Εργασία 6 – Κρυπτογραφία 2017

EXPLOITING ENCRYPTED COOKIES FOR FUN AND PROFIT

ΠΑΝΑΓΙΩΤΑ ΘΩΜΟΠΟΥΛΟΥ – Π14053

Περιεχόμενα

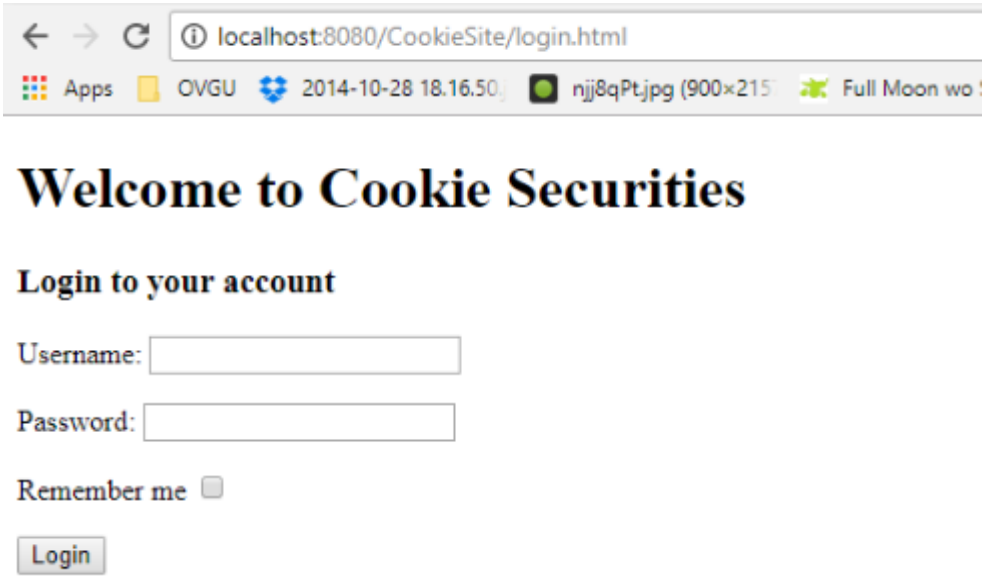
Εκφώνηση άσκησης....2
Εκτέλεση κώδικα....3
Επεξήγηση του κώδικα....5
Cookies σε CMS....6
Cookies σε Wordpress....6
Cookies σε Drupal....7
Cookies σε Joomla!....7

Εκφώνηση άσκησης

Φτιάξτε μία εφαρμογή η οποία χρησιμοποιεί encrypted cookies τα οποία να μην είναι ασφαλή ώστε να μπορείτε να τα εκμεταλλευτείτε μέσω αυτής της επίθεσης https://spring.io/blog/2014/01/20/exploiting-encrypted-cookies-for-fun-and-profit (περιέχει κώδικα). Στην συνέχεια κάντε default εγκαταστάσεις των joomla, drupal και wordpress και μελετήστε αν και κατά πόσον οι παραπάνω εφαρμογές εφαρμόζουν κάποια από αυτές τις μεθόδους και ποιες είναι οι απαραίτητες προϋποθέσεις για να το προσφέρουν.

Εκτέλεση κώδικα

Τρέχοντας τον κώδικα από το login.html αρχείο, βλέπουμε την σελίδα login:



Για να συνδεθούμε, βάζουμε τα στοιχεία:

Username: penny

Password: secret

Και επιλέγουμε το remember me, για να μπορέσει να δημιουργηθεί το cookie,

Welcome to Cookie Securities

Login to your account

Username: penny

Password: ••••••

Remember me

Login

και πατώντας Login, μεταφερόμαστε στην σελίδα index:

Welcome to the Cookie Securities

Would you like to log in as an Admin?

Yes!

Πατώντας “Yes!” μπορούμε να δούμε στο console ότι όντως συνδεθήκαμε ως admin:

```
Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (11 Οκτ 2017, 1:52:57 π.μ.)
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 905 ms
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.catalina.core.StandardService startInternal
INFO: Starting service [Catalina]
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/8.5.23
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Οκτ 11, 2017 1:52:59 ΠΜ org.apache.catalina.startup.Catalina start
INFO: Server startup in 503 ms
Hacked in as admin
```

Επεξήγηση του κώδικα

Από το αρχείο login.html, καλούμε την doPost μέθοδο της κλάσης Login.java με τα στοιχεία login ενός χρήστη.

```
<form action="Login" method="post">
```

Από εκεί, καλούμε την μέθοδο login της κλάσης Server.java η οποία αρχικά ελέγχει αν υπάρχει ο χρήστης με τα στοιχεία login, και δημιουργεί το encrypted cookie

```
String originalText = "username=" + username +
"&firstName="+user.getFirstName()+"&lastName="+user.getLastName();
byte[] encrypted = cipher.doFinal(originalText.getBytes());
byte[] iv = cipher.getIV();
return createEncryptedCookie(iv, encrypted);
```

Στην συνέχεια αποθηκεύουμε το cookie στο session,

```
session.setAttribute("cookie", userCookie);
```

και κάνουμε redirect στην κεντρική σελίδα μας, index.html.

```
response.sendRedirect("index.html");
```

Από εκεί, πατώντας το κουμπί “Yes!” καλούμε την doPost μέθοδο της κλάσης CookieHacking.java

```
<form action="CookieHacking" method="post">
```

Η οποία παίρνει το cookie του session του χρήστη και το αποθηκεύει στην μεταβλητή `userCookie`. Βρίσκουμε το Initialization Vector από το cookie μας:

```
byte[] originalIv = Utils.extractIv(userCookie);
```

και βρίσκουμε το IV του χρήστη Admin κάνοντας XOR με το IV μας και το όνομα χρήστη μας,

```
String pennyPlainTextFirstBlock = "username=penny&f";
```

και ύστερα με το όνομα χρήστη Admin:

```
String adminPlainTextFirstBlock = "username=admin&f";
```

Τέλος, δημιουργούμε το encryptedCookie του Admin με την χρήση του user cookie και το Admin IV,

```
String adminCookie = Utils.createEncryptedCookie(adminIv,
originalEncryptedText);
```

και εκτυπώνουμε το username του Admin στο console:

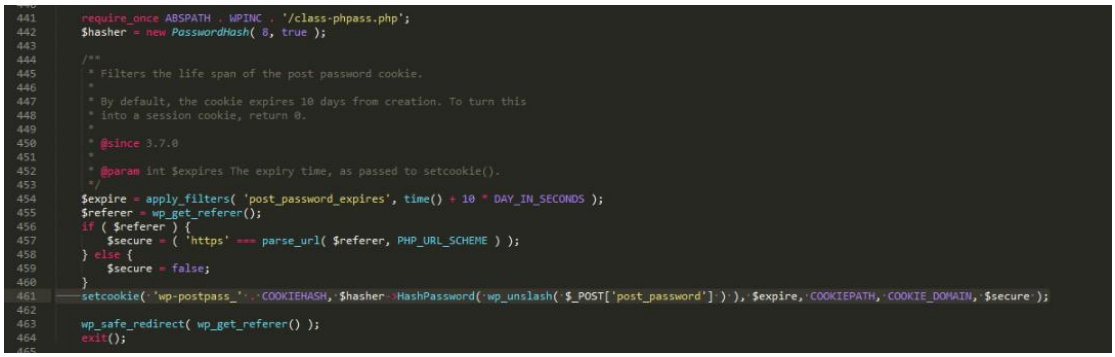
```
admin = server.getUsername(adminCookie);

System.out.println("Hacked in as " + admin);
```

Cookies σε CMS

Cookies σε Wordpress

Το Wordpress διαθέτει cookies για την αποθήκευση των δεδομένων login. Ένα μέρος του κώδικα βρίσκεται στο αρχείο wp-login.php:



```
    require_once ABSPATH . WPINC . '/class-phpass.php';
    $hasher = new PasswordHash( 8, true );

    /**
     * Filters the life span of the post password cookie.
     *
     * By default, the cookie expires 10 days from creation. To turn this
     * into a session cookie, return 0.
     *
     * @since 3.7.0
     *
     * @param int $expires The expiry time, as passed to setcookie().
     */
    $expire = apply_filters( 'post_password_expires', time() + 10 * DAY_IN_SECONDS );
    $referer = wp_get_referer();
    if ( $referer ) {
        $secure = ( 'https' === parse_url( $referer, PHP_URL_SCHEME ) );
    } else {
        $secure = false;
    }
    setcookie( 'wp-postpass_' . COOKIEHASH, $hasher->HashPassword( wp_unslash( $_POST['post_password'] ) ), $expire, COOKIEPATH, COOKIE_DOMAIN, $secure );

    wp_safe_redirect( wp_get_referer() );
    exit();
```

setcookie('wp-postpass_' . COOKIEHASH, $hasher->HashPassword(wp_unslash($_POST['post_password'])), $expire, COOKIEPATH, COOKIE_DOMAIN, $secure);

Από αυτήν τη γραμμή κώδικα, μπορούμε να καταλάβουμε ότι τα cookies κατακερματίζονται (hash). Από την στιγμή που τα cookies δεν είναι κρυπτογραφημένα, αλλά κατακερματισμένα, δεν γίνεται να αλλαχθεί το περιεχόμενο τους χωρίς να γνωρίζουμε το αρχικό -μη κατακερματισμένο- cookie. Άρα, ο μόνος τρόπος να κάνουμε log in ως admin είναι να δημιουργήσουμε το cookie του admin, γνωρίζοντας τον κωδικό του, και μετά να το περάσουμε από την συνάρτηση κατακερματισμού του Wordpress. Προφανώς, όμως, αν γνωρίζουμε τον κωδικό του admin, μπορούμε να κάνουμε έτσι κι αλλιώς log in. Επομένως, ο μόνος τρόπος να χρησιμοποιήσουμε την παραπάνω τεχνική με τα unsigned cookies θα ήταν αν τα cookies περνούσαν από κάποιο αλγόριθμο encryption, αντί για αλγόριθμο κατακερματισμού.

Cookies σε Drupal

Το Drupal δεν αποθηκεύει τα δεδομένα login σε cookies. Σε κάθε session πρέπει να γίνεται login.

Cookies σε Joomla!

Το Joomla! επίσης δεν αποθηκεύει τα δεδομένα login σε cookies.