованы.

3. Эксплуатируемое. Хотя бы одна уязвимость была использована и VM находится под угрозой.
4. Зомби. VM находится под контролем злоумышленника.

Измерение безопасности, смягчение действий атак и контрмеры. Здесь представлен выбор контрмер для полученного сценария атак. При обнаружении уязвимостей или подозрительных VM, некоторые контрмеры могут быть применены для ограничения возможностей злоумышленника. Контрмеры служат для защиты VM от компрометации и делают более заметным поведение атакующего для идентификации его действий.

Измерение метрик безопасности. Среди различных подходов, использование графа атак в качестве метрической модели безопасности для оценки рисков безопасности является хорошим выбором. Для оценки состояния и вероятности рисков сетевой безопасности текущей конфигурации сети, необходимы метрики безопасности в графе атак. После чего граф атак строит информацию об уязвимостях. Для внутреннего или внешнего узла, выбирается приоритетная вероятность равная вероятности угрозы, источник становится активным и препятствует эксплуатации уязвимости. Мы используем Gv, чтобы обозначить приоритет вероятности риска для корневого узла графа и обычно значению Gv присваивается высокая вероятность, например от 0,7 – 1.

Для использованного внутреннего узла, каждый шаг атаки есть узел e ϵ Nc и имеет вероятность эксплуатации уязвимости обозначенную как Gm[e]. Gm[e] назначается в зависимости от базовой метки CVSS. Базовая метка вычисляется на основе влияния и эксплуатационного фактора уязвимости. Базовая метка может быть получена непосредственно из национальной базы уязвимостей [5] (National Vulnerability Database, NVD) по идентификатору CVE

$$BS = (0.6 \times \text{IV} + 0.4 \times \text{E} - 1.5) \times \text{f(IV)}, \quad (1)$$

где $IV = 10.41 \times (1 - (1 - \text{C}) \times (1 - \text{I}) \times (1 - \text{A}))$,

$$E = 20 \times \text{AC} \times \text{AU} \times \text{AV}, \text{ и } f(IV) = \begin{cases} 0 \; if \; IV = 0; \\ else \; 1.176. \end{cases}$$

Величина воздействия IV вычисляется из трех основных параметров безопасности, а именно конфиденциальность (C), целостность (I) и доступность (A). Оценка уязвимости (E) состоит из вектора доступа (AV), сложности доступа (AC) и экземпляра аутентификации (AU). Значение базовой метки в диапазоне от 0 до 10. В графе атак присваиваем каждому внутреннему узлу с его базовой меткой значение, деленное на 10:

$$Gm[e] = \Pr(e = T) = BS(e)/10, \forall_e \in Nc. \quad (2)$$

В графе атак отношения между эксплуатируемыми атаками могут быть связывающими или разделяющими согласно своим условиям зависимости. Такие отношения могут быть представлены в виде условных вероятностей, когда вероятностный риск текущего узла определяется связью с его предшественниками и их вероятностными рисками. Мы предлагаем следующий вывод вероятности:

- Для любого шага атаки, узел n ∈ Nc с непосредственными предшественниками устанавливает W = parent(n)

$$\Pr(n|W) = Gm[n] \times \prod_{s \in W} \Pr(s|W); \quad (3)$$

- Для любой части узла n ∈ Nd с непосредственными предшественниками W = parent(n), и затем

$$\Pr(n|W) = 1 - \prod_{s \in W}(1 - \Pr(s|W)). \quad (4)$$

Условные вероятности назначены всем внутренним узлам в SAG и можно объединить значения рисков от всех предшественников для получения кумулятивной вероятности риска или абсолютной вероятности риска для каждого узла в соответствии с (5) и (6). На основе распределения условных вероятностей на каждом узле мы можем получить эффективную безопасность усиления плана или стратегии смягчения последствий.

- Для любого шага атаки, узел n ∈ Nc с непосредственными предшественниками устанавливает W = parent(n)

$$\Pr(n) = \Pr(n|W) \times \prod_{s \in W} \Pr(s); \quad (5)$$

- Для любой части узла n ∈ Nd с непосредственными предшественниками W = parent(n), и затем

$$\Pr(n) = 1 - \prod_{s \in W}(1 - \Pr(s)). \quad (6)$$

Стратегии смягчения последствий. Стратегии смягчения последствий основаны на метках безопасности, которые были определенны в предыдущем подразделе. Появляется возможность создания смягчающей стратегии в ответ на обнаруженные сигналы. Так же мы определяем термин область контрмер:

Определение 4. Область контрмер CM = {cm1, cm2,…cmn} есть набор контрмер. Каждая cm ∈ CM, как последовательность cm = (стоимость, изменение, состояние, эффективность), где:

1. Стоимость блока, описывающего расходы необходимые для применения контрмер в плане ресурсов и сложности эксплуатации, определяется в диапазоне от 1 до 5, более высокая метрика означает более высокую стоимость.
2. Изменение отрицательный эффект контрмер привносимый в SLA, и его значение колеблется от незначительного 1 к более значительному 5. Если контрмеры не имеют воздействия на SLA значение изменения равно 0.
3. Условие, является требованием для соответствующих контрмер.
4. Эффективность это процентная вероятность изменения узла, для которого эта контрмера применена.

В общем случае, есть множество контрмер, которые могут быть применены к облачной виртуальной сети в зависимости от доступных методов. Несколько общих контрмер для виртуальных сетей приведены в табл. 1.

Таблица 1

Возможные типы контрмер

№	Контрмера	Стоимость
1	Перенаправление трафика	3
2	Изоляция трафика	2
3	Глубокий анализ трафика	3
4	Создание правил фильтрации	2
5	Изменение MAC адреса	1
6	Изменение IP адреса	1
7	Блокировка порта	1
8	Программное исправление	4
9	Карантин	2
10	Реконфигурация сети	5
11	Изменение топологии сети	5

Стратегия изменения конфигурации сети в основном связана с действиями на втором и третьем уровнях сетевой модели OSI. На втором уровне, виртуальные мосты и виртуальные локальные сети, являющиеся основным компонентом для соединения двух VM в облачной виртуальной сети. Виртуальные мост является объектом, который соединяет VIFs (виртуальные интерфейсы). VM на разных мостах изолированы на втором уровне. VIFs на одном виртуальном мосту, но с разными тегами VLAN не могут общаться друг с другом напрямую. Основываясь на изоляции второго уровня, можно организовать реконфигурацию сети, для изоляции подозрительных VM. В результате этой контрмеры исключаются пути атак в графе, заставляя атакующего исследовать альтернативные пути. Третий уровень

реконфигурации это еще один способ исключить путь атаки. Через контроллер сети на каждом (Open Flow Switch OFS) или (Open Virtual Switch, OVS) может быть модифицирована таблица потоков для изменения топологии сети.

Следует отметить, подход реконфигурирования виртуальной сети на нижнем уровне имеет преимущество в том, что на верхних уровнях приложения будут испытывать минимальное воздействие. В частности такой подход возможен только при использовании программного коммутатора для автоматической реконфигурации в динамической среде. Такая контрмера, как изоляция трафика может быть реализована на OVS и OFS, ограничивая и перенаправляя подозрительные потоки. Когда в виртуальной сети обнаруживается подозрительная активность или сканирование портов, важно определить являются ли эти действия злонамеренными. В такой ситуации, изменение конфигурации сети заставит злоумышленника выполнить больше исследований, что в свою очередь сделает его поведение более заметным.

Выбор контрмер. Выбор контрмер описан следующим алгоритмом. На вход подается оповещение, граф атак G и область контрмер CM. Выбирается узел Valert, соответствующий порожденному оповещению. Прежде чем выбрать контрмеры мы рассчитываем расстояние от Valert до узла назначения, если расстояние больше чем пороговое значение, то выбор контрмер не выполняется, но обновляется ACG для отслеживания оповещений в системе. Для узла источника Valert, все достижимые узлы (включая исходный узел) собраны в набор T. Поскольку сигнал тревоги генерируется только после того, как злоумышленник выполнил действия, устанавливается вероятность Valert в 1 и вычисляются новые вероятности для всех его дочерних узлов в множестве T. Теперь для всех $t \in T$ применимы выбранные из CM контрмеры, и новые вероятности рассчитываются по эффективности выбранных контрмер. Изменение вероятности целевого узла дает преимущество для прикладного использования контрмеры. Следующим этапом вычисляется возврат инвестиций для каждой примененной контрмеры. Примененные к узлу контрмеры, дающие наименьшее значение возвращаемых инвестиций, считаются оптимальными. Наконец SAG и ACG так же обновляются перед завершением алгоритма. Сложность алгоритма $\mathcal{O}(|V \times |CM|)$, где |V| номер уязвимости и |CM| номер контрмеры.

Выводы. Мы предоставили комплексный подход для обнаружения и смягчения последствий атак в облачной виртуальной сети. Подход использует граф атак для обнаружения и прогнозирования атак, а так же возможности программирования виртуальных сетевых устройств, для обнаружения и защиты VM.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Armbrust M., Fox A., Griffith R.* A View of Cloud Computing // ACM Comm. – Apr. 2010. – Vol. 53, № 4. – C. 50-58.
2. *Keller E., Szefer J., Rexford J., Lee R. B.* NoHype: Virtualized Cloud Infrastructure without the Virtualization. Proc. 37th ACM Ann. Int'l Symp. Computer Architecture (ISCA '10), June 2010. – P. 350-361.
3. *Roy A., Kim D.S., Trivedi K.* Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees. Proc. IEEE Int'l Conf. Dependable Systems Networks (DSN '12), June 2012.
4. *Mell P., Scarfone K., Romanosky S.* Common Vulnerability Scoring System (CVSS)," http://www.first.org/cvss/cvss-guide.html, May 2010.
5. National Institute of Standards and Technology, "National Vulnerability Database, NVD," http://nvd.nist. gov, 2012.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Бабенко Людмила Климентьевна – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: blk@tsure.ru; 347928, г. Таганрог, ул. Чехова, 2, корпус "И"; тел.: 88634312018; кафедра безопасности информационных технологий; профессор.

Крамаров Леонид Сергеевич – e-mail: l.s.kramarov@gmail.com; кафедра безопасности информационных технологий; аспирант.

Babenko Lyudmila Klimentevna – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: blk@tsure.ru; Block “I”, 2, Chehov street, Taganrog, 347928, Russia; phone: +78634312018; the department of security of information technologies; professor.

Kramarov Leonid Sergeevich – e-mail: l.s.kramarov@gmail.com; the department of security of information technologies; postgraduate student.

УДК 004.056

Я.В. Тарасов, О.Б. Макаревич

МОДЕЛИРОВАНИЕ И ИССЛЕДОВАНИЕ НИЗКОИНТЕНСИВНЫХ DOS-АТАК НА BGP-ИНФРАСТРУКТУРУ*

Представлены результаты анализа и имитационного моделирования исследования атак типа «отказ в обслуживании» на различные сервисы хранения, обработки и передачи данных в сети Интернет. Основное внимание уделено имитации низкоинтенсивных DoS-атак на инфраструктуру глобального протокола динамической маршрутизации BGP (Border Gateway Protocol). В качестве программного обеспечения были выбраны apache2, nginx, quagga, debian, vmware и citrix xenserver с использованием самых популярных вариантов конфигурации. В некоторых случаях специально модифицировались конфигурационные параметры для снижения безопасности служб, что бы получить заранее предполагаемые результаты для сравнения. В рамках исследования были проанализированы уязвимости протоколов передачи данных транспортного и прикладного уровня, приводящие к возможности реализации угрозы низкоинтенсивных DoS-атак. Данное исследование должны наглядно продемонстрировать способы реализации атак данного типа на реальные вычислительные системы и вычислительные сети. Результатом проведения исследования является оценка критичности низкоинтенсивных атак типа отказ в обслуживании на инфраструктуру BGP.

Ннизкоинтенсивные DoS-атаки; Border Gateway Protocol; моделирование атак; безопасность вычислительных сетей.

Y.V. Tarasov, O.B. Makarevich

MODELING AND STUDY OF LOW-INTENSITY DOS-ATTACKS ON BGP-INFRASTRUCTURE

The results of the analysis and simulation study of attacks such as " denial of service " for a variety of services that store, process and transmit data to the Internet. Focuses on the simulation of low-intensity DoS-attacks on the infrastructure of the global dynamic routing protocol BGP (Border Gateway Protocol). As the software was chosen apache2, nginx, quagga, debian, vmware and citrix xenserver using the most popular configuration options . In some cases, specially modified configuration parameters to reduce the security services, which would get pre- expected results for comparison. The study analyzed the vulnerability of data communication protocols of the

* Работа выполнена при поддержке грантов РФФИ № 12-07-00013-а, РФФИ № 12-07-00014-а.