



CaptainCredz

Password Spraying in 2025

2025/07/02



Password spraying

What?

- **Password spraying:** few passwords, many users → *today's topic*
- ~~**Brute-force:**~~ few users, many passwords → *out of scope for today*



Password spraying

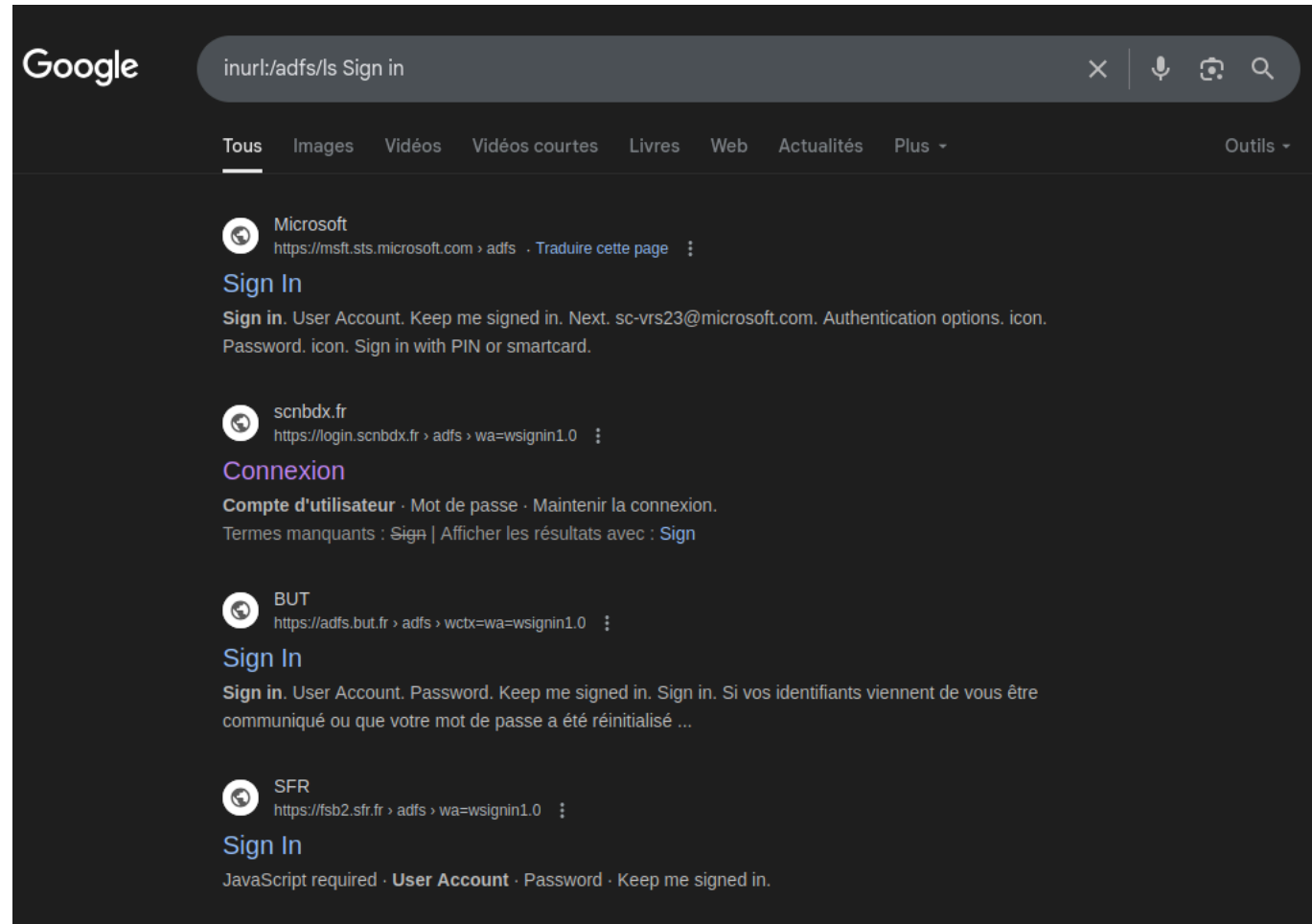
Why?

- **People are lazy!** On a >10k employees company, `<Company>2025!` might work
- **Companies migrate to cloud** & encourage remote work
- → often there is a company **authentication portal on the internet**
 - Microsoft SSO
 - Okta
 - Citrix
 - ADFS
 - Jira
 - Keycloak
 - ...



Password spraying

Why?



Password spraying

How?

- **Crawl the web:** Look for email addresses, employees names, ...
 - LinkedIn
 - RocketReach
 - Data breaches
 - ...
- **Create a small list of probable passwords**
 - `<Company>2025`
 - `<Company>2025!`
 - `Password123`
 - `<OfficeLocation>2025`
- **Start spraying!**



Flagged!





Flagged!

- **Why?**
 - Many auth failure
 - From the **same IP address**
 - In a **short time frame**
- **So what?**
 - Auth providers **start lying** (auth fails even with the correct password)
 - A success will be flagged as **suspicious**
 - **MFA** will probably be prompted on success
 - An **alert** will be raised and the **account locked** by the blue team



CaptainCredz

...to the rescue!

 synacktiv / captaincredz

114 ★

CaptainCredz is a modular and discreet password-spraying tool.



synacktiv

<https://github.com/synacktiv/captaincredz>



CaptainCredz

Spraying engine

- Fork from <https://github.com/knavesec/CredMaster> (for various reasons)
- Smart timings
 - More attempts on mornings when users log in legitimately
 - Slowly ramp up the cadence over the week
 - Be dead silent on week-ends
 - ...
- Can combine specific `user : password` lists and generic passwords
- Has a cache
- *Has a cool progress bar* 🕶️



- **Relies on *plugins*** (code specific to the auth portal)
- **Anyone can implement a new *plugin*** in 3 functions: `validate()`, `testconnect()`, `test_authenticate()`
- **Can trigger specific code (called *post-actions*)** depending on the attempt's result
 - Display cookies
 - Send a Slack notification
 - ...



CaptainCredz

Example run

```
{
  "plugins" : [{
    "name": "adfs",
    "args": { "url": "http://adfs.company.com/" },
    "proxy": null,
    "useragentfile": "useragents.lst",
    "req_timeout": 60
  }],
  "post_actions": {
    "display_cookies": { "trigger":["success"]}
  },
  "userfile" : "users.lst",
  "passwordfile" : "passwords.lst",
  "userpassfile" : "userpass.lst",
  "jitter" : 600,
  "delay_req" : 25,
  "delay_user" : 7200,
}
```



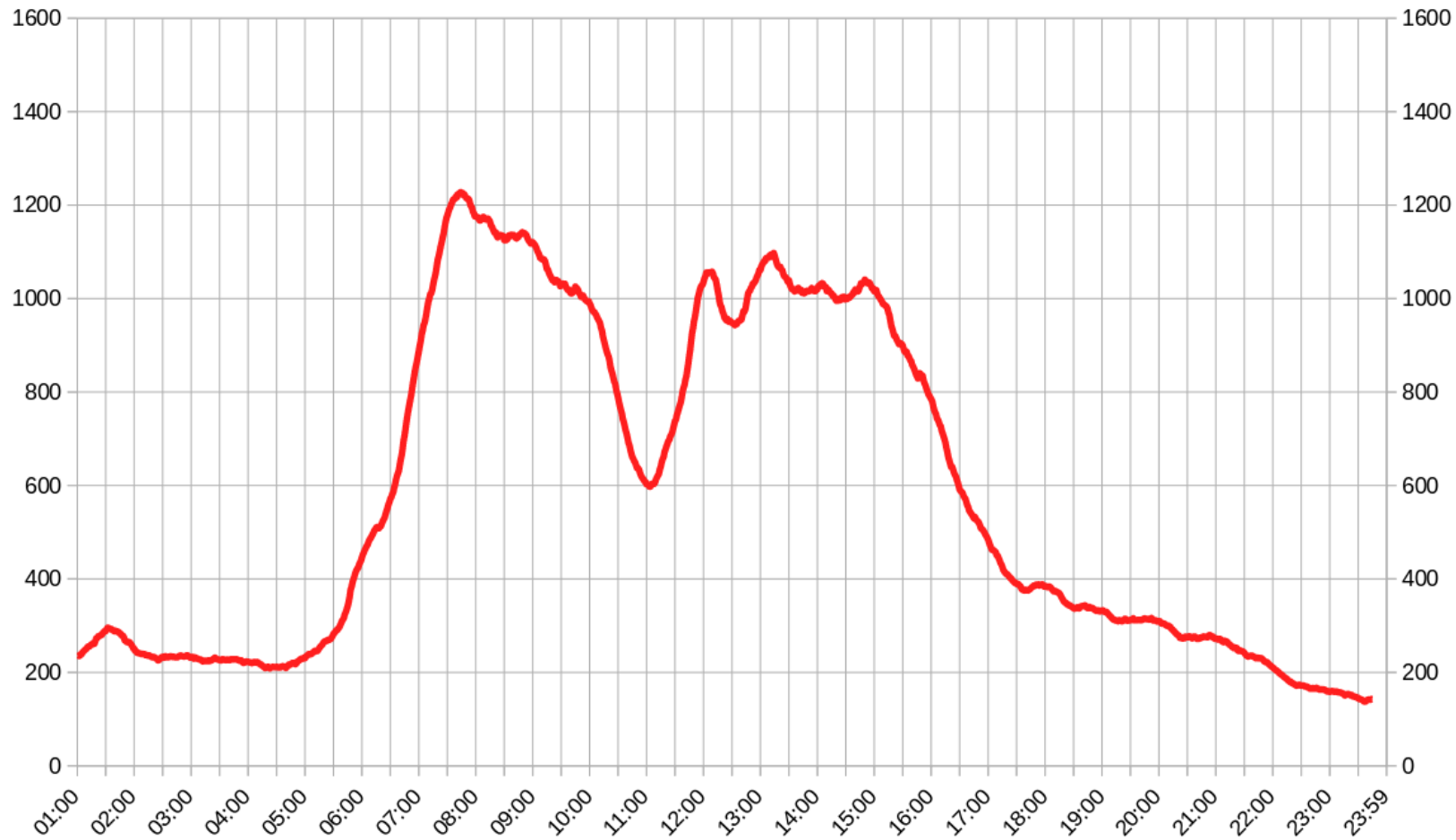
```
$ python3 captaincredz.py --config config.json
```

- **Delays depend on your target's size, habits and timezone**
 - Be patient
 - Try not to deviate *too much* from usual activity
 - How much is *too much*? Currently analyzing many Azure sign-in logs, stay tuned
- **Have a look at [synacktiv/IPspinner](#) to evade IP-based detection**
- **Was it worth the struggle? Yes!**
- **Multiple successes in redteam engagements** offering an initial access :)



CaptainCredz

Sign-in tendencies throughout a day



How to defend?

- **Use strong passwords**
 - Ban common words
 - Encourage the use of password managers
- **Or just don't use them!**
 - Yubikeys, mTLS, Passkeys, ...
- **Enforce multi-factor authentication**
- **Monitor and correlate authentication logs**





<https://www.linkedin.com/company/synacktiv>



<https://x.com/synacktiv>



<https://bsky.app/profile/synacktiv.com>



<https://synacktiv.com>