

Отчет по лабораторной работе №1-С «ЗАЩИТА НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ ПРЕДПРИЯТИЯ»

Кибербезопасность предприятия

Александрова У.В., Волгин И.А., Голощапов Я.В.,
Дворкина Е.В., Серегина И.А.

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	9
4.1 Обнаружение уязвимостей	9
4.2 Устранение уязвимостей и их последствий	21
4.3 Результат	28
5 Выводы	30
Список литературы	31

Список иллюстраций

4.1	Установка фильтров	10
4.2	Обнаружение входа в систему	11
4.3	Карточка инцидента	12
4.4	Детектирование создания неизвестного файла	12
4.5	Обнаружение файла с backdoor	13
4.6	Карточка инцидента	13
4.7	Обнаружение эксплуатации уязвимости	15
4.8	Обнаружение эксплуатации уязвимости через логи	16
4.9	Карточка уязвимости	16
4.10	Общее описание уязвимости	17
4.11	Обнаружение REST API	17
4.12	Упоминание пользователя в коде	18
4.13	Подозрительный пользователь с правами администратора	18
4.14	Карточка инцидента	18
4.15	SQL-запросы	19
4.16	Карточка инцидента	20
4.17	Источник информации об уязвимости	20
4.18	Пользователь dev1	21
4.19	Смена пароля	22
4.20	Запуск исполняемого файла в планировщике	23
4.21	Запуск исполняемого файла в планировщике	23
4.22	Путь к исполняемому файлу	24
4.23	Пример добавления кода, выводящего на экран надпись XSS в wiki-страницу	24
4.24	Содержимое файла redcloth3.rb	25
4.25	Исправления в файле redcloth3.rb	26
4.26	Успешное устранение уязвимости	26
4.27	Список пользователей Redmine	26
4.28	SQL-запрос	27
4.29	Содержимое файла quety.rb до исправления уязвимости	28
4.30	Содержимое файла quety.rb с исправлением уязвимости	28
4.31	Закрытые инциденты	28
4.32	Устраненные уязвимости и последствия	29

Список таблиц

1 Цель работы

Целью данной лабораторной работы является освоение практических навыков выявления, анализа и устранения уязвимостей информационных систем в рамках сценария «Защита научно-технической информации предприятия».

2 Задание

- Обнаружить, проанализировать и закрыть уязвимости:
 1. Слабый пароль пользователя;
 2. XSS (CVE-2019-17427);
 3. Blind SQL (CVE-2019-18890).
- Определить и устранить последствия эксплуатации уязвимостей:
 1. Developer backdoor (последствие уязвимости 1);
 2. Redmine User (последствие уязвимости 2).
- Разработать и применить меры по устраниению выявленных уязвимостей.

3 Теоретическое введение

VipNet IDS (Intrusion Detection System) — это система обнаружения вторжений для защиты информационных систем от несанкционированного доступа и атак. VipNet IDS предназначена для мониторинга сетевого трафика в реальном времени, выявления подозрительной активности, анализа сигнатур известных атак и аномального поведения, а также генерации оповещений и рекомендаций по реагированию. Система интегрируется с другими компонентами комплекса VipNet, обеспечивая многоуровневую защиту корпоративной инфраструктуры [6].

Слабый пароль пользователя — типичная проблема конфигурации, при которой используется легко угадываемый или стандартный пароль (например, «admin», «123456» и т.п.). Такая уязвимость позволяет злоумышленнику получить несанкционированный доступ к учётной записи и, как следствие, к системе в целом.

XSS (межсайтовый скрипting), CVE-2019-17427 — уязвимость в веб-приложении Redmine, позволяющая внедрять вредоносный JavaScript-код в веб-страницы, просматриваемые другими пользователями. Эксплуатация этой уязвимости может привести к краже сессионных cookie, подмене контента или выполнению действий от имени жертвы.

Blind SQL Injection, CVE-2019-18890 — уязвимость в Redmine, связанная с недостаточной фильтрацией пользовательского ввода, что позволяет злоумышленнику выполнять произвольные SQL-запросы к базе данных через слепые (blind) методы. Последствия включают извлечение, модификацию или удаление конфиденциальных данных, а также потенциальный выход за пределы

приложения (например, выполнение команд ОС при определённых условиях).

Developer backdoor — наличие скрытого (часто не задокументированного) входа, оставленного разработчиком, может позволить злоумышленнику обойти стандартные механизмы аутентификации и получить полный контроль над системой.

Redmine User — компрометация учётной записи пользователя Redmine (например, через XSS или слабый пароль) даёт доступ к проектной документации, задачам, исходному коду и другим конфиденциальным данным, что особенно критично в контексте защиты научно-технической информации предприятия.

4 Выполнение лабораторной работы

4.1 Обнаружение уязвимостей

Уязвимости и последствия будут детектироваться в основном с помощью ViPNet IDS NS, далее записываться в карточки инцидентов [5].

Для обнаружения актуальной подозрительной активности пользуемся фильтрами по дате, времени и важности (рис. 4.1).

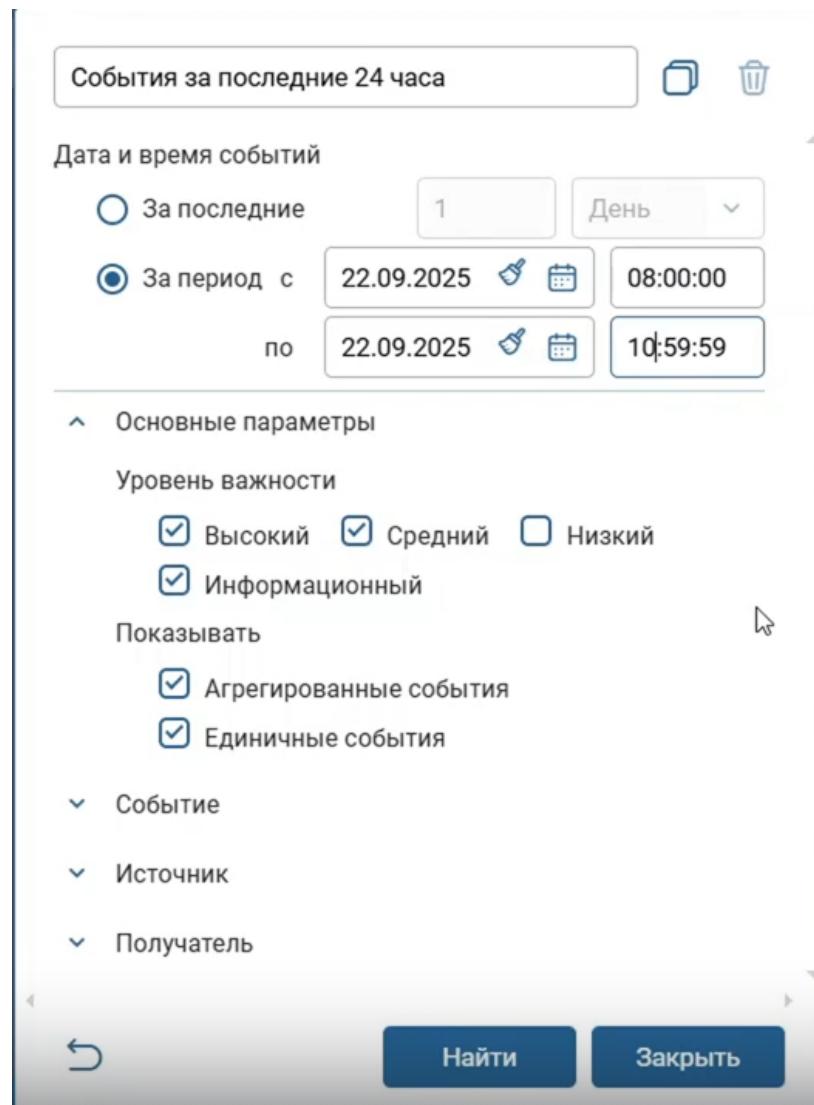


Рисунок 4.1: Установка фильтров

4.1.1 Обнаружение уязвимости «Слабый пароль пользователя»

По сценарию нам известно, что

1. Внутренний нарушитель подбирает пароль на файловый сервер и меняет существующий на сервере файл другим файлом с backdoor (дефектом алгоритма).
2. Пользователь Dev-1 загружает и запускает файл с backdoor.

3. Внутренний нарушитель получает контроль над компьютером пользователя Dev-1 и загружает скрипт для похищения учетных данных из браузера. Запускает данный скрипт, получает логин и пароль к Redmine.

Сначала было обнаружено сканирование системы и множество одинаковых событий со стороны 10.10.4.11 (узел менеджера) на 10.10.2.12 (файловый сервер), что может указывать на попытки подбора пароля. Далее Замечено событие класса «successful-admin», которое указывает на то, что пользователь успешно вошел в систему, подбрав пароль (рис. 4.2). Кроме того, видим действия между узлами 10.10.4.11, 10.10.2.12, 10.10.4.13 (Developer Workstation), которые также указывают, что был скомпрометирован пароль от компьютера Developer Workstation.

А в пакетах к этому событию увидим: Windows PowerShell running as user dev1 on DEV-ARM-01. Еще одно подтверждение.

The screenshot shows a security log interface with two main panes. The left pane displays a list of events over the last 24 hours, with columns for Date and time, Code of the event, ID, Rule name, and Class. Most events are related to SMB2 NT Create and ET POLICY. One event is highlighted in blue, showing it was triggered by a successful admin. The right pane is a detailed view of this specific event (ID 2020084) from 15.09.2025 at 00:11:09.801. It includes sections for General information (Date and time, Interface, Priority, Type, Protocol, Event ID), Analysis rule (Class, Group, Name, Description), and Text (Raw event data).

События	Событие 00:11:09.801 15.09.2025				
					Событие Источник Получатель Пакет
События за последние 24 часа					
Y_ Дата и время	Код события	Ко...	Название правила	Класс	
00:11:10.015 15.09.. 2025707	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:11:10.015 15.09.. 2025707	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:11:10.009 15.09.. 2044771	1		ET INFO PowerShell Command ...	misc-activity	
00:11:09.801 15.09.. 2038605	1		ET ATTACK_RESPONSE Nishan...	bad-unknown	
00:11:09.801 15.09.. 2020084	1		ET ATTACK_RESPONSE Micros...	successful-admin	
00:11:10.103 15.09.. 2025701	5		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:52.808 15.09.. 2025699	1		ET POLICY SMB Executable File...	bad-unknown	
00:10:52.808 15.09.. 2025699	1		ET POLICY SMB Executable File...	bad-unknown	
00:10:52.799 15.09.. 2025701	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:52.692 15.09.. 2025701	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:52.592 15.09.. 2025701	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:52.468 15.09.. 2025701	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:52.103 15.09.. 2025701	1		ET POLICY SMB2 NT Create An...	bad-unknown	
00:10:51.144 15.09.. 2025699	1		ET POLICY SMB Executable File...	bad-unknown	

Рисунок 4.2: Обнаружение входа в систему

Заполним карточку инцидента. Так как пароль был подобран, то уязвимость - слабый пароль пользователя. Рекомендации по устранению - изменить пароль на более сложный (рис. 4.3).

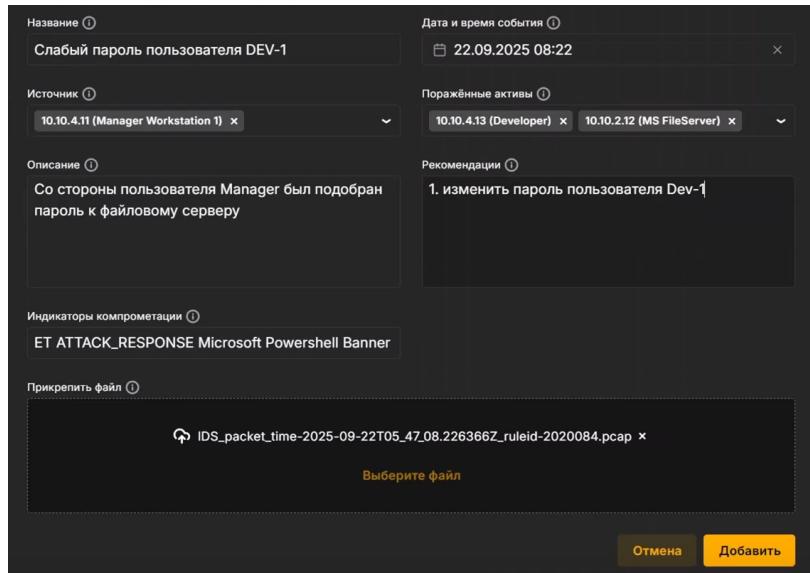


Рисунок 4.3: Карточка инцидента

4.1.2 Обнаружение последствия «Developer backdoor»

В последующих событиях можем видеть создание исполняемого файла на файловом сервере 10.10.2.12 и обращения к нему (рис. 4.4)

События				Событие 08:22:00.494 22.09.2025
				Событие Источник Получатель Пакет
				Общая информация
События за последние 24 часа	▼	▼	С	
...	Дата и время	Коли...	Название правила	
08:22:02.940 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:02.940 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:01.697 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:01.697 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:00.997 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:00.996 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:00.881 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.881 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.881 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.881 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.494 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.494 22.09.2025	1		ET POLICY SMB2 NT Create AndX Request For a .bat File	
08:22:00.456 22.09.2025	1		AM EXPLOIT SMB2 Possible Good Google < 9.7.0.4692.99 Use After ...	
08:22:00.046 22.09.2025	1		ET ATTACK_RESPONSE Nishang Invoke-PowerShellTcp Shell Pro...	

Рисунок 4.4: Детектирование создания неизвестного файла

Также находим событие класса trojan-activity (LaZagne), указывающее на часть атаки, запускающую программу для извлечения информации из браузера (рис. 4.5).

LaZagne – это приложение с открытым исходным кодом, используемое для получения множества паролей, хранящихся на локальном компьютере [4].

Событие	Событие 08:22:12.283 22.09.2025
События за последние 24 часа	
У... Дата и время IP Коли... Название правила	
08:22:24.625 22.09.2025 1 AM POLICY Requests Suspicious Python's User Agent	
08:22:24.581 22.09.2025 1 AM POLICY Requests Suspicious Python's User Agent	
08:22:24.581 22.09.2025 1 AM POLICY Requests Suspicious Python's User Agent	
08:22:24.529 22.09.2025 1 AM POLICY Requests Suspicious Python's User Agent	
08:22:24.528 22.09.2025 1 AM POLICY Requests Suspicious Python's User Agent	
08:22:21.331 22.09.2025 1 ET ATTACK_RESPONSE LaZagne Artifact Outbound in FTP	
08:22:21.331 22.09.2025 1 ET ATTACK_RESPONSE LaZagne Artifact Outbound in FTP	
08:22:12.283 22.09.2025 1 ET ATTACK_RESPONSE LaZagne Artifact Outbound in FTP	
08:22:06.270 22.09.2025 1 ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:06.269 22.09.2025 1 ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:02.940 22.09.2025 1 ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:02.940 22.09.2025 1 ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:01.697 22.09.2025 1 ET POLICY SMB2 NT Create AndX Request For an Executable File	
08:22:01.697 22.09.2025 1 FT POLICY SMB2 NT Create AndX Request For an Executable File	
« < Страница 3 > »	
Правила анализа	
Класс	trojan-activity
Группа	attack_response
Название	ET ATTACK_RESPONSE LaZagne Artifact Outbound in FTP
Описание:	Правило обнаруживает ответную (аторичную) часть атаки (например, попытку запустить произвольный код на ранее взломанный ресурс или отправку команды управления на такой ресурс) ¹²
Текст:	alert tcp \$HOME_NET any -> \$EXTERNAL_NET 1024: (msg:'ET ATTACK_RESPONSE LaZagne Artifact Outbound in FTP';flow: established,to_server;content: "The LaZagne Project";fast_pattern;reference: url/github.com/AlessandroZ/LaZagne;class type: trojan-activity;id: 2027151;rev: 2;metadata: affected_asset src affected_protocol microsoft_windows affected_vendor microsoft attack_target Client_Endpoint t confidence Medium created_at 2019-04-04 deployment Perimeter malware_famil LaZagne malware_family_Stealer signature_severity Major tias_category Exploration updated_at 2019-04-04)
Описание уязвимости	url: github.com/AlessandroZ/LaZagne

Рисунок 4.5: Обнаружение файла с backdoor

Заполним карточку инцидента. Первичные рекомендации по устранению последствия - удаление исполняемого файла с backdoor и остановка его работы (рис. 4.6).

Добавление инцидента	
Название ⓘ	Файл с backdoor
Дата и время события ⓘ	22.09.2025 08:22
Источник ⓘ	10.10.4.11 (Manager Workstation 1)
Поражённые активы ⓘ	10.10.4.13 (Developer)
Описание ⓘ	Запуск загруженного нарушителем файла с backdoor для похищения учетных данных из браузера
Индикаторы компрометации ⓘ	ET ATTACK_RESPONSE LaZagne Artifact Outbound
Рекомендации ⓘ	1. Удалить файл
Прикрепить файл ⓘ	IDS_packet_time-2025-09-22T05_22_12.283765Z_ruleid-2027151.pcap
Выберите файл	

Рисунок 4.6: Карточка инцидента

4.1.3 Обнаружение уязвимости «XSS (CVE-2019-17427)»

4. Внутренний нарушитель проводит атаку stored XSS для включения на Redmine сервере REST API. Вредоносный код записывается на Wikistrаницу проекта Dev1. Получив доступ к консоли администратора, внутренний нарушитель создает нового пользователя Redmine с правами администратора.

Далее видим события с источником - 10.10.4.11 (предполагаемый нарушитель, менеджер), получателем - 10.10.2.15 (сервер Redmine, к нему по сценарию нарушитель получил доступ). Это события AM EXPLOIN Possible Redmine <v4.0.4 XSS и событие AM EXPLOIT Generic Possible XSS in HTTP Body: они говорят о том, что эксплуатируется уязвимость Redmine, существующая в версиях до 4.0.4, позволяющая внедрять вредоносный JavaScript-код в веб-страницы, просматриваемые другими пользователями (рис. 4.7).

Также в пакетах к этому событию увидим:

```
POST /projects/dev1/wiki/Wiki/ HTTP/1.1
Host: redmine.ampire.corp
User-Agent: python-requests/2.31.0
...
Content-Type: multipart/form-data;
```

Это http-запрос на сервер Redmine с отправкой формы с данными

```
--8148d7a1c05d09d7c466c127bc05f0b7
Content-Disposition: form-data; name="content[text]"
<pre onfocusin="let xhr=new XMLHttpRequest;xhr.onreadystatechange=function(){if(xhr.readyState==4){var type='application/x-www-form-urlencoded'}},body:'utf8=?&authenticity_token=5e333333333333333333333333333333',type,'application/x-www-form-urlencoded'),n.onreadystatechange=function(){if(n.readyState==4){var type='application/x-www-form-urlencoded'}},body:'utf8=?&authenticity_token=5e333333333333333333333333333333',type,'application/x-www-form-urlencoded')--8148d7a1c05d09d7c466c127bc05f0b7
```

Content-Disposition: form-data; name="content[comments]"

--8148d7a1c05d09d7c466c127bc05f0b7

Content-Disposition: form-data; name="commit"

Save

--8148d7a1c05d09d7c466c127bc05f0b7

Content-Disposition: form-data; name="attachments[dummy][file]"; fil

Content-Type: application/octet-stream

А это изменение кода и коммит с сохранением (Save) изменений в коде.

The screenshot displays the NetworkMiner interface. The left pane shows a list of events with columns for Date & Time, IP, and Description. The right pane is a detailed view of a specific event from 08:47:32.437 on 22.09.2025. The detailed view includes sections for General Information (including source and destination IP, protocol, and priority), Rule Analysis (class: web-application-attack, group: exploit, name: AM EXPLOIT Possible Redmine < v4.0.4 XSS (CVE-2019-17427)), and Description (mentioning Python Requests User Agent). The event description in the detailed view states: "Правило обнаруживает в сетевом трафике программный код, позволяющий злоумышленнику внедрять скрипты на страницах сайта".

Рисунок 4.7: Обнаружение эксплуатации уязвимости

Также можем зайти через SSH подключение (работа в терминале) на сервер Redmine и проверить файл production.log (`cd /var/www/redmine-3.4.4/..., cat production log`) (рис. 4.8). В этом файле увидим сильно отличающиеся от остальных этапы, говорящие о внедрении JS кода.

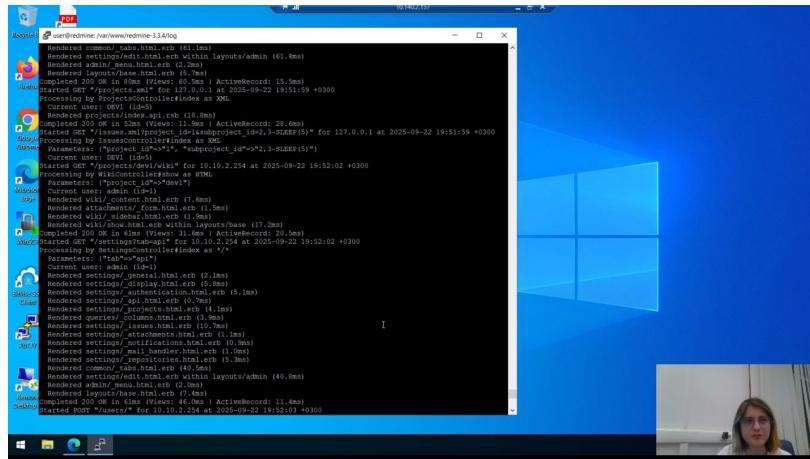


Рисунок 4.8: Обнаружение эксплуатации уязвимости через логи

Составляем карточку уязвимости (рис. 4.9), общее описание и рекомендации можно найти на сайте AMTIP [1] (рис. 4.10).

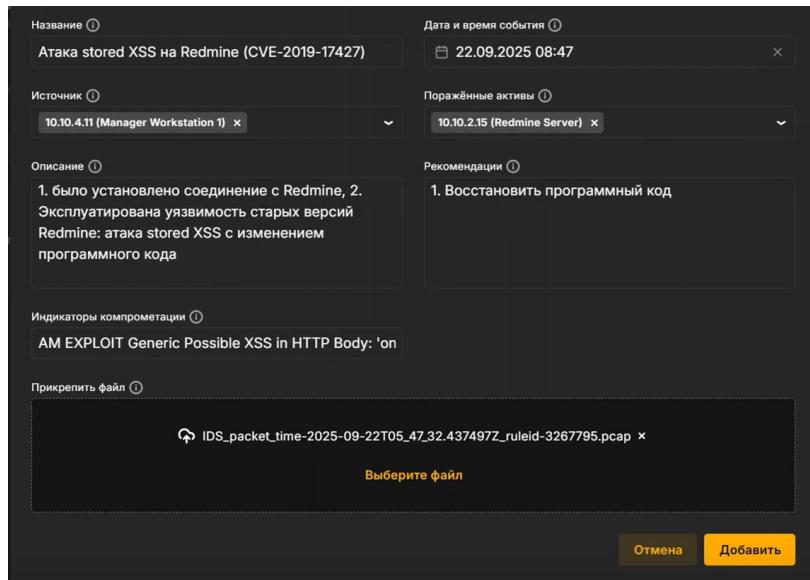


Рисунок 4.9: Карточка уязвимости

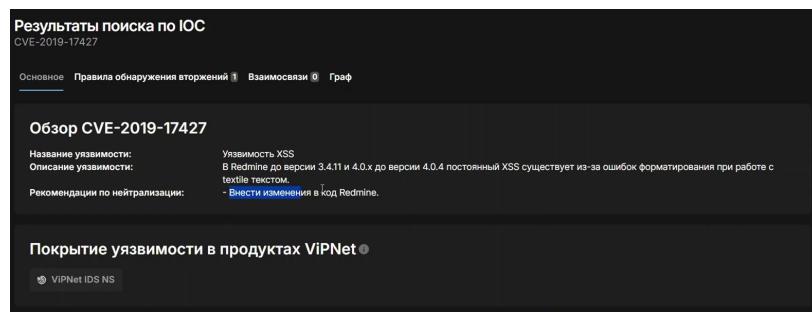


Рисунок 4.10: Общее описание уязвимости

4.1.4 Обнаружение последствия «Redmine User»

Уже в ходе работы на сервере Redmine (подключение через WEB) было замечено следующее:

При переходе в проект DEV1 во вкладке wiki включен REST API (рис. 4.11), а также в коде упоминается пользователь с именем hacker (рис. 4.12), этого же странного пользователя мы обнаружим и в списках пользователей, кроме того, у него есть права администратора (рис. 4.13). Этот же код мы видели в пакете события.

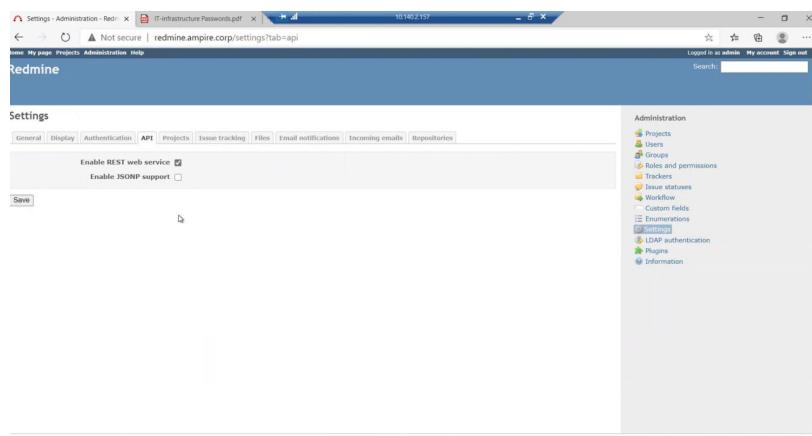


Рисунок 4.11: Обнаружение REST API

Рисунок 4.12: Упоминание пользователя в коде

Login	First name	Last name	Email	Administrator	Created	Last connection
admin	Redmine	Admin	admin@example.net	<input checked="" type="checkbox"/>	02/13/2020 02:10 PM	09/22/2025 06:49 PM
DEV1	John	Doe	dev1@ampire.corp	<input checked="" type="checkbox"/>	02/17/2020 08:18 AM	09/22/2025 07:24 PM
DEV2	Jane	Dow	jane@flow@ampire.corp	<input checked="" type="checkbox"/>	02/19/2020 12:31 PM	09/22/2025 07:24 PM
hacker	hacker	hacker	hacker@hacker.ru	<input checked="" type="checkbox"/>	09/22/2025 08:25 AM	

Рисунок 4.13: Подозрительный пользователь с правами администратора

Заполняем карточку инцидента, в рекомендациях удаление нового пользователя (рис. 4.14).

Рисунок 4.14: Карточка инцидента

4.1.5 Обнаружение уязвимости «Blind SQL (CVE-2019-18890)»

5. Внутренний нарушитель ожидает, когда администратор просмотрит страницу с внедренным вредоносным кодом.
 6. Внутренний нарушитель проводит Blind SQL-инъекцию, получает доступ к данным конфиденциального проекта.

Мы видим в событиях большое количество SQL запросов типа `SELECT SLEEP` и `SELECT FROM` от сервера 10.10.4.11 (нарушитель) на сервер Redmine 10.10.2.15, количество и постоянство которых настораживает (рис. 4.15).

События							
Сборка за последние 24 часа				Фильтр			
У	Дата и время	Кол-во	Название правила	Класс	IP-адрес источника	IP-адрес получателя	Направл...
•	08:47:33.142 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT SLEEP' query	client-side-exploit	10.10.2.254	10.10.2.15	→ ↵ ↵
•	08:47:33.142 22.09.2025	1	ET WEB_SERVER_SQL Injection Select Sleep Time Delay	web-application	10.10.2.254	10.10.2.15	↑ ↵ ↵
•	08:47:33.142 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT FROM' query	client-side-exploit	10.10.4.11	10.10.2.15	↑ ↵ ↵
•	08:47:33.142 22.09.2025	1	ET WEB_SERVER_SQL Injection Select Sleep Time Delay	web-application	10.10.4.11	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT SLEEP' query	web-application	10.10.2.254	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT FROM' query	client-side-exploit	10.10.2.254	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	ET WEB_SERVER_SQL Injection Select Sleep Time Delay	web-application	10.10.2.254	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT SLEEP' query	web-application	10.10.4.11	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	ET WEB_SERVER_SQL Injection Select Sleep Time Delay	client-side-exploit	10.10.4.11	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	AM SQL Generic SQL в HTTP URI: 'SELECT FROM' query	web-application	10.10.4.11	10.10.2.15	↑ ↵ ↵
•	08:47:33.009 22.09.2025	1	ET WEB_SERVER_SQL Injection Select Sleep Time Delay	web-application	10.10.4.11	10.10.2.15	↑ ↵ ↵

Рисунок 4.15: SQL-запросы

В пакетах к запросам увидим:

```
GET /issues.xml?project_id=1&subproject_id=2,3-IF%20((select%20desc  
Host: redmine.ampire.corp  
User-Agent: python-requests/2.31.0
```

Этот пакет указывает на попытку BLIND-SQL-инъекции через API Redmine, с целью извлечения данных из таблицы issues (например, описаний задач с заголовками, начинающимися на SECRET). Использование SLEEP(5) позволяет злоумышленнику косвенно определять, выполняются ли условия, на основе времени ответа сервера.

При заполнении карточки уязвимости (рис. 4.16) ссылаемся на гитхаб [2] (рис. 4.17).

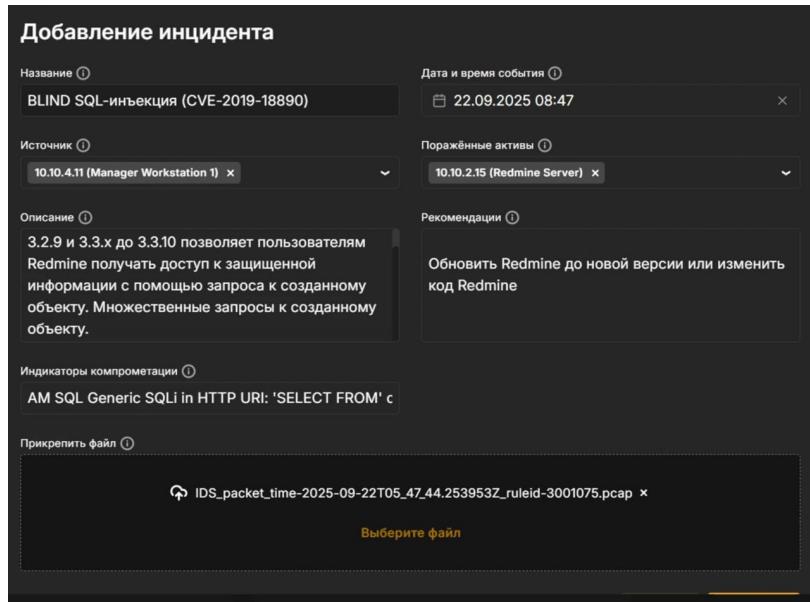


Рисунок 4.16: Карточка инцидента



Рисунок 4.17: Источник информации об уязвимости

4.2 Устранение уязвимостей и их последствий

4.2.1 Уязвимость Слабый пароль пользователя DEV-1

Для закрытия уязвимости меняем пароль на более сложный, не содержащийся в словаре. Для изменения пароля на сервере MS Active Directory подключаемся к нему через удаленный рабочий стол. Открываем на нем «Active Directory Users and Computers» через комбинацию Win+R и ввод в поиск dsa.msc, переходим в users, находим dev1, меняем пароль для пользователя dev1 (рис. 4.18), (рис. 4.19).

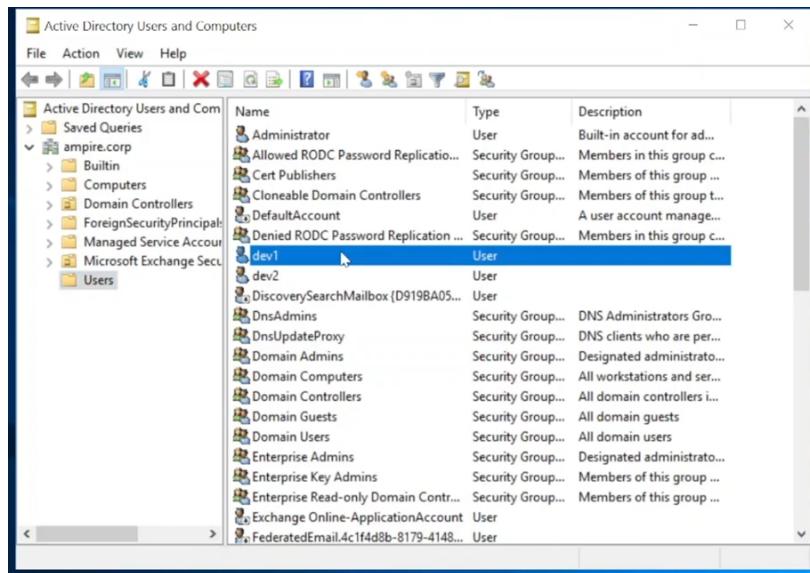


Рисунок 4.18: Пользователь dev1

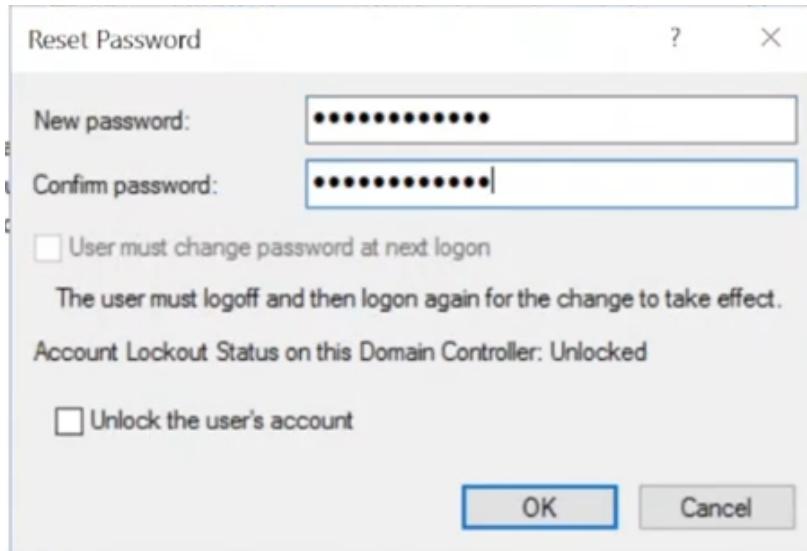


Рисунок 4.19: Смена пароля

4.2.2 Последствие «Developer backdoor»

Через удаленный рабочий стол переходим на сервер Developer (10.10.4.13), на сервере через Win+R и ввод в поиск taskschd.msc сможем открыть планировщик задач (рис. 4.20).

Нарушитель может создать сервис, автоматически запускающий исполняемый файл, который устанавливает Reverse Shell подключение. Новая задача, записанная нарушителем, находится на узле Developer 1 в планировщике задач (рис. 4.21).

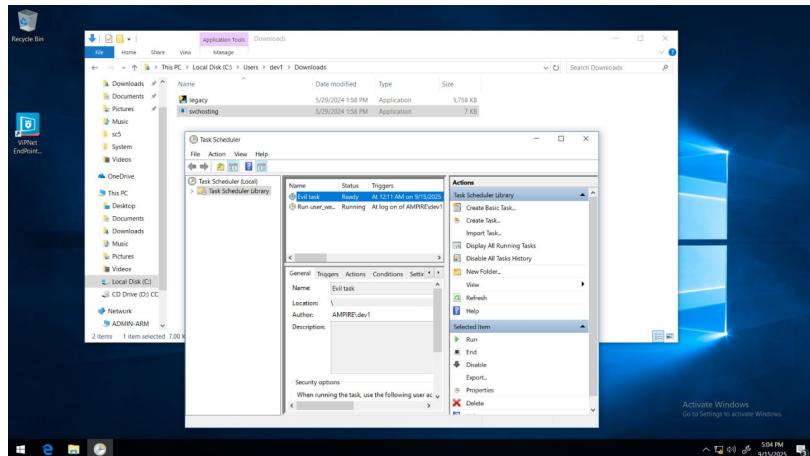


Рисунок 4.20: Запуск исполняемого файла в планировщике

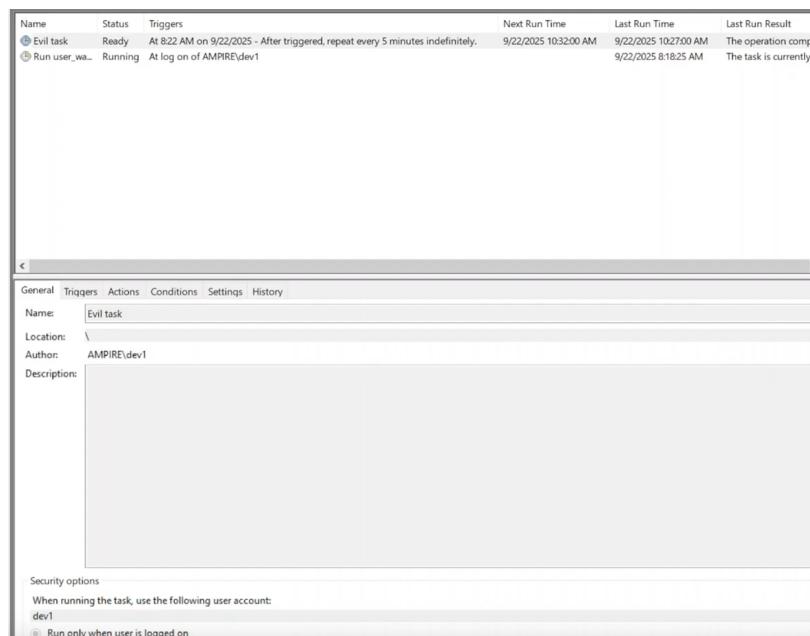


Рисунок 4.21: Запуск исполняемого файла в планировщике

Удаляем задачу в планировщике, и теперь надо удалить файл. Во вкладке action мы обнаружили путь к файлу C:\Users\dev1\Downloads\svchosting.exe, благодаря этому нашли его и удалили.

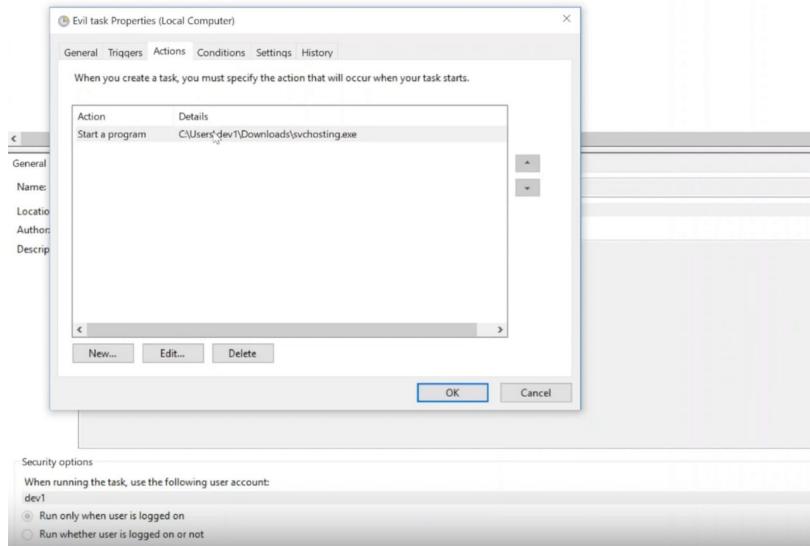


Рисунок 4.22: Путь к исполняемому файлу

4.2.3 Уязвимость атака XSS

Для устранения этой уязвимости нам нужно будет зайти на Redmine и через веб-браузер, и через консоль по ssh подключению.

В Redmine до версии 4.0.4 постоянный XSS существует из-за ошибок форматирования при работе с textile текстом. В данном сценарии используется для включения REST API для эксплуатации следующей уязвимости (рис. 4.23).

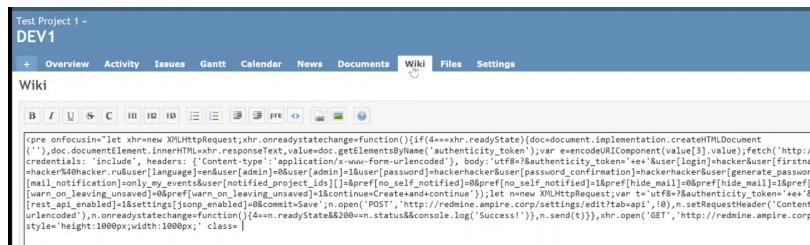
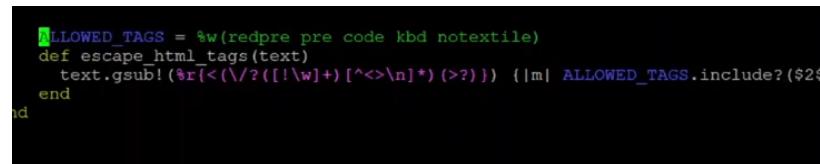


Рисунок 4.23: Пример добавления кода, выводящего на экран надпись XSS в wiki-страницу

Для устранения этой уязвимости необходимо внести изменения в код Redmine. Необходимо найти обработку текста wiki-страницы при наличии в тексте

html-тегов. Из описания уязвимости понятно, что необходимо найти библиотеку для преобразования textile разметки в html. В Redmine за данное преобразование отвечает файл redcloth3.rb. Для устранения изменим в нем (все изменения через nano \var\www\redmine-3.3.4\lib\redcloth3.rb) следующие строки (рис. 4.24). Удаляем тег pre из разрешенных тегов, версию правильного кода можем подсмотреть на github: <https://github.com/redmine/redmine/blob/master/lib/redcloth3.rb> [3].



```
ALLOWED_TAGS = %w(redpre pre code kbd notextile)
def escape_html_tags(text)
  text.gsub!(%r{<(/?(!\w)+[^>]\n*)(>?)}) do |m|
    if ALLOWED_TAGS.include?($2) && $3.present?
      "<#$1#$3"
    else
      "<#$1#{'>' unless $3.blank?}"
    end
  end
end
```

Рисунок 4.24: Содержимое файла redcloth3.rb

Код, исправляющий уязвимость CVE-2019-17427:

```
ALLOWED_TAGS = %w(redpre pre code kbd notextile)
def escape_html_tags(text)
  text.gsub!(%r{<(/?(!\w)+[^>]\n*)(>?)}) do |m|
    if ALLOWED_TAGS.include?($2) && $3.present?
      "<#$1#$3"
    else
      "<#$1#{'>' unless $3.blank?}"
    end
  end
end
```

```

ALLOWED_TAGS = %w(redpre pre code kbd notextile)
def escape_html_tags(text)
  text.gsub!(%r{<(\/?((!\w)+[^<>\n]*)(>?)}) do |m|
    if ALLOWED_TAGS.include?($2) && $3.present?
      "<#${$1}#${$3}""
    else
      "&lt;#${$1}#{'>' unless $3.blank?}"
    end
  end
end
end

```

Рисунок 4.25: Исправления в файле redcloth3.rb

После внесения изменений необходимо перезапустить службу веб-сервера sudo systemctl restart nginx.service. Видим, что уязвимость успешно устранена, так как изменилось отображение страницы на Redmine Wiki (рис. 4.26).



Рисунок 4.26: Успешное устранение уязвимости

4.2.4 Последствие «Redmine User»

Нарушитель создал пользователя на портале Redmine. Пользователь, обладающий правами администратора, имеет неограниченный доступ к пользовательской базе. Для обнаружения добавления нового привилегированного пользователя заходим в консоль администратора Redmine, переходим в раздел «Administration» – «Users» и смотрим список существующих пользователей (рис. 4.27).

Login	First name	Last name	Email	Administrator	Created	Last connection
admin	Redmine	Admin	admin@example.net	✓	02/13/2020 01:10 PM	09/22/2025 08:22 PM
DEV1	John	Doe	dev1@amprre.corp		02/17/2020 08:18 AM	09/22/2025 08:32 PM
DEV2	Jane	Dow	janedow@amprre.corp	✓	02/19/2020 12:31 PM	09/22/2025 08:32 PM
hacker	hacker	hacker	hacker@hacker.ru	✓	09/22/2025 08:25 AM	09/22/2025 08:25 AM

Рисунок 4.27: Список пользователей Redmine

Удаляем пользователя hacker через веб-интерфейс, успешно нейтрализуя

последствие Redmine User.

4.2.5 Уязвимость Blind SQL-инъекция

Эксплуатируемая уязвимость – CVE-2019-18890.

В Redmine до версии 3.2.9 и 3.3.x до версии 3.3.10 уязвимость позволяет пользователям Redmine получать доступ к защищенной информации с помощью сгенерированного объектного запроса. Уязвимость реализуется посимвольным перебором с замером времени ответа. В данном сценарии данная уязвимость используется для получения конфиденциальной информации из БД, минуя разграничение доступа Redmine. Время прихода пакета является индикатором: при запоздании пакета – символ подбран верно, иначе – перебор продолжается (рис. 4.28). Попробуем сделать SQL запрос на WEB сервер Redmine SLEEP(2) (с помощью ссылки `http://redmine.ampire.corp/issues.xml?project_id=1&subprojec SLEEP(2)`), увидим, что время ожидания ответа возрастает почти до 9 секунд, что свидетельствует об инъекции.

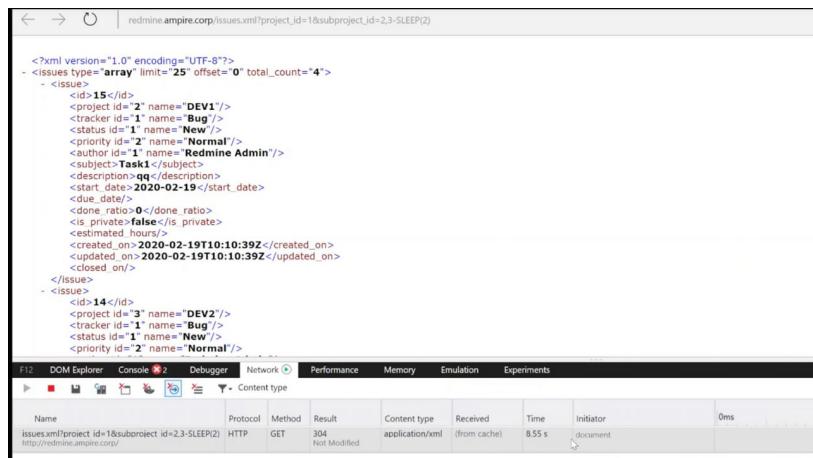


Рисунок 4.28: SQL-запрос

Для устранения этой уязвимости необходимо внести изменения в код Redmine. Снова через SSH заходим на Redmine с помощью консоли. Исходя из адреса и запроса следует найти местоположение скрипта, где происходит обработка

параметра `subproject_id`. В данном случае рассмотрен файл `\var\www\redmine-3.3.4\app\models\query.rb` [3].

Вносим изменения в код, добавляя фильтрацию значений (рис. 4.29), (рис. 4.30), и после перезапуска веб-сервера через команду `sudo systemctl restart nginx.service` уязвимость устраняется. (Зачеркнут продублированный инцидент)

```
# include the selected subprojects
ids = [project.id] + values_for("subproject_id").each(&:to_i)
project_clauses << "#{Project.table_name}.id IN (%s)" % ids.join(',')
when '***'
```

Рисунок 4.29: Содержимое файла query.rb до исправления уязвимости

```
# include the selected subprojects
ids = [project.id] + values_for("subproject_id").map(&:to_i)
project_clauses << "#{Project.table_name}.id IN (%s)" % ids.join(',')
when '***'
```

Рисунок 4.30: Содержимое файла query.rb с исправлением уязвимости

4.3 Результат

После выполнения всех приведенных выше действий получим, что мы закрыли все обнаруженные инциденты (рис. 4.31), и, соответственно, устранили уязвимости и последствия, предполагаемые этой лабораторной работой (рис. 4.32).

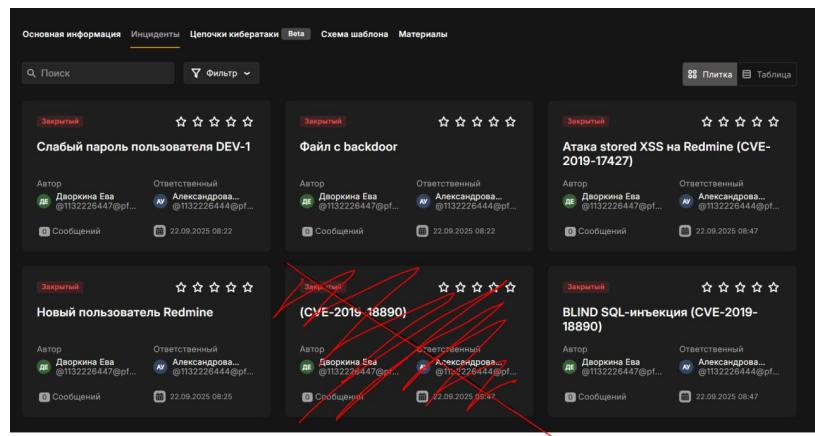


Рисунок 4.31: Закрытые инциденты

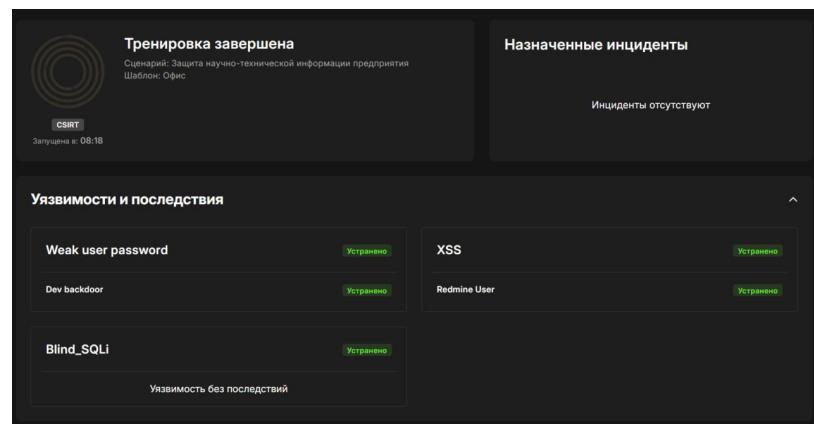


Рисунок 4.32: Устраниенные уязвимости и последствия

5 Выводы

В ходе выполнения лабораторной работы были успешно достигнуты поставленные цели: освоены практические навыки выявления, анализа и устранения типовых уязвимостей информационной системы. В рамках сценария «Защита научно-технической информации предприятия» были обнаружены и закрыты критические уязвимости и их последствия эксплуатации.

Список литературы

- [1] *AMTIP.*
- [2] *CVE-2019-18890 POC (Proof of Concept).*
- [3] *Redmine.*
- [4] *The LaZagne Project.* 2015.
- [5] *Программный комплекс обучения методам обнаружения, анализа и устранения последствий компьютерных атак "Ampire". Сценарий №5 «ЗАЩИТА НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ ПРЕДПРИЯТИЯ».*
- [6] *Сетевой сенсор системы обнаружения атак программно-аппаратный комплекс ViPNet IDS NS 3.* Infotechs. infotechs. 321 с.