**БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2021. – № 2 (101).** – 122–135

*МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ*

УДК 004.056 DOI: 10.17212/2782-2230-2021-2-122-135

# HONEYPOT КАК ИНСТРУМЕНТ СОЗДАНИЯ ЭФФЕКТИВНОЙ ЗАЩИЩЕННОЙ СИСТЕМЫ*

А.Б. АРХИПОВА[1], Д.Р. КАРЕВСКИЙ[2]

[1] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент кафедры защиты информации. E-mail: arhipova@corp.nstu.ru*
[2] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, лаборант кафедры защиты информации. E-mail: kaf_zi@corp.nstu.ru*

Honeypot-системы были разработаны для поиска и изучения действий злоумышленников в скомпрометированной системе. Термин honeypot используется для системы, которая была настроена с намерением быть скомпрометированной (поэтому обычно содержит более старое ПО с уязвимостями безопасности либо имеет дыры в безопасности, связанные с неправильной настройкой ПО) и для получения информации о методах и инструментах злоумышленника.

Honeypot-система может снизить количество ложных срабатываний, выдаваемых другими средствами защиты информации, внедренными в систему, такими как IDS/IPS-системы, различные антивирусы. Это делает чрезвычайно эффективным использование таких систем для обнаружения атак. Организации, которые могут регистрировать тысячи предупреждений в день с использованием традиционных технологий, будут регистрировать только сто предупреждений с помощью honeypot-систем. Honeypot, с другой стороны, могут быть легко использованы для выявления и захвата новых, более изощренных атак, придуманных сообществом черных хакеров. Honeypot могут легко обнаружить новые атаки, потому что любое действие против такой системы является аномалией. Таким образом, honeypot можно использовать для сбора, управления и анализа большего количества данных об атаках.

Honeypot-система также может быть использована для получения информации о хакерской деятельности в рамках определения методов работы злоумышленников и, как результат, становится превентивной мерой против реально защищенной системы. На ранних этапах хакер сканирует сеть для поиска уязвимых компьютеров, в результате чего обнаруживает приманку, которая намеренно уязвима для привлечения атак. Если в дальнейшем злоумышленник попытается подключиться к honeypot, система немедленно обнаружит и зафиксирует действие, потому что обычный пользователь не должен взаимодействовать с системой.

* Статья получена 22 февраля 2021 г.

В представленной работе были рассмотрены теоретические аспекты honeypot-систем, представлены классификации honeypot по различным основаниям. Представлена архитектура honeypot-системы, предназначенной для исследования поведения злоумышленника после его проникновения внутрь корпоративной системы, как инструмент реализации комплексной эффективной защищенной системы организации.

**Ключевые слова:** honeypot-система, информационная безопасность, киберугрозы, защищенная система, системы обнаружения вторжений, уязвимости CVE, среда виртуализации VMware с ОС Ubuntu, архитектура защищенной honeypot-системы

## ВВЕДЕНИЕ

В настоящее время увеличение использования сетевых ресурсов сопровождается ростом объема проблем безопасности. Новые угрозы и уязвимости обнаруживают каждый день и затрагивают пользователей и компании на критических уровнях – от вопросов конфиденциальности до финансовых потерь. Одним из важных аспектов защиты в таком случае является анализ методов и действий, выполняемых злоумышленником.

За последние десятилетия было разработано большое количество инструментов для защиты от атак, с которыми сталкивается большинство коммерческих и некоммерческих организаций.

Один из наиболее часто используемых инструментов – межсетевой экран (firewall), который представляет собой программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами.

Система обнаружения вторжений (IDS, Intrusion Detection Systems) – это еще один тип таких инструментов, использующийся для обнаружения некоторых типов вредоносной активности, которая может нарушить безопасность компьютерной системы. К такой активности относят сетевые атаки против уязвимых сервисов, атаки, направленные на повышение привилегий, неавторизованный доступ к важным файлам, а также действия вредоносного программного обеспечения. Но этим инструментам зачастую не хватает возможности обнаружить новые угрозы, такие как уязвимости нулевого дня, основанные на атаках нулевого дня. Также эти инструменты не могут собрать больше информации о злонамеренных действиях злоумышленника, его полезной нагрузке (payload), доступных эксплоитах (exploit) и навыках ввиду отсутствия сигнатур атак нулевого дня в соответствующих базах [1, 2].

В рамках повышения уровня защиты любой организации и создания эффективных защищенных систем разработчики средств безопасности предпринимают попытки формирования наиболее полной базы данных о доступных уязвимостях, об имеющихся эксплоитах и действиях злоумышленников.

Исследовательские организации и образовательные учреждения по безопасности уже анализируют методы и следы сообщества «черных» хакеров. Решением проблемы может явиться создание honeypot-систем [1–3].

## 1. АНАЛИЗ СОВРЕМЕННЫХ HONEYPOT-СИСТЕМ

Изначально honeypot-системы были разработаны для поиска и изучения действий злоумышленников в скомпрометированной системе. Термин «honeypot» используется для системы, которая была настроена с намерением быть скомпрометированной (поэтому обычно содержит более старое ПО с уязвимостями безопасности либо имеет дыры в безопасности, связанные с неправильной настройкой ПО), и для получения информации о методах и инструментах злоумышленника.

Honeypot-система может снизить количество ложных срабатываний, выдаваемых другими средствами защиты информации, внедренных в системе, такими как IDS/IPS-системы, различные антивирусы. Это делает чрезвычайно эффективным использование таких систем для обнаружения атак. Организации, которые могут регистрировать тысячи предупреждений в день с использованием традиционных технологий, будут регистрировать только сто предупреждений с помощью honeypot-систем. Honeypot, с другой стороны, могут быть легко использованы для выявления и захвата новых, более изощренных атак, придуманных сообществом черных хакеров. Honeypot могут легко обнаружить новые атаки, потому что любое действие против такой системы является аномалией. Таким образом, honeypot можно использовать для сбора, управления и анализа большего количества данных об атаках.

Honeypot-система также может быть использована для получения информации о хакерской деятельности в рамках определения методов работы злоумышленников и, как результат, становится превентивной мерой против реально защищенной системы. На ранних этапах хакер сканирует сеть для поиска уязвимых компьютеров, в результате чего обнаруживает приманку, которая намеренно уязвима для привлечения атак. Если в дальнейшем злоумышленник попытается подключиться к honeypot, система немедленно обнаружит и зафиксирует действие, потому что обычный пользователь не должен взаимодействовать с системой.

Анализ литературы позволил классифицировать honeypots по разным основаниям. В частности, в зависимости от уровня взаимодействия honeypots подразделяются на три типа: с высоким, низким и средним уровнями взаимодействия [2].

Honeypot с высоким уровнем взаимодействия представляет собой полноценную систему для взаимодействия. Это означает, что honeypot с высокой степенью взаимодействия не просто имитирует операционную систему, сервисы, процессы и функции, а является максимально приближенной к реальной системе. Эта система предполагает получение злоумышленником полного контроля над системой honeypot, что может быть использовано для получения дополнительной информации об инструментах, тактике и мотивах атакующего [3].

Система honeypot с высоким уровнем взаимодействия может отвлекать внимание злоумышленника максимально эффективно от реальных ресурсов компании, поскольку предполагает расход большого количества ресурсов для обнаружения и компрометации данной системы.

У системы honeypot с высоким взаимодействием также есть недостатки. Например, злоумышленники могут использовать honeypot, которая была взломана, для проведения новых атак на другие системы, доступные в сети.

Система honeypot с низким уровнем взаимодействия обычно предоставляет только имитацию определенных сервисов. В результате honeypot с низким уровнем взаимодействия только эмулирует службу и регистрирует данные, которые в результате будут записаны на жесткий диск, однако не обеспечивает доступ к другим ресурсам на компьютере. Honeypot-система с низким уровнем взаимодействия имеет ряд преимуществ. Во-первых, данная система достаточно проста в настройке и обслуживании. Во-вторых, не требует значительных вычислительных мощностей и не может быть полностью скомпрометирована хакером. В результате риск использования honeypot с низким уровнем взаимодействия намного меньше, чем при использовании honeypot с высоким уровнем взаимодействия. С другой стороны, к недостаткам honeypot-систем с низким уровнем взаимодействия относят факт того, что происходит имитация реальной системы без предоставления злоумышленникам настоящей корневой оболочки.

Honeypot среднего взаимодействия. Этот тип honeypot создает иллюзию ложной операционной системы, с которой атакующий может взаимодействовать [4, 5], тем самым записывая все действия злоумышленника. Honeytrap является одной из honeypot-систем со средним взаимодействием.

Honeypot-системы могут работать как на реальных, так и на виртуальных машинах. В зависимости от этого выделяют физические и виртуальные системы honeypot. Так, физические системы теоретически обеспечивают больший уровень изоляции системы от остальной сети, но при этом требуют гораздо больших ресурсов для работы. Виртуальные же системы требуют меньше ресурсов, а также гораздо более удобны в настройке и обслуживании.

## 2. АРХИТЕКТУРА И РЕАЛИЗАЦИЯ ЗАЩИЩЕННОЙ HONEYPOT-СИСТЕМЫ

Архитектура системы предполагает наличие следующих ключевых особенностей [6–17]:

1) система honeypot должна удерживать злоумышленника в системе в течение длительного времени, что означает, что система должна быть максимально реалистичной;

2) система honeypot не должна раскрывать реальную сеть организации (граница honeypot должна быть прочной);

3) система не должна содержать официально опубликованных уязвимостей CVE, чтобы атакующие с большей вероятностью стали применять неопубликованные атаки нулевого дня на систему honeypot;

4) honeypot-система должна работать в среде виртуализации VMware с ОС Ubuntu с базовой конфигурацией и службами, требующими мониторинга.

Идея разработки заключается в создании системы, идентичной существующим, предполагающей многопоточную работу несколько различных служб.

Допущение: наличие уязвимости, используя которую злоумышленник имеет возможность проникнуть в систему.

Архитектура защищенной honeypot-системы предполагает два основных этапа – выбор служб для реализации honeypot-системы и установку (настройку) служб.

В качестве первой ступени привлечения внимания злоумышленников выступает установка и настройка веб-сервера. Далее следует работа на уровне FTP-сервера в рамках обмена файлами между компьютерами по локальной сети и Интернету. На сервере настроен только анонимный доступ без возможности загрузки / скачивания файлов, FTP-сервер будет доступен на порте X.

Другой системной службой, работающей в системе, является MySQL-сервер. MySQL-сервер, как система управления базами данных, будет доступен на порте Y.

Также в разрабатываемой системе будет работать системная служба SSH, которая позволит производить удаленное управление операционной системой. SSH-сервер будет доступен на порте Z.

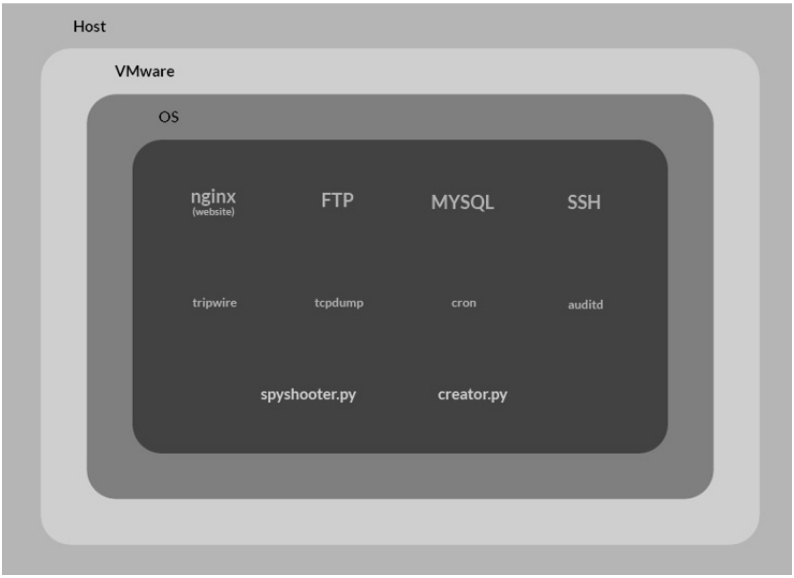
Для всех вышеперечисленных служб необходимым условием является открытие портов для имитации реальной системы.

В результате успешной brute-force атаки на службу SSH злоумышленник попадет на машину под пользователем с минимальными правами.

С этого момента все действия, которые предпримет злоумышленник, будут записаны при помощи различных средств: сниффер сетевых пакетов tcpdump (запишет и сохранит весь сетевой трафик), подсистема аудита auditd (запишет и сохранит журналы логирования системы), скрипт на языке Python для записи экрана (запишет экран в качестве видео со всеми действиями злоумышленника). После этого все файлы будут отправлены на удаленное хранилище.

Графически структура разработанной honeypot-системы представлена на рисунке.

При обращении потенциального злоумышленника к FTP-серверу отображается список файлов с разными именами, расширениями, размерами и датами создания. Это становится возможным за счет работы заранее прописанного скрипта, написанного на языке Python для привлечения внимания злоумышленника. Скрипт считывает имена и расширения из заранее подготовленных файлов согласно параметрам разработчика. Также он меняет размер и дату создания файла на случайную, чтобы они выглядели максимально правдоподобно.




Общая структура разработанной honeypot-системы

General structure of the developed honeypot system

Листинг скрипта может быть представлен в следующем виде:

```
import ....
BASE_DIR = 'docs'  # в какую папку сохранять файлы
WORDS_FILE = 'words.txt'
EXTENSION_FILE = 'extensions.txt'
SIZE_FROM = 1024  # размер файла не меньше значения (в байтах)
SIZE_TO = 8192  # размер файла не больше значения (в байтах)
def log(s):
    now = datetime.datetime.now()
    msg = '{}: {}'.format(now.strftime("%Y-%m-%d %H:%M:%S"), s)
    print(msg)
def load_file_content(fn):
    content = None
    try:
        with open(fn, 'r', encoding='utf-8') as file:
            content = file.read()
    except:
.....
    return content.splitlines()
def read_lists():
    if not os.path.isfile(WORDS_FILE):
        log(f'Не найден файл со списком слов!')
        sys.exit()
    words_list = load_file_content(WORDS_FILE)
    words_list = [x for x in words_list if x]
    if not words_list:
        log('Не удалось получить список слов!')
        sys.exit()

    if not os.path.isfile(EXTENSION_FILE):
        log(f'Не найден файл со списком расширений!')
        sys.exit()
    extensions_list = load_file_content(EXTENSION_FILE)
    extensions_list = [x for x in extensions_list if x]
    if not extensions_list:
        log('Не удалось получить список расширений!')
        sys.exit()
```

```
    return words_list, extensions_list
creator.py
def create_file(fn):
    file_size = random.randrange(SIZE_FROM, SIZE_TO)
    try:
        with open(fn, 'wb') as f:
            f.seek(file_size - 1)
            f.write(b'\0')
            return True
    except:
        log(f'Ошибка создания файла {fn}')
        sys.exit()

def change_time_creation(fn):
    file_year = str(random.randint(2000, 2020))
    file_month = str(random.randint(1, 7)).zfill(2)
    ……
```

Для документации действий злоумышленников в honeypot-системе был написан скрипт на языке Python, который с заданным интервалом делал скриншоты виртуальной машины и, сравнивая с предыдущим скриншотом, определял, происходит там что-то или нет. При выявлении активности скрипт включал запись экрана.

Алгоритм работы скрипта состоит из следующих действий:

1) подключение злоумышленника к системе при помощи SSH;

2) запуск скрипта;

3) формирование скриншота экрана;

4) дублирование скриншота экрана через фиксированный промежуток времени;

5) если скриншоты различны в результате изменения структуры:

5.1) старт записи экрана на определенное время;

5.2) сохранение записи в определенную папку и файл;

6) если ничего не произошло, то переходим к пункту 2.

Листинг скрипта может быть представлен в следующем виде:

```
def read_settings():
    config = configparser.ConfigParser()
    config.read("config.ini")
```

```
    def log(s):
        now = datetime.datetime.now()
        msg = '{}: {}'.format(now.strftime("%Y-%m-%d %H:%M:%S"), s)
        print(msg)
    def start_record(fn):
        log('Начата запись...')
        screen_size = (X1 - X0, Y1 - Y0)
        fourcc = cv2.VideoWriter_fourcc(*"XVID")
        out = cv2.VideoWriter(fn, fourcc, FPS, screen_size)
        cycle_time = FPS * DURATION
        for _ in range(cycle_time):
            img = pyautogui.screenshot(region=(X0, Y0, X1, Y1))
            frame = np.array(img)
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            out.write(frame)

        cv2.destroyAllWindows()
        out.release()
        log('Завершена запись видео.')
    def main():
        log('Начало работы!')
        read_settings()
        base_screenshot = pyautogui.screenshot(region=(X0, Y0, X1, Y1))
        base_screenshot = cv2.cvtColor(np.array(base_screenshot),
cv2.COLOR_RGB2BGR)
        while True:
            new_screenshot = pyautogui.screenshot(region=(X0, Y0, X1, Y1))
            new_screenshot = cv2.cvtColor(np.array(new_screenshot),
cv2.COLOR_RGB2BGR)
            log('Сделан новый скриншот...')
            res = cv2.absdiff(base_screenshot, new_screenshot)
            res = res.astype(np.uint8)
            percentage = round((np.count_nonzero(res) * 100) / res.size, 2)
            log(f'Разница между изображениями: {percentage}')
```

## ЗАКЛЮЧЕНИЕ

Концепция honeypot-систем является важным дополнением к сфере информационной безопасности. Разработанная honeypot-система предназначена для исследования поведения злоумышленника после его проникновения внутрь корпоративной системы. Для злоумышленников использовались такие системные службы, как веб-сервер Nginx, FTP-сервер, SSH-сервер для удаленного доступа к honeypot-системе, MYSQL-сервер, а также ряд утилит и скриптов. Разработанная honeypot-система предоставляет необходимый функционал для успешной работы, также ее архитектура позволяет при необходимости расширить спектр возможностей системы и задать вектор дальнейшего развития.

## СПИСОК ЛИТЕРАТУРЫ

1. *Diebold P., Hess A., Schäfer G.* A honeypot architecture for detecting and analyzing unknown network attacks // 14th Kommunikation in Verteilten Systemen (KiVS 2005), Kaiserslautern, 28. Februar – 3. März. – Berlin; Heidelberg: Springer, 2005. – P. 245–255.

2. An experimental study of SSH attacks by using honeypot decoys / E. Kheirkhah, S.M.P. Amin, H.A.J. Sistani, H. Acharya // Indian Journal of Science and Technology. – 2013. – Vol. 6. – P. 5567–5578.

3. *Tiwari R., Jain A.* Design and analysis of distributed honeypot system // International Journal of Computer Applications. – 2012. – Vol. 55, N 13. – P. 20–23.

4. *Mahajan S., Adagale A.M., Sahare C.* Intrusion detection system using Raspberry PI Honeypot in network security // International Journal of Engineering Science and Computing. – 2016. – Vol. 6. – P. 2792–2795.

5. *Provos N., Holz T.* Virtual honeypots: from botnet tracking to intrusion detection. – Upper Saddle River: Addison-Wesley, 2007.

6. Hack like no one is watching: using a honeypot to spy on attackers / L. Liu, K. Mahar, C. Virdi, H. Zhou // MIT Computer and Network Security Term Projects, 2016. – URL: http://docplayer.net/21979034-Hack-like-no-one-is-watching-using-ahoneypot-to-spy-on-attackers.html (accessed: 30.05.2021).

7. Applying deception mechanisms for detecting sophisticated cyber attacks / A Research Paper by TopSpin Security. – October, 2016. – URL: https://library.cyentia.com/report/report_002229.html (accessed: 30.05.2021).

8. *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. – P. 110–124. – DOI: 10.1504/IJSN.2009.023430.

9. *Jones H.M.* The restrictive deterrent effect of warning messages on the behavior of computer system trespassers. PhD Thesis / University of Maryland. – College Park, 2014.

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

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

12. *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.

13. *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). – Heidelberg: Springer, 2007. – DOI: 10.1007/978-3-540-74320-0_11. – (Lecture Notes in Computer Science; vol. 4637).

14. *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.

15. *Barros A.* DLP and honeytokens. – 2007, August 27. – URL: http://blog.securitybalance.com/2007/08/dlp-and-honeytokens.html (accessed: 30.05.2021).

16. *Sobesto B.* Empirical studies based on honeypots for characterizing attackers behaviour. PhD Thesis / University of Maryland. – College Park, 2015.

17. *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. – DOI: 10.1145/3099012.3099016.

***Архипова Анастасия Борисовна***, кандидат технических наук, доцент кафедры защиты информации Новосибирского государственного технического университета. Основные направления научных исследований – программное обеспечение научных задач, управление в социально-экономических систе-

мах, информационная безопасность. Имеет более 60 публикаций. E-mail: arhipova@corp.nstu.ru

***Каревский Данила Русланович***, лаборант кафедры защиты информации Новосибирского государственного технического университета. Основные направления научных исследований – программное обеспечение научных задач, информационная безопасность. E-mail: kaf_zi@corp.nstu.ru

DOI: 10.17212/2782-2230-2021-2-122-135

**Honeypot as a tool for creating an effective secure system***

**A.B. Arkhipova[1], D.R. Karevskiy[2]**

[1] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, candidate of technical sciences, associate professor of the Department of Information Security. E-mail: arhipova@corp.nstu.ru*
[2] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, laboratory assistant of the Department of Information Security. E-mail: kaf_zi@corp.nstu.ru*

In the presente work, the theoretical aspects of honeypot systems were considered, and the classification of honeypots on various grounds was presented. The architecture of a honeypot system is presented, designed to investigate the behavior of an attacker after his penetration into the corporate system, as a tool for implementing a complex effective secure system of the organization.

**Keywords:** honeypot system, information security, cyber threats, protected system

## REFERENCES

1. Diebold P., Hess A., Schäfer G. A honeypot architecture for detecting and analyzing unknown network attacks. *14th Kommunikation in Verteilten Systemen (KiVS 2005)*, Kaiserslautern, 28. Februar – 3. März. Berlin, Heidelberg, Springer, 2005, PP. 245–255. (In German).

2. Kheirkhah E., Amin S.M.P., Sistani H.A.J., Acharya H. An experimental study of ssh attacks by using honeypot decoys. *Indian Journal of Science and Technology*, 2013, vol. 6, pp. 5567–5578.

3. Tiwari R., Jain A. Design and analysis of distributed honeypot system. *International Journal of Computer Applications*, 2012, vol. 55, no. 13, pp. 20–23.

* *Received 22 February 2021.*

4. Mahajan S., Adagale A.M., Sahare C. Intrusion detection system using Raspberry PI Honeypot in network security. *International Journal of Engineering Science and Computing*, 2016, vol. 6, pp. 2792–2795.

5. Provos N., Holz T. *Virtual honeypots: from botnet tracking to intrusion detection*. Upper Saddle River, Addison-Wesley Professional, 2007.

6. 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 at: http://docplayer.net/21979034-Hack-like-no-one-is-watching-using-ahoneypot-to-spy-on-attackers.html (accessed 30.05.2021).

7. *Applying deception mechanisms for detecting sophisticated cyber attacks*. A research paper by TopSpin Security. October, 2016. Available at: https://library.cyentia.com/report/report_002229.html (accessed 30.05.2021).

8. 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. 110–124. DOI: 10.1504/IJSN.2009.023430.

9. Jones H.M. *The restrictive deterrent effect of warning messages on the behavior of computer system trespassers. PhD Thesis*. University of Maryland. College Park, 2014.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. Barros A. *DLP and honeytokens*. 2007, August 27. Available from: http://blog.securitybalance.com/2007/08/dlp-and-honeytokens.html (accessed 30.05.2021).

16. Sobesto B. *Empirical studies based on honeypots for characterizing attackers behaviour. PhD Thesis*. University of Maryland. College Park, 2015.

17. 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. DOI: 10.1145/3099012.3099016.

Для цитирования:

*Архипова А.Б., Каревский Д.Р.* Honeypot как инструмент создания эффективной защищенной системы // Безопасность цифровых технологий. – 2021. – № 2 (101). – С. 122–135. – DOI: 10.17212/2782-2230-2021-2-122-135.

For citation:

Arkhipova A.B., Karevskiy. Honeypot kak instrument sozdaniya effektivnoy zashishennoy systemi [Honeypot as a tool for creating an effective secure system]. *Bezopastnost cifrovih technologiy = Digital security*, 2021, no. 2 (101), pp. 122–135. DOI: 10.17212/2782-2230-2021-2-122-135.