# МАСШТАБИРУЕМОЕ HONEYPOT-РЕШЕНИЕ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В КОРПОРАТИВНЫХ СЕТЯХ

**А.В. Красов**[1], **Р.Б. Петрив**[1*], **Д.В. Сахаров**[1], **Н.Л. Сторожук**[1], **И.А. Ушаков**[1]

[1]Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация
*Адрес для переписки: roman.petriv@mail.ru

**Информация о статье**
УДК 004.056
Статья поступила в редакцию 26.07.2019

**Ссылка для цитирования:** Красов А.В., Петрив Р.Б., Сахаров Д.В., Сторожук Н.Л., Ушаков И.А. Масштабируемое Honeypot-решение для обеспечения безопасности в корпоративных сетях // Труды учебных заведений связи. 2019. Т. 5. № 3. С. 86–97. DOI:10.31854/1813-324X-2019-5-3-86-97

**Аннотация:** *В статье анализируются тенденции в области применения технологий защиты сетей, использующих Honeypot-решения для выявления и исследования поведения нарушителей в целях выработки мер противодействия атакам. Предложено масштабируемое решение, протестированное на исследовательском стенде на основе технологий Microsoft Azure. Выполнен стресс-тест предложенного решения с использованием DDoS-атаки.*

**Ключевые слова:** *корпоративные сети, облачные вычисления, Deception Technologies (технологии «обманки»), Honeypot-решение, безопасность инфокоммуникационных сетей, киберугрозы, DDoS-атака.*

## Введение

Несмотря на существенное развитие и накопленный опыт (так называемый Best Practice) применения технологий безопасности инфокоммуникационных сетей, наблюдается устойчивый рост количества, разнообразия и результативности кибератак. Многие крупные компании (Yahoo, Uber, Equifax) за последние три года столкнулись как с серьезными утечками данных, так и нарушениями в работе инфраструктуры вследствие злонамеренных действий. Согласно статистике [1], в первой половине 2019 г. только вредоносных приложений для мобильных устройств блокировалось ежедневно в среднем более 24 000, и их количество имеет тенденцию к росту. Также наблюдается устойчивый рост числа атак на IoT-инфраструктуру (более 600 % в 2017 г., 400–500 % по разным оценкам в 2018 г.), атак с помощью программ-вымогателей (в т. ч. шифровальщиков – WannaCry, Petya и др).

Масштабы кибератак продолжают расти, создавая угрозу конфиденциальности, целостности и доступности информационных активов, а также репутации организаций. «Ландшафт угроз» [2], вне всякого сомнения, станет более сложным, а атаки более целенаправленными, ориентированными на уязвимые места конкретной цели с реализацией обхода имеющихся средств защиты [3, 4].

Уже несколько лет преступные синдикаты работают подобно стартапам. Как другие успешные стартапы, они становятся более зрелыми и расширяют свои возможности. У киберпреступников уже давно есть заказчики, организована кооперация и специализация в различных областях: от написания вредоносного ПО до хостинга, тестирования, оказания курьерских услуг и т. д.

Вредоносные программы также совершенствуются их разработчиками и уже способны обходить элементы машинного обучения в развернутых системах киберзащиты [5]. Следующим шагом может стать использование передовых инструментов машинного обучения для автоматизации выбора хакерами своих целей, их наименее защищенных и оптимальных для реализации кибератак участков.

Растущее количество и разнообразие угроз требует постоянного совершенствования мер защиты, разработки и использования новых механизмов и методов противодействия кибератакам. Актуальной является задача разработки гибких унифицированных решений. Безусловно, выработка эффективных решений в значительной степени зависит от того, насколько исследованы и могут быть спрогнозированы действия нарушителей.

Анализ инцидентов безопасности является чрезвычайно важным источником информации для формирования мер противодействия, однако не может в достаточной мере обеспечивать своевременность, адекватность и эффективность (в том числе экономическую) мер противодействия, поскольку строится на изучении произошедших событий.

В случае успешной кибератаки нарушителя она может быть расследована с получением максимально полных данных о ее характере и способе проведения, но существенный ущерб уже может быть необратим. В случае своевременного реагирования и предотвращения атаки ущерба удается избежать, однако получить более подробную информацию о нарушителе и его потенциале крайне сложно. Решением данной задачи может быть использование приемов и методов отвлечения нарушителя на ложные цели, то есть использование так называемых *Deception Technologies* (технологий «обманки»), направленных на введение в заблуждение нарушителя.

**Назначение и перспективность применения технологий «обманки» в корпоративных сетях**

Большинство современных центров защиты информации оснащены технологиями, фиксирующими инциденты безопасности [6–9]. Но современная ситуация с защитой периметра сети все чаще рассматривается специалистами как вопрос времени. Отмечается тенденция, при которой злоумышленники обнаруживаются спустя все более существенное время нахождения внутри сети. Однако недостаточно найти и удалить злоумышленника из сети, и на основе анализа его поведения выработать решения. Следует более полно изучить противника, чтобы понять его, а для этого нужно предоставить ему больше времени для совершения злонамеренных действий.

Поскольку его действия несут все в себе угрозу, необходимо принять меры, позволяющие специалистам по безопасности изучать злоумышленника в воспринимаемой им как естественной среде (оставаясь при этом незамеченными для него), то есть в имитированной среде, где невозможно нанесение реального ущерба [10–11]. При этом следует как можно дольше удерживать злоумышленника в имитированной среде для изучения его действий, что может достигаться использованием множества технологий «обманки», которые выглядят весьма естественно, но являются приманками, содержат скрытые механизмы оповещения и др. [12].

Распределение ловушек и приманок целесообразно осуществлять по всей сети и на конечных точках, которые представляются информационными активами организации. Как только злоумышленник подключается к «среде обмана», выдается предупреждение с высокой точностью, что дает организации возможность принять решение – либо быстро исключить его из сети, либо изучить его методы и поведение в контролируемой среде. Доказано, что использование технологий «обманки» резко сокращает время пребывания злоумышленника в среде до того, как он будет обнаружен средствами контроля безопасности [10].

Одним из эффективных направлений технологий «обманки» является использование Honeypot, так называемых «медовых» ловушек, которые привлекательно выглядят для киберпреступника, но в реальности чувствительных данных не содержат, а только имитируют их наличие и возможность доступа. Иными словами, «обманка» с помощью Honeypot – проактивная защита, которая делает атаку более трудной для выполнения, и в то же время делает возможным нанесение ответного удара по злоумышленнику. В процессе взаимодействия с Honeypot атакующий раскрывает свои приемы, средства и возможности; может быть идентифицирован в дальнейшем при анализе массивов данных об аналогичных инцидентах [11]. Успешность применения технологий Honeypot-«обманки» в значительной степени зависит от того, насколько реалистично активы сети выглядят при попытке доступа извне, а также от того, насколько эффективно реализованы механизмы наблюдения и анализа происходящих в них событий.

В последнее время ряд исследователей склонны использовать термин Honeypot для обозначения совокупности использующих данный подход технологий, как концептуальное направление для различных прикладных решений.

**Концепция Honeypot: решения, эволюция, противодействие, требования**

В настоящее время абсолютно универсального решения, которое использует концепцию Honeypot, не существует. Если рассматривать ее как имитационную модель реального вычислительного процесса, то к любому программно-аппаратному решению на базе этой концепции применимо ограничение Тьюринга. Последнее требует применения субоптимальных (рациональных) подходов к реализации Honeypot-решений в контексте более конкретной задачи. Так появились решения на базе Honeypot, которые разделяются по объектам, категориям и области применения в сети (таблица 1).

Таблица отражает типичные технологические Honeypot-решения, составлена на основе анализа их функциональных возможностей и результатов тестирования, описанных в литературе с учетом специфики задач, и соответственно, имитируемых объектов. Функциональные возможности Honeypot-продуктов могут расширяться, области применения пересекаться, в т. ч. вследствие специфики сетей.

**ТАБЛИЦА 1. Обзор Honeypot-«обманок» по областям применения [13]**

| Технологические решения | Объект «обманки» | Категория | Хост | Сеть |
|---|---|---|---|---|
| Fake Honeypot | Honeypot | Server | ✗ | ✓ |
| Honeyentries | Table, data set | Database | ✓ | ✗ |
| MTD | Topo., net. interf., memory, arch. | Versatile | ✓ | ✓ |
| Honeyword | Password | Authentica-tion | ✓ | ✗ |
| Honeyaccount | User account | Authentica-tion | ✓ | ✗ |
| Honeyfile | (Cloud-)File | File system | ✓ | ✓ |
| Honeypatch | Vulnerability | Server | ✓ | ✓ |
| – | Memory | Server | ✓ | ✗ |
| – | Metadata | File | ✓ | ✗ |
| HoneyURL | URL | File | ✗ | ✓ |
| Honeymail | E-Mail adress | File | ✗ | ✓ |
| Honeypeople | Social network profile | File | ✗ | ✗ |
| Honeyport | Network port | Server | ✗ | ✓ |
| Decep. web server | Error codes, Robot.txt | Server | ✗ | ✓ |
| OS interf. | System call | Server | ✓ | ✗ |

Изначально Honeypot развивались, чтобы соответствовать наиболее актуальным угрозам, которые прогнозировались с учетом данных об инцидентах безопасности, и ориентировались в большей степени на узкие прикладные задачи конкретных сетей. По мере усложнения сетей и вычислительных систем, увеличению числа возможных угроз, усложнились задачи и для Honeypot. Их разработка стала циклическим процессом, так как требовала постоянного совершенствования для все более эффективного привлечения злоумышленников, используемых ими инструментов и вредоносных программ.

Вредоносные программы, захваченные на Honeypot, анализируются ретроспективно с целью последующей переработки приманки. С этой точки зрения вклад Honeypot в безопасность считается реактивным, если имитация реального объекта проста и предполагает только регистрацию действий, т. е. выявление не очень квалифицированного нарушителя. Простые Honeypot имитируют поведение реальных систем, однако даже беглый взгляд, например, на процесс и результаты сканирования, позволяет более квалифицированному нарушителю увидеть, что цель имеет аномальное количество открытых портов со службой удаленного управления (рисунок 1).

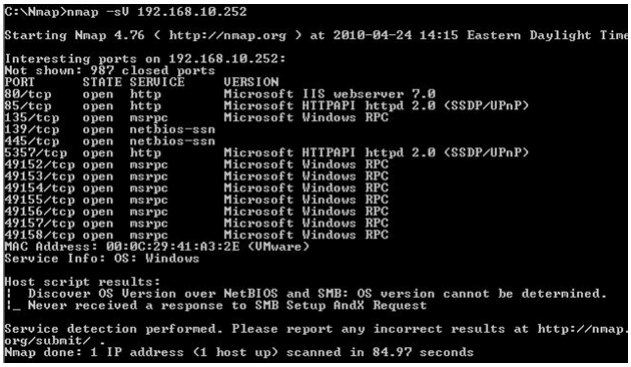


**Рис. 1. Реакция Honeypot, имитирующая открытые порты MS Windows при сканировании с помощью nmap**

Следующим этапом развития Honeypot стала более реальная имитация на основе паттернов реакции реальных систем. Но с началом использования в сетях Honeypot и развития решений на базе этой концепции, злоумышленниками стали разрабатываться и применяться инструментальные средства, основанные на различных методах детектирования (таблица 2, где mitigation – способ снижения эффективности метода детектирования).

**ТАБЛИЦА 2. Обзор методов противодействия Honeypot [13]**

| Метод детектирования | Содержание, детали | Мишень | Mitigation |
|---|---|---|---|
| Временное поведение | Измерение RTT, чтобы выявить корреляции между IP-адресами | honeyed, virtual honeypots | Имитация времени поведения |
| Stack fingerprinting | Отправка поврежденных пакетов и анализ ответов | Имитация стеков связи | Реализация полного TCP / IP-стека |
| Функциональное зондирование | Использование предоставленных функций и проверка статуса | SMTP и DNS | Реализация полной функциональности |
| Поведение системных вызовов | Аномалии во временном поведении или локализации памяти | Система Linux | Имитация времени поведения, KASLR |
| Сетевой трафик | Анализ трафика сети RX и TX, например, количество байтов | Сетевая фильтрация данных, например, Sebek | Hinder network monitoring, VMI, Proxy |
| Обнаружение UML | Вывод dmesg, сетевое устройство, / proc /, структура памяти | UML-изоляция хоста | Инструменты манипуляции для показа информации |
| Обнаружение VMware | Аппаратное обеспечение, например, MAC-адрес, вход / выход бэкдор | Изоляция хоста на основе VMware | Настройка оборудования, патч I/O бэкдор |
| Обнаружение отладчика | Использование функции ptrace (), функции IsDebuggerPresent () или поиска в памяти для 0xCC | e.g. Cuckoo | – |
| Семантический разрыв | Управление структурой данных ядра | VMI | – |
| Кастомизация | Поиск строк по умолчанию | – | Настройка системы |

В целом методы противодействия, основанные на концепции Honeypot-решениям, вырабатываются на базе выделения и анализа недостатков в моделировании сетевого объекта, выявления более примитивного в сравнении с реальными объектами поведения. Многие инструменты для создания Honeypot представляют собой ПО с открытым кодом, что облегчает их изучение злоумышленниками.

Honeypot-решения, как и технологии «обманки» в целом, в значительной степени основываются на человеческой психологии, поэтому единственным способом противостоять их обнаружению является развитие Honeypot по пути усложнения поведения, использования гибких масштабируемых конфигураций, которые могут быть построены на основе функционала и инструментов с открытым кодом.

Honeypot-решения в этом смысле должны быть в большей степени проактивны, максимально правдоподобно взаимодействуя с нарушителем, удерживая и провоцируя дополнительные действия, направляя по многовариантным ложным путям, предоставляя на пути проникновения в ложную систему задачи по преодолению сравнимых с реальной системой уровней сложности.

В числе характеристик таких решений целесообразно ввести параметр с условным названием *имитационная достоверность*, который должен оцениваться относительно 100-процентной вероятности того, что «медовые» ловушки не были определены как ложные активы и компоненты «обманки». Эффективность с учетом имитационной достоверности при этом можно рассматривать как результативность обнаружения атак, при которой они были обнаружены, но имитационный характер цели не был раскрыт злоумышленником.

### Обзор результатов экспериментальных исследований применения технологии Honeypot-«обманок» на функционирующей инфраструктуре

В [14–26] подробно описан ряд «полевых» исследований технологий «обманки» с использованием Honeypot в университетских сетях. В этих статьях авторы провели анализ поведения как реальных, так и ирреальных (экспертов по безопасности) злоумышленников в уязвимой системе: они предоставили им атакуемую систему, в которой использовали ресурсы Honeypot, и следили за их поведением.

В первом раунде своего исследования Д. Франхольц и соавторы [14] предложили серверные Honeypot-приложения: ложные баннеры, фальшивый файл robots.txt, поддельные сообщения об ошибках, адаптивная задержка и honey-files. Было проверено 1 200 посещений злоумышленников. Во втором раунде авторы [15] проанализировали поведение злоумышленников, отслеживаемое в течение 222 дней с помощью шести Honeypots, развернутых на одном клиенте и пяти серверах веб-хостинга. Используемыми LIhoneypot отслеживались и контролировались более 12 млн. посещений. В Honeypot-объектах использовались протоколы honeytokenTP, honey-tokenTPS, FTP, POP3, SMTP, SSH и Telnet. Также имитировались протоколы Bacnet и Modbus для исследования угроз промышленным приложениям.

М. Лазаров и соавторы [16] преднамеренно поместили фиктивную конфиденциальную информацию в электронные таблицы Google, в т. ч. IP-адреса, призванные заманить злоумышленников. Было отслежено 174 клика, 44 посещения 39-ти уникальных IP-адресов.

Л. Лю и соавторы [17] следовали аналогичному подходу. Но в их эксперименте ключи SSH были выложены в свободный доступ на github, а Honeypot реализовывался на основе Cowrie. Контролировалось порядка 31 000 уникальных паролей и поведение пользователя после входа в систему в течение двух недель. Далее был выполнен анализ распределения наиболее часто использованных для осуществления атаки паролей из ежегодно публикуемыми SplashData списков наиболее употребительных паролей. Результат показал, что в 76 % атаки производились с IP-адресов, которые специально использовались для атаки и были квалифицированы как источники угроз сервисом IP Intelligence с вероятностью 1,0.

В [18] описан более сложный эксперимент: сформирована имитация сети организации, содержащая учетные записи пользователей, почтовые данные, документы, профили браузера и другие информационные ресурсы. В сеть были введены различные ловушки и приманки. Было организовано тестирование в форме квеста по принципу CTF (*от англ.* Capture the Flag – захват флага), в рамках которого специально привлеченным для этой цели экспертам по безопасности предлагалось осуществить проникновение в сеть и собрать некий актив из пяти расположенных в разных местах компонентов. Параллельно исследовалась реакция сети на автоматические атаки вредоносного ПО. В результате 100 % атак и людей, и машин были обнаружены как минимум одним из использованных средств, что доказало эффективность построенного Honeypot-решения, н вместе с тем и необходимость развития технологий симуляции в направлении усложнения поведения. Было отмечено, что, несмотря на не очень эффективные действия вначале, благодаря «knowledge gap» (разрыву между реальными знаниями атакующих и их самооценкой), этот разрыв по мере взаимодействия с объектами сети в ходе эксперимента существенно сокращался (что в частности демонстрировалось снижением числа используемых команд). А это позволяло прогнозировать относительно быструю выработку ими механизма выявления фиктивных активов при сохранении слож-

ности модели на первоначально предложенном уровне. Эксперимент также показал, что лучшие результаты были у атакующих, внимательно ознакомившихся с условиями задания (в которые намеренно были включены подсказки, что дополнительно акцентировало значимость наличия у атакующего предварительной информации для успешного проведения атаки).

Поведение атакующих после вторжения анализировалось в несколько ином аспекте и в более ранних работах. Целью эксперимента в [19] было определить характеристики подключений и классифицировать атаки, а также исследовать реакцию атакующих на предупреждающие баннеры. Для этого использовался набор данных Джонс [20], собранный с помощью HIhoneypot и содержащий 1 548 обращений от 478 атакующих. В эксперименте для развертывания Dionaea Honeypots использовались незадействованные IP-адреса сети университета Мэриленд; в ходе почти полугодового эксперимента 624 сеанса контролировались с помощью инструмента Honeypot Spy.

В аналогичном направлении (исследование реакции на баннеры) Д. Маймон и соавторы [21] выполнили два эксперимента, длительностью в 2 и 6 месяцев. В этих экспериментах использовались 86 и 502 компьютера соответственно. При подключении некоторые из них представляли предупреждающие баннеры. Рабочие станции также содержали различные уязвимые точки входа, созданные с использованием Sebek и OpenVZ в качестве шлюза, которые привлекли 1 058 и 3 768 вторжений соответственно.

Е. Хейрхак и соавторы [22] проанализировали учетные данные, использованные в ходе попыток логического подключения. 8 подключений Honeypot, которые были включены в SSH, представлены шести разным сетям университетского городка в течение семи недель. Было зафиксировано 98 180 соединений с 1 153 уникальных IP-адресов в 79 странах.

В [23, 24] Х. Цзян и соавторы применяли различные виды Honeypots, а именно: Dionaea, Mwcollector, Amun и Nepenthes, – которые использовались, для статистического анализа паттернов атак. Было выделено 5 периодов, в течение которых 166 атакующих проводили атаки на уязвимые сервисы: SMB, NetBIOS, honeytokenTP, MySQL и SSH.

С. Лорен и соавторы [25] проанализировали поведение злоумышленника на основе шаблонов нажатия клавиш. При этом были отслежены, как индикаторы для различных типов поведения, 24 различных действия злоумышленников.

Honeypot в количестве 3 единиц с различными конфигурациями, установленные на протяжении восьми месяцев в университетских сетях, были доступны через SSH и захватили 20 335 набранных команд в течение 1 171 сеанса атак.

Схожий подход был использован в [26]: 4 Linux honeypot были введены в университетскую сеть, доступную через SSH в течение 24 дней, с легко угадываемыми учетными данными. Действия злоумышленников контролировались с помощью syslog-ng (для захвата команд, быстрого доступа к системным вызовам) и Sebek (для сбора нажатий клавиш). Было собрано 269 262 попыток атак с 229 уникальных IP-адресов.

Следует отметить, что кроме различного рода Honeypot, использующих встроенные механизмы контроля, также часто применяются реальные системы с расширенными возможностями мониторинга, которые по определению относятся к HIhoneypot. Только две из описанных выше работ [15, 16] используют существенно различающиеся технологии «обманки», а наиболее сложный эксперимент, как представляется, описан в [18].

В обзоре [13] приводятся выполненные на основе опросов оценки вышерассмотренных (кроме [18]) и некоторых других экспериментов и использованных в них решений. Последние оценивались по таким признакам, как интерактивность, масштабируемость, юридические или этические соображения, тип, развертывание, преимущества и недостатки по сравнению с другими видами технологии защиты, качество и тип данных и полученные значения, тип ресурса технологии «обманки», технический способ развертывания и вид технологии «обманки», возможности обнаружения и предотвращения обнаружения и растяжимость. Оценка производилась по системе «да/нет» без более детальных характеристик. Будучи весьма условной, сравнительная оценка, тем не менее, отчетливо показала наряду с отсутствием идеального (или стремящегося к нему) решения, также и отсутствие решений, в которых бы сочетались интерактивность, масштабируемость и механизм предотвращения обнаружения.

Следует отметить, что все три характеристики взаимосвязаны и взаимопротиворечивы, и в некоторых рассмотренных решениях не могут сочетаться в силу имевшихся аппаратных ограничений. Как представляется, их сочетание может быть реализовано в Honeypot-решении на основе облачных вычислений. Масштабируемые устойчивые Honeypot-решения в облачной среде гипотетически могут также использоваться в качестве активной защиты для противодействия атакам, нацеленным на снижение производительности сети (DDoS-атаки), и могут быть реализованы с использованием подходов, изложенных в [27–29].

В контексте исследований технологий «обманки» и Honeypot-решений следует отметить также работы [30–34].

**Предлагаемое решение**

В интересах исследования и демонстрации возможностей Honeypot-решения для обеспечения безопасности, в корпоративных сетях был разработан специальный лабораторный стенд, созданный и протестированнный внутри системы Microsoft Azure. В качестве его основы был выбран Windows Server 2016 с развернутым веб-сервером Microsoft IIS. Устойчивость Honeypot исследовалась в условиях DDoS-атаки типа «TCP SYNFLOOD», для эмуляции которой развернут образ Kali linux для управления botnet-сетью (рисунок 2).

Веб-серверу с точки зрения вычислительных ресурсов выделено 1 ядро и 2 ГБ оперативной памяти. При генерации виртуальной машины (ВМ) в скрипте были прописаны условия автоматического создания реплик (instances) сервера и обработки трафика от него через подготовленный балансировщик. В рассматриваемой конфигурации ограничений по решаемым задачам и типам операционных систем нет.

Windows Server и IIS были выбраны из экономических и практических соображений, среди которых – простота эмуляции IIS web-сервера средствами honeyd. Windows server достаточно просто изобразить, поскольку он имеет значительное количество легко сканируемых портов, по которым можно определить, какие конкретно службы и сервисы запущены на публичном сервере. Однако слишком большое число портов может отпугнуть злоумышленника, так как крайне редко встречается в реальности.

Соответственно на honeyd реализована эмуляция базовых сервисов (сетевые сокеты 80/tcp, 443/tcp, 8080/tcp, 3389/tcp); также есть возможность получить базовые ответы. Обеим ВМ дана возможность соединяться с внешней средой с использованием «белых» (публичных) IP-адресов для возможности удаленного доступа к ВМ внутри стенда, а также для возможности реализации собственно DDoS-атаки, которая в исследовании реализована следующим образом. В качестве ботнет-сети выступают ВМ с установленной операционной системой Ubuntu и утилитой msfconsole, Command and Control (C&C); сервер Kali включает настроенный оркестратор Ansible. Из среды Kali через Ansible отдается команда о генерации TCP SYN-сообщений.

Для соответствующей подготовки ВМ был написан скрипт, в котором указана возможность масштабирования; сам же сервер создается один. При отработке системы реагирования на атаку предусмотрена возможность создать новые instances (реплики) с похожим набором параметров относительно шаблона ВМ. Сами реплики, как было указано ранее, являются Honeypot. При создании им выделяется меньше ресурсов: 1 ядро и 512 МБ оперативной памяти, но при этом настраивается сетевая карта с высокой производительностью (до 1 Гб/с). В скрипте на языке JSON указаны правила создания сетей и адаптеров ВМ, а также правила создания балансировщика нагрузки. В том числе в шаблоне указано ограничение на количество генерируемых ВМ, наличие публичных адресов и правила взаимодействия с балансировщиком. Также указаны правила прямого проброса интересующего трафика.

Балансировщик нагрузки реализован по принципу NAT-трансляций (*от англ.* Network Address Translation, преобразование сетевых адресов) в конкретный сетевой сокет реплики виртуальной машины с единым внешним IP-адресом (рисунок 3) и распределяет новые входящие потоки данных, которые поступают на внешний интерфейс для экземпляров внутреннего пула в соответствии с правилами и проверками работоспособности.

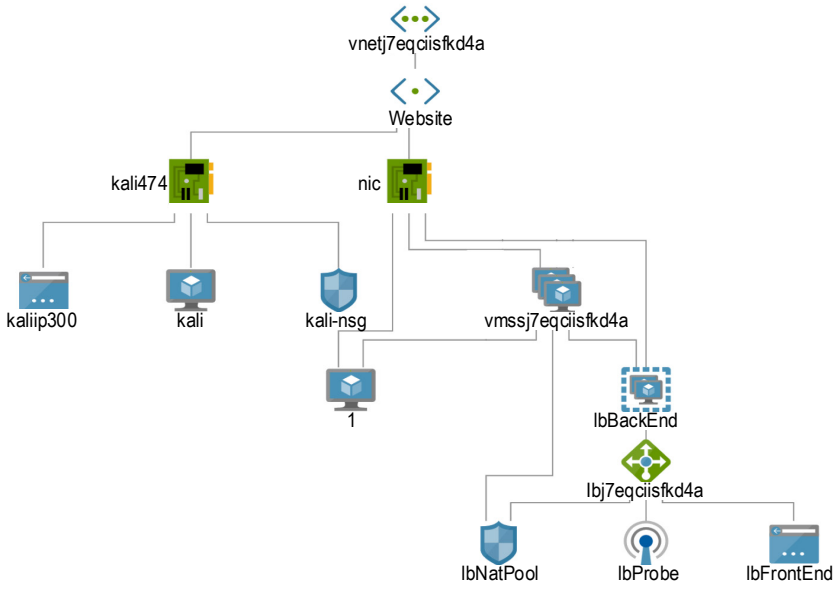


**Рис. 2. Топология лабораторного стенда**

Помимо этого, общедоступный балансировщик может предоставлять исходящие соединения для ВМ в виртуальной сети путем преобразования их частных IP-адресов в публичные. Все остальные сетевые сокеты при обработке входящего трафика он блокирует для того, чтобы не отпугнуть атакующего во время сканирования сетевой части веб-сервера.

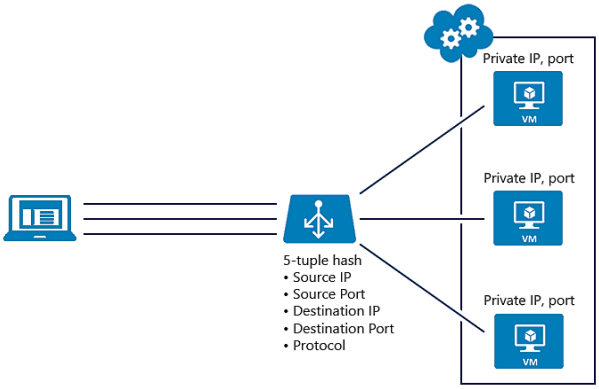


**Рис. 3. Принцип балансировки нагрузки**

По умолчанию балансировщик для сопоставления потоков с доступными серверами использует 5-элементный хэш, состоящий из IP-адреса источника, порта источника, IP-адреса назначения, порта назначения и номера протокола IP. Возможно создать сходство с конкретным исходным IP-адресом, выбрав двух- или трехэлементный хэш для данного правила. Все пакеты одного и того же потока пакетов поступают в один и тот же экземпляр за интерфейсом с балансировкой нагрузки. С помощью балансировщика можно создать входящее правило NAT для переноса трафика с определенного порта определенного IP-адреса внешнего интерфейса на определенный порт определенного внутреннего экземпляра внутри виртуальной сети. Это достигается тем же распределением на основе хэшей, что и для распределения нагрузки. Распространенными сценариями для этой возможности являются сеансы RDP (*от англ.* Remote Desktop Protocol, протокол удаленного рабочего стола) и SSH (*от англ.* Secure Shell, безопасная «оболочка») с отдельными экземплярами ВМ. Также можно сопоставить несколько внутренних конечных точек с различными портами на одном IP-адресе внешнего интерфейса.

Балансировщик нагрузки напрямую не взаимодействует с транспортным, пользовательским или прикладным уровнем, но может поддерживать любой сценарий TCP- или UDP-приложения. Он не прерывает и не инициирует потоки, взаимодействует с полезной нагрузкой потока, не предоставляет функции шлюза прикладного уровня, и протокольные «рукопожатия» всегда происходят непосредственно между клиентом и экземпляром ВМ. На каждую конечную точку отвечает только ВМ; ответ на запрос к внешнему интерфейсу является ответом, генерируемым серверной ВМ. При успешной проверке подключения к внешнему интерфейсу проверяется сквозное подключение по крайней мере к одной внутренней виртуальной машине. Публичный балансировщик отображает общедоступный IP-адрес и номер порта входящего трафика на частный IP-адрес и номер порта ВМ и, наоборот, для ответного трафика от ВМ. Применяя правила балансировки нагрузки, можно распределять определенные типы трафика между несколькими ВМ или службами.

При создании копии ВМ автоматически создаются новые правила; сами ВМ создаются прозрачно для администратора и эмулируют поведение на основе определенных параметров, выявленных системой по заданному паттерну, так же есть возможность генерировать автоматические сообщения администратору на почту или в SIEM-систему (*от англ.* Security Information and Event Management, системы управления событиями информационной безопасности).

### Тестирование и анализ результатов

Как уже упоминалось, для исследования устойчивости решения была выбрана DDoS-атака, которая осуществлялась по открытому 80-му порту веб-сервера IIS. Отправной точкой послужил развернутый образ Kali linux, выступающий в качестве C&C-сервера, с которого была отдана команда на старт атаки, и некоторое количество хостов начали генерировать TCP-SYN. Хосты при генерации SYN-flood-а используют случайные порты источника и могут отправлять до 1 Гб/с трафика. Трафик попадает на балансировщик, которому предоставлена максимальная производительность сети (40 Гб/с). В скрипте прописано правило мониторинга загрузки процессора (CPU) и указаны пороговые значения, при которых происходит создание новых реплик ВМ и, соответственно, их удаление при спаде нагрузки.

После старта атаки происходит нарастание нагрузки на ВМ, который идет во время ожидания реакции системы на триггер для создания новой копии (рисунок 4). Осуществляется горизонтальное масштабирование сервиса, но сам сервис не увеличивает свою производительность – увеличивается только сетевая емкость за счет добавления «медовых» ловушек. И за счет увеличения производительности сети происходит снижение загрузки на основную реплику веб-сервера. В скрипте прописано правило: при наличии загрузки процессора более 90 % в течение 5 минут следует создать реплику Honeypot.

После создания одного нового инстанса происходит проверка правила и, если загрузка все еще больше 90 %, происходит создание еще одной реплики до тех пор, пока загрузка не упадет. Чтобы излишне не использовать вычислительные ресур-

сы, создано правило по удалению реплики при снижении средней загрузки по CPU.

На представленном графике (рисунок 5) видно, что при получении достаточно большого объема трафика возрастает нагрузка на CPU, вследствие чего происходит деградация конечного сервиса, которым могут пользоваться клиенты компании. Но после отработки реакции на триггер система начинает создавать новые реплики веб-сервера, и уже через 13 минут после атаки становится заметным линейный характер спада нагрузки на CPU сервера. Чем больше реплик генерируется в автоматическом режиме, тем меньше оказывается влияние на реальную инфраструктуру.

С точки зрения топологии (рисунок 6) рассматриваемой сети реплики веб-сервера находятся за балансировщиком трафика, и для пользователей конечного сервиса их количество незаметно. Но при этом имеется возможность снять нагрузку с ВМ и гарантировать определенный уровень сервиса на сайте компании.

Результат эксперимента показал возможность оказывать сервис клиентам в условиях активной защиты от атаки DDoS типа «TCP SYNFLOOD». Сегодня на рынке систем информационной безопасности существует несколько готовых решений, которые могут предложить не только горизонтальное масштабирование сервиса, но и вертикальное, позволяют не только в пределах облачной инфраструктуры разворачивать реплики ВМ. Имеются решения, которые не предоставляют автоматическое масштабирование среды, но дают возможность исследовать тип атаки и ее воздействие на сервис.

Предлагаемое решение может сочетать в себе масштабируемость, устойчивость и интерактивность, поскольку поведение Honeypot может формироваться на основе обновляемых паттернов, в него могут быть добавлены дополнительные механизмы мониторинга и др. Использование облачных ресурсов, аналогичных используемым реальной сетью в облачной среде, позволяет создавать масштабируемые Honeypot-решения с высокой имитационной достоверностью.

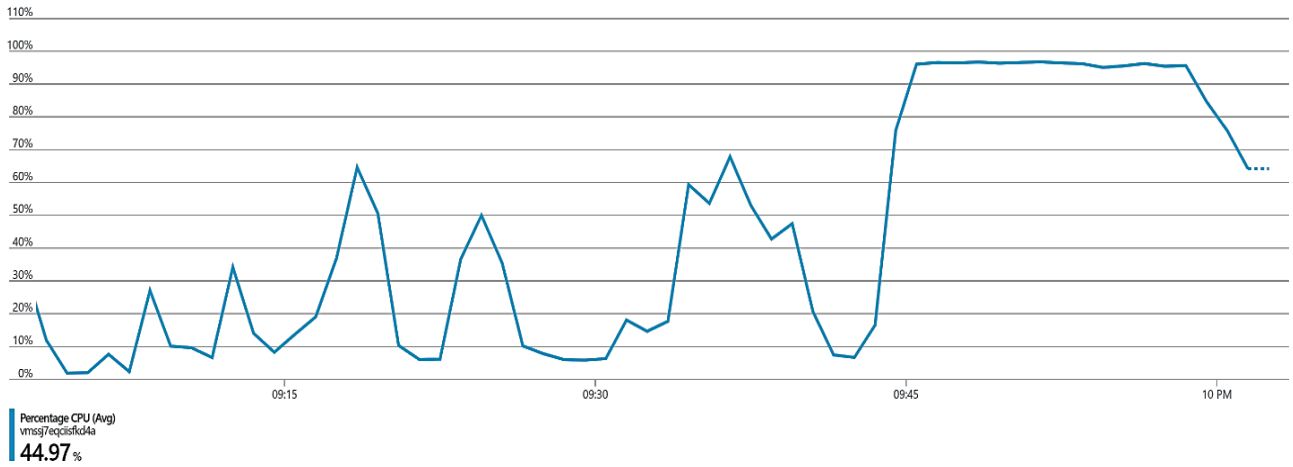


**Рис. 4. График нагрузки CPU в начале атаки**

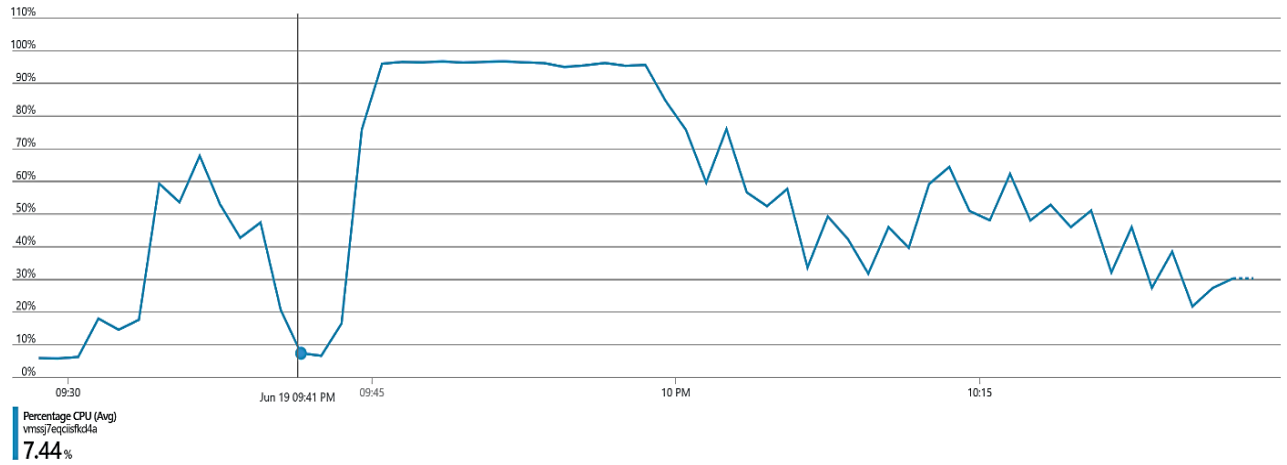


**Рис. 5. Снижение нагрузки в зависимости от количества реплик**

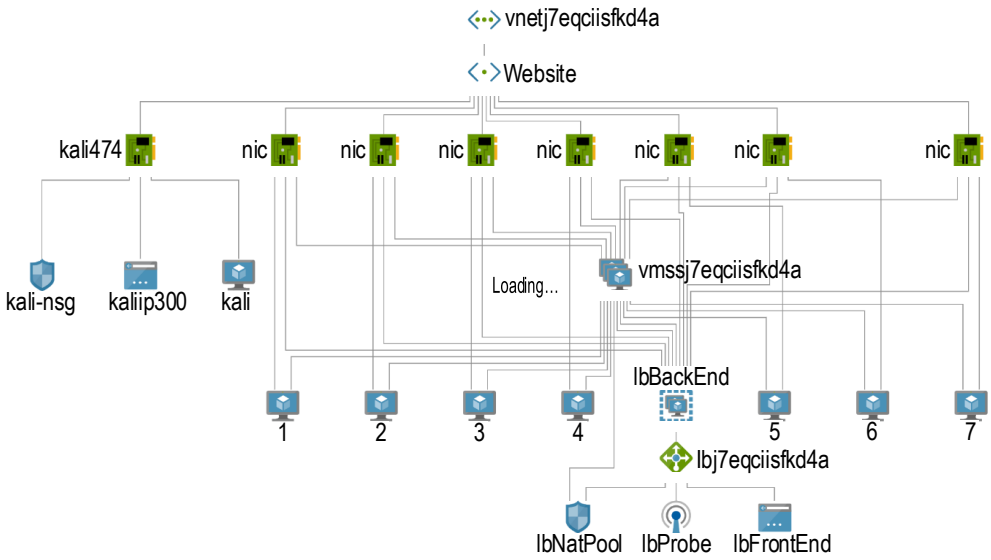


**Рис. 6. Финальная топология**

## Заключение

В статье рассмотрен процесс создания масштабируемого Honeypot-решения, и исследована его работа с помощью стенда в среде Microsoft Azure. Использование технологии «обманки» потребовало разработки приложения, позволяющего применить механизмы копирования оконечного устройства и дублирование сервиса на его основе. Предложенный подход может быть расширен и масштабирован в качестве действующего решения в корпоративных сетях, построенных с использованием технологий облачных вычислений. В статье также отмечена необходимость разработки и применения системы показателей эффективности Honeypot-решений и комплексных решений с использованием технологий «обманки» на всех этапах создания для достижения высокой эффективности их практического использования.

**Список используемых источников**

1. 60 Must-Know Cybersecurity Statistics for 2019 // Varonis. URL: https://www.varonis.com/blog/cybersecurity-statistics (дата обращения 26.04.2019)

2. Буйневич М.В., Владыко А.Г., Доценко С.М., Симонина О.А. Организационно-техническое обеспечение устойчивости функционирования и безопасности сети связи общего пользования. СПб.: СПбГУТ, 2013. 192 с.

3. Буйневич М.В, Васильева И.Н, Воробьев Т.М, Гниденко И.Г, Егорова И.В, Еникеева Л.А и др. Защита информации в компьютерных системах. СПб.: Изд-во СПбГЭУ, 2017. 163 с.

4. Израилов К.Е. Анализ состояния в области безопасности программного обеспечения // II Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (Санкт-Петербург, Россия, 27–28 февраля 2013). СПб.: СПбГУТ, 2013. С. 874–877.

5. Израилов К.Е. Модель прогнозирования угроз телекоммуникационной системы на базе искусственной нейронной сети // Вестник ИНЖЭКОНа. Серия: Технические науки. 2012. № 8(59). С. 150–153.

6. Покусов В.В. Особенности взаимодействия служб обеспечения функционирования информационной системы // Информатизация и связь. 2018. № 5. С. 51–56.

7. Buinevich M., Fabrikantov P., Stolyarova E., Izrailov K., Vladyko A. Software Defined Internet of Things: Cyber Antifragility and Vulnerability Forecast // Proceedings of the 11th International Conference on Application of Information and Communication Technologies (AICT, Moscow, Russia, 20–22 September 2017). Piscataway, NJ: IEEE, 2017. PP. 293–297. DOI:10.1109/ICAICT.2017.8687021

8. Kotenko I., Kuleshov A., Ushakov I. Aggregation of elastic stack instruments for collecting, storing and processing of security information and events // Proceedings of the SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, San Francisco, USA, 4–8 August 2017). Piscataway, NJ: 2017. DOI:10.1109/UIC-ATC.2017.8397627

9. Котенко И.В., Левшун Д.С., Чечулин А.А., Ушаков И.А., Красов А.В. Комплексный подход к обеспечению безопасности киберфизических систем на основе микроконтроллеров // Вопросы кибербезопасности. 2018. № 3(27). С. 29–38. DOI:10.21681/2311-3456-2018-3-29-38

10. Котенко И.В., Ушаков И.А., Пелёвин Д.В., Овраменко А.Ю. Гибридная модель базы данных NoSQL для анализа сетевого трафика // Защита информации. Инсайд. 2019. № 1(85). С. 46–54.

11. Котенко И.В., Ушаков И.А. Использование технологий больших данных для мониторинга инцидентов информационной безопасности // Юбилейная XV Санкт-Петербургская Международная Конференция «Региональная Информатика (РИ-2016)» (Санкт-Петербург, Россия, 26–28 октября 2016). СПб.: СПОИСУ, 2016. С. 168–169.

12. Ушаков И.А., Котенко И.В., Крылов К.Ю. Анализ методик применения концепции больших данных для мониторинга безопасности компьютерных сетей // IX Санкт-Петербургская межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2015)» (Санкт-Петербург, Россия, 28–30 октября 2015). СПб.: СПОИСУ, 2015. С. 75–76.

13. Fraunholz D., Anton S.D., Lipps C., Reti D., Krohmer D., Pohl F, et al. Demystifying Deception Technology: A Survey. 2018. DOI:10.13140/RG.2.2.30392.65288

14. Fraunholz D., Zimmermann M., Schotten H.D. Towards Deployment Strategies for Deception Systems // Advances in Science, Technology and Engineering Systems Journal. 2017. Vol. 2. Iss. 3. PP. 1272–1279.

15. Fraunholz D., Schotten H.D. Defending Web Servers with Feints, Distraction and Obfuscation // Proceedings of the International Conference on Computing, Networking and Communications (ICNC, Maui, USA, 5–8 March 2018). Piscataway, NJ: IEEE, 2018. DOI:10.1109/ICCNC.2018.8390365

16. Lazarov M., Onaolapo J., Stringhini G. Honey Sheets: What Happens to Leaked Google Spreadsheets? // Proceedings of the 9th Workshop on Cyber Security Experimentation and Test (CSET, Austin, USA, 8 August 2016). Berkeley: USENIX, 2016. URL: https://www.usenix.org/system/files/conference/cset16/cset16-paper-lazarov.pdf (дата обращения 17.09.2019)

17. Liu L., Mahar K., Virdi C., Zhou H. Hack Like no One is Watching: Using a Honeypot to Spy on Attackers. MIT Computer and Network Security Term Projects, 2016. URL: http://docplayer.net/21979034-Hack-like-no-one-is-watching-using-a-honeypot-to-spy-on-attackers.html (дата обращения 17.09.2019)

18. Applying Deception Mechanisms for Detecting Sophisticated Cyber Attacks // A Research Paper by TopSpin Security. October 2016. URL: https://www.cyentia.com/library-item/applying-deception-mechanisms-for-detecting-sophisticated-cyber-attacks (дата обращения 14.06.2019)

19. Robin B., Cukier M. An evaluation of connection characteristics for separating network attacks // International Journal of Security and Networks. 2009. Vol. 4. Iss. 1-2. PP. 110–24. DOI:10.1504/IJSN.2009.023430

20. Jones H.M. The Restrictive Deterrent Effect of Warning Messages on the Behavior of Computer System Trespassers. PhD Thesis. College Park: University of Maryland, 2014.

21. Maimon D., Alper M., Sobesto B., Cuckier M. Restrictive deterrent effects of a warning banner in an attacked computer system // Criminology. 2014. Vol. 52. Iss. 1. DOI:10.1111/1745-9125.12028

22. Kheirkhah E., Amin S.M.P., Sistani H.A., Acharya H.S. An experimental study of SSH attacks by using Honeypot Decoys // Indian Journal of Science and Technology. 2013. Vol. 6. Iss. 12. PP. 5567–5578.

23. Jiang X., Wang X., Xu D. Stealthy Malware Detection Through VMM-based "Out-of-the-box" Semantic View Reconstruction // Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS, Alexandria, USA). New York: ACM Press, 2007. DOI:10.1145/1315245.1315262

24. Jiang X., Wang X. "Out-of-the-box" Monitoring of VM-Based High-Interaction Honeypots // Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID, Gold Goast, Australia, 5–7 September 2007). Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007. Vol. 4637. DOI:10.1007/978-3-540-74320-0_11

25. Laurén S., Rauti S., Leppänen V. An Interface Diversified Honeypot for Malware Analysis // Proceedings of the 10th European Conference on Software Architecture Workshops (ECSAW, Copenhagen, Denmark, 28 November – 02 December 2016). New York: ACM Press, 2016. DOI:10.1145/2993412.2993417

26. Al-Shaer E., Duan Q., Jafarian J. Random Host Mutation for Moving Target Defense // Proceedings of the 8th International Conference on Security and Privacy in Communication Networks (SecureComm, Padua, Italy, 3–5 September 2012). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Berlin: Springer, 2012. Vol. 106. DOI:10.1007/978-3-642-36883-7_19

27. Krasov A.V., Levin M.V., Shterenberg S.I., Isachenkov P.A. Traffic flow management model in software-defined networks with unequal load metric // H&ES Research. 2016. Vol. 8. Iss. 4. PP. 70–74.

28. Красов А.В., Левин М.В., Цветков А.Ю. Управление сетями передачи данных с изменяющейся нагрузкой // Всероссийская научная конференция по проблемам управления в технических системах. СПб.: Санкт-Петербургский государственный электротехнический университет, 2015. № 1. С. 141–146.

29. Красов А.В., Левин М.В., Штеренберг С.И., Исаченков П.А. Методология управления потоками трафика в программно-определяемой адаптивной сети // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2016. № 4. С. 3–8.

30. Whaley B. Toward a general theory of deception // Journal of Strategic Studies. 1982. Vol. 5. Iss. 1. PP. 178–192. DOI:10.1080/01402398208437106

31. Barros A. DLP and honeytokens. 2007. URL: http://blog.securitybalance.com/2007/08/dlp-and-honeytokens.html (дата обращения 17.09.2019)

32. Spitzner L. The Honeynet Project: Trapping the Hackers. URL: https://pdfs.semanticscholar.org/ec08/e8c4537db092da8c1fd239f2d9fe189d56d6.pdf (дата обращения 17.09.2019)

33. Sobesto B. Empirical Studies based on Honeypots for Characterizing Attackers Behaviour. PhD Thesis. College Park: University of Maryland, 2015.

34. Sentanoe S., Taubmann B., Reiser H.P. Virtual Machine Introspection Based SSH Honeypot // Proceedings of the 4th Workshop on Security in Highly Connected IT Systems (SHCIS, Neuchatel, Switzerland, 19–22 June 2017). New York: ACM Press, 2017. DOI:10.1145/3099012.3099016

* * *

# SCALABLE HONEYPOT SOLUTION FOR CORPORATE NETWORKS SECURITY PROVISION

**A. Krasov[1], R. Petriv[1*], D. Sakharov[1], N. Storozhuk[1], I. Ushakov[1]**

[1]The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

**Article info**
The article was received 26 July 2019

**For citation:** Krasov A., Petriv R., Sakharov D., Storozhuk N., Ushakov I. Scalable Honeypot Solution for Corporate Networks Security Provision. *Proceedings of Telecommunication Universities*. 2019;5(3):86–97. (in Russ.) Available from: https://doi.org/10.31854/1813-324X-2019-5-3-86-97

**Abstract:** *Trends in modern security technologies with honeypot technologies use are analyzed to detect and explore intruders behavior for counteract measures development. Scalable solution proposed and tested within Microsoft Azure exploratory installation. DDoS attack stress test of the solution is performed.*

**Keywords:** *corporate networks, cloud computing, deception technologies, honeypot technologies, infocommunication networks security, cyberthreats, DDoS attack.*

**References**

1. Varonis. *60 Must-Know Cybersecurity Statistics for 2019*. Available from: https://www.varonis.com/blog/cybersecurity-statistics [Accessed 26th April 2019]

2. Buinevich M.V., Vladyko A.G., Dotsenko S.M., Simonina O.A. *Organizational and technical provision of functioning and security of public communications network*. St. Petersburg: The Bonch-Bruevich Saint-Petersburg State University of Telecommunications Publ.; 2013. 192 p. (in Russ.)

3. Buinevich M.V., Vasilyeva I.N., Vorobev T.M., Gnidenko I.G., Egorova I.V., Enikeeva L.A., et al. *Zashchita informatsii v kompiuternykh sistemakh* [Information Security in Computer Systems]. St. Petersburg: Saint-Petersburg State University of Economics Publ.; 2017. 163 p. (in Russ.)

4. Izrailov K. Analysis of the Security State of Software. *Proceedings of the IId International Conference on Infotelecommunications in Science and Education, 27–28 February 2013, St. Petersburg, Russia*. St. Petersburg: The Bonch-Bruevich Saint-Petersburg State University of Telecommunications Publ.; 2013. p.874–877. (in Russ.)

5. Izrailov K. Model of Forecasting the Telecommunication System Threats on the Basis of the Artificial Neural Network. *Vestnik INZHEKONa. Seriia: Tekhnicheskie nauki*. 2012;8(59):150–153. (in Russ.)

6. Pokusov V.V. Features of Interaction of Services for the Operation of the Information System. *Informatizatsiia i sviaz*. 2018;5:51–56. (in Russ.)

7. Buinevich M., Fabrikantov P., Stolyarova E., Izrailov K., Vladyko A. Software Defined Internet of Things: Cyber Antifragility and Vulnerability Forecast. *Proceedings of 11th International Conference on Application of Information and Communication Technologies, AICT, 20–22 September 2017, Moscow, Russia*. Piscataway, NJ: IEEE; 2017. p.293–297. Available from: https://doi.org/10.1109/ICAICT.2017.8687021

8. Kotenko I., Kuleshov A., Ushakov I. Aggregation of elastic stack instruments for collecting, storing and processing of security information and events. *Proceedings of the SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, 4–8 August 2017, San Francisco, USA*. Piscataway, NJ; 2017. Available from: https://doi.org/10.1109/UIC-ATC.2017.8397627

9. Kotenko I., Levshun D., Chechulin A., Ushakov I., Krasov A. Integrated Approach to Provide Security of Cyber-Physical Systems Based on Microcontrollers. *Voprosy kiberbezopasnosti*. 2018;3(27):29–38. (in Russ.) Available from: https://doi.org/10.21681/2311-3456-2018-3-29-38

10. Kotenko I.V., Ushakov I.A., Pelevin D.V., Ovramenko A. Yu. Hybrid NoSQL Database Model for Analysis of Network Traffic. *Zaŝita informacii. Inside*. 2019;1(85):46–54. (in Russ.)

11. Kotenko I.V., Ushakov I.A. Using big data technologies to monitor information security incidents. *Proceedings of the XV Anniversary St. Petersburg Internatonal Conference "Regional Informatics (RI-2016)", 26–28 October 2016, St. Petersburg, Russia.* St. Petersburg.: Sankt-Peterburgskoe obshchestvo informatiki vychislitelnoi tekhniki sistem sviazi i upravleniia; 2016. p.168–169. (in Russ.)

12. Ushakov I.A., Kotenko I.V., Крылов К. Yu. Analysis of methods of application of the big data concept for monitoring the security of computer networks. *Proceedings of the IX St. Petersburg Interregional Conference "Information Security of Russian Regions (ISRR-2015)", 28–30 October 2015, St. Petersburg, Russia.* St. Petersburg.: Sankt-Peterburgskoe obshchestvo informatiki vychislitelnoi tekhniki sistem sviazi i upravleniia; 2015. p.75–76. (in Russ.)

13. Fraunholz D., Anton S.D., Lipps C., Reti D., Krohmer D., Pohl F, et al. *Demystifying Deception Technology: A Survey*. 2018. Available from: https://doi.org/10.13140/RG.2.2.30392.65288

14. Fraunholz D., Zimmermann M., Schotten H.D. Towards Deployment Strategies for Deception Systems. *Advances in Science, Technology and Engineering Systems Journal*. 2017;2(3):1272–1279.

15. Fraunholz D., Schotten H.D. Defending Web Servers with Feints, Distraction and Obfuscation. *Proceedings of the International Conference on Computing, Networking and Communications, ICNC, 5–8 March 2018, Maui, USA*. Piscataway, NJ: IEEE; 2018. Available from: https://doi.org/10.1109/ICCNC.2018.8390365

16. Lazarov M., Onaolapo J., Stringhini G. Honey Sheets: What Happens to Leaked Google Spreadsheets? *Proceedings of the 9th Workshop on Cyber Security Experimentation and Test, CSET, 8 August 2016, Austin, USA*. Berkeley: USENIX; 2016. Available from: https://www.usenix.org/system/files/conference/cset16/cset16-paper-lazarov.pdf [Access 17th September 2019]

17. Liu L., Mahar K., Virdi C., Zhou H. *Hack Like no One is Watching: Using a Honeypot to Spy on Attackers*. MIT Computer and Network Security Term Projects. 2016. Available from: http://docplayer.net/21979034-Hack-like-no-one-is-watching-using-a-honeypot-to-spy-on-attackers.html [Access 17th September 2019]

18. *Applying Deception Mechanisms for Detecting Sophisticated Cyber Attacks. A Research Paper by TopSpin Security*. October 2016. Available from: https://www.cyentia.com/library-item/applying-deception-mechanisms-for-detecting-sophisticated-cyber-attacks [Access 14th June 2019]

19. Robin B., Cukier M. An evaluation of connection characteristics for separating network attacks. *International Journal of Security and Networks*. 2009;4(1-2):110–24. Available from: https://doi.org/10.1504/IJSN.2009.023430

20. Jones H.M. *The Restrictive Deterrent Effect of Warning Messages on the Behavior of Computer System Trespassers*. PhD Thesis. College Park: University of Maryland; 2014.

21. Maimon D., Alper M., Sobesto B., Cuckier M. Restrictive deterrent effects of a warning banner in an attacked computer system. *Criminology*. 2014;52(1). Available from: https://doi.org/10.1111/1745-9125.12028

22. Kheirkhah E., Amin S.M.P., Sistani H.A., Acharya H.S. An experimental study of SSH attacks by using Honeypot Decoys. *Indian Journal of Science and Technology*. 2013;6(12):5567–5578.

23. Jiang X., Wang X., Xu D. Stealthy Malware Detection Through VMM-based "Out-of-the-box" Semantic View Reconstruction. *Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS, Alexandria, USA*. New York: ACM Press; 2007. Available from: https://doi.org/10.1145/1315245.1315262

24. Jiang X., Wang X. "Out-of-the-box" Monitoring of VM-Based High-Interaction Honeypots. *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection, RAID, 5–7 September 2007, Gold Goast, Australia. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer; 2007. vol.4637. Available from: https://doi.org/10.1007/978-3-540-74320-0_11

25. Laurén S., Rauti S., Leppänen V. An Interface Diversified Honeypot for Malware Analysis. *Proceedings of the 10th European Conference on Software Architecture Workshops, ECSAW, 28 November – 02 December, 2016, Copenhagen, Denmark*. New York: ACM Press; 2016. Available from: https://doi.org/10.1145/2993412.2993417

26. Al-Shaer E., Duan Q., Jafarian J. Random Host Mutation for Moving Target Defense. *Proceedings of the 8th International Conference on Security and Privacy in Communication Networks, SecureComm, 3–5 September 2012, Padua, Italy. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Berlin: Springer; 2012. vol.106. Available from: https://doi.org/10.1007/978-3-642-36883-7_19

27. Krasov A.V., Levin M.V., Shterenberg S.I., Isachenkov P.A. Traffic flow management model in software-defined networks with unequal load metric. *H&ES Research*. 2016;8(4):70–74.

28. Krasov A.V., Levin M.V., Tcvetkov A.I. Upravlenie setiami peredachi dannykh s izmeniaiushcheisia nagruzkoi [Management of Data Networks with Variable Load]. *Vserossiiskaia nauchnaia konferentsiia po problemam upravleniia v tekhnicheskikh sistemakh* [All-Russian Scientific Conference on Management Problems in Technical Systems]. St. Petersburg: Saint-Petersburg Electrotechnical University Publ.; 2015. iss.1. p.141–146. (in Russ.)

29. Krasov A.V., Levin M.V., Shterenberg S.I., Isachenkov P.A. Methodology Research on the Efficiency of the Traffic Flow Management Method Based on the Information About the Load of Software-Defined Networks with Unequal Route Metric. *Vestnik of St. Petersburg State University of Technology and Design. Series 1. Natural and technical science*. 2016;4:3–8. (in Russ.)

30. Whaley B. Toward a general theory of deception. *Journal of Strategic Studies*. 1982:5(1):178–192. DOI:10.1080/01402398208437106

31. Barros A. *DLP and honeytokens*. 2007. Available from: http://blog.securitybalance.com/2007/08/dlp-and-honeytokens.html [Access 17th September 2019]

32. Spitzner L. *The Honeynet Project: Trapping the Hackers*. URL: https://pdfs.semanticscholar.org/ec08/e8c4537db092da8c1fd239f2d9fe189d56d6.pdf [Access 17th September 2019]

33. Sobesto B. *Empirical Studies based on Honeypots for Characterizing Attackers Behaviour*. PhD Thesis. College Park: University of Maryland; 2015.

34. Sentanoe S., Taubmann B., Reiser H.P. Virtual Machine Introspection Based SSH Honeypot. *Proceedings of the 4th Workshop on Security in Highly Connected IT Systems, SHCIS, 19–22 June 2017, Neuchatel, Switzerland*. New York: ACM Press; 2017. Available from: https://doi.org/10.1145/3099012.3099016