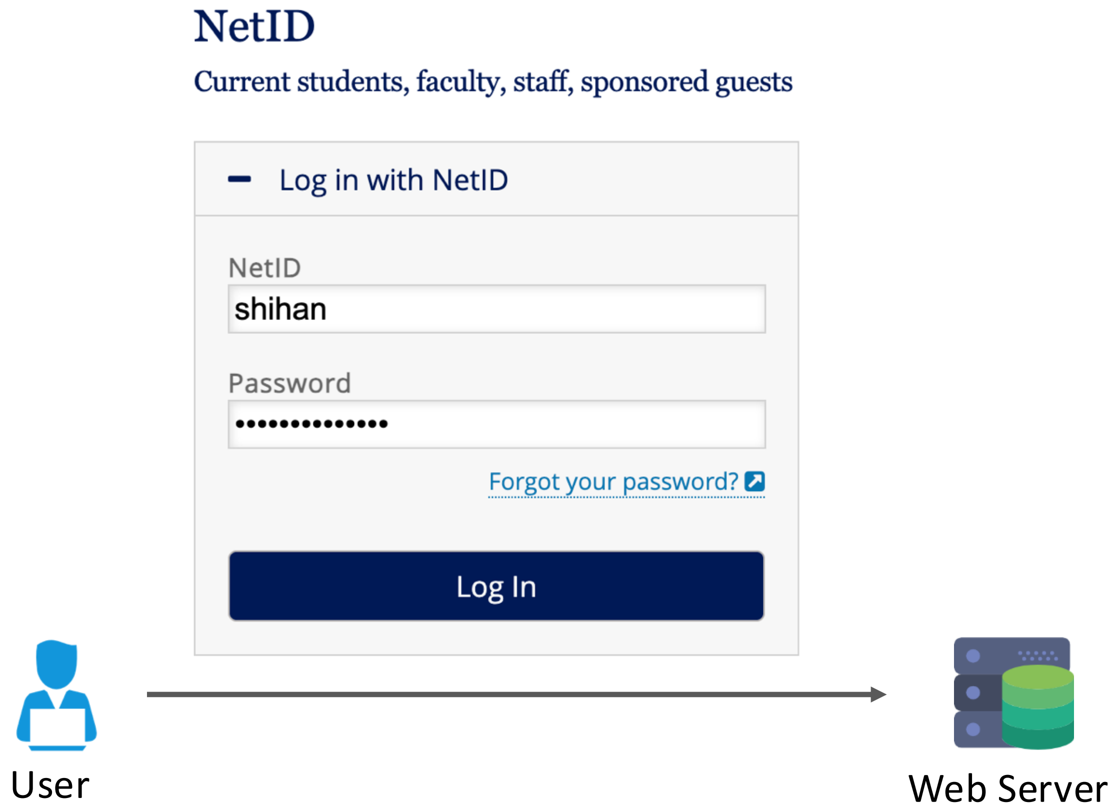
PreAcher: Secure and Practical Password Pre-Authentication by Content Delivery Networks

Shihan Lin[1], Suting Chen[2], Yunming Xiao[3], Yanqi Gu[4],
Aleksandar Kuzmanovic[2], Xiaowei Yang[1]

[1]Duke University, [2]Northwestern University, [3]University of Michigan, [4]UCI

Motivation

- Password login is prevalent on the Internet



NetID

Current students, faculty, staff, sponsored guests

− Log in with NetID

NetID: shihan

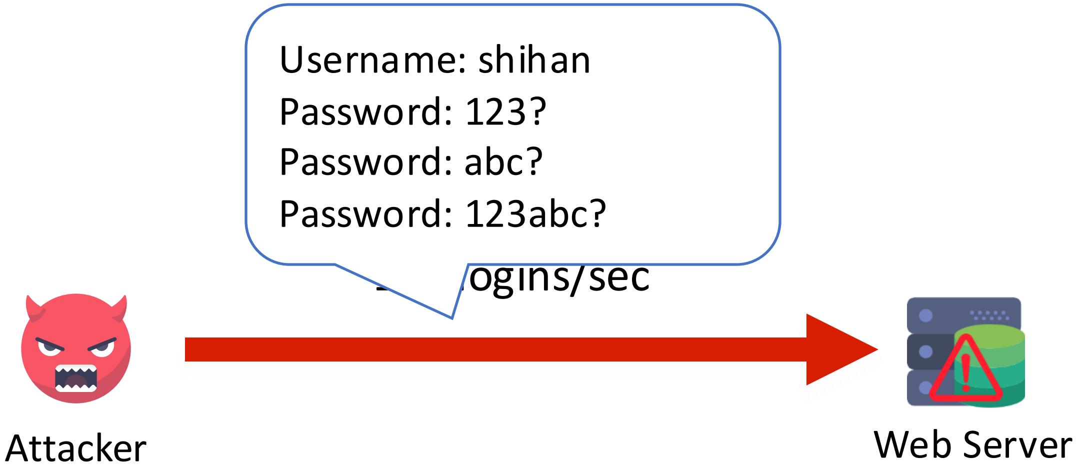
Password: ••••••••••••••

Forgot your password?

Log In

Password Login is Vulnerable to ADoS Attacks

- Application-layer Denial of Service (ADoS)
  - Exhaust a server's CPU by a sending *small* amount of login traffic
  - DDoS usually takes a *large* amount of traffic
- ADoS happens under credential stuffing attacks
  - Attackers repeatedly try many passwords to crack user accounts
  - Akamai reports 280 million suspicious logins per day*

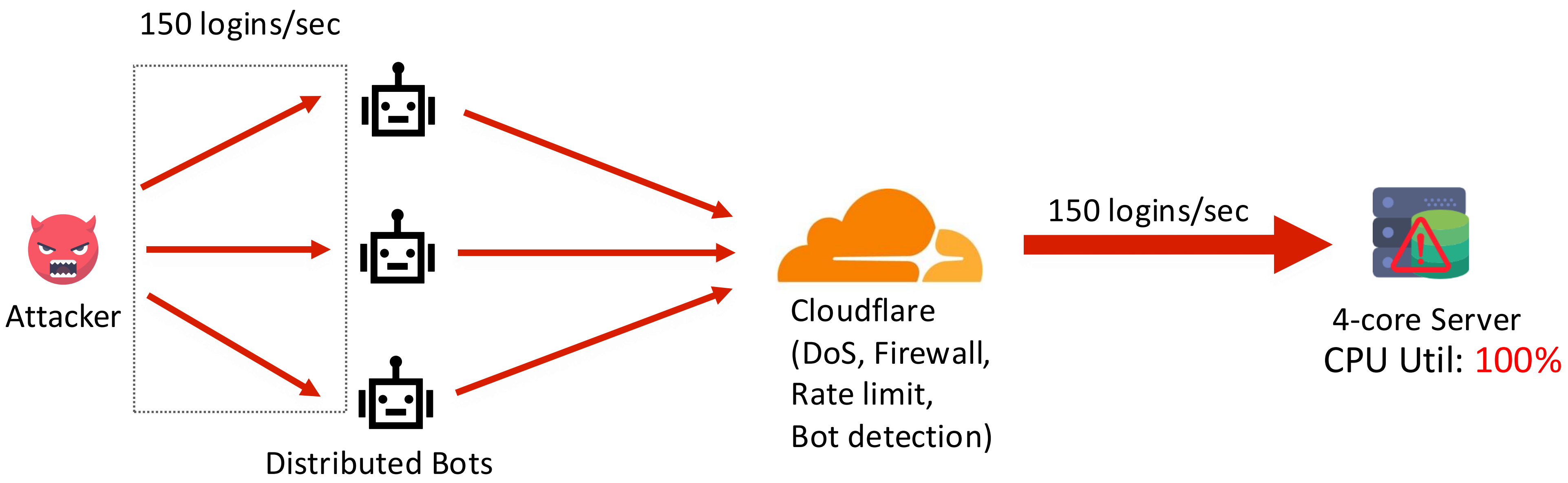


Username: shihan
Password: 123?
Password: abc?
Password: 123abc?

*https://www.akamai.com/glossary/what-is-credential-stuffing

ADoS Proof of Concept

- Bypass existing defense provided by Cloudflare

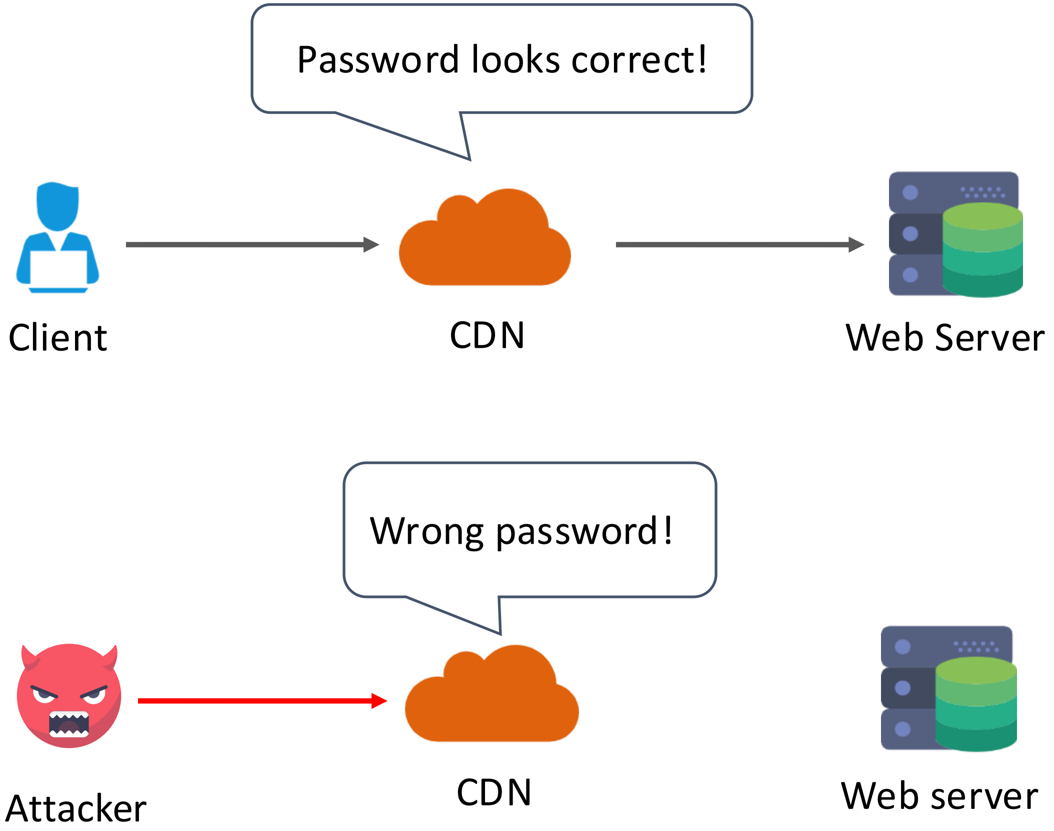


Existing Solutions & Limitations

Solutions	DoS Prevention	Password Secrecy/ Account Security	Usability
CAPTCHA	◑	●	○
Two-factor authentication (2FA)	○	●	○
Rate limit	○	●	●
Bot detection	◑	◑	●
Single Sign-On (SSO)	●	◑	●

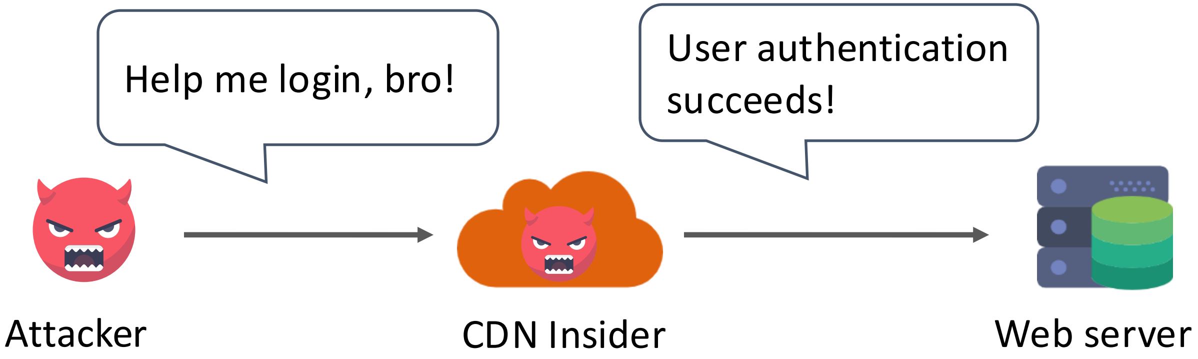
Key Insight of PreAcher

- **Pre-Authentication on a CDN without exposing passwords to the CDN**
  - filter out invalid logins without correct passwords

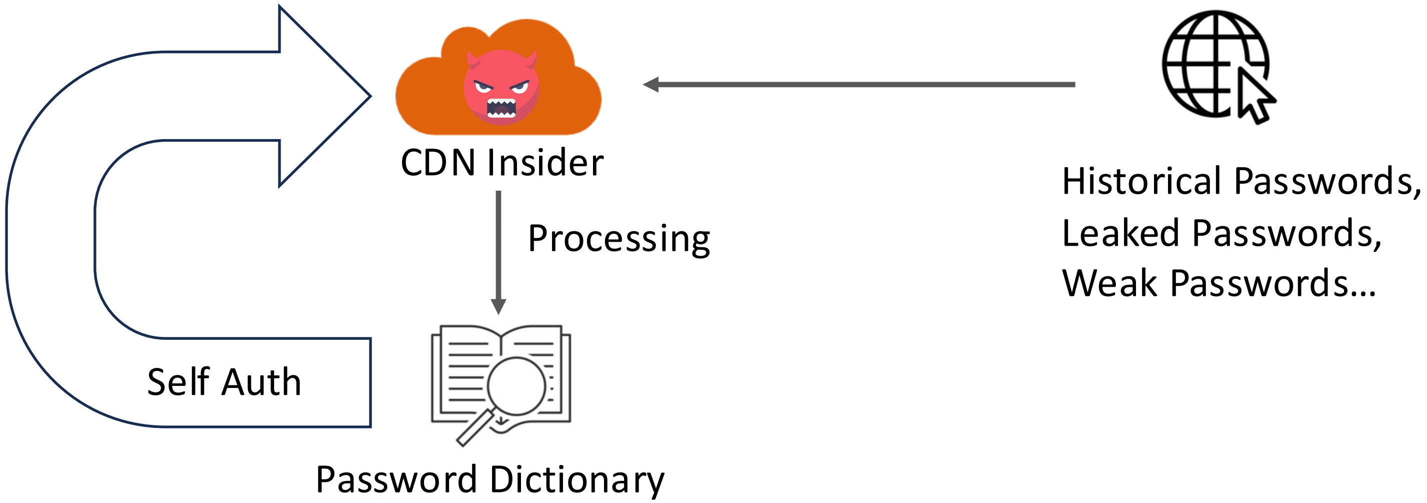


Unfortunately…

- Attackers inside CDNs may impersonate users

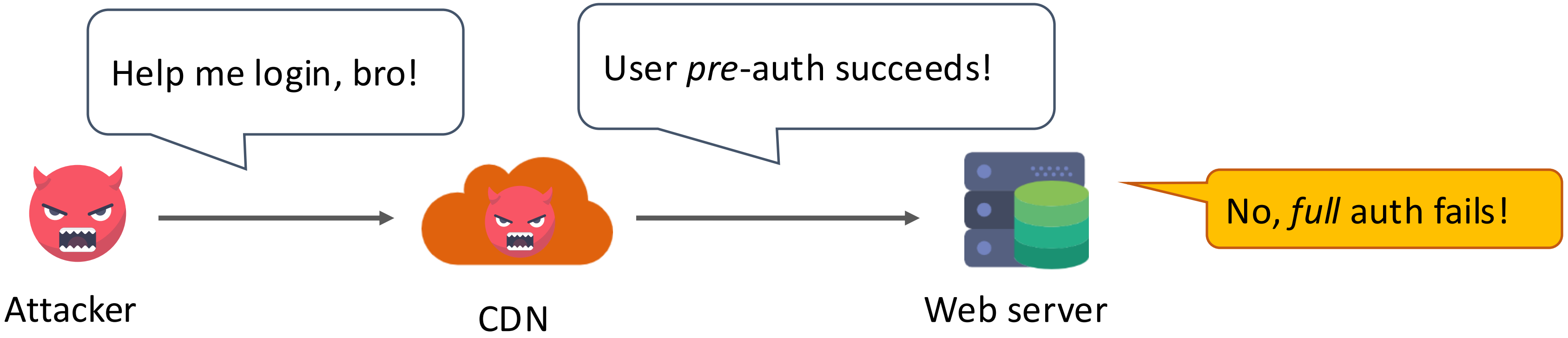


- Offline Dictionary Attacks (ODAs) by attackers inside a CDN



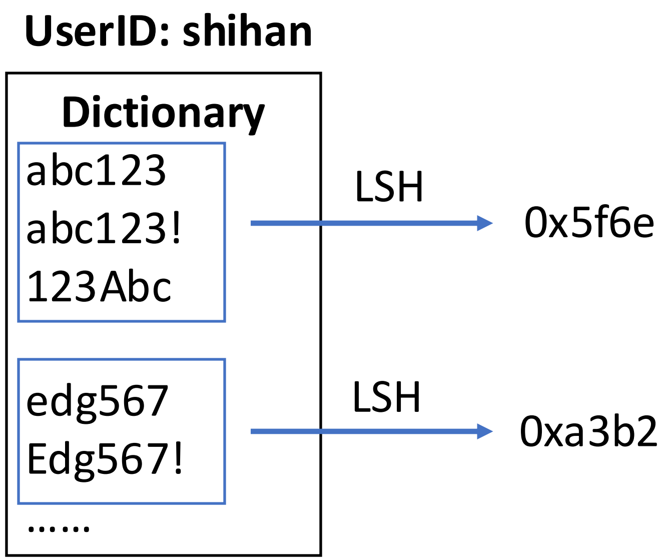
Solution: Pre-Auth + Full Auth

- Attackers inside CDNs may impersonate users
  - Involve the server to double check the authentication
  - ***Pre***-authentication on the CDN
  - ***Full*** authentication on the server



Solution: Locality-Sensitive Hashing (LSH)

- Offline Dictionary Attacks (ODAs) by attackers inside a CDN
  - *Observation*: Passwords in a dictionary share certain similarities
  - *Idea*: Group similar passwords into one *pseudo*-password for ***Pre-Auth***
    - Locality-Sensitive Hashing (LSH)



UserID: shihan

Dictionary

abc123
abc123!
123Abc

LSH

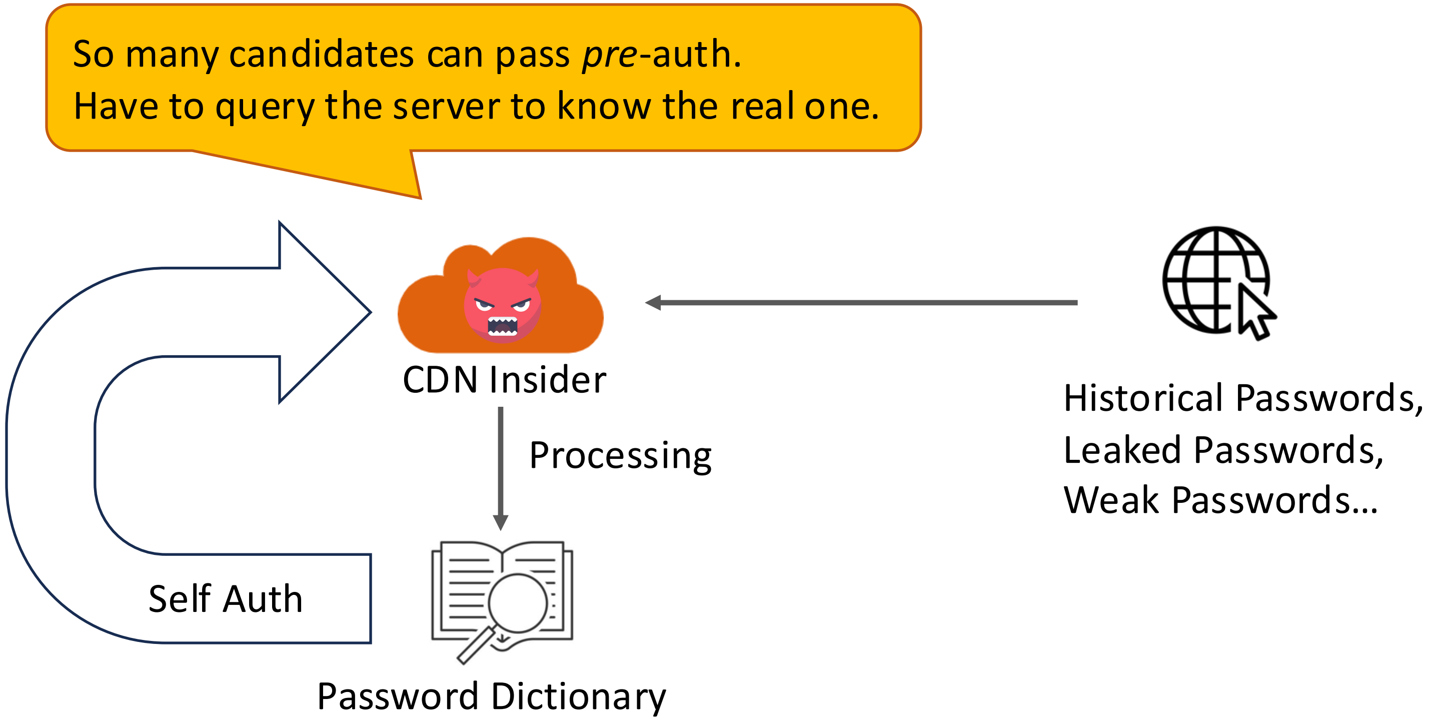
0x5f6e

edg567
Edg567!

……

LSH

0xa3b2



So many candidates can pass *pre*-auth.
Have to query the server to know the real one.

CDN Insider

Historical Passwords,
Leaked Passwords,
Weak Passwords…

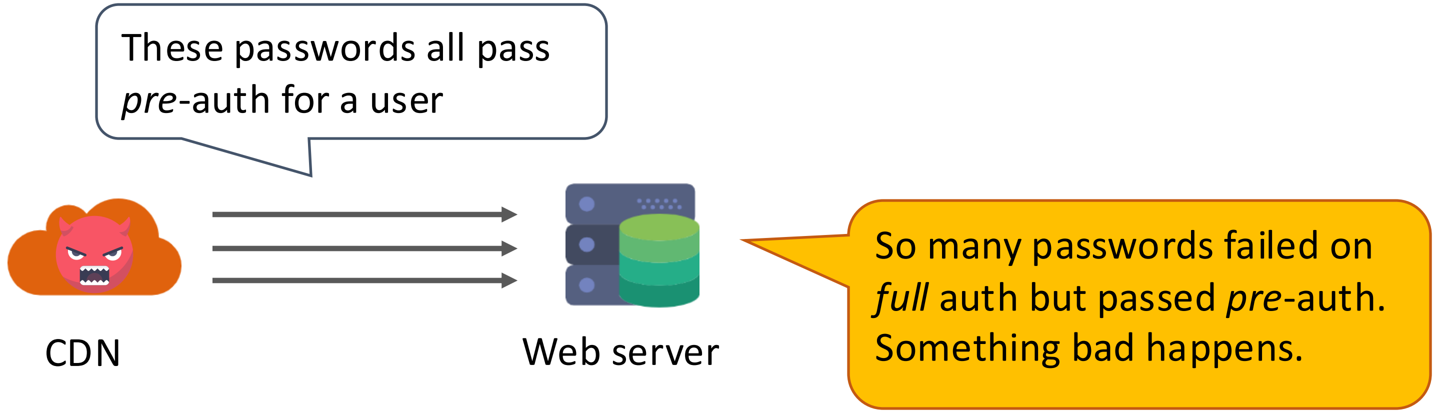
Processing

Self Auth

Password Dictionary

What if the CDN tries many online queries?

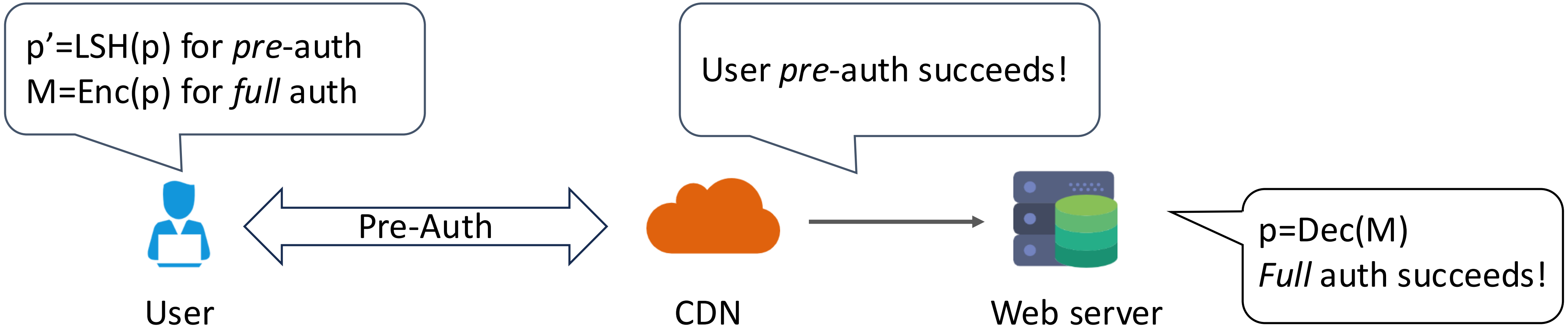
- Detect the attacks when the server receives >Q failed full authentication queries



These passwords all pass *pre*-auth for a user

So many passwords failed on *full* auth but passed *pre*-auth. Something bad happens.

Put All Together: PreAcher



Implementation & Deployment

- Server operations
  - C++ library
  - Implemented by web developers
- Client operations
  - JavaScript library
  - Imported by web developers into webpages
- CDN operations
  - Serverless computing service on CDNs
  - JavaScript code snippet
  - Deployed by web developers

**Web developers can unilaterally deploy PreAcher
to protect their websites!**

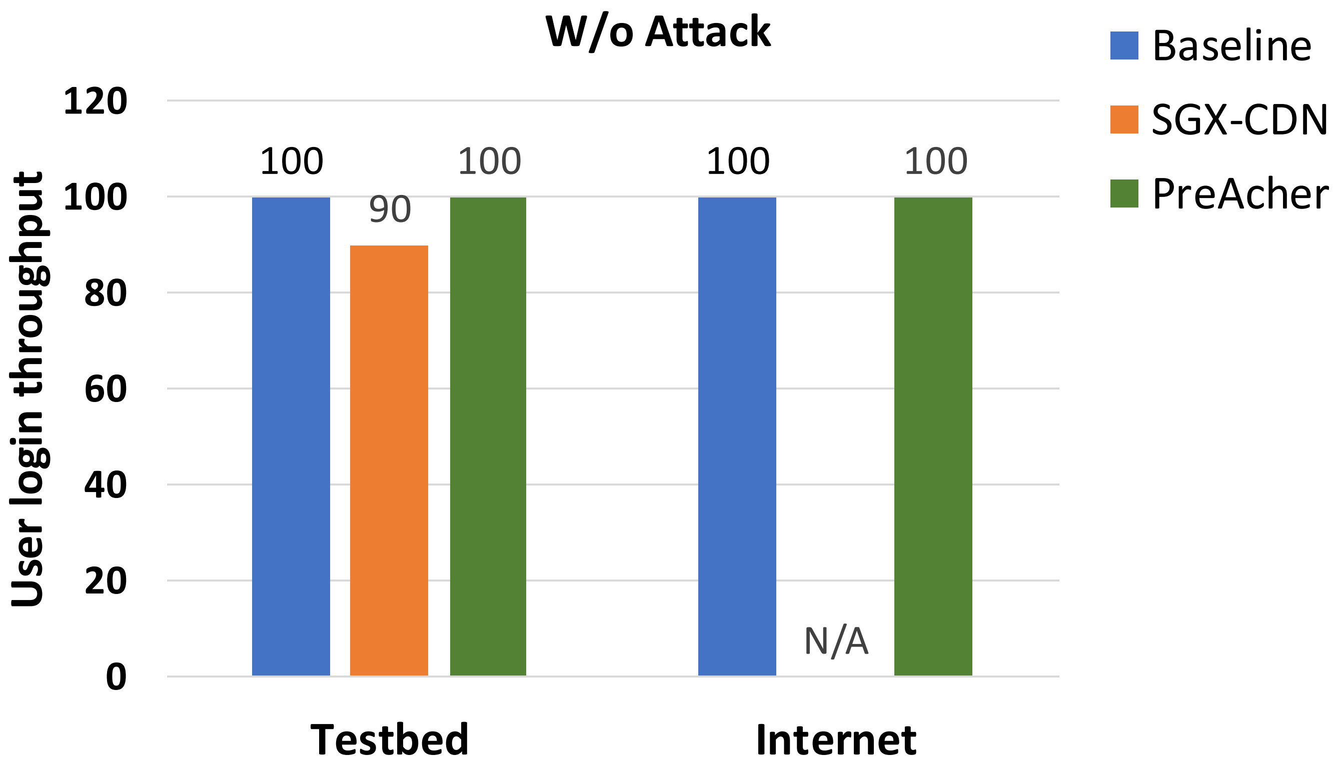
Evaluation

- Testbed experiment
  - Use Azure VMs as the client, server, CDN
- Internet experiment
  - Use Azure VMs as the client and server
  - Use Cloudflare as the CDN
- Two strawman solutions
  - Baseline
    - A CDN simply forwards every login request to the server
  - SGX-CDN
    - Use SGX on a CDN to fully authenticate users to filter out invalid logins

Evaluation: ADoS Defense

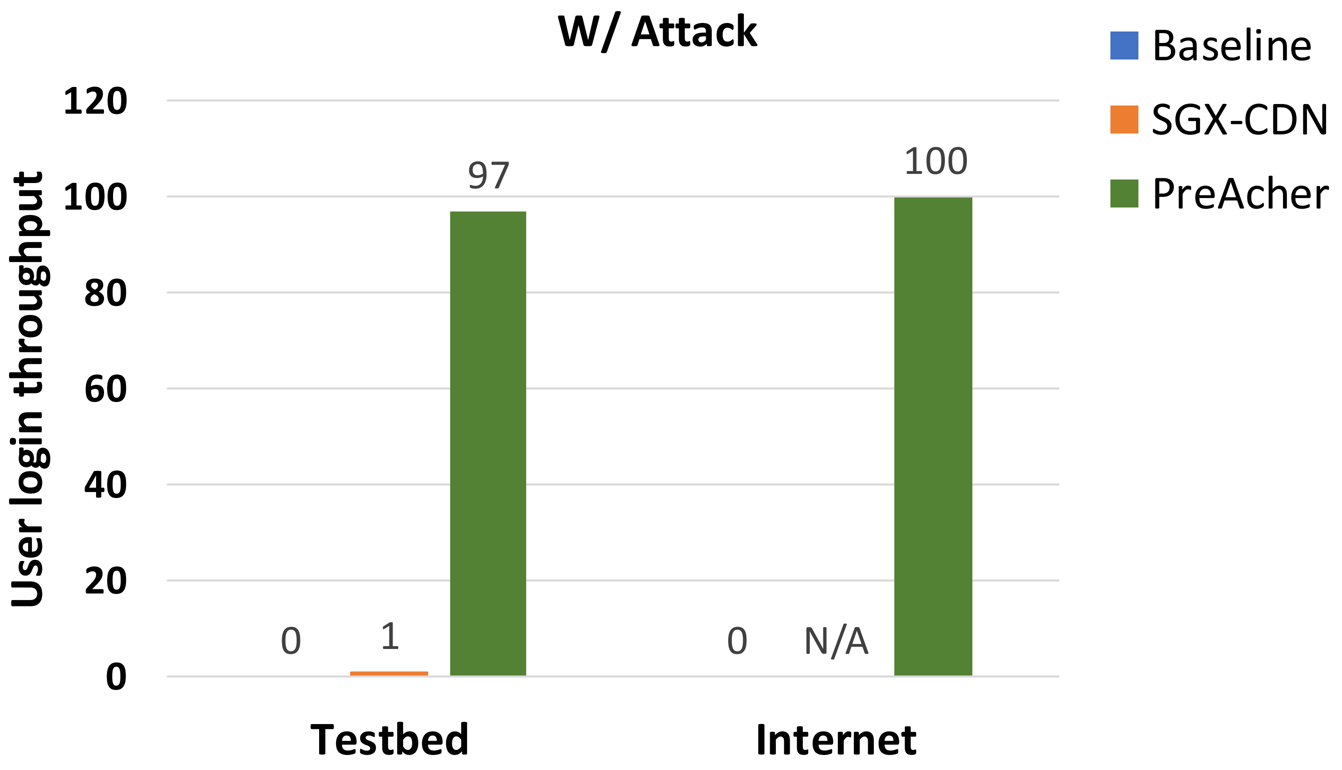
- User traffic: 100 valid logins/sec

- Measure the throughput of valid logins



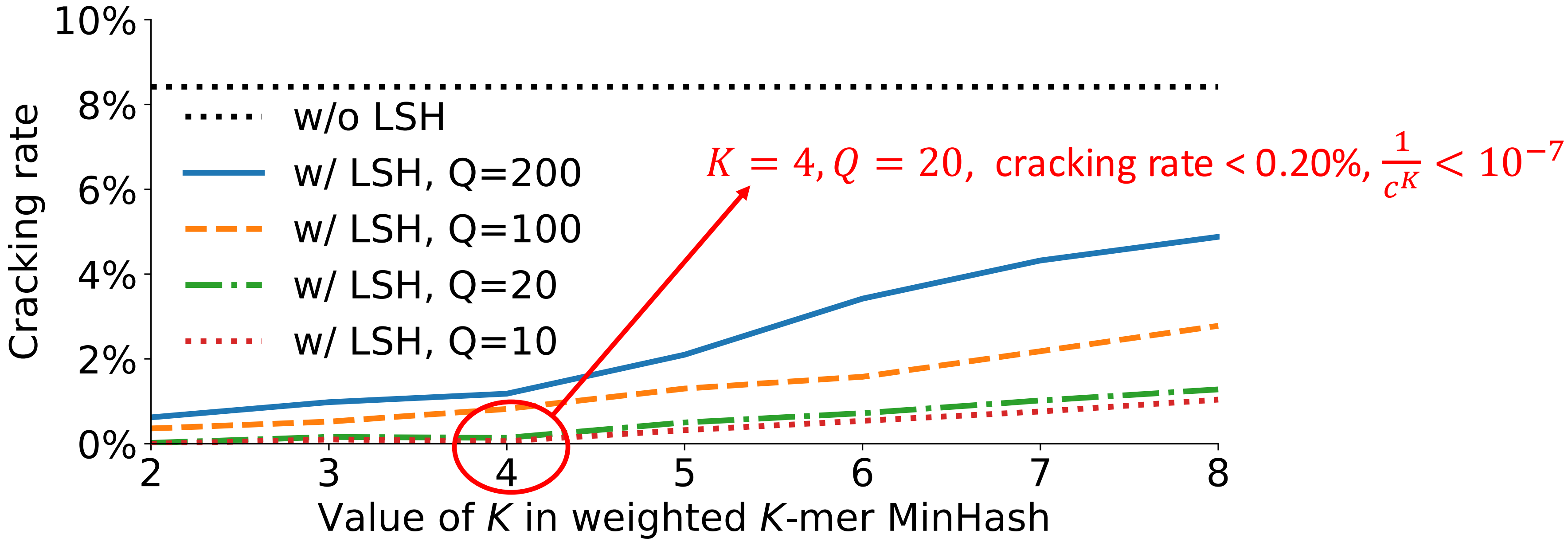
Evaluation: ADoS Defense

- User traffic: 100 valid logins/sec
- Attack traffic: 400 invalid logins/sec
- Measure the throughput of valid logins



Evaluation: LSH Efficacy

- Simulate an attacker inside a CDN to crack 5000 user accounts
  - Use *pass2path* (S&P’19) algorithm and 4iQ dataset to generate password dictionaries
  - Compute the *cracking rate* of user accounts when PreAcher is deployed
- K: The parameter of LSH algorithm (Weighted K-mer MinHash)
- Q: The failure number after which will the server report an ODA
- Trade off between ADoS defense and ODA prevention
  - The percentage of Invalid logins that pass pre-auth: $\frac{1}{c^K}$, c is the alphabet size



$K = 4, Q = 20$, cracking rate < 0.20%, $\frac{1}{c^K} < 10^{-7}$

Conclusion: PreAcher

- Securely pre-authenticate users on CDNs
  - Filter out malicious login requests without correct passwords
  - CDNs cannot access or guess user passwords
- Immediate deployable on the Internet by web server unilaterally
  - JavaScript for client operations
  - Serverless computing for CDN operations
- Code: https://github.com/SHiftLin/NSDI2025-PreAcher

Thank you!

Q&A

NSDI2025-PreAcher