Internet Security 2009W

Protokoll

Weak Security Platform

Manuel Mausz, Matr. Nr. 0728348
manuel-tu@mausz.at
Aldin Rizvanovic, Matr. Nr. 0756024
e0756024@student.tuwien.ac.at

Wien, am 12. Oktober 2009

Inhaltsverzeichnis

Cryptographic: Basic....3
Cryptographic: Intermediate....4
Browser: Cookie....5
Browser: User Agent....6
SSI: WordCount....7
PHP: Simple Gallery....8
PHP: Login....9
PHP: File Viewer....10
PHP: Unix Calendar....12
PHP: Simple Blog....13
PHP: Simple Secured Gallery....15
JavaSE: Login....17
JavaSE: Login Advanced....19
JavaEE5 (Seam): Webshop....21
JavaEE5 (Seam): Webshop Reloaded....23
Screenshot: Alle Aufgabe gelöst....25

Cryptographic: Basic

Bei der Eingabe der Zeichen „aaa“ lieferte uns der Algorithmus die Ausgabe „abc“. Bei der Eingabe der Zeichen „abc“ lieferte er „ace“. Daraus erkannten wir, dass der Algorithmus immer den ASCII-Wert des Buchstaben um seine Position (beginnend bei 0) erhöhte. Wir implementierten die umgekehrte Version dieses Algorithmus in einem kleinen php-Programm und führten es mit dem vorgegebenem Wert „fmqumft“ aus. Als Ausgabe bekamen wir das richtige Ergebnis „florian“.

Implementierung:

```php
<?php

$input = $argv[1];
$output = "";

for($i = 0; $i < strlen($input); $i++)
{
  $char = $input{$i};
  $output .= chr(ord($char) - $i);
}

print "input: ".$input."\n";
print "output: ".$output."\n";

?>
```

Ausführung:

```
manuel@www:~/uni/isec> php wsp_enc1.php fmqumft
input: fmqumft
output: florian
manuel@www:~/uni/isec>
```

Cryptographic: Intermediate

Wie beim ersten Beispiel versuchten wir durch eingeben von „abc“ und Ergebnis „eca“ den Algorithmus zu verstehen. Nach ein paar versuchen wurde schnell klar, der zweite Algorithmus liefert das selbe Ergebnis wie der erste, jedoch wird die Ausgabe zusätzlich noch invertierte. Wir änderten unser php-Programm dementsprechend ab und gelangten zum richtigen Ergebnis „michael“.

Implementierung:

```php
<?php

$input = strrev($argv[1]);
$output = "";

for($i = 0; $i < strlen($input); $i++)
{
  $char = $input{$i};
  $output .= chr(ord($char) - $i);
}
$output = $output;

print "input: ".$input."\n";
print "output: ".$output."\n";

?>
```

Ausführung:

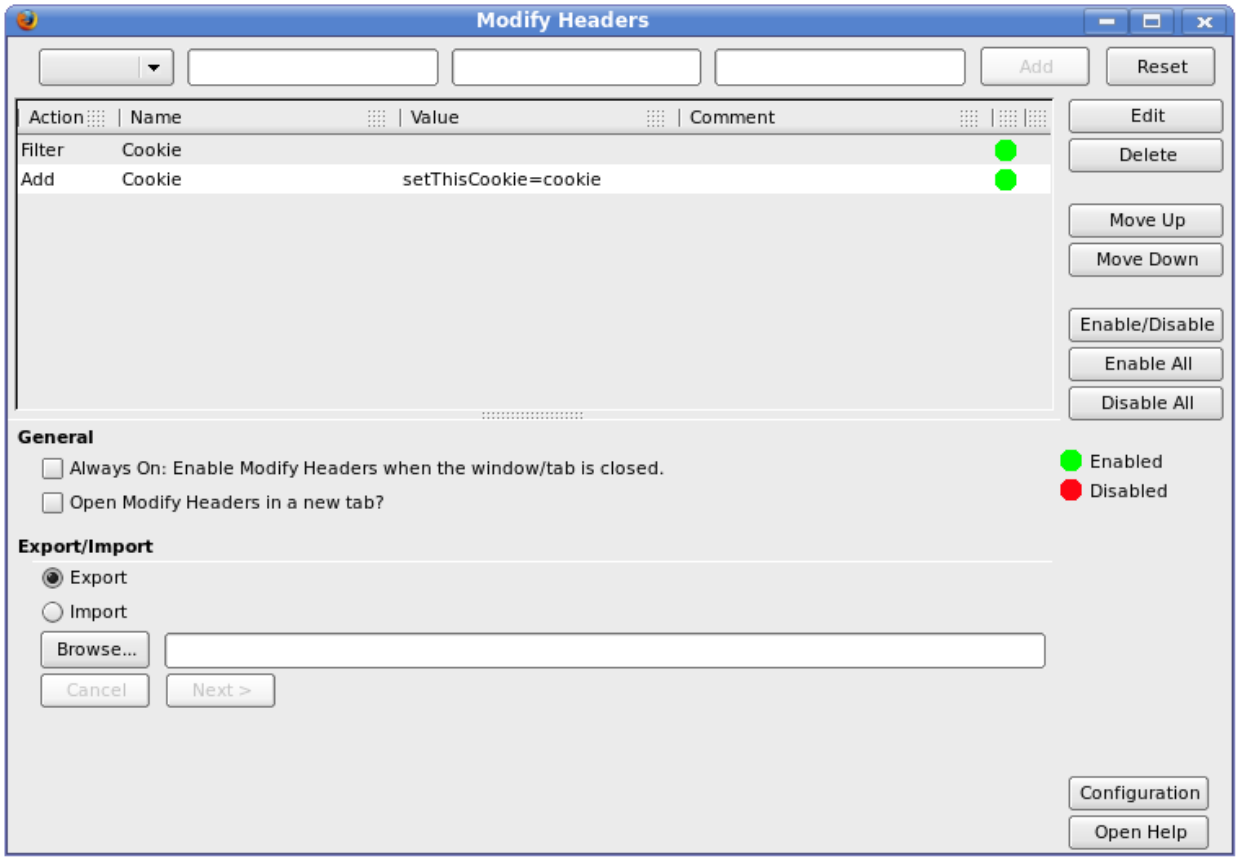
```
manuel@www:~/uni/isec> php wsp_enc2.php rjekejm
input: mjekejr
output: michael
manuel@www:~/uni/isec>
```

Browser: Cookie

Nachdem wir uns mit dem Prinzip von Cookies vertraut gemacht haben, entdeckten wir durch Zuhilfenahme eines Addons für Firefox das Cookie„setThisCookie“ mit dem Wert „not+initialized“. Den Wert des Cookies haben wir nach mehrmaligen Versuchen auf den richtigen Wert „cookie“ gesetzt und konnten somit das Beispiel lösen.

Cookie geändert (via Modifikation der HTTP-Header):

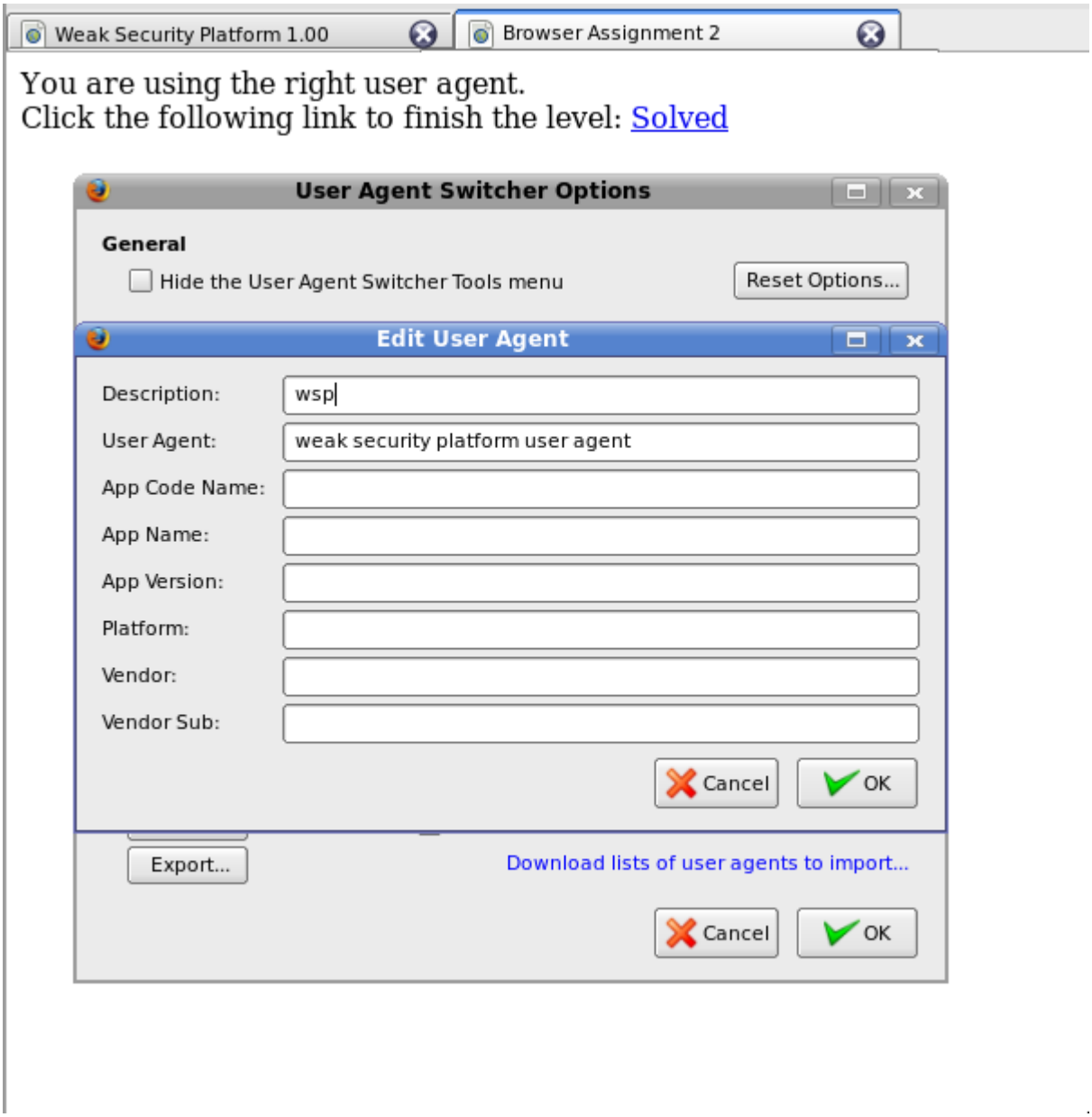
Cookie setThisCookie set correctly.
Click the following link to finish the level: Solved



Browser: User Agent

Durch die Fehlermeldung merkten wir, dass der User Agent des Browsers nicht berechtigt ist, die Seite zu betrachten. Daher haben wir mit einem Addon für Firefox den User Agent wie gefordert auf „weak security platform user agent“ gestellt und das Beispiel damit gelöst.

Änderung des User Agents via Firefox-Addon:



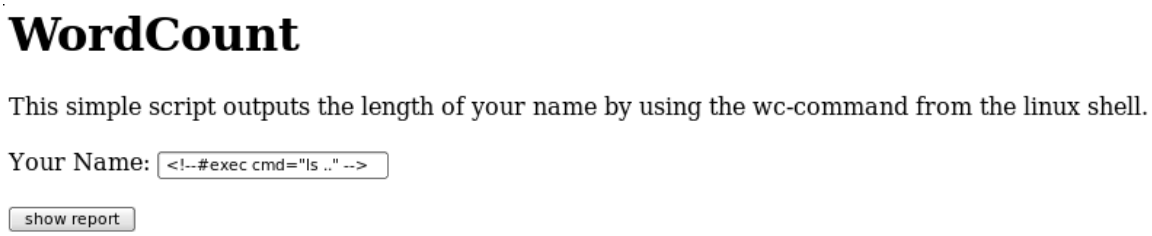
SSI: WordCount

Zunächst haben wir uns im Internet über SSI erkundigt um die verfügbaren Befehle und die richtige Syntax zu verstehen.

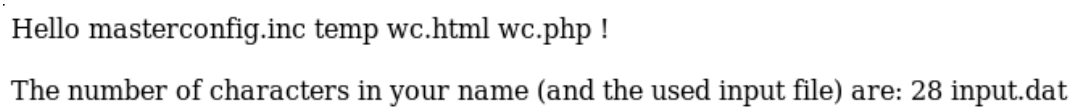
Syntax: *<!--#befehl parametername1="wert" parametername2="wert" -->*

Durch Kombination des SSI-Befehls „exec“ und des Linux-Befehls „ls“, haben wir die im Verzeichnis liegenden Dateien aufgelistet bekommen und eine „masterconfig.inc“ Datei gefunden. Um dessen Inhalt zu erhalten, haben wir den SSI-Befehl „include“ verwendet.

SSI-Befehl: *exec cmd=“ls ..“*

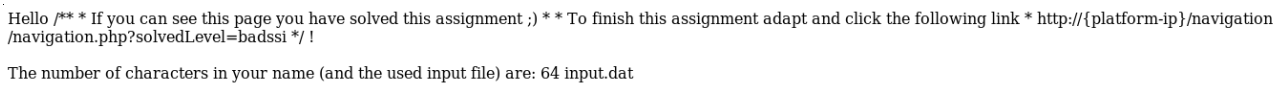


Ausgabe:



SSI-Befehl: *include virtual=“/inetsec-ssi-as1/masterconfig.inc“*

Ausgabe:



PHP: Simple Gallery

Da es uns möglich war die Verzeichnisliste des Ordners am Webserver zu sehen, erkannten wir, dass sich die gesuchte Datei „config.php“ im selben Verzeichnis wie die Bilder befindet. Nachdem wir die Download-URL eines Bildes änderten, konnten wir uns statt eines Bildes die gesuchte Datei herunterladen, in der die Lösung stand.

Downloadurl:

Inhalt der Datei „config.php“:

```php
<?php

  /**
   * If you can see this page after download as an image you solved
   * this assignment ;)
   *
   * To finish this assignment adopt and click the following link
   * http://{platform-ip}/navigation/navigation.php?solvedLevel=ppesSEPP
   */

  $thumbext = "_small.jpg";

  $db_host = "localhost";
  $db_port = "3306";
  $db_user = "weak-security-fan";
  $db_pass = "1234";
?>
```

PHP: Login

Durch ändern der URL konnten wir, wie im Beispiel zuvor, zur Verzeichnisliste gelangen, in welcher sich eine Datei namens „config.inc“ befindet. In dieser befindet sich sowohl der Username als auch das Passwort, welches wir zur Lösung des Beispiels benötigten.

URL zur Verzeichnisliste:

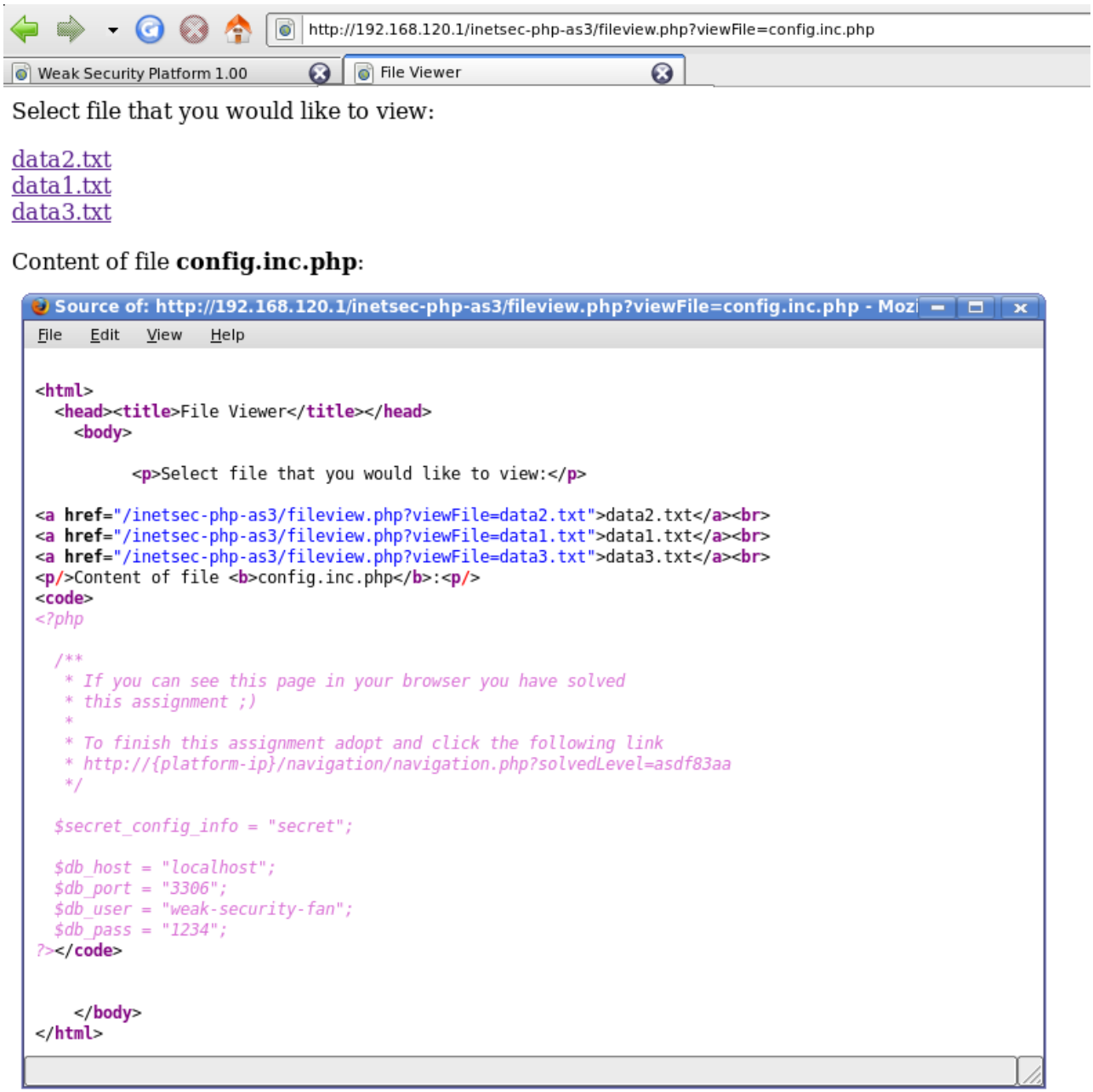
Inhalt der Datei „config.inc“:

PHP: File Viewer

Durch anklicken eines der drei Dateien und betrachten der URL, bemerkt wir, dass ein FileViewer existiert, der die Inhalte der Dateien anzeigt. Zusätzlich entdeckten wir in der Verzeichnisliste eine Datei „config.inc.php“. Durch Abänderung der URL wurde uns die Datei angezeigt. Da die Datei PHP-Code enthält, der in PHP-Tags (ähnlich der XML-Dekleration) eingebettet ist und vom Browser nicht direkt interpretiert werden kann, muss der PHP-Code über den Quelltext der Webseite betrachtet werden.

Verzeichnisliste:

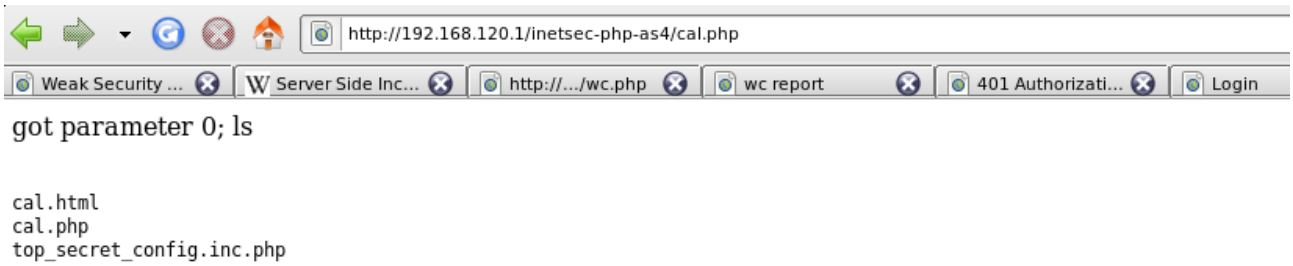
Inhalt der Datei „config.inc.php“:



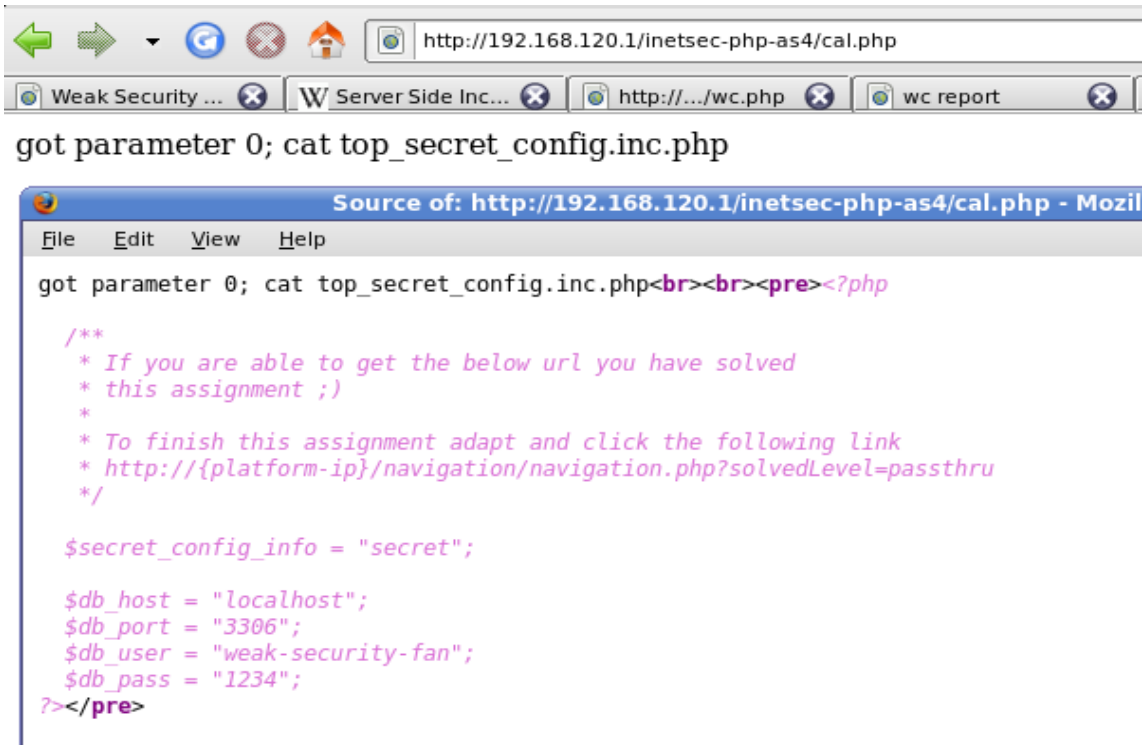
PHP: Unix Calendar

Nachdem wir eine Jahreszahl eingegeben haben , lieferte uns das Skript, wie erwartet, einen Kalender zu dieser Eingabe. Da es unsere Eingabe als Parameter benutzt hat, versuchten wird den Befehl „ls“ an die Eingabe anzuhängen - mit Erfolg. Wir entdecken die Datei „top_secret_config_inc.php“, die wir uns mit dem Befehl „cat“ ausgeben ließen. Der Inhalt der Datei war wieder im Quelltext der Webseite zu suchen.

Ausgabe mit der Eingabe *“0; ls“:*



Ausgabe mit der Eingabe *“0; cat top_secret_config.inc.php“:*



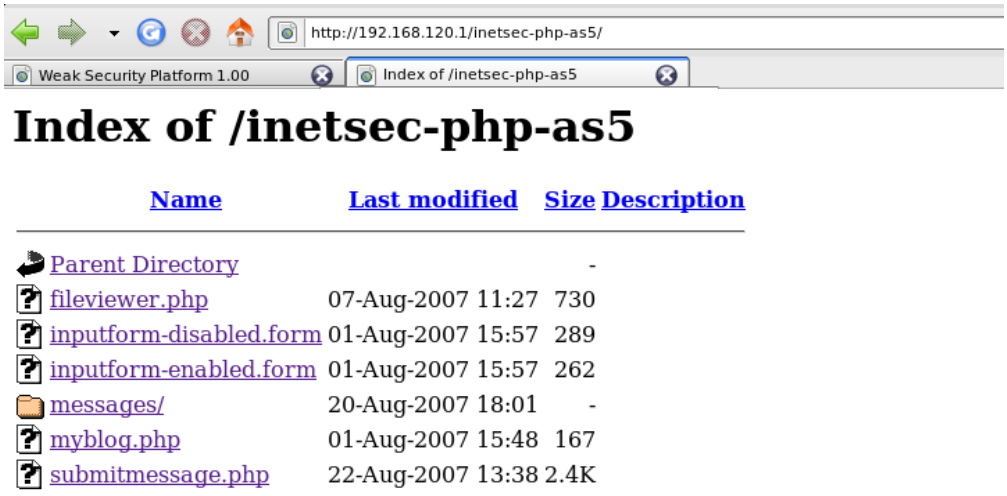
PHP: Simple Blog

Ein Blog bei dem sich drei, bereits vorhandene Einträge per FileViewer betrachten lassen. In der Verzeichnisliste entdeckten wir die Dateien „inputform-disable.form“, „inputform-enable.form“, „myblog.php“ sowie ein Verzeichnis namens „message“, in welcher sich die gespeicherten Blognachrichten befinden.

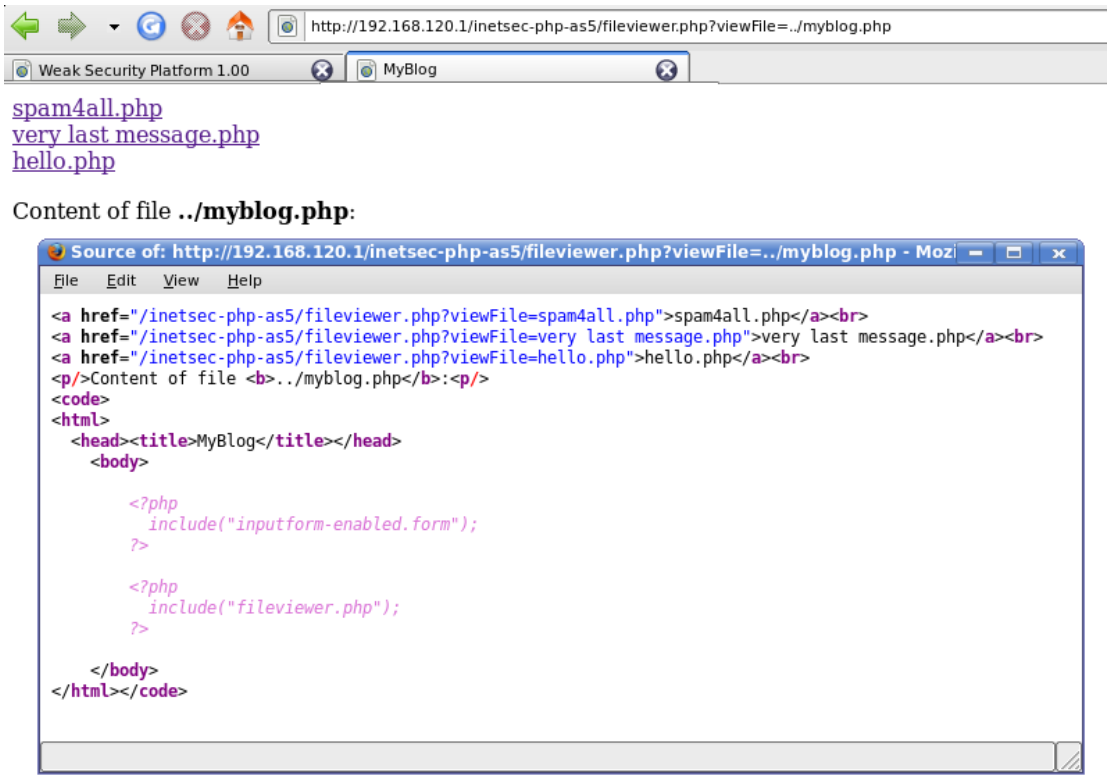
Durch den im Blog integrierten FileViewer und Anpassung der URL, konnten wir uns den Inhalt der Datei „myblog.php“ ausgeben lassen. Die PHP-Datei inkludiert die Datei „inputform-enable.form“. Ziel musste es also sein, die Datei über das Formular zu überschreiben.

Nach einer Testeingabe stellten wir fest, dass das Subjectfeld als Namensgebung für die neu zu speichernde Datei dient. Durch die Eingabe von „../myblog“ im Subjectfeld gelangt man in den übergeordneten Ordner von „message“ und überschreibt damit die Datei „myblog.php“. Um das Beispiel zu lösen musste lediglich noch der, entsprechend der Aufgabe, angepasste Quelltext der Datei „myblog.php“ angegeben.

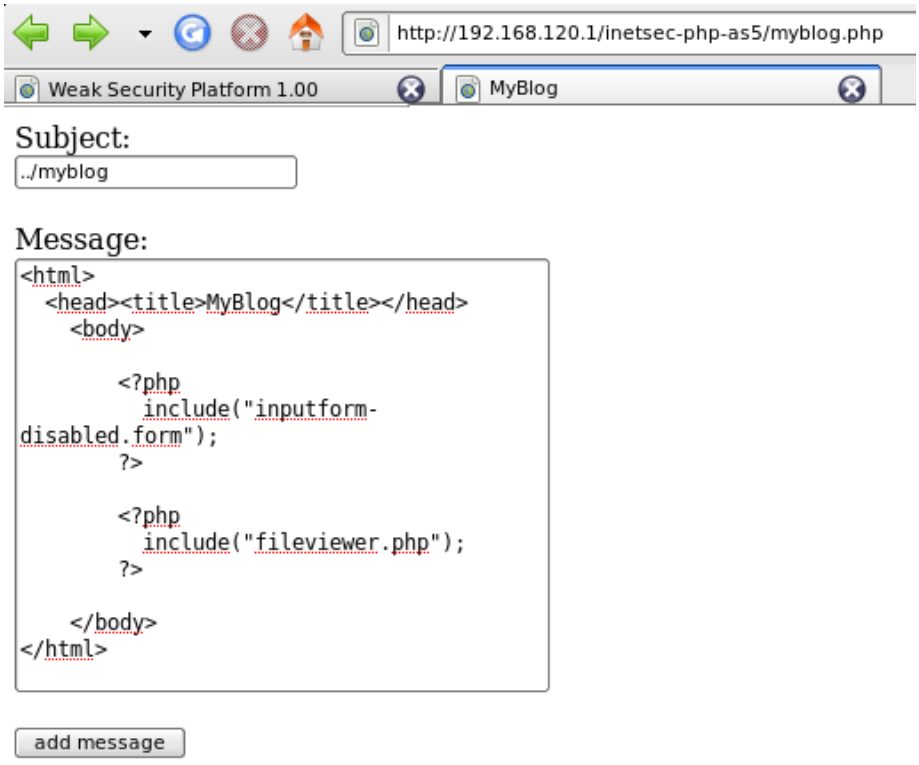
Verzeichnisliste:



Ausgabe der Datei „../myblog.php“ über den FileViewer:



Überschreiben des Inhalts der Datei „myblog.php“:



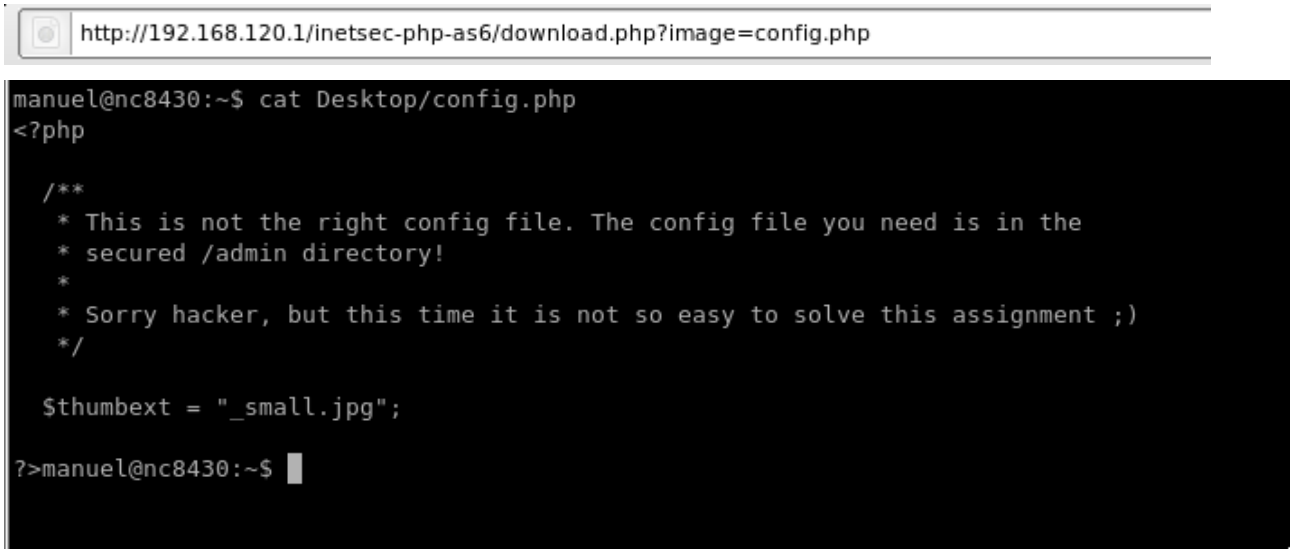
PHP: Simple Secured Gallery

Zunächst haben wir versucht, wie bei Aufgabe #7, die offensichtliche Konfigurationsdatei herunterzuladen, die wir in der Verzeichnisliste entdeckt haben. Doch diesmal war die Lösung nicht so leicht. Doch wir erhielten zumindest die Information, dass sich die Konfigurationsdatei, in einem geheimen Verzeichnis „/admin“ befindet. Der Wechsel in das Verzeichnis wurde jedoch durch eine Benutzer-/Passwortabfrage verhindert.

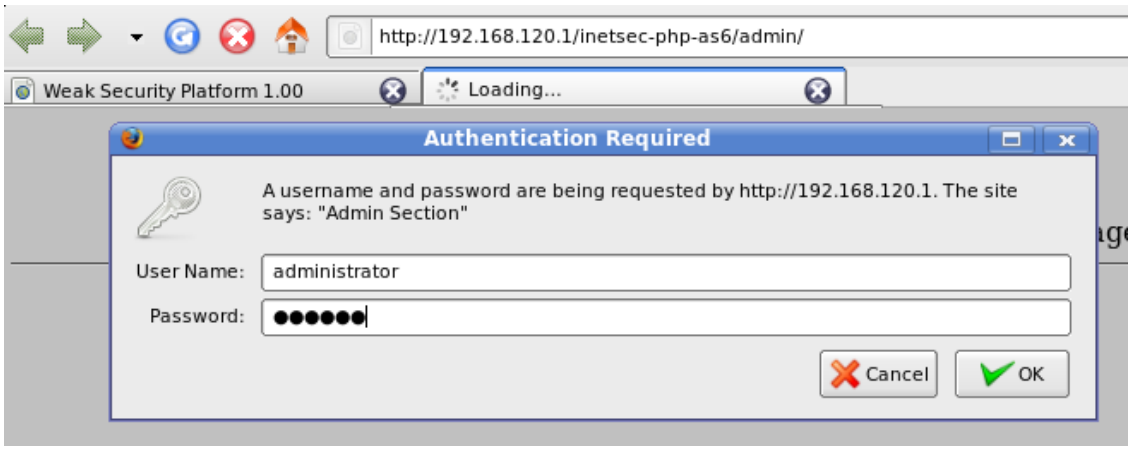
Üblicherweise speichert man die Zugangsdaten der Personen, die Zugriff haben, in der Datei „.htpasswd“. Diese Datei haben wir, wie gewohnt, über die Bildbetrachtungsfunktion der Gallery heruntergeladen. Da das Passwort MD5-verschlüsselt ist, mussten wir es mit dem Programm JohnTheRipper (Dictionary-Attack & Brute-Force) knacken.

Mit den Zugangsdaten zum Verzeichnis, erhielten wir gleichzeitig Namen der gesuchten Konfigurationsdatei, die wir zu guter letzt ebenfalls über die Bildbetrachtungsfunktion heruntergeladen haben.

URL zum Download sowie Inhalt der Datei „config.php“:



Passwortabfrage:



URL zum Download sowie Inhalt der Datei „admin/.htpasswd“


```
manuel@nc8430:~$ cat admin_.htpasswd
administrator:$1$AAODv...$gXPqGkIO3Cu6dnclE/sok1
manuel@nc8430:~$
```

Aufruf und Ausgabe von John:

```
manuel@nc8430:~/john-1.7.3.4/run$ ./john ~/admin_.htpasswd
Loaded 1 password hash (FreeBSD MD5 [32/64 X2])
shadow           (administrator)
guesses: 1  time: 0:00:00:00 100% (2)  c/s: 3882  trying: ranger - shadow
manuel@nc8430:~/john-1.7.3.4/run$
```

Verzeichnisliste des Verzeichnis „/admin“:

URL zum Download der endgültigen Konfigurationsdatei:

JavaSE: Login

Die Seite enthält ein JavaApplet namens „JavaAs1.class“, die ganz gewöhnlich heruntergeladen werden kann und den kompilierten Javabytecode enthält. Um den ursprünglichen Javacode zu erhalten, wird dieser dekompiliert. Das Passwort „compilation“ lässt sich anschließend einfach durch Analyse des Javacodes gewinnen.

```
manuel@nc8430:~$ zip JavaAs1.jar JavaAs1.class
  adding: JavaAs1.class (deflated 47%)
manuel@nc8430:~$ java -cp jode-1.1.2-pre1.jar jode.decompiler.Main JavaAs1.jar
Jode (c) 1998-2001 Jochen Hoenicke <jochen@gnu.org>
JavaAs1
/* JavaAs1 - Decompiled by JODE
 * Visit http://jode.sourceforge.net/
 */
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.net.MalformedURLException;
import java.net.URL;

import javax.swing.JApplet;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class JavaAs1 extends JApplet implements ActionListener
{
    private String password;
    private JPanel loggedInPanel;
    private JButton loginButton;
    private JButton finishButton;
    private JPasswordField passwordField;

    public void init() {
        password = "p";
        JavaAs1 javaas1 = this;
        String string = javaas1.password;
        StringBuilder stringbuilder = new StringBuilder;
        ((UNCONSTRUCTED)stringbuilder).StringBuilder(string);
        javaas1.password = stringbuilder.append("il").toString();
        JavaAs1 javaas1_0_ = this;
        String string_1_ = javaas1_0_.password;
        StringBuilder stringbuilder_2_ = new StringBuilder;
        ((UNCONSTRUCTED)stringbuilder_2_).StringBuilder(string_1_);
        javaas1_0_.password = stringbuilder_2_.append("ation").toString();
        password = new StringBuilder("com").append(password).toString();
        password = new StringBuilder("de").append(password).toString();
        loggedInPanel = new JPanel(new BorderLayout());
        JTextArea textArea = new JTextArea();
        textArea.setEditable(false);
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);
        textArea.setText
            ("Very important information ...\nVery  information ...\nVery important
information ...\n\n\nCongratulations, you solved this assignment. Click the following
button to finish ...");
```

```
        loggedInPanel.add(new JScrollPane(textArea), "Center");
        loggedInPanel.add(finishButton = new JButton("Finish"), "South");
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(passwordField = new JPasswordField(20));
        getContentPane().add(loginButton = new JButton("Login"));
        finishButton.addActionListener(this);
        loginButton.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == loginButton) {
            if (new String(passwordField.getPassword()).equals(password)) {
                getContentPane().removeAll();
                setContentPane(loggedInPanel);
                validate();
            }
        } else if (e.getSource() == finishButton) {
            try {
                getAppletContext().showDocument
                    (new URL
                     ("http", getCodeBase().getHost(),
                      "/navigation/navigation.php?solvedLevel=olrtb7z"));
            } catch (MalformedURLException malformedurlexception) {
                /* empty */
            }
        }
    }
}

Decompiled 1 classes.
manuel@nc8430:~$ ==> decompilation
```

JavaSE: Login Advanced

Wie beim vorigen Beispiel erfuhren wie, dass das JavaApplet seinen Code aus einer „JavaAs2.class“ bezog und diese sich in einem „javaas2.jar“ Archiv befindet. Wieder konnten wir dieses jar-Archiv herrunterladen und die darin befindliche class-Datei dekomplilieren. Aus dem Javacode erhielten wir ein als MD5HEX-String verschlüsseltes Passwort, welches wir nur noch mit JohnTheRipper (gepatched mit „giant patch“) entschlüsseln mussten.

```
manuel@nc8430:~$ java -cp jode-1.1.2-pre1.jar jode.decompiler.Main javaas2.jar
Jode (c) 1998-2001 Jochen Hoenicke <jochen@gnu.org>
JavaAs2
/* JavaAs2 - Decompiled by JODE
 * Visit http://jode.sourceforge.net/
 */
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.net.MalformedURLException;
import java.net.URL;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import javax.swing.JApplet;
import javax.swing.JButton;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class JavaAs2 extends JApplet implements ActionListener
{
    private static final String pwdHash = "1a1dc91c907325c69271ddf0c944bc72";
    private JPanel loggedInPanel;
    private JButton loginButton;
    private JButton finishButton;
    private JPasswordField passwordField;

    public void init() {
        loggedInPanel = new JPanel(new BorderLayout());
        JTextArea textArea = new JTextArea();
        textArea.setEditable(false);
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);
        textArea.setText
            ("Very important information ...\nVery important information ...\nVery
important information ...\n\n\nCongratulations, you solved this assignment. Click the
following button to finish ...");
        loggedInPanel.add(new JScrollPane(textArea), "Center");
        loggedInPanel.add(finishButton = new JButton("Finish"), "South");
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(passwordField = new JPasswordField(20));
        getContentPane().add(loginButton = new JButton("Login"));
        finishButton.addActionListener(this);
        loginButton.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == loginButton) {
            try {
```

```
                String enteredPassword
                    = new String(passwordField.getPassword());
                MessageDigest md = MessageDigest.getInstance("MD5");
                md.update(enteredPassword.getBytes());
                byte[] md5byteArray = md.digest();
                StringBuffer md5HexString = new StringBuffer();
                for (int i = 0; i < md5byteArray.length; i++)
                    md5HexString.append(toHexString(md5byteArray[i]));
                if (md5HexString.toString().equalsIgnoreCase
                    ("1a1dc91c907325c69271ddf0c944bc72")) {
                    getContentPane().removeAll();
                    setContentPane(loggedInPanel);
                    validate();
                }
            } catch (NoSuchAlgorithmException ex) {
                JOptionPane.showMessageDialog(this,
                                              new StringBuilder
                                                  ("Internal Error: ").append
                                                  (ex.getMessage())
                                                  .toString());
            }
        } else if (e.getSource() == finishButton) {
            try {
                getAppletContext().showDocument
                    (new URL
                     ("http", getCodeBase().getHost(),
                      "/navigation/navigation.php?solvedLevel=zumk30oO"));
            } catch (MalformedURLException malformedurlexception) {
                /* empty */
            }
        }
    }

    private String toHexString(byte b) {
        int value = (b & 0x7f) + (b < 0 ? 128 : 0);
        String ret = value < 16 ? "0" : "";
        ret = new StringBuilder(ret).append(Integer.toHexString(value))
                  .toString();
        return ret;
    }
}

Decompiled 1 classes.
```

Aufruf und Ausgabe von JohnTheRipper:

```
manuel@nc8430:~$ echo "doesntmatter:1a1dc91c907325c69271ddf0c944bc72" > mypw
manuel@nc8430:~$ ./john-1.7.3.4/run/john --format=raw-MD5 mypw
Loaded 1 password hash (Raw MD5 [raw-md5])
pass             (doesntmatter)
guesses: 1  time: 0:00:00:00 100% (2)  c/s: 36500  trying: nichole - peggy
```

JavaEE5 (Seam): Webshop

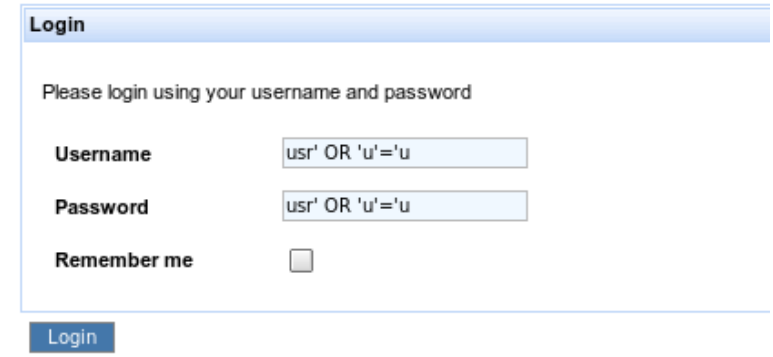
Der erste Teil der Aufgabe bestand darin, sich mit nicht vorhandenen Zugangsdaten einzuloggen. Da in der Angabe bereits SQL erwähnt wurde, lag die Vermutung einer SQL-Injection nahe.

Wir nahmen folgendes Query an: *"SELECT * FROM usertable WHERE username='" + username + "' AND password='" + password + "';*

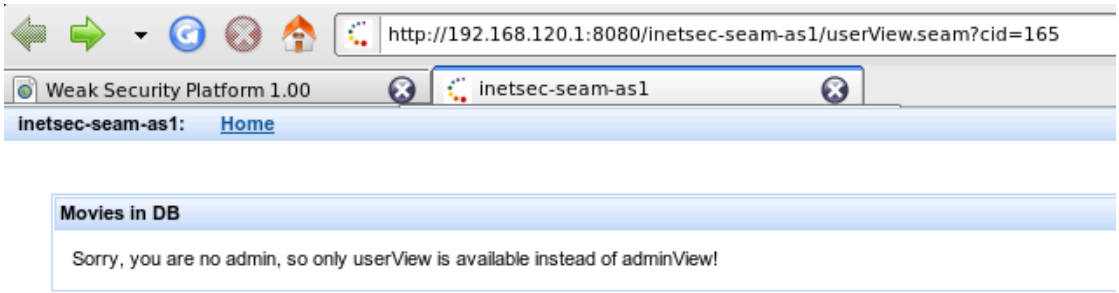
Ausgehend von diesem Query, mussten wir für einen erfolgreichen Login, unsere Eingabe so anpassen, dass beide Zweige der Konjunktion „wahr" werden. Dies brachte uns zu der Lösung „*usr' OR 'u'='u'*".

Eingeloggt im Webinterface musste lediglich noch die URL angepasst werden, um das Admininterface zu erreichen.

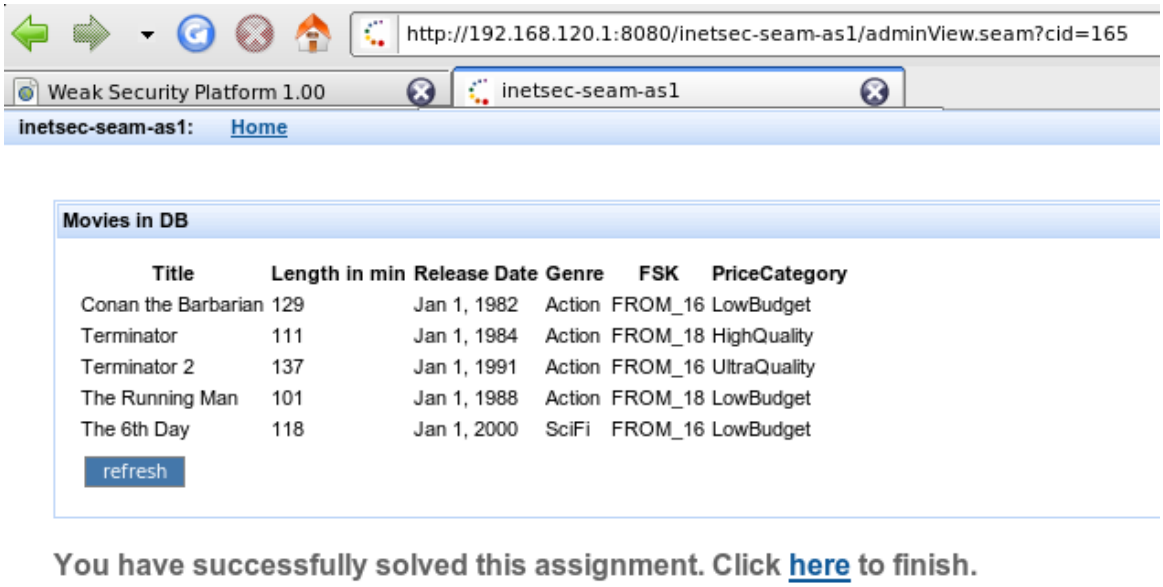
Einloggen durch SQL-Injection:



Userinterface:



Admininterface:

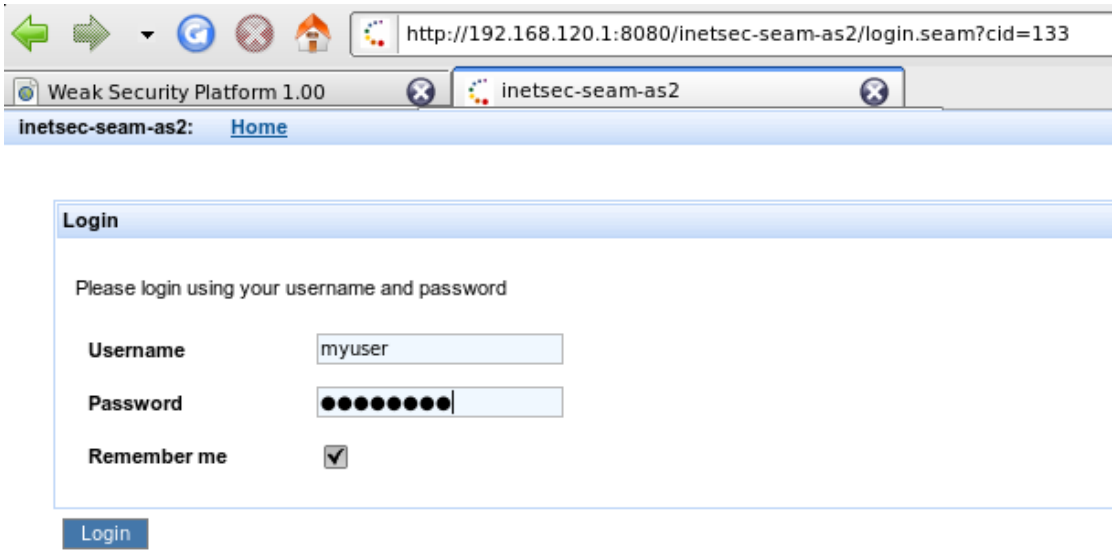


JavaEE5 (Seam): Webshop Reloaded

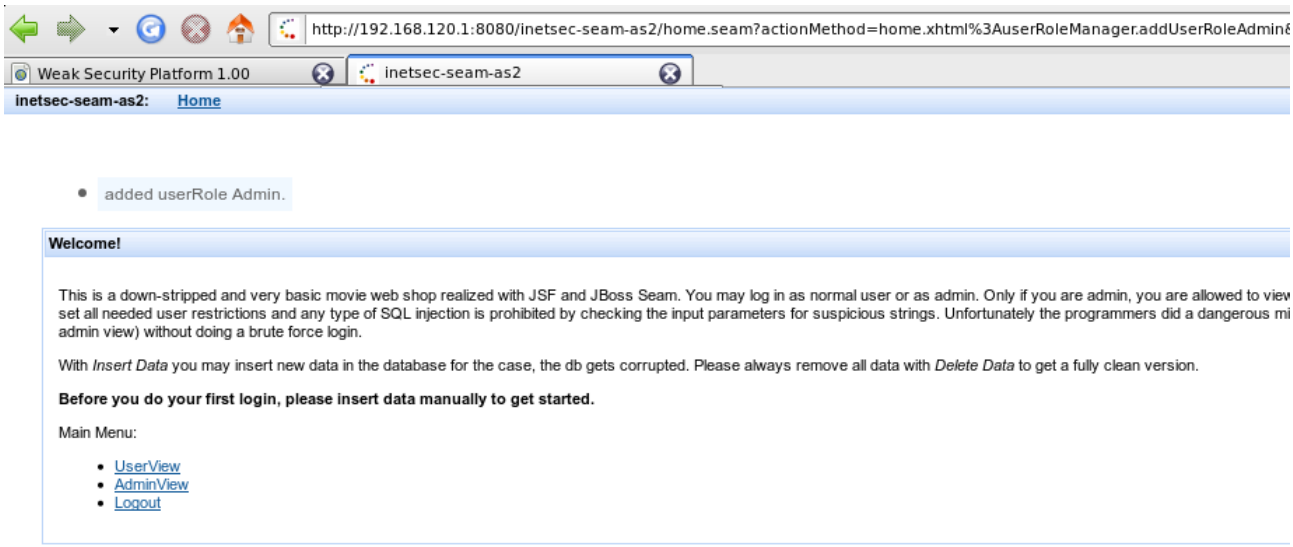
Nach kurzer Analyse des bereitgestellten Sourcecodes, war schnell klar, dass es eine eigene öffentliche Seite zur Erstellung eines Benutzers gibt. Mit diesem konnten wir uns erfolgreich einloggen, jedoch besaßen wir nur Benutzerrechte. Laut Codeanalyse erhält jedoch ausschließlich der Benutzer „admin“, den es bereits in der Datenbank gibt und der nicht neu angelegt werden kann, Adminrechte. Die Rechte werden über den internen Aufruf einer Funktion zugewiesen. Diese Funktion ist jedoch auch über HTTP aufrufbar, womit wir selbst in der Lage waren, uns Adminrechte zuzuweisen und die Aufgabe somit lösten.

Erzeugen eines neuen Benutzers:

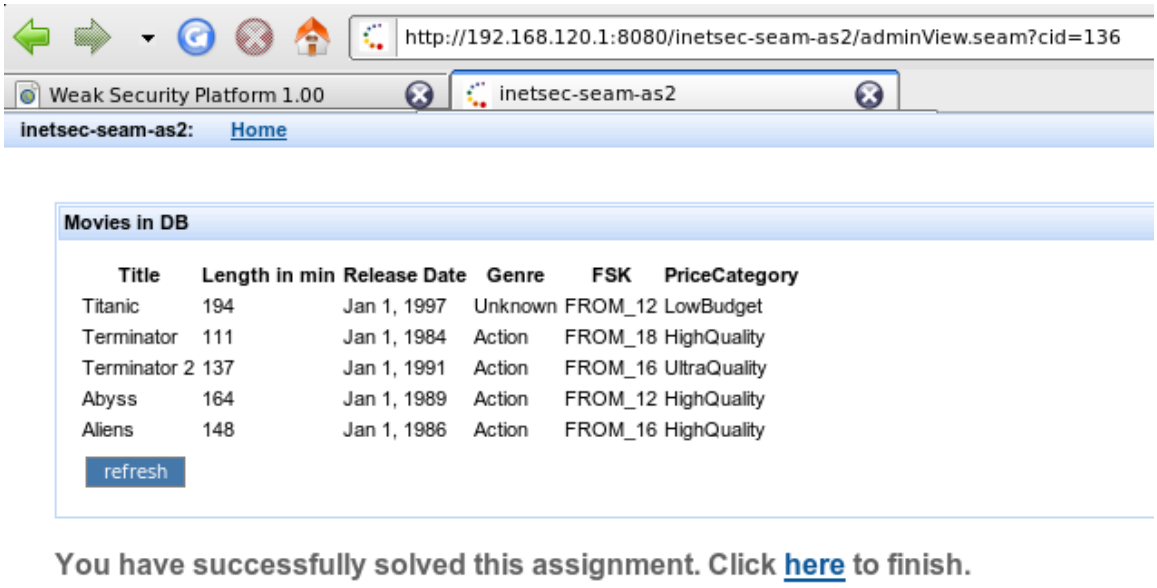
Login mit diesem Benutzer:



Aufruf der Funktion „userRoleManager::addUserRoleAdmin“ über HTTP:



Admininterface:



Screenshot: Alle Aufgabe gelöst

Weak Security Platform 1.00

Name	Description	Solved
Cryptographic: Basic	Try to break a simple password encryption algorithm.	
Cryptographic: Intermediate	This is an enhanced version of the basic algorithm.	
Browser: Cookie	Set the right value of the right cookie.	
Browser: User Agent	Only users with the special *weak security platform user agent* can solve this assignment.	
SSI: WordCount	Try to get the config file using bad SSI programming.	
PHP: Simple Gallery	A weak implementation of a PHP-based image gallery. Secure information is stored in a config.php file.	
PHP: Login	Retrieve the login data.	
PHP: File Viewer	Get the configuration file which stores secure information.	
PHP: Unix Calendar	Try to trick that simple script of an unix calendar to get the relevant content of the secret config file.	
PHP: Simple Blog	Disable the form fields by using the web interface.	
PHP: Simple Secured Gallery	A security improved version of the first PHP assignment. Try to get the config file again. Hint: This assignment is using the security mechanisms of the popular Apache 2 webserver. Useful Tool	
JavaSE: Login	Get the login password.	
JavaSE: Login Advanced	Security enhanced version of the previous assignment. Get the login password.	
JavaEE5 (Seam): Webshop	Break the login and get admin access. (HQL/SQL Injection) **This assignment is hosted on a JBoss Application Server. It may take some minutes to get ready after booting the system.**	
JavaEE5 (Seam): Webshop Reloaded	Get admin access by using not proberly secured application functions. (Code Analyzing) Source Code **This assignment is hosted on a JBoss Application Server. It may take some minutes to get ready after booting the system.**	

Vienna University of Technology, 2007