

Отчёт по лабораторной работе №1

Кибербезопасность предприятия

Абакумова Олеся,
Герра Гарсия Максимиано Антонио,
Канева Екатерина,
Клюкин Михаил,
Ланцова Яна,
НФИбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Уязвимость 1. SQL-инъекция.	6
2.2	Последствие уязвимости 1	12
2.3	Уязвимость 2. Отключенная защита антивируса.	13
2.4	Последствие уязвимости 2. Admin meterpreter.	16
3	Выводы	19

Список иллюстраций

2.1	Найдено сканирование.	6
2.2	Описание	7
2.3	Найден exploit	7
2.4	Добавление инцидента по уязвимости 1.	8
2.5	Подключение к удалённому ПК.	9
2.6	Заходим на веб-портал.	9
2.7	Командная строка.	10
2.8	Поиск уязвимости.	10
2.9	Параметры уязвимой функции.	11
2.10	Исправляем уязвимость.	11
2.11	Устранили уязвимость 1.	12
2.12	Список установленных соединений.	12
2.13	Разрыв соединения с нарушителем.	12
2.14	Последствия первой уязвимости устранены.	13
2.15	Создали запись об инциденте.	14
2.16	Удаление записи DisableAntiSpyware в реестре.	14
2.17	Интерфейс Windows Defender.	15
2.18	Включение Real-time Protection.	15
2.19	Антивирус работает.	16
2.20	Соединение с машиной нарушителя.	16
2.21	Остановка процесса.	17
2.22	Логи подключений по RDP и успешная аутентификация.	17
2.23	Изменение пароля администратора.	17
2.24	Удаление привилегированного пользователя.	18
2.25	Все уязвимости устранены.	18

Список таблиц

1 Цель работы

Устранить уязвимости и последствия информационных систем Компании.

2 Выполнение лабораторной работы

2.1 Уязвимость 1.SQL-инъекция.

Начнём выполнение лабораторной, сразу зайдём в список событий и найдём информацию о произошедшем (рис. 2.1):

События

События за последние 24 часа

У...

Дата и время

ИД

Код собит...

К...

Название правила

Класс

13:20:34.348 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:20:05.108 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:19:28.013 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:19:00.016 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:18:19.167 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:17:51.424 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:17:22.625 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:17:21.578 19...	3227018	1	ET SCAN Behavioral Unusu...	network-scan
13:16:43.019 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:16:16.084 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:15:35.470 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:15:08.779 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit
13:14:36.825 19...	3200655	1	AM EXPLOIT Possible Goog...	client-side-exploit

« < Страница 1 > »

Событие 13:17:21.578 19.09.2025

Событие

Источник

Получатель

Пакет

Тип события	Сигнатурное событие
Протокол	TCP
Код события	3227018

Правило анализа

Класс

Группа

Название

Описание:

Текст:

Описание уязвимостей

network-scan

scan

ET SCAN Behavioral Unusually fast Terminal Server Traffic Potential Scan or Infection (Inbound) var1

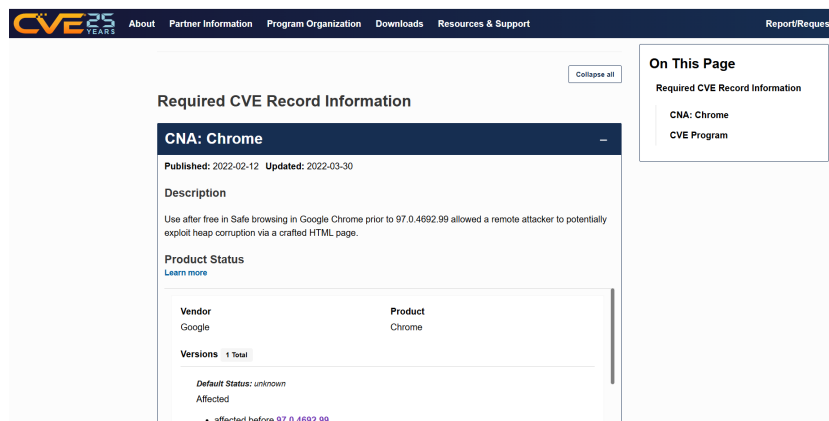
Правило обнаруживает факт сканирования

alert tcp SEXTERNAL_NET any -> \$HOME_NET 3389 (msg: "ET SCAN Behavioral Unusually fast Terminal Server Traffic Potential Scan or Infection (Inbound) var1";flow: to_server;flags: S,12;threshold: type both, track by_src, count 20, seconds 360;reference: url:doc.emergingthreats.net/2001972;classtype: network-scan;sid: 3227018;rev: 3;metadata: affected_asset dst, affected_os any, affected_product n/a, affected_vendor n/a, attack_target Client_and_Server, tlas_category Scan)

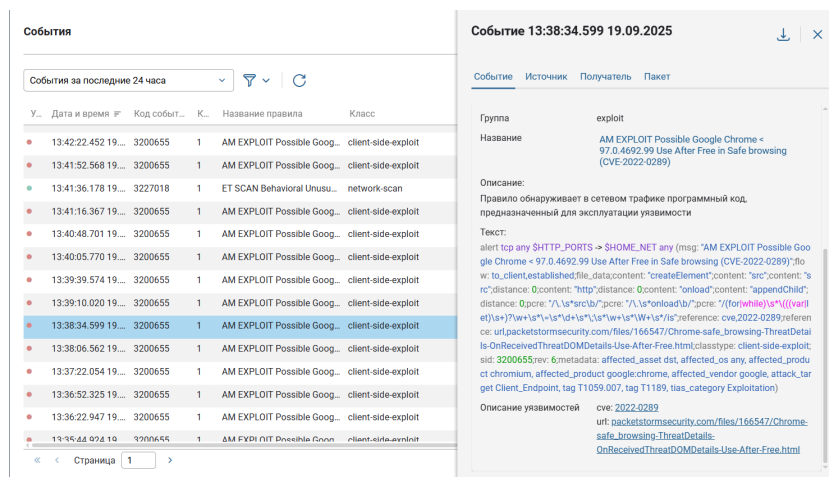
url: doc.emergingthreats.net/2001972

Рис. 2.1: Найдено сканирование.

Посмотрели описание на CVE (рис. 2.2):



Делаем вывод. Нарушитель проводит сканирование сети 195.239.174. Пока его мотивы неясны.



Найденные улики дают нам понять, что нарушитель сканирует веб-сервер на предмет SQL-инъекций утилитой sqlmap. Нарушитель генерирует php reverse shell, используя найденную SQL-инъекцию, загружает вредоносный файл на веб-сервер. Нарушитель генерирует письмо с вредоносным вложением и отправляет администратору. Администратор в свою же очередь открывает письмо

и тогда запускает вредоносный скрипт. Об этом как раз так свидетельствует наличие программного кода в сетевом трафике. Как итог, нарушитель получает контроль над компьютером администратора и meterpreter-сессию.

Изучили уязвимость, добавили инцидент (рис. 2.4):

Добавление инцидента

Название ⓘ
SQL-инъекция

Дата и время события ⓘ
19.09.2025 13:17

Источник ⓘ
195.239.174.96 (CMS Drupal) x

Пораженные активы ⓘ
10.10.4.11 (Manager Workstation 1) x

Описание ⓘ
Нарушитель проводит сканирование сети 195.239.174.0/24 и находит веб-сервер. Далее сканирует веб-сервер на предмет SQL-инъекций утилитой sqlmap. Нарушитель генерирует php

Рекомендации ⓘ
Известно, что \$id является уязвимым параметром, следует проверять тип данного параметра. Требуется найти место кода, где данный параметр считывается из GET запроса.

Индикаторы компрометации ⓘ
Сканирование сети

Прикрепить файл ⓘ
IDS_packet_time-2025-09-19T10_17_21.578737Z_ruleid-3227018.pcap x
Выберите файл

Рис. 2.4: Добавление инцидента по уязвимости 1.

На узле Web Server PHP находится уязвимый веб-сервер на 80 порту. Нарушитель использует данную уязвимость для загрузки и для выполнения php reverse shell. Для этого мы подключаемся к удаленному рабочему столу (рис. 2.5):

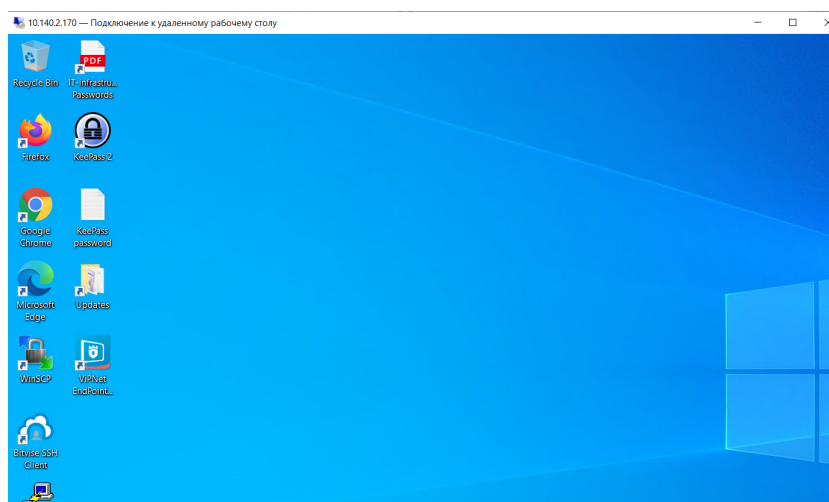


Рис. 2.5: Подключение к удалённому ПК.

С удалённого ПК зашли на веб-портал (рис. 2.6):

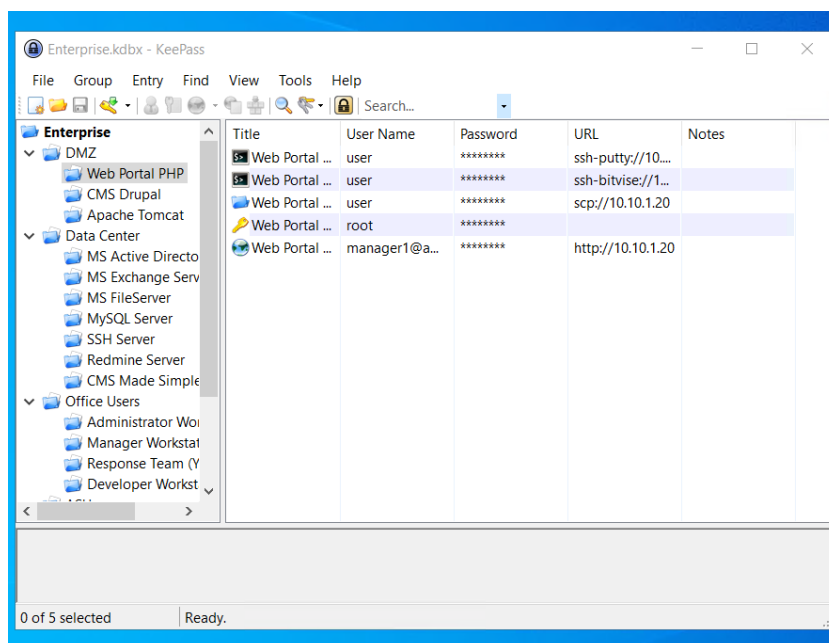
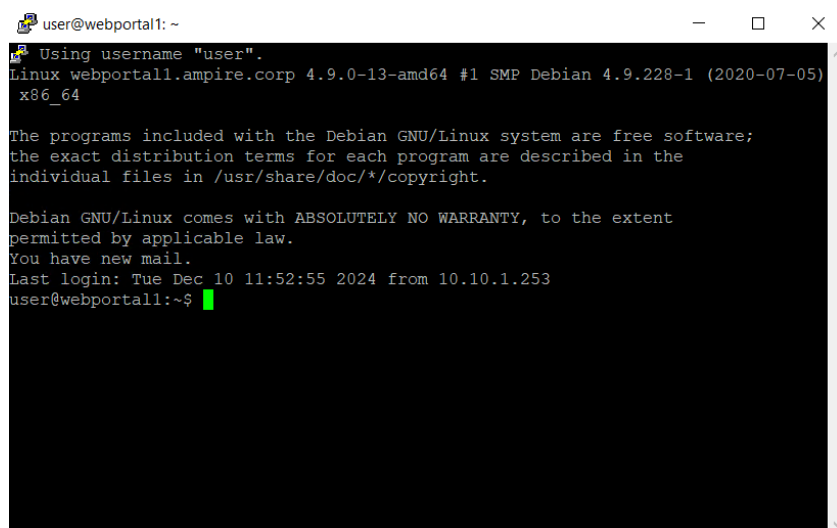


Рис. 2.6: Заходим на веб-портал.

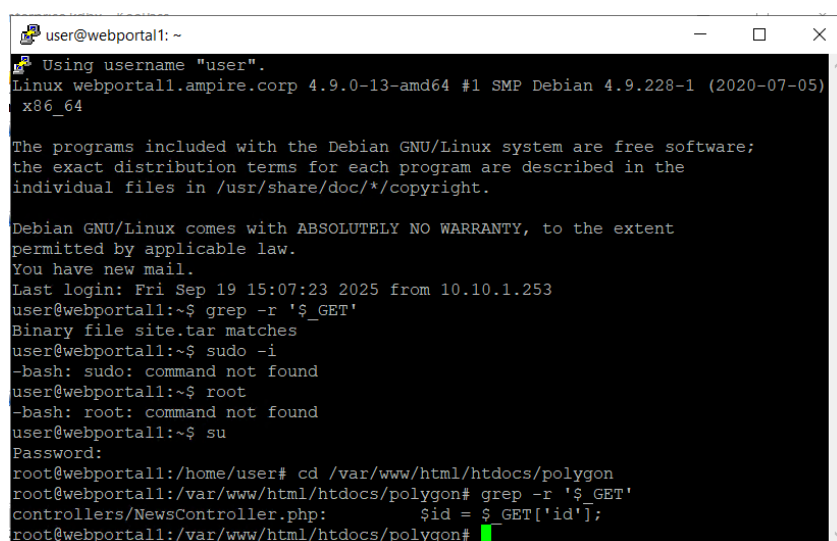
Далее для устранения уязвимости зашли в командную строку (рис. 2.7):



```
user@webportal1: ~  
Using username "user".  
Linux webportal1.ampire.corp 4.9.0-13-amd64 #1 SMP Debian 4.9.228-1 (2020-07-05)  
x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
You have new mail.  
Last login: Tue Dec 10 11:52:55 2024 from 10.10.1.253  
user@webportal1:~$
```

Рис. 2.7: Командная строка.

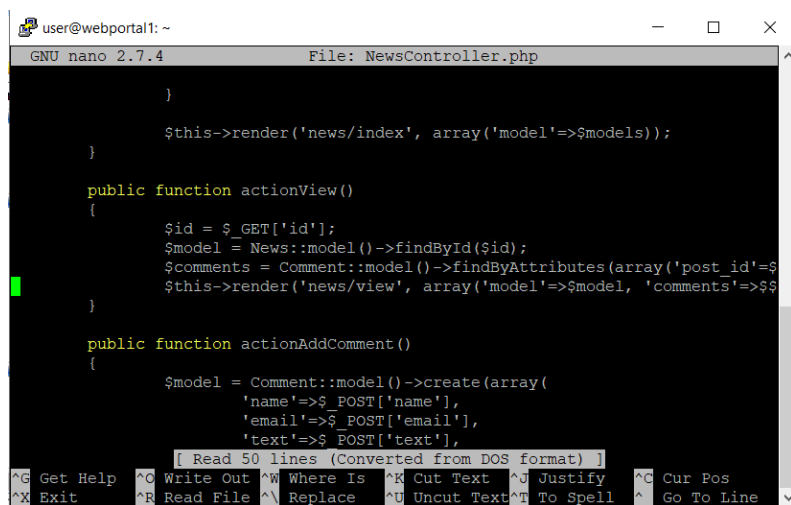
Известно, что `4id$` является уязвимым параметром, следует проверить тип данного параметра. Требуется найти место кода, где данный параметр считывается из GET запроса (рис. 2.8):



```
user@webportal1: ~  
Using username "user".  
Linux webportal1.ampire.corp 4.9.0-13-amd64 #1 SMP Debian 4.9.228-1 (2020-07-05)  
x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
You have new mail.  
Last login: Fri Sep 19 15:07:23 2025 from 10.10.1.253  
user@webportal1:~$ grep -r '$_GET'  
Binary file site.tar matches  
user@webportal1:~$ sudo -i  
-bash: sudo: command not found  
user@webportal1:~$ root  
-bash: root: command not found  
user@webportal1:~$ su  
Password:  
root@webportal1:/home/user# cd /var/www/html/htdocs/polygon  
root@webportal1:/var/www/html/htdocs/polygon# grep -r '$_GET'  
controllers/NewsController.php:         $id = $_GET['id'];  
root@webportal1:/var/www/html/htdocs/polygon#
```

Рис. 2.8: Поиск уязвимости.

Нашли место с уязвимостью - она в функции `actionView` (рис. 2.9):



```
user@webportal1: ~
GNU nano 2.7.4 File: NewsController.php

    }

    $this->render('news/index', array('model'=>$models));
}

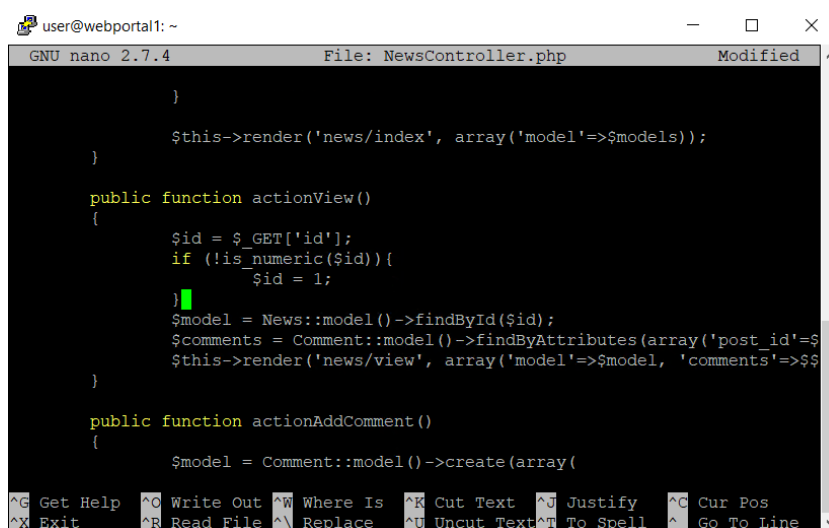
public function actionView()
{
    $id = $_GET['id'];
    $model = News::model()->findById($id);
    $comments = Comment::model()->findByAttributes(array('post_id'=>$
    $this->render('news/view', array('model'=>$model, 'comments'=>$$

    }

    public function actionAddComment()
    {
        $model = Comment::model()->create(array(
            'name'=>$_POST['name'],
            'email'=>$_POST['email'],
            'text'=>$_POST['text'],
            [ Read 50 lines (Converted from DOS format) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Рис. 2.9: Параметры уязвимой функции.

Устраним уязвимость, добавляя проверку типа параметра \$id (рис. 2.10):



```
user@webportal1: ~
GNU nano 2.7.4 File: NewsController.php Modified
    }

    $this->render('news/index', array('model'=>$models));
}

public function actionView()
{
    $id = $_GET['id'];
    if (!is_numeric($id)){
        $id = 1;
    }
    $model = News::model()->findById($id);
    $comments = Comment::model()->findByAttributes(array('post_id'=>$
    $this->render('news/view', array('model'=>$model, 'comments'=>$$

    }

    public function actionAddComment()
    {
        $model = Comment::model()->create(array(
            'name'=>$_POST['name'],
            'email'=>$_POST['email'],
            'text'=>$_POST['text'],
            [ Read 50 lines (Converted from DOS format) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Рис. 2.10: Исправляем уязвимость.

Проверили, что инцидент устранён (рис. 2.11), теперь будем устранять последствие:

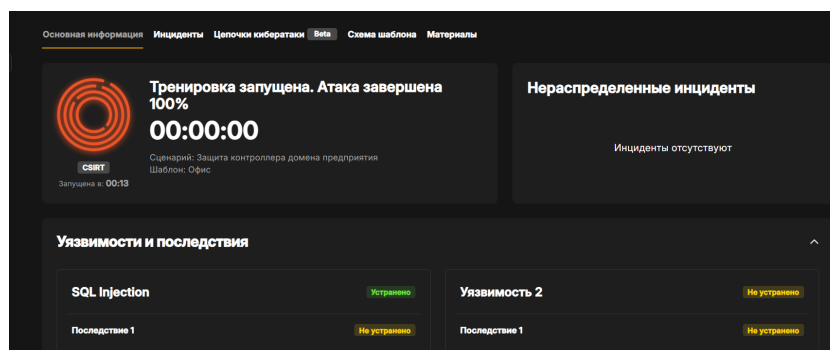


Рис. 2.11: Устранили уязвимость 1.

2.2 Последствие уязвимости 1

Нарушитель устанавливает shell сессию с веб-порталом РНР. Для обнаружения последствия необходимо проверить сокет уязвимой машины при помощи утилиты ss с ключами -tp (рис. 2.12):

```

user@webportal1: ~
components controllers images js shell.php
config css index.php models views
root@webportal1:/var/www/html/htdocs/polygon# cd controllers
root@webportal1:/var/www/html/htdocs/polygon/controllers# ls
NewsController.php SiteController.php
root@webportal1:/var/www/html/htdocs/polygon/controllers# nano NewsController.php
root@webportal1:/var/www/html/htdocs/polygon/controllers# ss -tp
State Recv-Q Send-Q Local Address:Port Peer Address:Port
ESTAB 0 0 10.10.1.20:tpoxy 10.10.1.253:7960
users: (("server",pid=644,fd=8))
ESTAB 0 0 10.10.1.20:44342 195.239.174.11:1085
users: (("chisel.sh",pid=7822,fd=11))
ESTAB 0 0 10.10.1.20:56240 10.10.1.25:5044
users: (("filebeat",pid=695,fd=5))
ESTAB 0 0 10.10.1.20:51720 195.239.174.11:4444
users: (("chisel.sh",pid=7822,fd=3), ("sh",pid=7821,fd=3), ("f0NNR",
pid=7252,fd=3))
ESTAB 0 272 10.10.1.20:ssh 10.10.1.253:24376
users: (("sshd",pid=7351,fd=4), ("sshd",pid=7342,fd=4))
ESTAB 0 0 10.10.1.20:58298 10.10.2.17:25004
users: (("epp_agentd",pid=32334,fd=37))
root@webportal1:/var/www/html/htdocs/polygon/controllers#

```

Рис. 2.12: Список установленных соединений.

Завершаем сессию нарушителя с помощью команды ss и параметром -K (рис. 2.13):

```

root@webportal1:/var/www/html/htdocs/polygon/controllers# ss -K dst '195.239.174.11' dport = 4444
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
tcp ESTAB 0 0 10.10.1.20:51720 195.239.174.11:4444
root@webportal1:/var/www/html/htdocs/polygon/controllers#

```

Рис. 2.13: Разрыв соединения с нарушителем.

Таким образом, устранили последствия первой уязвимости и закрыли инцидент (рис. 2.14):

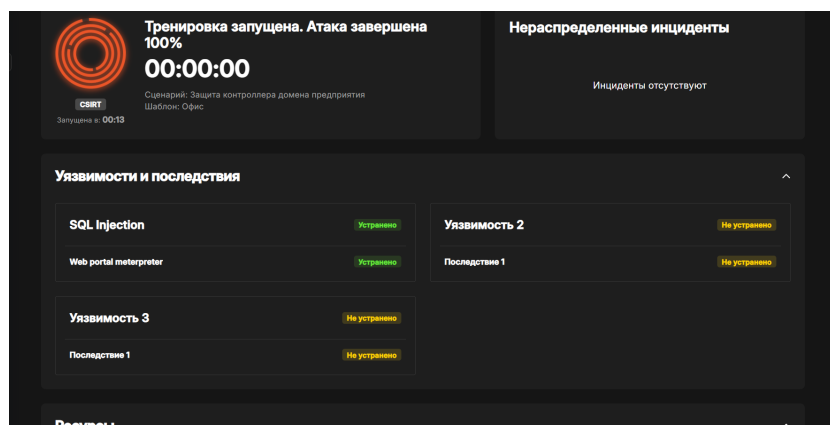


Рис. 2.14: Последствия первой уязвимости устранены.

2.3 Уязвимость 2. Отключенная защита антивируса.

На узле администратора выключена защита в реальном времени Windows Defender, что дает нарушителю возможность получить контроль над компьютером администратора при запуске им вредоносного скрипта diag.ps1.

Создаём запись об инциденте (рис. 2.15):

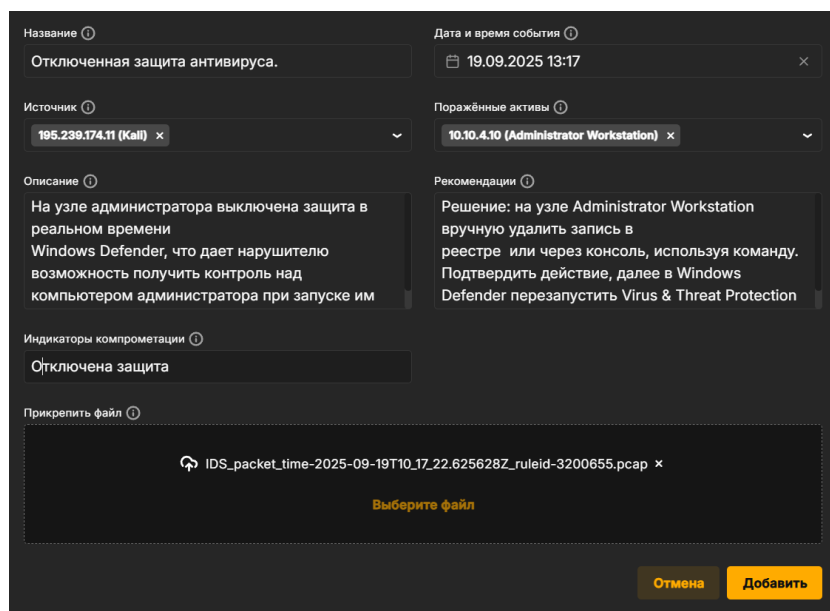


Рис. 2.15: Создали запись об инциденте.

Далее, поскольку на узле Administrator Defender антивирус был отключен, вручную удаляем запись (рис. 2.16), подтверждаем действие, в Windows Defender перезапускаем Virus & Threat Protection (рис. 2.17), включаем Real-time Protection (рис. 2.18), проверяем, что всё работает (рис. 2.19):

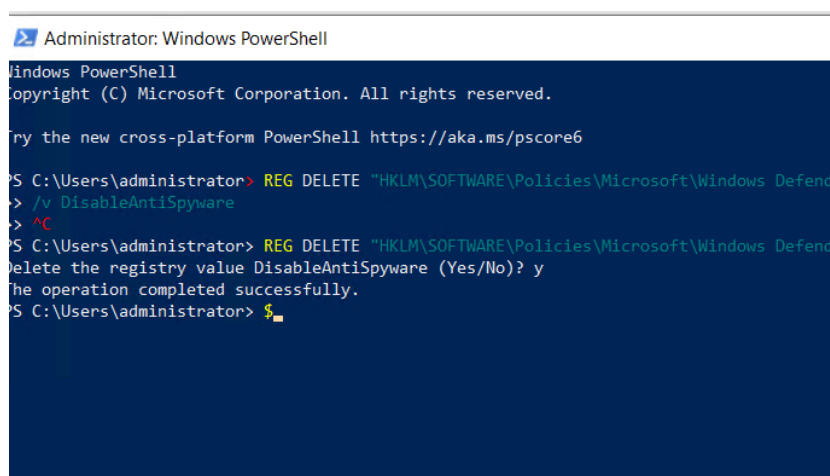


Рис. 2.16: Удаление записи DisableAntiSpyware в реестре.

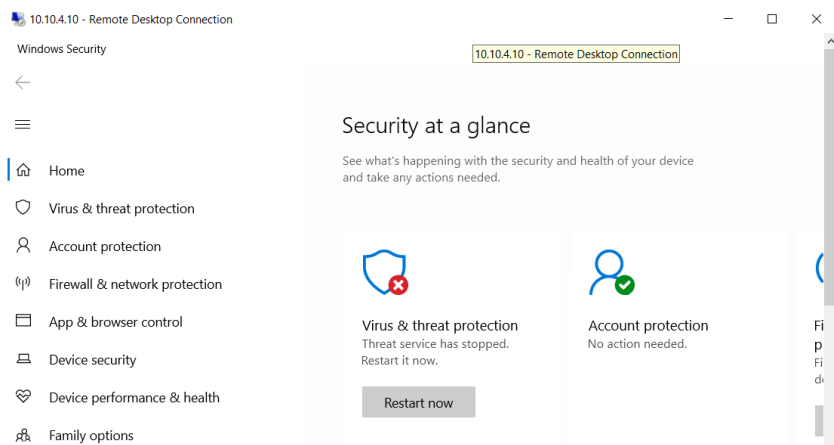


Рис. 2.17: Интерфейс Windows Defender.

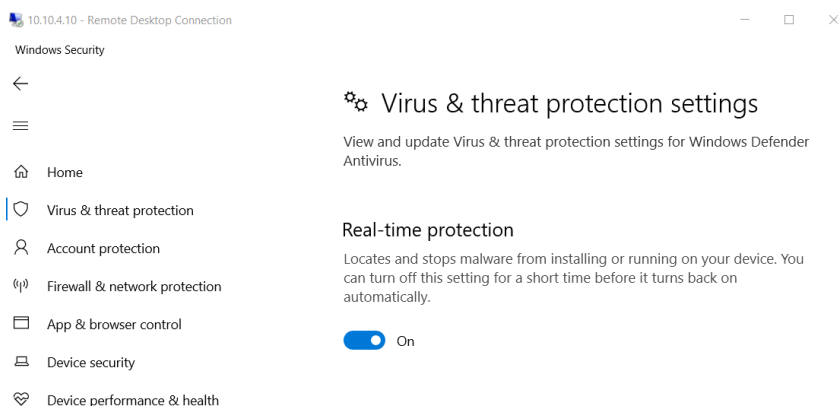


Рис. 2.18: Включение Real-time Protection.

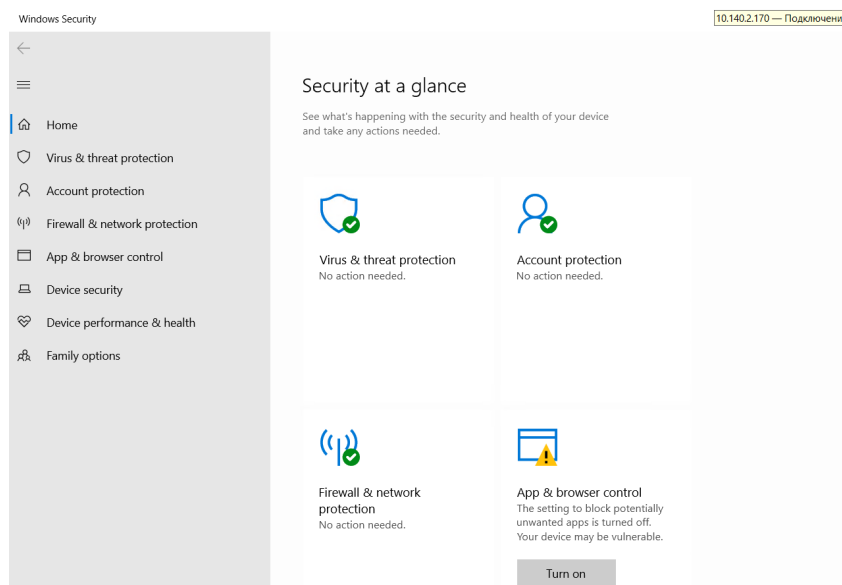


Рис. 2.19: Антивирус работает.

После выполненных действий необходимо перезагрузить Windows.

2.4 Последствие уязвимости 2. Admin meterpreter.

Далее устраняем последствия. Проверяем, что сейчас соединение нарушителя есть (рис. 2.20):

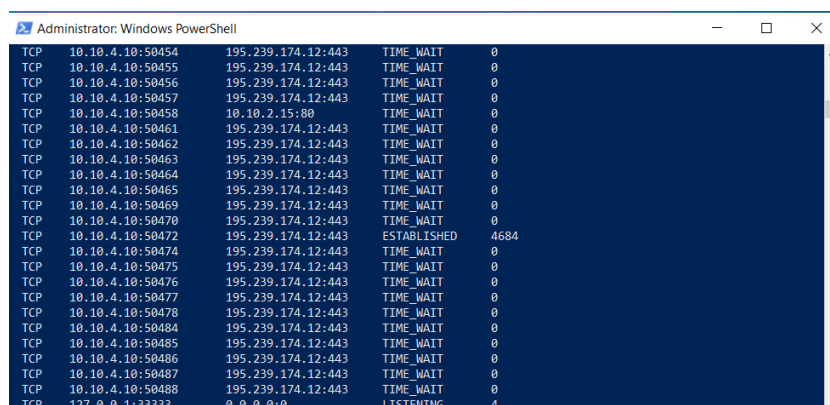


Рис. 2.20: Соединение с машиной нарушителя.

Разрываем соединение с помощью команды taskkill (рис. 2.21):


```
PS C:\Users\administrator> taskkill /f /pid 4684
SUCCESS: The process with PID 4684 has been terminated.
PS C:\Users\administrator>
```

Рис. 2.21: Остановка процесса.

Далее устраняем проблему, связанную со слабым паролем. На узле MS Active Directory установлен слабый пароль к учётной записи администратора, что позволяет нарушителю подобрать пароль, смотрим логи подключений по RDP (рис. 2.22):

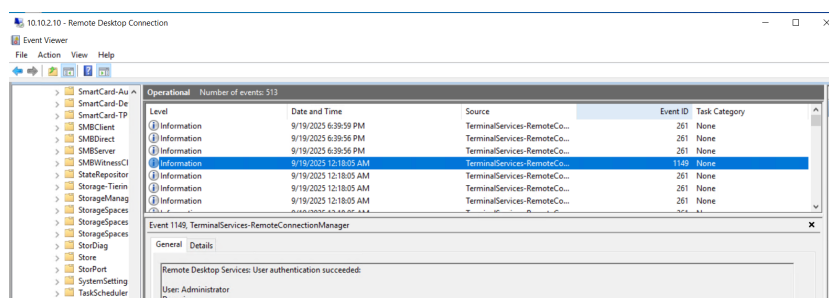


Рис. 2.22: Логи подключений по RDP и успешная аутентификация.

Устраним уязвимость, сменив пароль администратора (рис. 2.23):

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> net user Administrator *
Type a password for the user:
Retype the password to confirm:
The passwords do not match.

Type a password for the user:
Retype the password to confirm:
The command completed successfully.

PS C:\Users\Administrator>
```

Рис. 2.23: Изменение пароля администратора.

Переходим к устранению последствий. Был создан новый привилегированный пользователь, находим его в Administrative Tools - Active Directory Users and computers, во вкладке Users и удаляем (рис. 2.24):

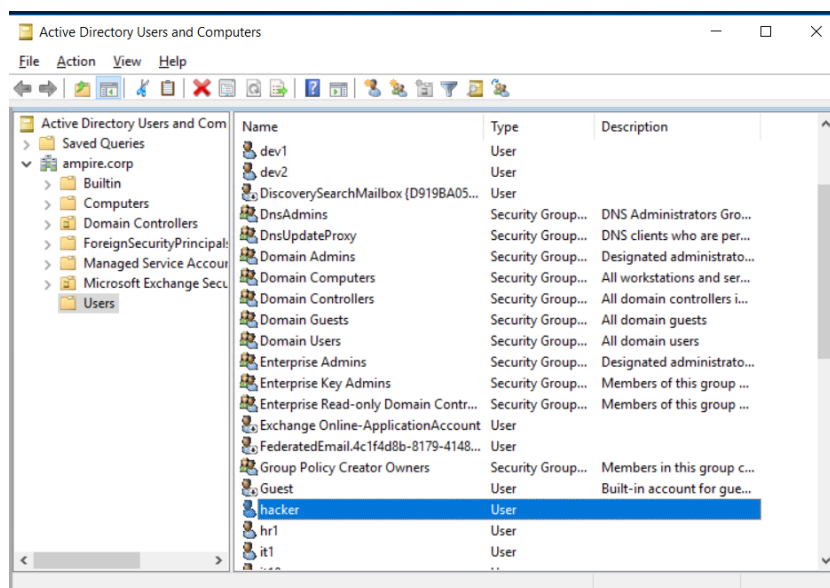


Рис. 2.24: Удаление привилегированного пользователя.

Проверяем, что все уязвимости и их последствия устранены (рис. 2.25):

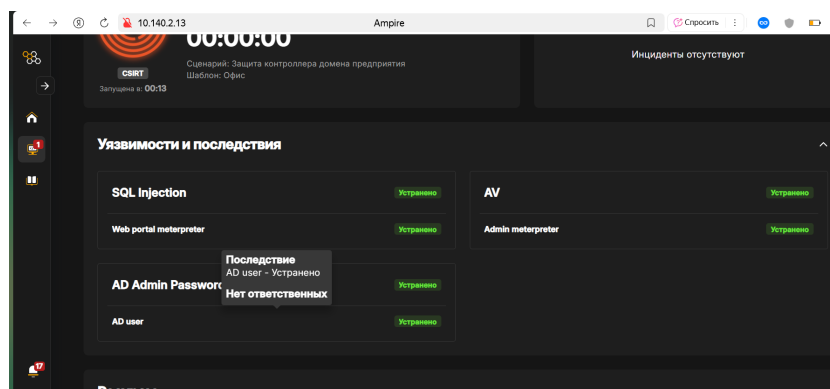


Рис. 2.25: Все уязвимости устранены.

3 Выводы

Устранили уязвимости сайта Компании.