

Отчет по лабораторной работе №2

Дисциплина: Кибербезопасность предприятия

Боровиков Даниил	Хрусталев Влад
Гисматуллин Артём	Чесноков Артёмий
Коннова Татьяна	Нефедова Наталья
Уткина Алина	Бансимба Клодели

Содержание

1	Задание. Сценарий №5.	5
2	Последовательность действий нарушителя	6
3	Выполнение лабораторной работы	7
3.1	Перечень уязвимостей и последствий	7
3.1.1	Слабый пароль пользователя	7
3.1.2	Последствие Dev backdoor	7
3.1.3	Уязвимость 2: XSS (CVE-2019-17427)	10
3.1.4	Последствие: Redmine User	12
3.1.5	Уязвимость 3: Blind SQL-инъекция (CVE-2019018890)	13
4	Общие выводы	16

Список иллюстраций

3.1	Файл svchosting.exe в папке Downloads	8
3.2	Задание “Evil task” в планировщике задач	9
3.3	Настройки задания с путем к вредоносному файлу	10
3.4	Вредоносный XSS-код на Wiki-странице	11
3.5	Исходный код файла redcloth3.rb	11
3.6	Исправленная версия файла redcloth3.rb	12
3.7	Перезапуск службы nginx	12
3.8	Список пользователей Redmine с пользователем “hacker”	13
3.9	Подтверждение удаления пользователя	13
3.10	Логи Redmine с SQL-инъекцией	14
3.11	Исправление в файле query.rb	14
3.12	Главная страница	15
3.13	Карточки инцидентов и последствий	15

Список таблиц

1 Задание. Сценарий №5.

ЗАЩИТА НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ ПРЕДПРИЯТИЯ

Внешний нарушитель умеет использовать инструментарий для проведения компьютерных атак, знает техники постэксплуатации.

Средство обнаружения вторжений – программно-аппаратный комплекс для обнаружения вторжений в информационные системы ViPNet IDS NS.

Автоматическое выявление инцидентов на основе интеллектуального анализа событий информационной безопасности – программно-аппаратный комплекс ViPNet TIAS.

ViPNet EPP применяется для защиты отдельных компонентов информационной инфраструктуры организаций – персональных компьютеров пользователей и корпоративных серверов.

Специализированный контроль сетевой безопасности и предотвращение проникновения, упрощающие централизованное управление сетью – Security Onion. Security Onion связывает воедино три основные функции: - полный захват пакетов; - обнаружение сетей и конечных точек; - мощные инструменты анализа.

2 Последовательность действий нарушителя

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

3 Выполнение лабораторной работы

3.1 Перечень уязвимостей и последствий

3.1.1 Слабый пароль пользователя

Обнаружение: На файловом сервере использовался слабый пароль учетной записи dev1.

Устранение: Пароль изменен на сложный через Active Directory Users and Computers.

3.1.2 Последствие Dev backdoor

Обнаружение:

- В папке Downloads пользователя dev1 обнаружен файл svchosting.exe
- В планировщике задач создано задание Evil task с автозапуском при входе пользователя
- Задание настроено на выполнение каждые 5 минут

Устранение:

- Удалено задание Evil task из планировщика задач
- Удален файл svchosting.exe из папки Downloads (рис. 3.1 - рис. 3.3)

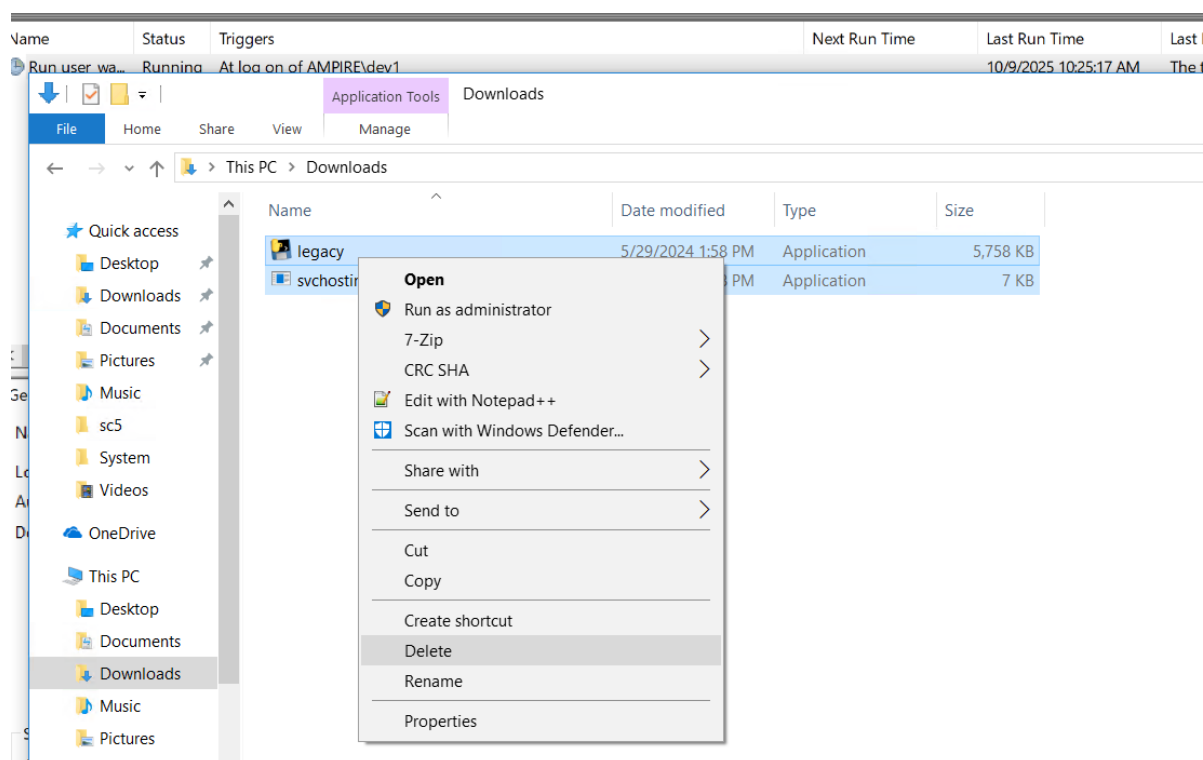


Рис. 3.1: Файл svchosting.exe в папке Downloads

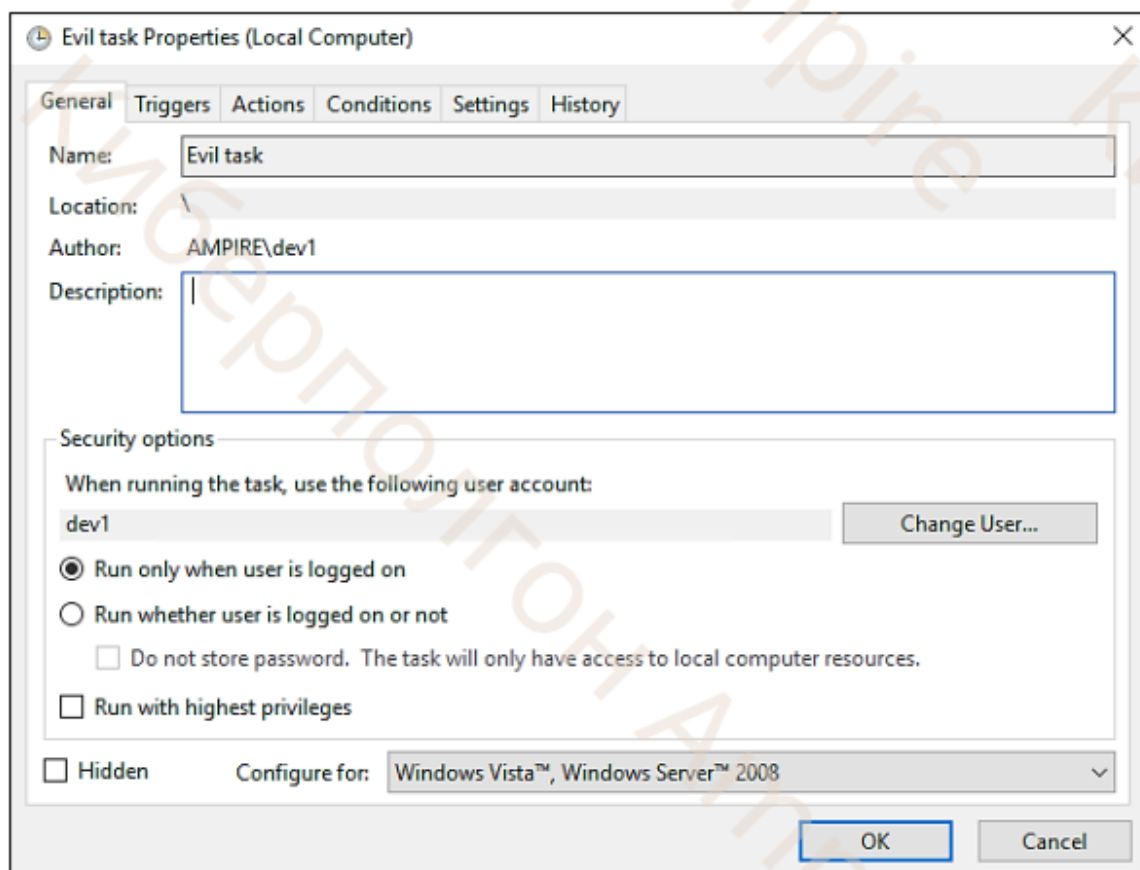


Рис. 3.2: Задание “Evil task” в планировщике задач

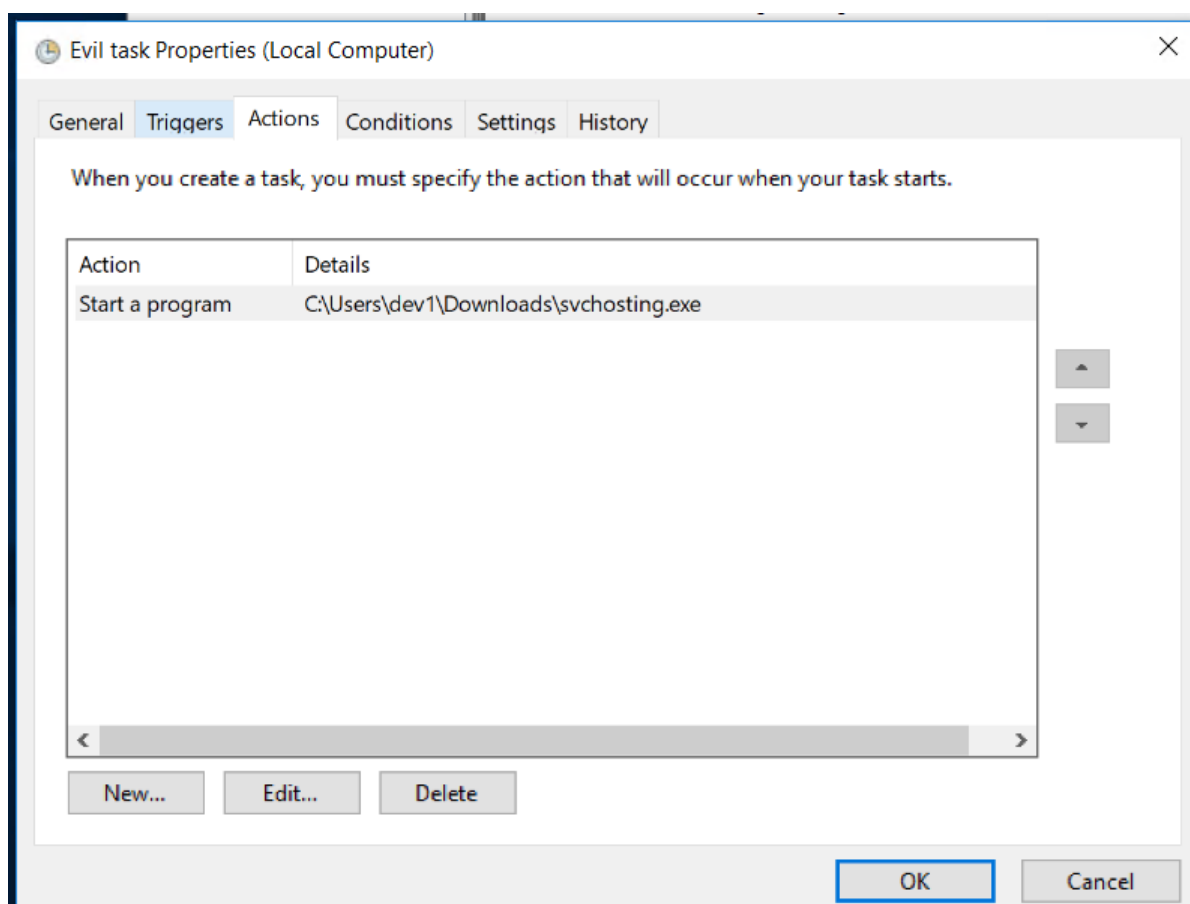


Рис. 3.3: Настройки задания с путем к вредоносному файлу

3.1.3 Уязвимость 2: XSS (CVE-2019-17427)

Обнаружение: На Wiki-странице проекта DEV1 обнаружен сложный XSS-код, который:

- Создает пользователя “hacker” с правами администратора
- Включает REST API в настройках Redmine
- Использует событие onfocusin для автоматического выполнения

Устранение:

- В файле redcloth3.rb исправлена обработка HTML-тегов
- Удален тег pre из списка разрешенных тегов (ALLOWED_TAGS)

- Перезапущена служба nginx: `sudo systemctl restart nginx.service` (рис. 3.4 - рис. 3.7)

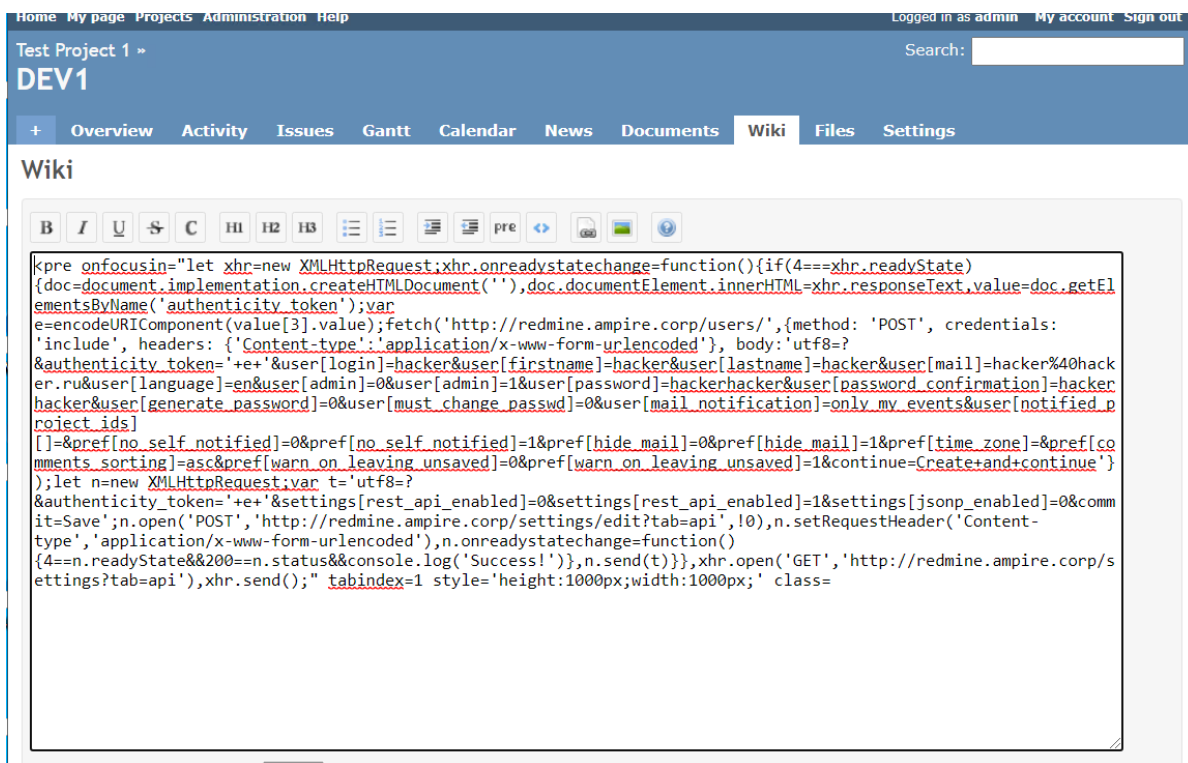


Рис. 3.4: Вредоносный XSS-код на Wiki-странице

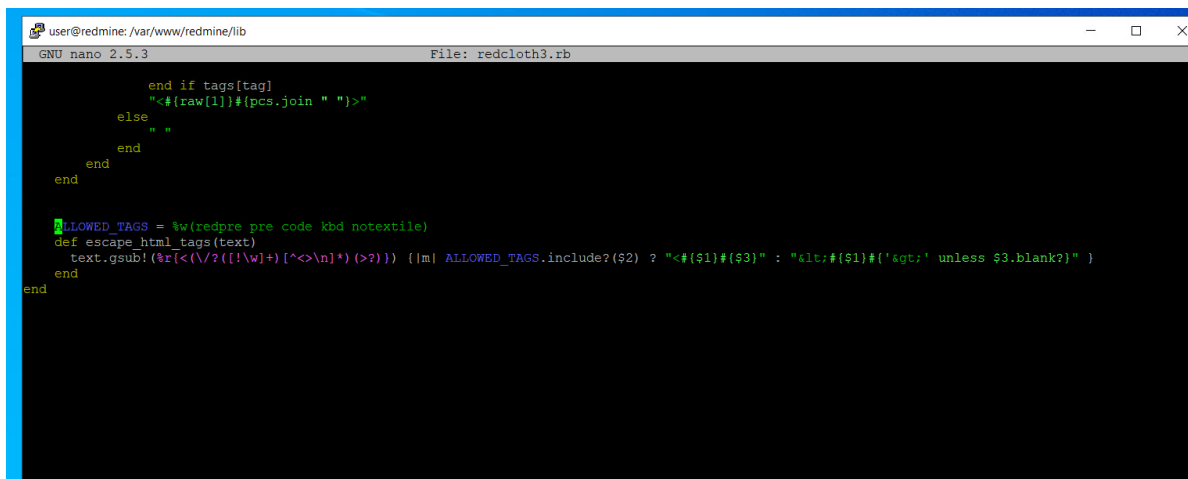
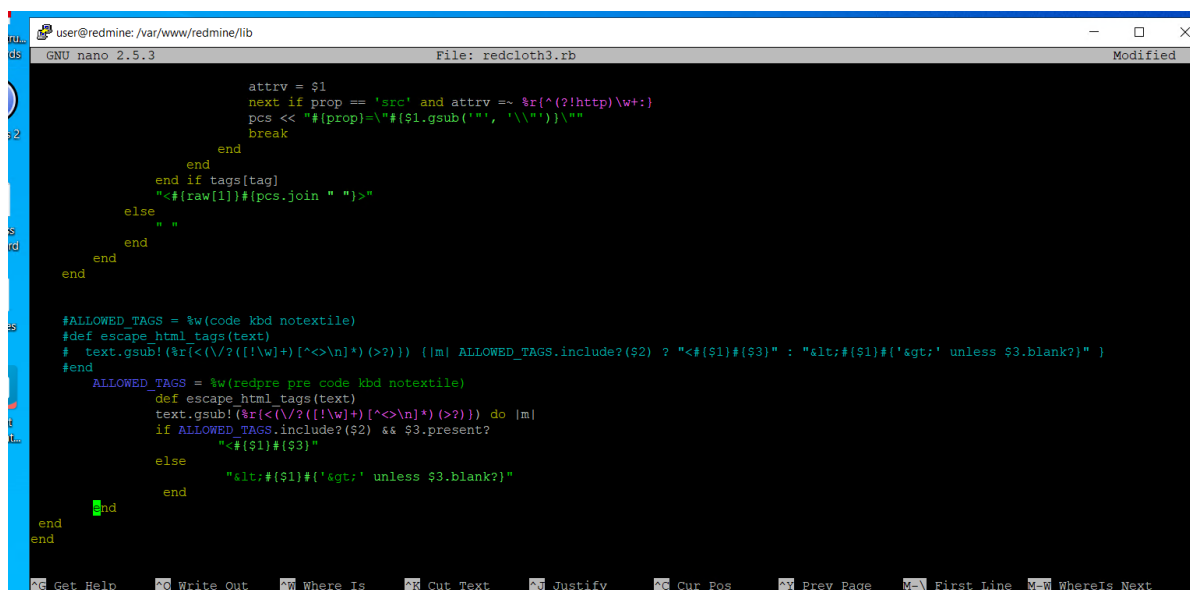


Рис. 3.5: Исходный код файла redcloth3.rb

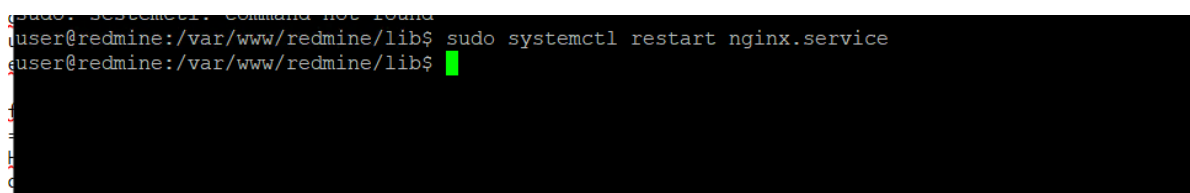


```
user@redmine: /var/www/redmine/lib
GNU nano 2.5.3 File: redcloth3.rb Modified

    attrv = $1
    next if prop == 'src' and attrv =~ %r{^(?!http)\w+:}
    pcs << "#{prop}=\"#{ $1.gsub('"', '\\\"') }\""
    break
  end
end
end if tags[tag]
  "<#{raw[1]}#{pcs.join " "}">"
else
  " "
end
end
end
end

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

Рис. 3.6: Исправленная версия файла redcloth3.rb



```
user@redmine:/var/www/redmine/lib$ sudo systemctl restart nginx.service
user@redmine:/var/www/redmine/lib$
```

Рис. 3.7: Перезапуск службы nginx

3.1.4 Последствие: Redmine User

Обнаружение: В Redmine создан пользователь “hacker” с email hacker@hacker.ru (рис. 3.8)

Устранение: Пользователь “hacker” удален через веб-интерфейс Redmine (рис. 3.9)

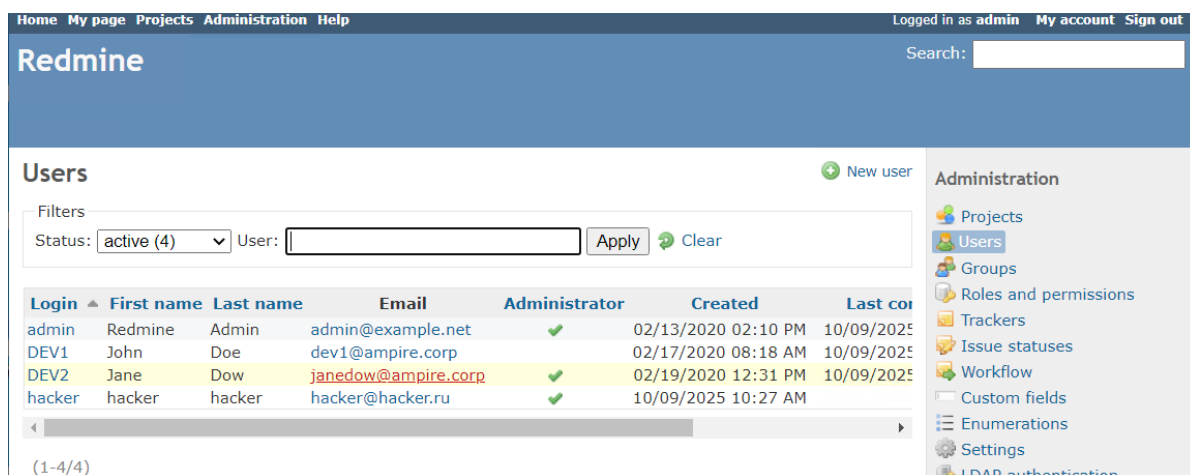


Рис. 3.8: Список пользователей Redmine с пользователем “hacker”

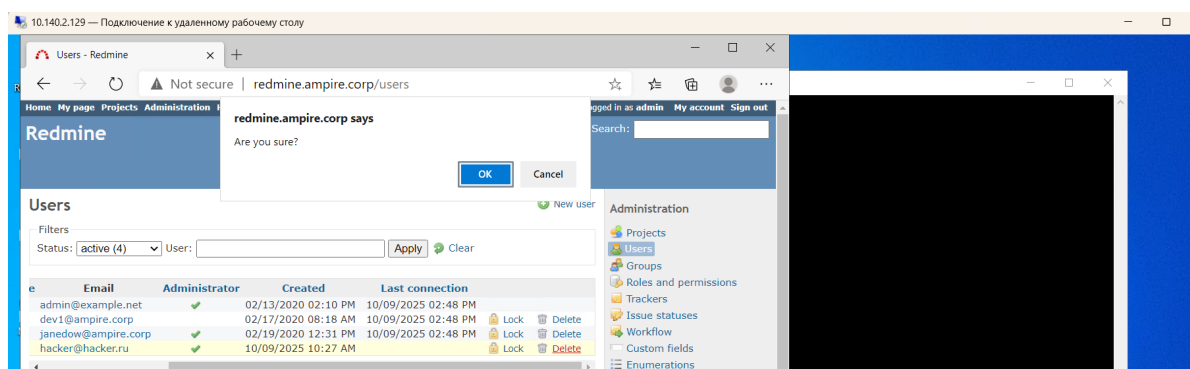


Рис. 3.9: Подтверждение удаления пользователя

3.1.5 Уязвимость 3: Blind SQL-инъекция (CVE-2019018890)

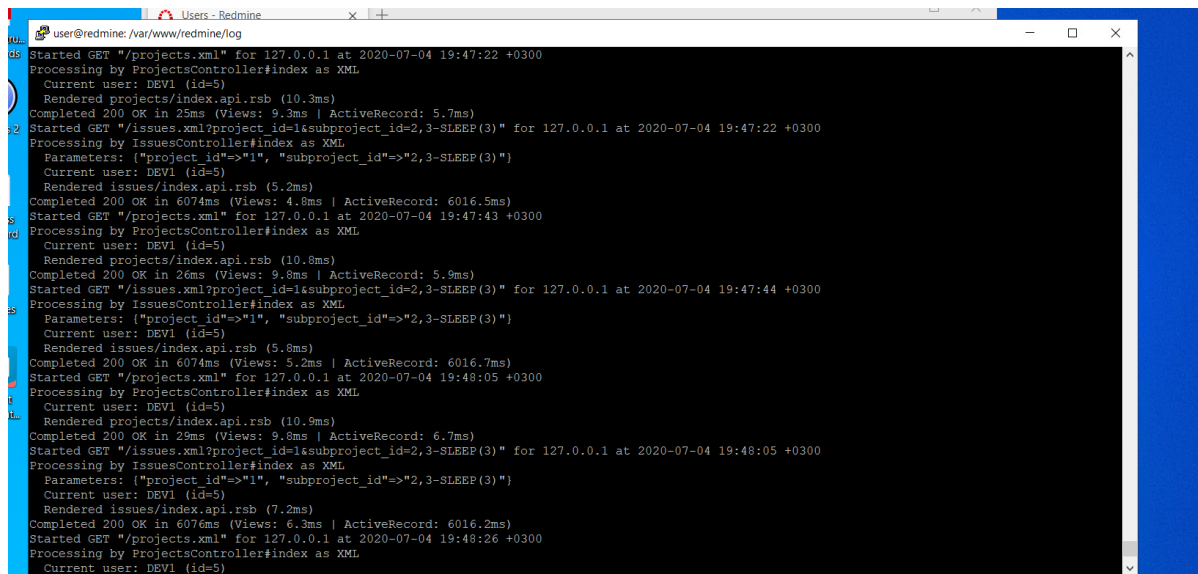
Обнаружение:

- В логах Redmine зафиксированы запросы с SLEEP(3) в параметре subproject_id
- Время выполнения запросов увеличилось до 6074 мс (вместо обычных ~30 мс) (рис. 3.10)

Устранение:

- В файле query.rb закомментирован уязвимый код обработки subproject_id

- Добавлена фильтрация входных параметров (рис. 3.11)

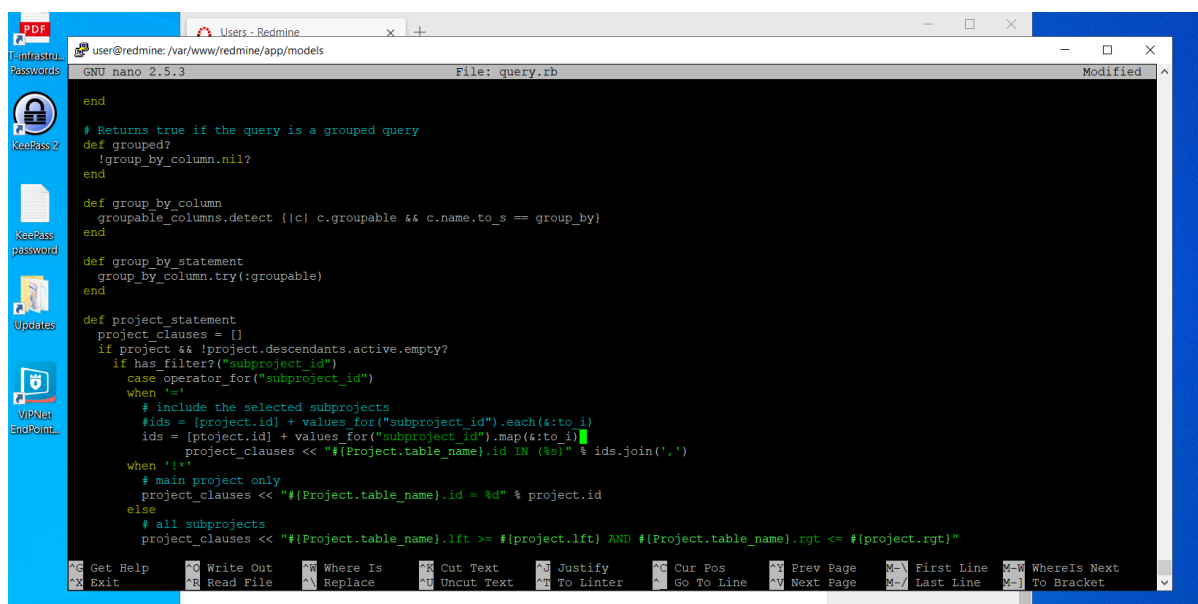


```

user@redmine: /var/www/redmine/log
Started GET "/projects.xml" for 127.0.0.1 at 2020-07-04 19:47:22 +0300
Processing by ProjectsController#index as XML
Current user: DEV1 (id=5)
Rendered projects/index.api.rsb (10.3ms)
Completed 200 OK in 25ms (Views: 9.3ms | ActiveRecord: 5.7ms)
Started GET "/issues.xml?project_id=1&subproject_id=2,3-SLEEP(3)" for 127.0.0.1 at 2020-07-04 19:47:22 +0300
Processing by IssuesController#index as XML
Parameters: {"project_id"=>"1", "subproject_id"=>"2,3-SLEEP(3)"}
Current user: DEV1 (id=5)
Rendered issues/index.api.rsb (5.2ms)
Completed 200 OK in 6074ms (Views: 4.8ms | ActiveRecord: 6016.5ms)
Started GET "/projects.xml" for 127.0.0.1 at 2020-07-04 19:47:43 +0300
Processing by ProjectsController#index as XML
Current user: DEV1 (id=5)
Rendered projects/index.api.rsb (10.8ms)
Completed 200 OK in 26ms (Views: 9.8ms | ActiveRecord: 5.9ms)
Started GET "/issues.xml?project_id=1&subproject_id=2,3-SLEEP(3)" for 127.0.0.1 at 2020-07-04 19:47:44 +0300
Processing by IssuesController#index as XML
Parameters: {"project_id"=>"1", "subproject_id"=>"2,3-SLEEP(3)"}
Current user: DEV1 (id=5)
Rendered issues/index.api.rsb (5.8ms)
Completed 200 OK in 6074ms (Views: 5.2ms | ActiveRecord: 6016.7ms)
Started GET "/projects.xml" for 127.0.0.1 at 2020-07-04 19:48:05 +0300
Processing by ProjectsController#index as XML
Current user: DEV1 (id=5)
Rendered projects/index.api.rsb (10.9ms)
Completed 200 OK in 29ms (Views: 9.8ms | ActiveRecord: 6.7ms)
Started GET "/issues.xml?project_id=1&subproject_id=2,3-SLEEP(3)" for 127.0.0.1 at 2020-07-04 19:48:05 +0300
Processing by IssuesController#index as XML
Parameters: {"project_id"=>"1", "subproject_id"=>"2,3-SLEEP(3)"}
Current user: DEV1 (id=5)
Rendered issues/index.api.rsb (7.2ms)
Completed 200 OK in 6076ms (Views: 6.3ms | ActiveRecord: 6016.2ms)
Started GET "/projects.xml" for 127.0.0.1 at 2020-07-04 19:48:26 +0300
Processing by ProjectsController#index as XML
Current user: DEV1 (id=5)

```

Рис. 3.10: Логи Redmine с SQL-инъекцией



```

GNU nano 2.5.3 File: query.rb Modified
end

# Returns true if the query is a grouped query
def grouped?
  !group_by_column.nil?
end

def group_by_column
  groupable_columns.detect {|c| c.groupable && c.name.to_s == group_by}
end

def group_by_statement
  group_by_column.try(:groupable)
end

def project_statement
  project_clauses = []
  if project && !project.descendants.active.empty?
    if has_filter?("subproject_id")
      case operator_for("subproject_id")
      when '='
        # include the selected subprojects
        ids = [project.id] + values_for("subproject_id").each(&:to_i)
        ids = [project.id] + values_for("subproject_id").map(&:to_i)
        project_clauses << "#{Project.table_name}.id IN (#{ids.join(',')})"
      when '!=', '>', '<'
        # main project only
        project_clauses << "#{Project.table_name}.id = #{project.id}"
      else
        # all subprojects
        project_clauses << "#{Project.table_name}.lft >= #{(project.lft)} AND #{Project.table_name}.rgt <= #{(project.rgt)}"
      end
    end
  end
end

```

Рис. 3.11: Исправление в файле query.rb

Общий результат выполненной работы: (рис. 3.12 - рис. 3.13)

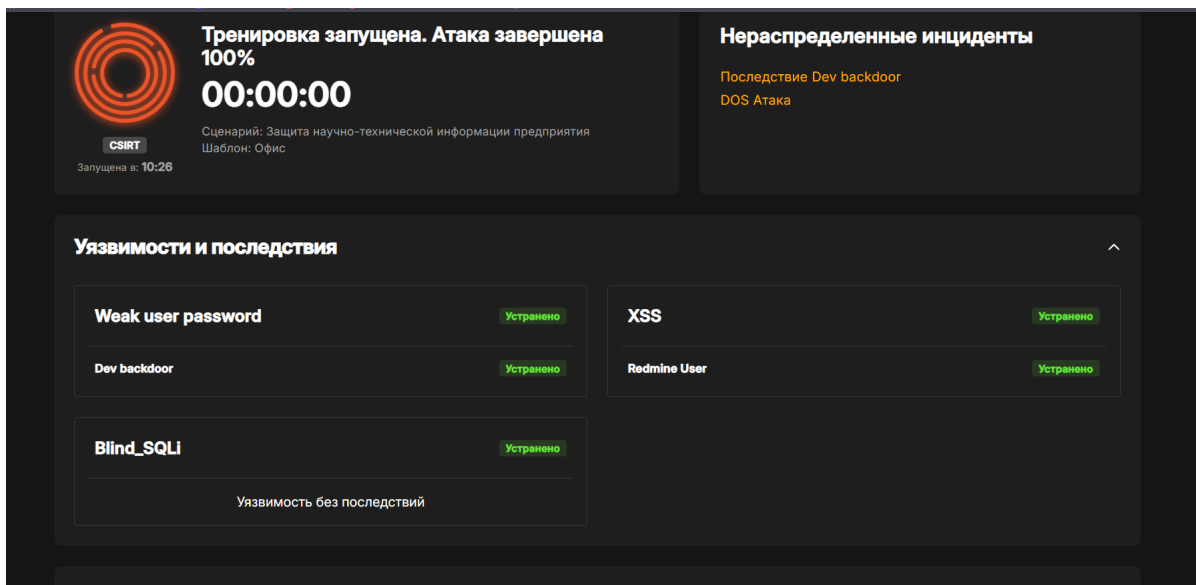


Рис. 3.12: Главная страница

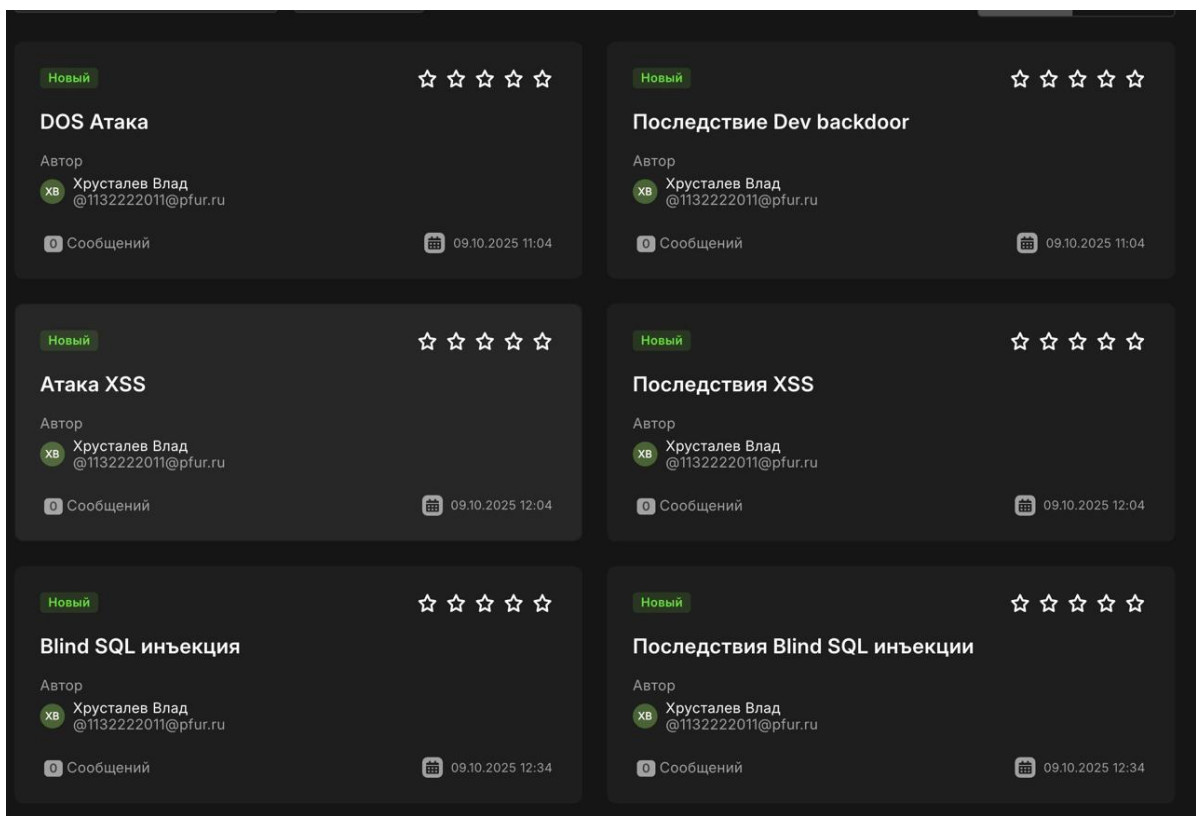


Рис. 3.13: Карточки инцидентов и последствий

4 Общие выводы

В рамках учебно-практического занятия на базе программного комплекса обучения методам обнаружения, анализа и устранения последствий компьютерных атак «Amprige» мы выполнили сценарий №5 «Защита научно-технической информации предприятия». Внутренний нарушитель, используя слабые пароли и уязвимости в веб-приложении Redmine, осуществил комплексную атаку с целью получения доступа к конфиденциальной информации. Нарушитель применил техники внедрения backdoor, эксплуатации XSS-уязвимости и слепой SQL-инъекции для создания привилегированного пользователя и несанкционированного доступа к данным. Уровень сложности сценария — 8 (из 10). Мы успешно выявили уязвимости, проанализировали последствия атаки, устранили их и отработали методы детектирования с использованием инструментов ViPNet IDS NS, ViPNet TIAS и Security Onion, а также освоили методики исправления исходного кода приложений для устранения уязвимостей.