DJEye: מ-XSS בפורום ועד להשתלטות על רחפן

מאת עודד ואנונו, דיקלה ברדה ורומן זאיקין

הקדמה

תעשיית הרחפנים מוערכת בכ-127 מיליארד דולר, ונשלטת ברובה על ידי חברת הרחפנים הסינית DJI. החברה מחזיקה בנתח שוק גלובלי של כ-70% מכלל התעשייה, ופועלת ביותר ממאה מדינות.
(https://dronedj.com/2017/09/19/dji-dominates-drone-industry)

החברה שולטת הן בשוק הרחפנים לשימוש פרטי והן בשוק המקצועי, שבו רחפנים משמשים גופים ביטחוניים, כוחות שיטור, גופי תקשורת, חברות תעשייתיות, חברות חקלאות וחברות בנייה, בפיקוח ובתחזוקה של תשתיות חיוניות.

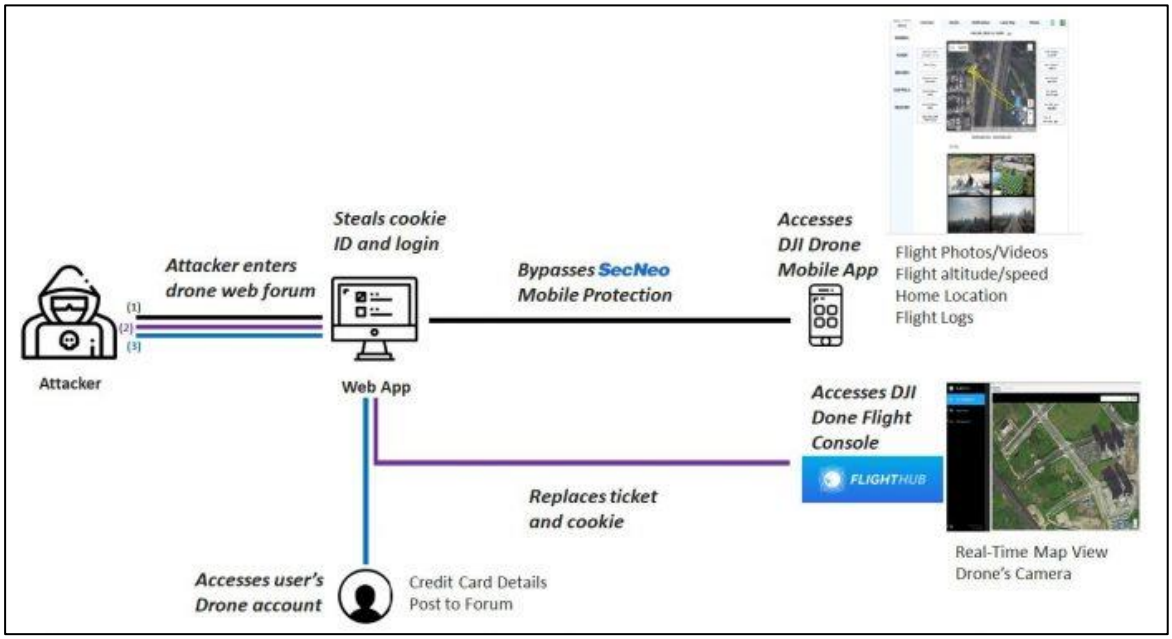
במחקר שבצענו על DJI בצ'ק פוינט, מצאנו ליקוי אבטחה אשר מעניק לתוקף גישה לחשבון DJI של המשתמש מבלי שהמשתמש יהיה מודע לכך. ובכך להקנות לתוקף גישה אל:

- תמונות וסרטונים שצולמו בזמן הטסת הרחפן, כמו כן מסלול הטיסה ובאילו זוויות התבצע הצילום.
- צפייה במצלמה בשידור חי, כמו כן צפייה במסלול הרחפן וקבלת שמע, במידה והמשתמש משתמש במערכת DJI FlightHub.
- מידע המשויך לחשבון המשתמש כגון כרטיסי אשראי, מייל וכל מידע אחר אשר שמור בפרופיל המשתמש.

ליקוי האבטחה שאיתרנו מתחיל בפורום של DJI וממשיך לתוך כל התשתיות של DJI הכוללות:

- כל פלטפורמות ה-Web של DJI כגון – החנות, הפורום וחשבון המשתמש.
- גישה לאפליקציות DJI GO ו-DJI Go 4 כמו כן לאפליקציית DJI Pilot ולכל המידע השמור בה.
- גישה וניהול הרחפנים במערכת DJI FlightHub המשמש ארגונים ברחבי העולם.

את ליקוי האבטחה דיווחנו ל-DJI בחודש מרץ והם הגדירו את ליקוי האבטחה כסיכון גבוה אך הסבירות לניצול נמוכה מכיוון שהניצול של ליקוי האבטחה מסובך יחסית.



[סיכום שלושת הממצאים בגרף פשוט]

ליקוי אבטחה בפורום של DJI

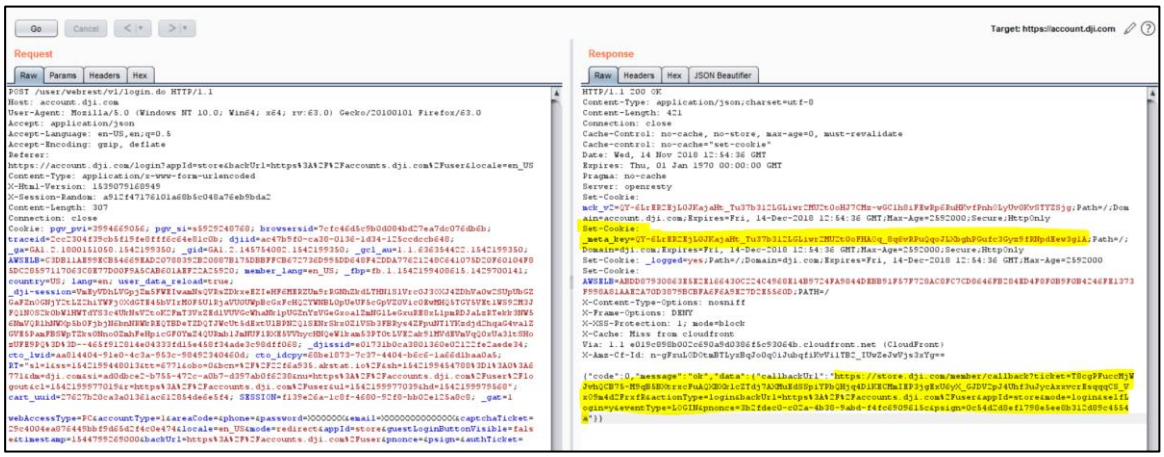
התחלנו את המחקר שלנו על ידי חקירת תהליך ההזדהות של המשתמש מול המערכת, והתמקדמו בניסיון להבין כיצד DJI מנהלת את זהות המשתמש בין כל הפלטפורמות השונות. כמו כן ניסנו להבין באיזו שיטת אימות בחרו מסך כל השיטות הקיימות כיום לצורך אימות משתמש.

על מנת לעשות זאת פתחנו burp suite והסתכלנו על כל התעבורה העוברתף, לאחר אבחון תעבורה זריז זהינו תעבורה ל-3 התתי מתחמים הבאים:

- forum.dji.com
- account.dji.com
- store.dji.com

ניסינו להבין כיצד מתבצע האימות בין כל תתי המתחמים, הריי לא נדרש אימות מחדש בין כל תת מתחם בעת המעבר בין המערכות.

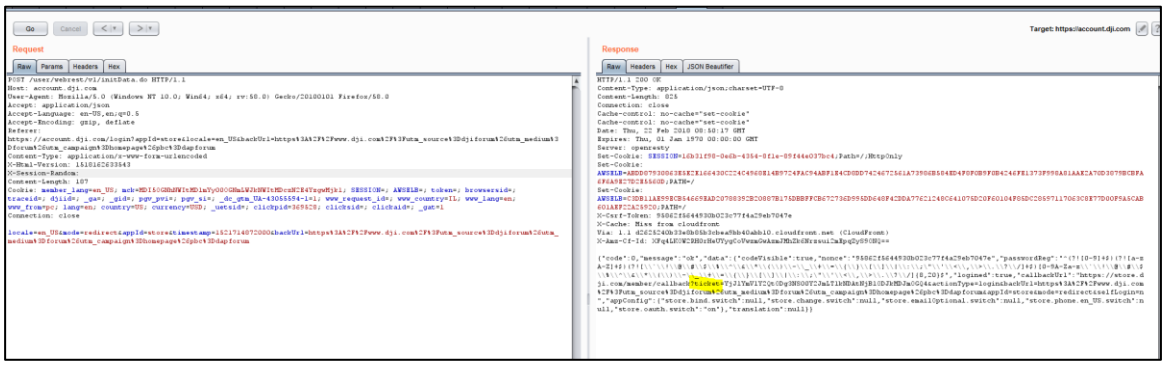
שמנו לב שברגע שמשתמש עושה התחברות בכל אחד מתתי המתחמים הוא מעובר ל-account.dji.com, שם הוא מזין את שם המשתמש והסיסמא שלו וכתשובה הוא מקבל redirect ticket ועוגייה בשם _meta_key:



ברגע שמתבצעת ההפניה הוא מקבל עוד מספר עוגיות שמאפיינות את החשבון שלנו.

```
Connection: close
Cache-Control: no-cache
Date: Wed, 14 Nov 2018 13:37:58 GMT
Location: https://accounts.dji.com/user
Server: openresty
Set-Cookie: user_data_reload=true; domain=.dji.com; path=/; expires=Fri, 14 Dec 2018 13:37:58 -0000
Set-Cookie: country=US; domain=.dji.com; path=/; expires=Wed, 14 Nov 2018 14:37:58 -0000
Set-Cookie: cart_uuid=dffe8990ca3801360df3122fe2aede34; domain=.dji.com; path=/; expires=Thu, 14 Nov 2019 13:37:58 -0000
Set-Cookie: cart_data_qty=0; domain=.dji.com; path=/; expires=Thu, 14 Nov 2019 13:37:58 -0000
Set-Cookie: user_nick_name=DroneIT%27sMe%21; domain=.dji.com; path=/; expires=Fri, 14 Dec 2018 13:37:58 -0000
Set-Cookie: user_avatar_url=%2F%2Fasset1.djicdn.com%2Fuploads%2Favatars%2Fthumb_5524ea2c-c46c-4497-9bcc-936bf404ad14.jpg; do
Set-Cookie:
_dji-session=Y28wUXdScThQWlVnMGUlbmtQWTI0T0M2TGNuSWVtb3B6UXF1WkdJYmNtNGVRbFAxbEZrcFFYd200dEc1YlhER1ViSGZRdWdmRmd2MTlGVWhvYUS
DBGQklZbmhFZ2xDMGJ4U0ozbUJvTzVPUW4zNmtNNEVXV3NuMnAwRkF6V2pDdGhIZThTME1JTFlnSm5I0EpCWDRSczVKQThudldBbHhNY3ZzUXljUGNLbnJEb2Y2T
5b772085894b8d7867a38a4d39eb583865e; domain=.dji.com; path=/; Http0nly
Strict-Transport-Security: max-age=86400; includeSubdomains
Via: nw, 1.1 9128c49d19c76fd86ec4c647434ccb0a.cloudfront.net (CloudFront)
X-Content-Type-0ptions: nosniff
X-Downgrade: 0
X-Frame-0ptions: ALL0WALL
X-NW-Proxy-Latency: 0
X-NW-Upstream-Latency: 62
X-Request-Id: 30900bc7-17e9-4371-9334-f0c53406f14d
X-Runtime: 0.053376
X-Via: w nr fr b
X-XSS-Protection: 1; mode=block
X-Cache: Miss from cloudfront
X-Amz-Cf-Id: bwt3cqR673-2zz5Be7bSsoiD0nsn9lnGB-WQV38asd_W8AGo314BTA==
```

כך המשתמש מתחבר לפורטל המבוקש, ברגע שמשתמש רוצה לעבור בין פורטל, לדוגמה מ-forum ל-store מתבצעת בקשה ל-initData.do עם העוגייה mck אשר מאפיינת את חשבון המשתמש.



ביצוע פנייה שכזאת בכל מעבר בין תת מתחם היא די לא שגרתית, מכיוון שלדפדפן יש את מזהה החשבון שהוא בעצם mck וביצוע redirect עם ticket בכל פעם תחת אותו מתחם לדעתנו הוא די מיותר.

בדרך כלל מבצעים פניות מסוג זה כאשר מתבצעות פניות בין מתחמים שונים אשר יש למצוא דרך לאפיין את החשבון כמו בביצוע אימות באמצעות Auth2.

לאחר שבחנו את מנגנון האימות הבנו שעלינו למצוא דרך להשיג את העוגייה mck, לשם כך עברנו על כל תתי המתחמים וחיפשנו האם קיים תת מתחם כלשהו ששם העוגייה אינה מוגנת באמצעות HTTPOnly.

כך הגענו לתת המתחם – forum.dji.com, במתחם זה לאמיתו של דבר לא מצאנו את העוגייה mck אך מצאנו עוגייה דומה אשר נקראת meta_key_, עוגייה זו מכילה תבנית זהה ל-mck כך שהנחנו שמדובר באותו הדבר.

לאחר שיטוט קצר בפורום, מצאנו את הפניה הבאה:

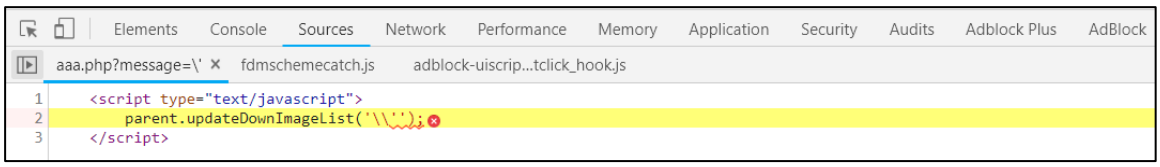
```
https://forum.dji.com/forum.php?mod=ajax&action=downremoteimg&message=
```

אם ננסה לבצע XSS ניתקל במנגנון סינון שנראה כמו addslashes שמוסיף את התו "\" לפני כל סימן מיוחד כדי להפוך אותו לטקסט, אך שמנו לב שלא מתבצעת הוספת התו "\" לתו "\" עצמו כמו בפקודה addslashes של php, כך שאפשר להניח שאנו עומדים בפני הקוד הבא:

```php
<?php

$data = addcslashes($_GET['message'], '/"\'');
echo <<<EOL
    <script type="text/javascript">
        parent.updateDownImageList('$data');
    </script>
EOL;

?>
```

אתם מוזמנים להעתיק את הקוד הזה לדף php ולנסות לנצל את ה-XSS בעצמכם, שימו לב שהפונקציה updateDownImageList לא מוגדרת בכוונה, כנראה הגענו לאיזה דף שנטען ב-iframe ול-parent שלו יש את הפונקציה הזאת ב-scope.

כדי לבצע XSS עלינו תחילה לנסות לצאת מתוך הגרשיים, כך שאם נזין \' הפונקציה addcslashes תוסיף לנו \ לגרש שלנו ובגלל שהוספנו \ בעצמנו הוא יהפוך לטקסט מה שיראה כך:



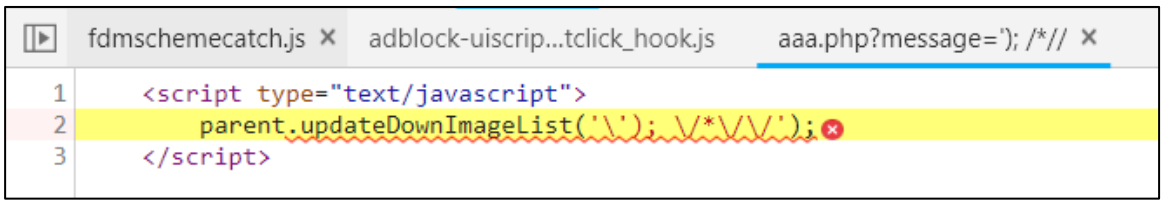
כעת עלינו למצוא דרך לטפל בצורה תקינה בתווים (' שנשארו ב-JavaScript על פי Mozilla Developers:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar

קיימים 2 סוגי הערות בשפת JavaScript:

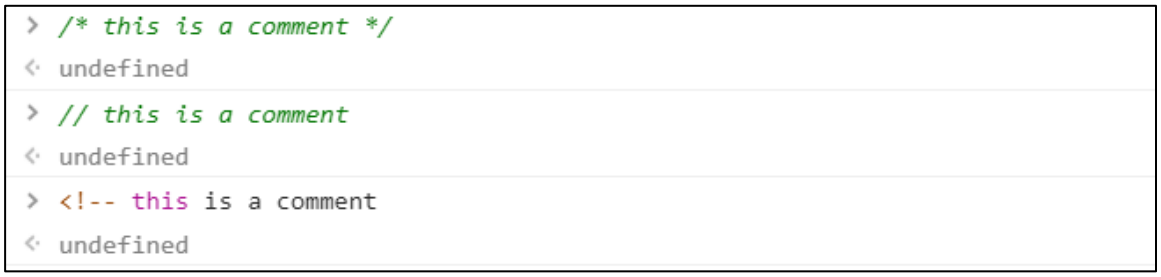
- //
- /* */

אך אם נשתמש בהם נראה שיש סינון לתו "/" שאנו צריכים על מנת לבצע הערה:

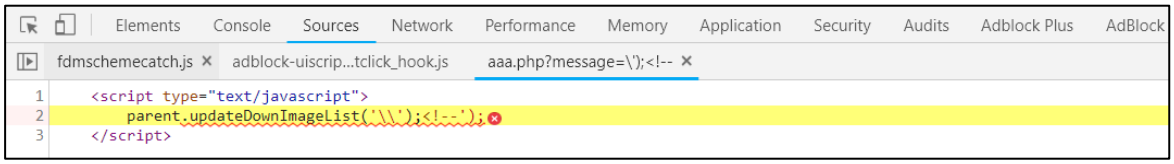


אז איך בכל זאת ניתן לנצל את ה-XSS הזה? התשובה היא באמצעות הערה שלישית שלא הרבה משתמשים בה או בכלל יודעים שיש אפשרות כזאת ב-JavaScript מכיוון שאינה מופיעה באתרי פיתוח ב-JavaScript.

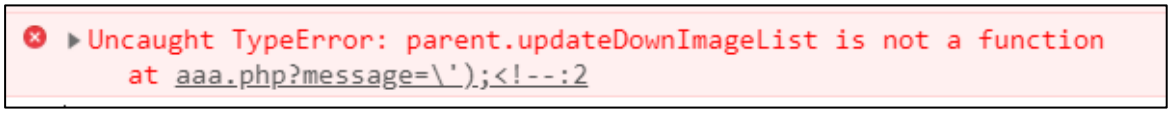
ב-JavaScript קיימת הערה נוספת והיא בעצם שימוש בחלק של הפתיחה של ההערה של HTML מה שנראה כך <!-- סוג זה של הערה זהה ל-// מה שיראה כך:



הערה זו שונה מהערת HTML מכיוון שאין לה את החלק השני שסוגר אותה "-->" בדומה ל-"/* */" ולכן היא הערה של שורה בודדת כגון הערה "//". נשתמש בהערה בצורה הבאה: **<!-- ;(\'** מה שיראה כך:



השגיאה שאנו רואים בדפדפן כעת היא:



לכן עלינו להגדיר את הפונקציה הזאת בעצמנו בצורה הבאה:

```
\'); function updateDownImageList (data) { } <!--
```

אם נכניס את הקוד הזה בדפדפן נראה שאין לנו שגיאות והכל עובד בצורה תקינה, כעת לתוך הפונקציה updateDownImageList(data) נוסיף את הקוד הזדוני שלנו שיגנוב את העוגייה בשבילנו, מה שיראה כך:

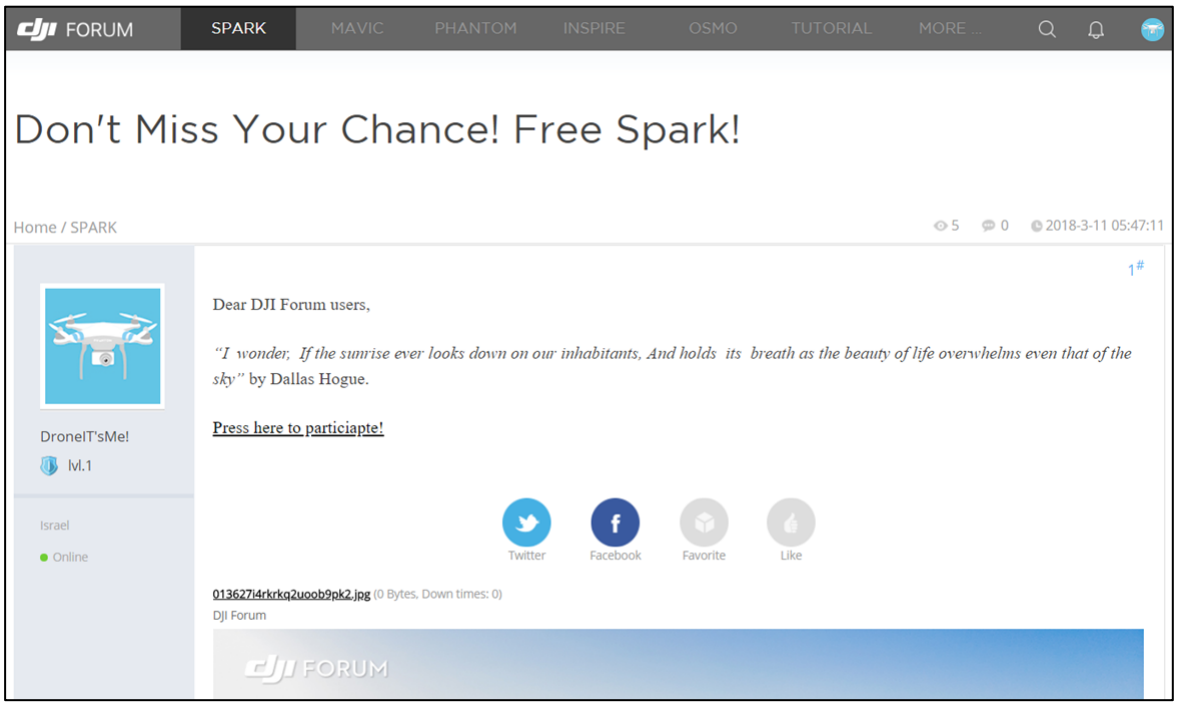
```
\'); function updateDownImageList (data) { alert(document.cookie); } <!-
-
```

ואם נכניס את זה לדפדפן נקבל את כל העוגיות:

נקודה נוספת שיש לציין היא שה-XSS שמצאנו מתאים כמו כפפה ליד עבור הגדרת הפורום של DJI אשר אומרת שבפורום של DJI ניתן ליצור קישורים רק לדפים או תמונות בתוך הפורום עצמו. בגלל שמצאנו את ה-XSS בתוך הפורום אנו יכולים להסתיר אותו בצורה מושלמת ואף להכניס אותו לתוך הודעות שניתן לפרסם בפורום.

כבונוס ה-XSS הזה גם עוקף את ה-XSS Auditor של chrome מכיוון שהוא מוזרק באופן ישיר לתוך קוד ה-JavaScript.

עד כאן הכל טוב, איך אנו ממשיכים מכאן להשגת חשבונות של משתמשים? בתור התחלה ניתן לפרסם מודעה מרתקת בפורום אשר תגרום למשתמשים ללחוץ עליה וכך נשיג את העוגיות שלהם (כמובן שפרסמנו הודעה ב-Draft כדי לא לגרום למשתמשי הפורום באמת ללחוץ על הלינקים שלנו):



במידה ומישהו לוחץ על הלינק אנו מקבלים את העוגייה meta_key שהיא בעצם זהה לעוגייה mck בפניה שהצגנו קודם לכן לקבלת ה-token.

החלפת עוגיות באופן ישיר לא עוזר להגיע לחשבון של המטרה שלחצה על הלינק ועלינו להשיג את ה-token של המשתמשים לשם כך, התהליך נראה כך:

1. יוצרים חשבון ב-DJI.com משלנו.
2. מתחברים לחשבון שיצרנו בפורום של DJI.
3. לוחצים על מעבר לתת מתחם אחר של DJI כגון store.dji.com מה שיגרום לשליחת initData.do.
4. משנים בחבילה את העוגייה mck לעוגייה meta_key שקיבלנו ב-XSS.
5. צד השרת יחזיר לנו ticket עבור החשבון של הקורבן ואנחנו בפנים.

פריצה לחשבון המשתמש מאפשר לתוקף לקבל את כל הנתונים הבאים:

- כל פרטי המשתמש הנמצאים תחת accounts.dji.com.
- פרסום הודעות בפורום בשם המשתמש לצורך פריצה לחשבונות נוספים.
- רכישת מוצרים ב-store.dji.com במידה ומחובר כרטיס אשראי לחשבון.

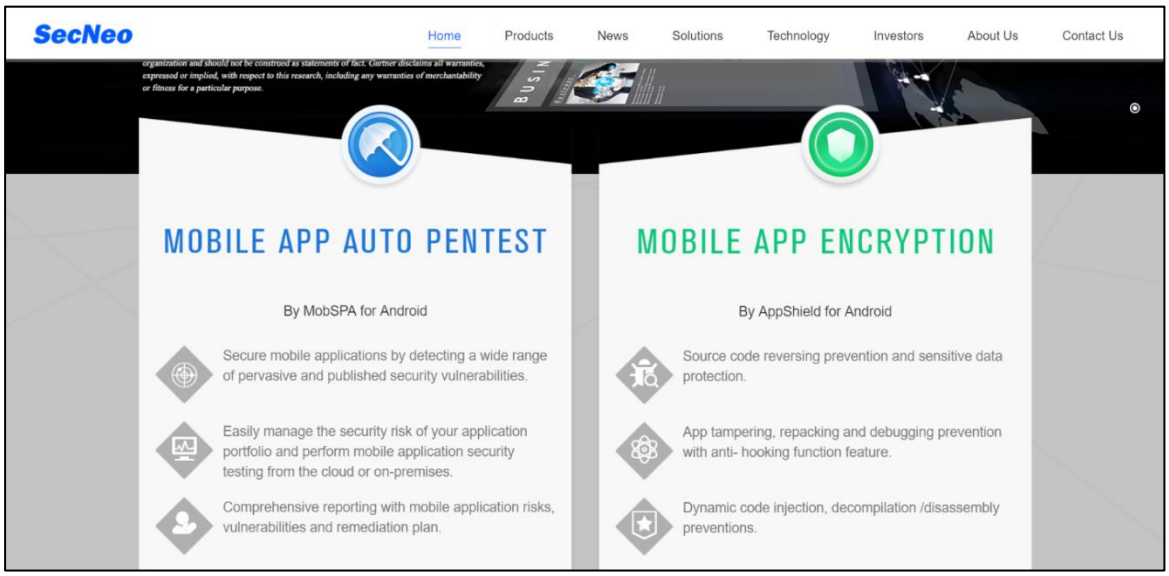
מעקף הגנות באפליקציה של DJI

על מנת להשתלט על חשבון משתמש באפליקציה של DJI היה עלינו לנתח את התעבורה שיוצאת מהאפליקציה.

מה שלא התאפשר לכאורה מכיוון ש-DJI השתמשו ב-SSL Pining שהיה עלינו לעקוף.

התחלנו בפריסת האפליקציה כדי לראות את שיטת המימוש על מנת שנוכל לעקוף אותה אך הופעתנו לגלות שהפריסה לא הצליחה וכל מה שאנו רואים זה ספריה הנקראת SecNeo.com.

אחרי חיפוש קצר בגוגל איתרנו את החברה שמספקת ל-DJI שירותי הגנה:



זו הפעם הראשונה שאנו נתקלים בחברה הזאת אך במבט חטוף נראה שמדובר בהגנה די רצינית:

- הגנה מפני הנדסה לאחור של קוד האפליקציה הן סטטי והן דינמי.
- הגנה מפני hooking.
- הגנה מפני צפייה בקוד המקור של האפליקציה.
- הצפנה של כל הנתונים הנשמרים בדיסק הקשיח.

הסתכלנו מעט על האפליקציה והיה נראה ש-DJI פשוט משתמשים בכל השירותים שהחברה הזאת מספקת, הכל מוצפן ולא שגרתי.

התחלנו את המעקף של SecNeo על ידי חיבור frida לטלפון והצגת כל התהליכים הפעילים:

```
frida-ps -U
```

ברשימת האפליקציות מצאנו את האפליקציה dji.go.v4 וניסינו להתחבר אליה באמצעות הקוד הנאיבי הבא:

```
import frida, sys


jscode = """
    Java.perform(function() {
        console.log("working");
    });
"""

process = frida.get_usb_device().attach('dji.go.v4')
script = process.create_script(jscode)
script.load()
sys.stdin.read()
```

כמובן שקיבלנו מיד סירוב ולא הצלחנו להתחבר לאפליקציה. איך יכול להיות שקיבלנו סירוב? יכול להיות שכבר יש אפליקציה נוספת שמחוברת לאפליקציה הזאת ב-debug?

הסתכלנו שוב ב:

```
frida-ps -U
```

ואכן כן! האפליקציה רצה פעמיים:

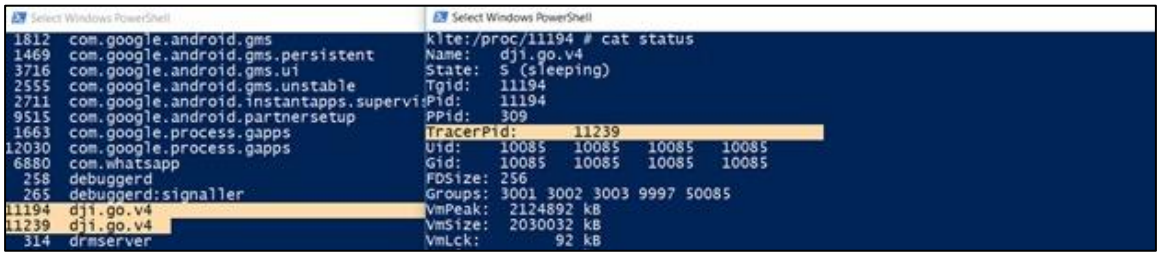
ובעצם מחוברת לעצמה ב-debug, מערכת ההפעלה מאפשרת רק לתוכנה 1 להתחבר ב-debug ולכן אנחנו לא הצלחנו להתחבר בעצמנו.

נראה שמדובר במנגנון ה-anti-debugging של SecNeo, בחנו מעט את האפשרויות שעומדות בפנינו:

- להתחיל לבצע הנדסה לאחור לכל הקוד של SecNeo
- לנסות תחילה באגים לוגיים פשוטים לפני שנכניס את עצמנו לפרויקט די ארוך

התחלנו לחשוב על באגים לוגיים אפשריים ואז עלינו על זה! בחנו מי בעצם ב-TracerPid ושמנו לב ש-SecNeo בצעו את ה-debugging הפוך!

במקום קודם כל ליצור instance שלהם ואז ליצור ישר את האפליקציה תחת debugging הם קודם כל מריצים את האפליקציה האמיתית של DJI ורק אז מריצים את ה-instance שלהם ומתחברים אליה ב-debugging.



[מכיוון שדיווחנו במרץ והממצא תוקן התמונה הבאה היא משקפת את המצב אחר התיקון ולכן נראה שה-debugging תקין]

אם נסתכל על הקובץ /proc/1194/status נראה שהאפליקציה הזאת מחוברת ל-11239 ב-debugging. מכיוון ש-SecNeo בצעו את ה-debugging שלהם מתוך ה-instance השני הם יצרו race condition שמאפשר לתוקף להתחבר לאפליקציה לפניהם, ניסינו לנצח אותם ב-race וכל פעם שהצלחנו האפליקציה פשוט קרסה.

כנראה מכיוון שה-instance השני לא הצליח להתחבר ב-debugging אז זה גרם לשגיאה מסוימת. לאחר מספר ניסיונות הבנו שאין בכלל צורך לנצח את ה-instance השני במרוץ מכיוון שאין צורך במרוץ כלל!

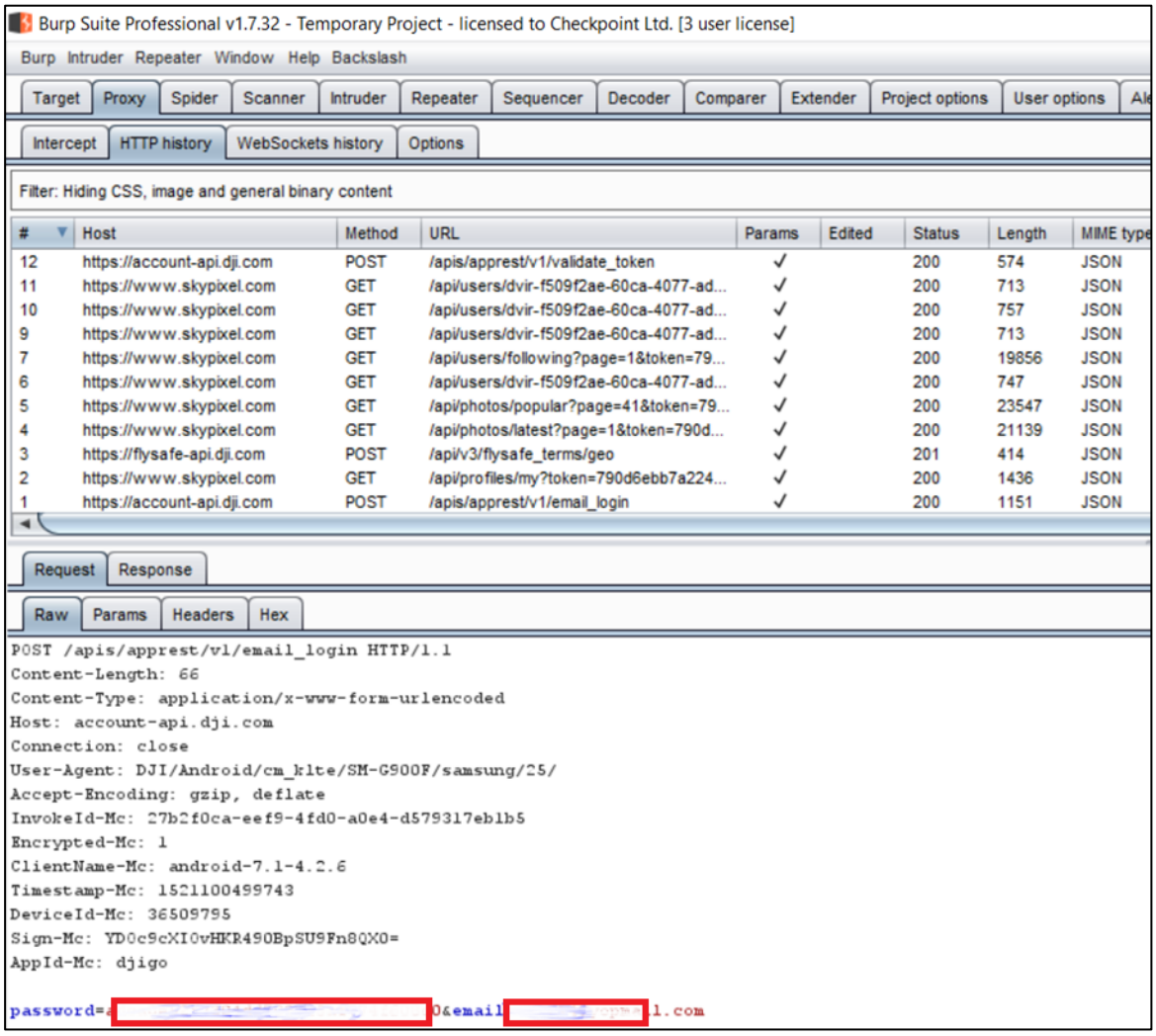
פשוט נפתח את האפליקציה במצב suspended וזהו, כך כתבנו את הקוד הבא:

```
import frida
import sys

jscode = """
    Java.perform(function (){
        console.log("working!");
    });
"""

process = frida.get_usb_device()
pid = process.spawn(["dji.go.v4"])

session = process.attach(pid)

script = session.create_script(jscode)
script.load()
process.resume(pid)

sys.stdin.read()
session.detach()
```

הרצנו את הקוד והאפליקציה עלתה בצורה תקינה! אף קיבלנו את ההדפסה working!.

עד כאן אנו מצליחים להזריק את הקוד שלנו לאפליקציה וכל מה שנאשר לנו לעשות זה פשוט עקפנו את ה-SSL pinning באמצעות frida.

לאחר שעקפנו את ה-SSL Pinning הצלחנו לראות את כל התעבורה שיוצאת מהאפליקציה:



לאחר שהזנו שם משתמש וסיסמא ראינו שהתשובה לפנייה שלנו מכילה את ה-meta_key שאותו אנחנו מכירים ועוד token כלשהו שלא ראינו קודם לכן:

```
{code:0,message:ok,data:{nick_name:XXXXXXXXX,cookie_name:_meta_key,cookie_key:NTUxNjM2ZTXXXXXXXXXXXXXXXNmI2NjhmYTQ5NGMz,active:false,email:XXXXXXXX@gmail.com,token:XXXXXXXXX2139,validity:15275XXXXX,user_id:9629807625XXXXXX,register_phone:,area_code:,inner_email:false,subscription:false,vipLevel:null,vipInfos:[]}}
```

הרצנו חיפוש באמצעות burp suite על כל החבילות שנשלחו בין האפליקציה לצד השרת על מנת שנוכל לאתר את כל הפניות שאנו מקבלים בהם את ה-token הזה כתשובה לבקשה.

כך מצאנו את הפנייה mobile.php אשר מאפשרת לנו באמצעות meta_key_ לקבל token, ואת ה-meta_key_ אנו יכולים להשיג בפורום:

```
GET /source/plugin/mobile/mobile.php?module=forumdisplay&submodule=checkpost HTTP/1.1

Host: forum.dji.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0

Accept: */*

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: https://forum.dji.com/home.php?mod=spacecp&ac=avatar

X-NewRelic-ID: XXXlJXGwIGXXXFVSBAQP

X-Requested-With: XMLHttpRequest

Cookie: www_lang=en; lang=en; country=US; currency=USD; _ga=;
traceid=7c5XXXX8dXXXd4f8d7XXXX1d; djiid=93f3XXXX-fXXX-0XXX-XXXX-1XXXXXXXdda;
user_data_reload=false; cart_data_qty=0; www_user_nick_name=XXXX;
_djissid=XXXXXXXXXX35da9d1238d5599630; _gid=; www_request_id=4e075903XXXX007-8e04-
dXXXXXXe381XXX2013580; www_from=pc; browsersid=f832f8de61cXXXXXfef2fXX7dXXb641;
_logged=yes; _meta_key=MTIyNmM2NGYtYzQXXX00XXXXXXXXXXXXXXXXXXXXXXXXXXX;
cart_uuid=7cX0XXXX135da9XXXX5599630; user_nick_name=XXXX;
user_avatar_url=%XXXXXXXXX.djicdn.XXXXXXXXXXXXXXXX-03c3-4ab5-XXXXXXXXXX.jpg; _dji-session=
```

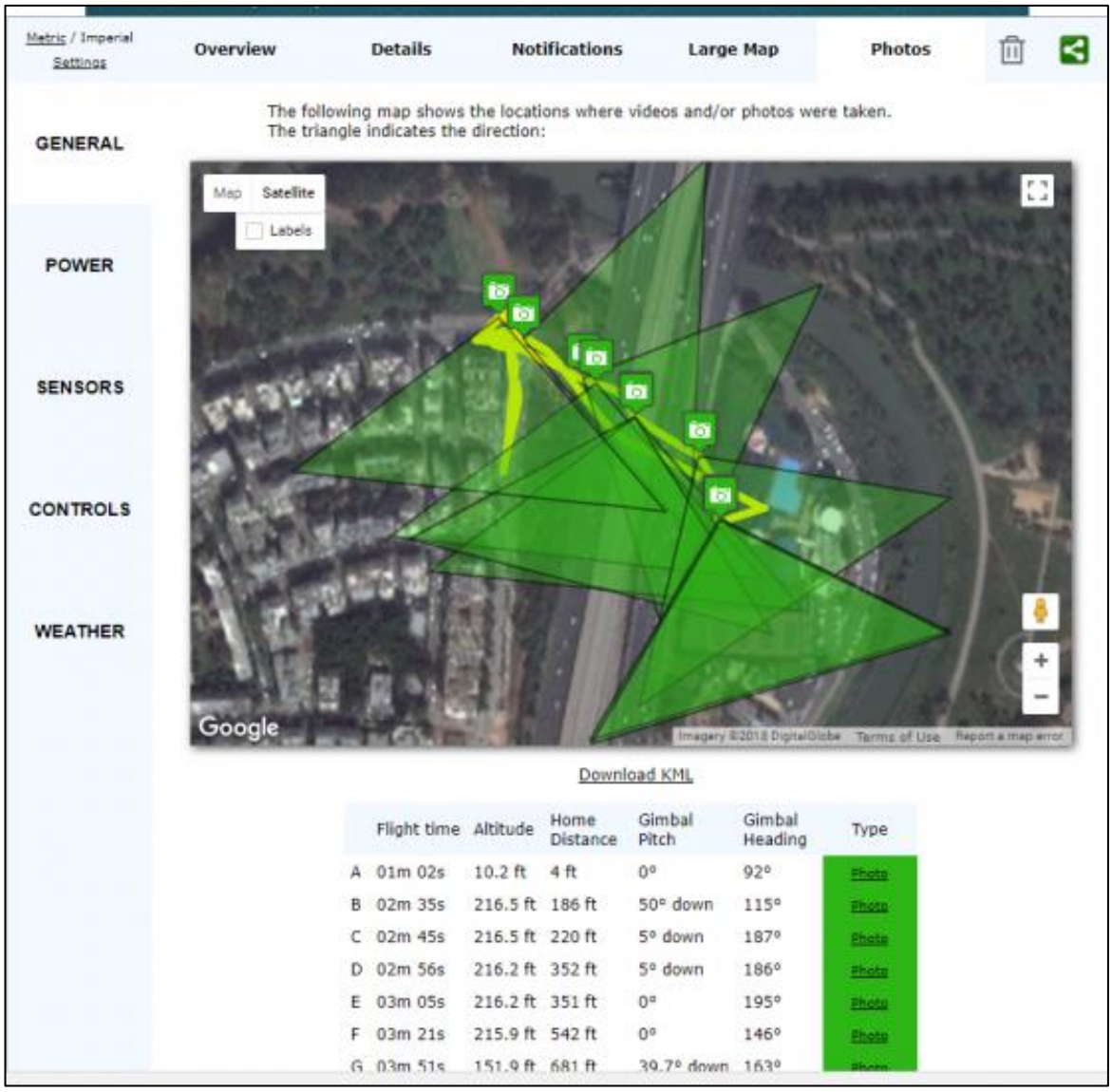
mobile.php חבילה הדומה ל-initData.do אשר באמצעות mck מייצרת לנו ticket. כך שכדי לפרוץ לחשבון מובייל עלינו לבצע את השלבים הבאים:

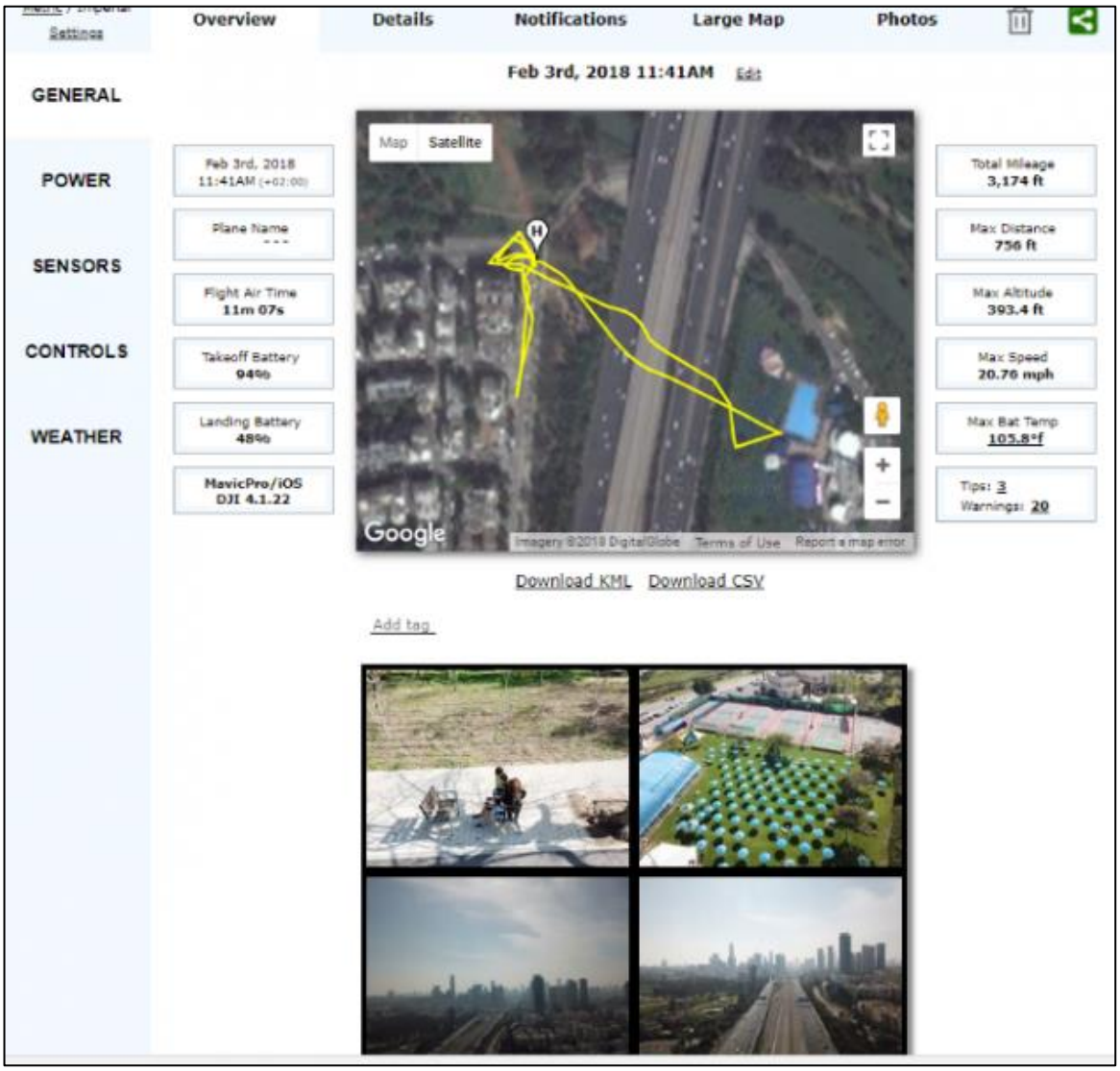
1. להשתמש ב-frida כדי לעקוף את ה-SSL Pinning.
2. להשיג meta_key של המטרה שלנו ולשלוח בקשה ל-mobile.php כדי לקבל גם את ה-token שלו.
3. להתחבר עם החשבון שלנו ולשנות את ה-token וה-meta_key שחוזר בבקשת ה-email_login.

פריצה לחשבון המובייל מאפשר לתוקף לקבל את הנתונים הבאים:

- צפייה בכל מסלולי הטיסה.
- צפייה בכל התמונות שצולמו עם הרחפן.
- צפייה בזוויות הצילום של הרחפן.

דוגמאות:




Overview
Details
Notifications
Large Map
Photos
Feb 3rd, 2018 11:41AM
Edit
GENERAL
POWER
SENSORS
CONTROLS
WEATHER
Feb 3rd, 2018 11:41AM
Plane Name
Flight Air Time 11m 07s
Takeoff Battery 94%
Landing Battery 48%
MavicPro/iOS DJI 4.1.22
Map
Satellite
Google
Total Mileage 3,174 ft
Max Distance 756 ft
Max Altitude 393.4 ft
Max Speed 20.76 mph
Max Bat Temp 105.8°f
Tips: 3
Warnings: 20
Download KML
Download CSV
Add tag

השתלטות על רחפנים תעשייתים (DJI FlightHub)

DJI FlightHub זו פלטפורמה למגזר העסקי אשר מאפשרת לנהל צי שלם של רחפנים מכל מקום בעולם. באמצעות הפלטפורמה ניתן לצפות ולשמוע את הרחפנים בזמן אמת, לקבל נקודות ציון ותמונה על גבי מפה.

תעשיות רבות בעולם משתמשות במערכת זו לצורך ניהול צי הרחפנים שלהם, למידע נוסף מוזמנים לצפות בסרטון התדמית של DJI לגבי מערכת ה-DJI FlightHub:

https://www.youtube.com/watch?v=D7W2DQshmxY

במערכת ה-DJI FlightHub קיימת רמת הרשאות אשר מנוהלת בצורה הבאה:

- לכל ארגון יש admin שאחראי על פלטפורמת FlightHub. יש לו גישה לכל המידע שנשמר בפלטרפומה, כגון רישומי טיסות, צפייה של וידאו ושמיעת סאונד בזמן אמת, ויכולת לנהל את כל הארגון.
- לכל צוות יש captain שגם לו יש הרשאה להיכנס לפלטרופמת ה-FlightHub והוא יכול נהל את כל הטייסים ולצפות ברחפנים שלהם.
- בנוסף יש רמת הרשאה pilot שזה בעצם טייס שיכול לנהל רק את הרחפן של עצמו.

כך שאם נצליח לתקוף admin או captain נוכל לצפות ברחפנים ולשמוע סאונד בזמן אמת של כל צי הרחפנים הצוותי או של כל הארגון.

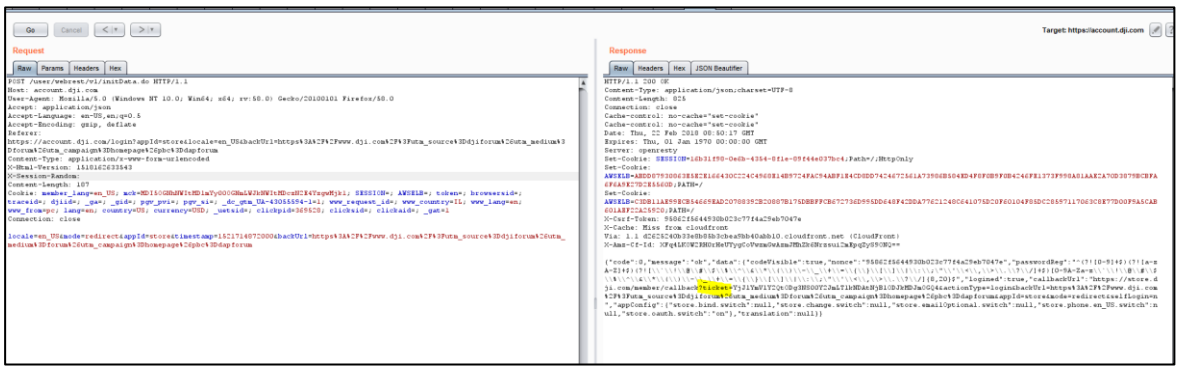
בחנו את תהליך ההזדהות של המערכת והופתענו לגלות שמדובר באותה הלוגיקה כמו בכל שאר מערכות ה-DJI. העוגייה החשובה כאן היא mck או meta_key אשר בעזרתה אנו יכולים להשיג את כל שאר המידע שאנו צריכים כדי להתחבר לחשבון של המשתמש.

במידה והצלחנו להשיג את העוגייה דרך הפורום, עלינו להשיג גם כאן את ה-ticket לחשבון.

ניסינו לבצע בדיוק את אותם השלבים כפי שביצענו בהתחברות ל-forum או ל-store.

תפסנו את בקשת initData.do ושינינו את עוגיית mck ל-mck של המשתמש שאנו רוצים לתקוף, אך מסיבה כלשהי, הפעם לא קיבלנו ticket בתשובה לבקשה.

הסתכלנו שוב על בקשת ה-initData.do:



אם תשימו לב, תראו שבמידע שאנו מעבירים בגוף ההודעה קיים פרמטר הנקרא appId, בבקשה המקורית התוכן שלו הינו Flighthubprod2.

ברגע שאנו מחליפים את ה-mck ומשאירים את ה-appId צד השרת לא מחזיר לנו ticket, אך אם נשנה את ה-appId ל-store או כל דבר אחר תחת dji.com נקבל ticket שבמפתיע עובד גם עבור DJI FlightHub.

אז כל מה שהתוקף צריך לעשות על מנת לפרוץ לחשבון משתמש הינו:

1. לשלוח בקשת initData.do עם ה-appId=store ועוגיית mck של המטרה אותה הוא רוצה לפרוץ, בתגובה התוקף יקבל ticket שהוא ישתמש בו בשלב הבא.
2. לאחר מכן, על התוקף להתחבר לחשבון שלו ב-DJI FlightHub ובעת הכניסה לשנות את ה-mck ל-mck של המטרה ובתגובה שחוזרת מצד השרת לשנות את ticket ל-ticket שקיבלנו בסעיף 1.
3. התוצאה היא מעבר לחשבון של המטרה אותה רצינו לתקוף.

חשוב לציין שהמטרה לא תקבל התרעה על הכניסה לחשבון ולתוקף תיהיה גישה לכל הנתונים בחשבון, כגון צפייה בשידור חי במצלמת הרחפנים, שמע, מידע על הרחפן עצמו כגון מיקום ועוד.

סיכום

שימוש ב-SSO בין מערכות רבות מצד אחד נוח וקל לניהול, אך מצד שני ברגע שתוקף יצליח להשיג פיסת מידע קריטית הוא יצליח להתחבר לכל שאר המערכות של הארגון.