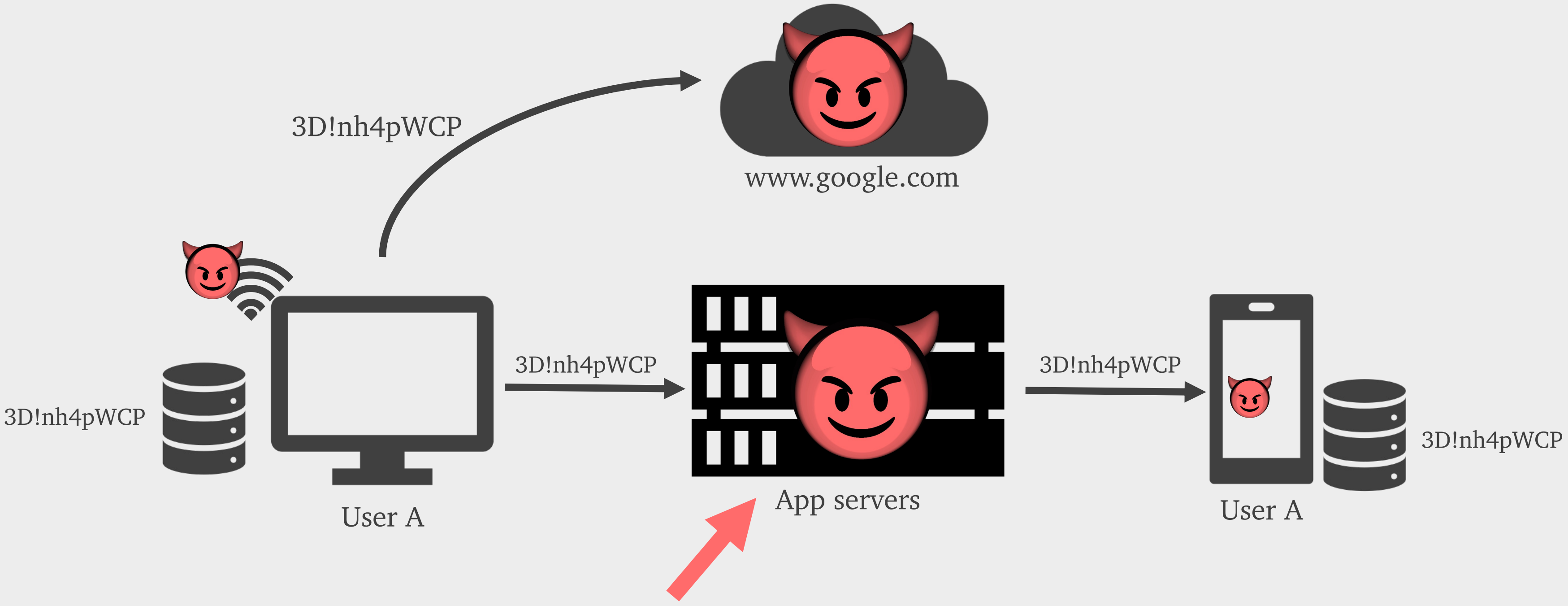
Exploiting Leakage in Password Managers via Injection Attacks

Andrés Fábrega, Armin Navamari, Rachit Agarwal, Ben Nassi, Thomas Ristenpart

Cornell University | CORNELL TECH

PASSWORD MANAGERS



Cannot read or manipulate user data

New directions in password managers:

new advanced features

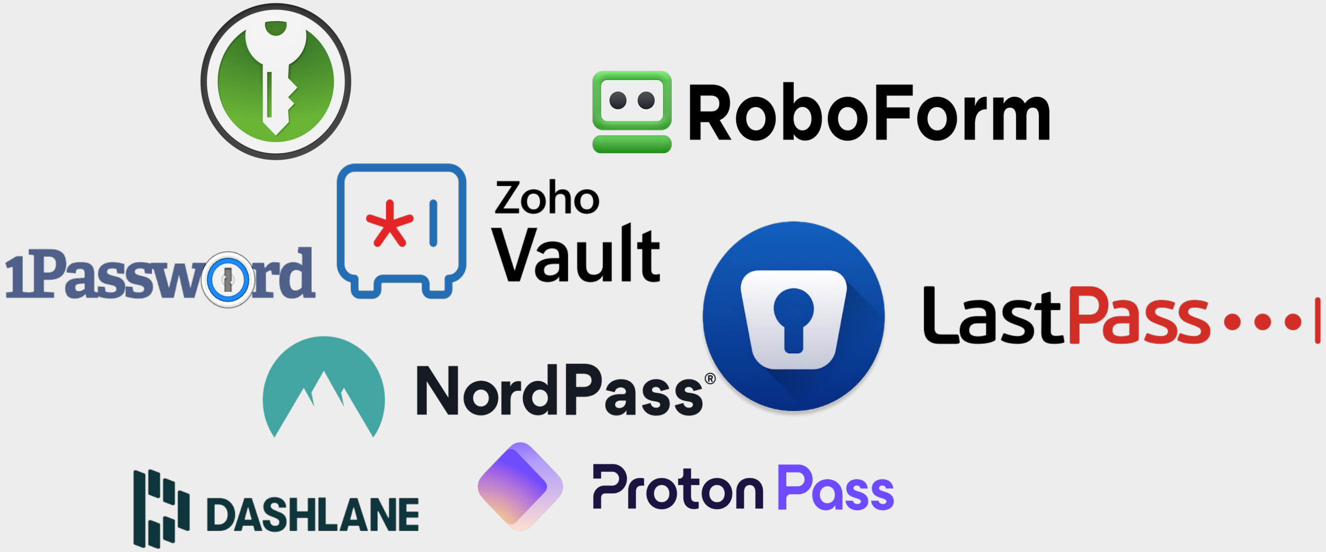
increasing app complexity

THIS WORK: INJECTION ATTACKS

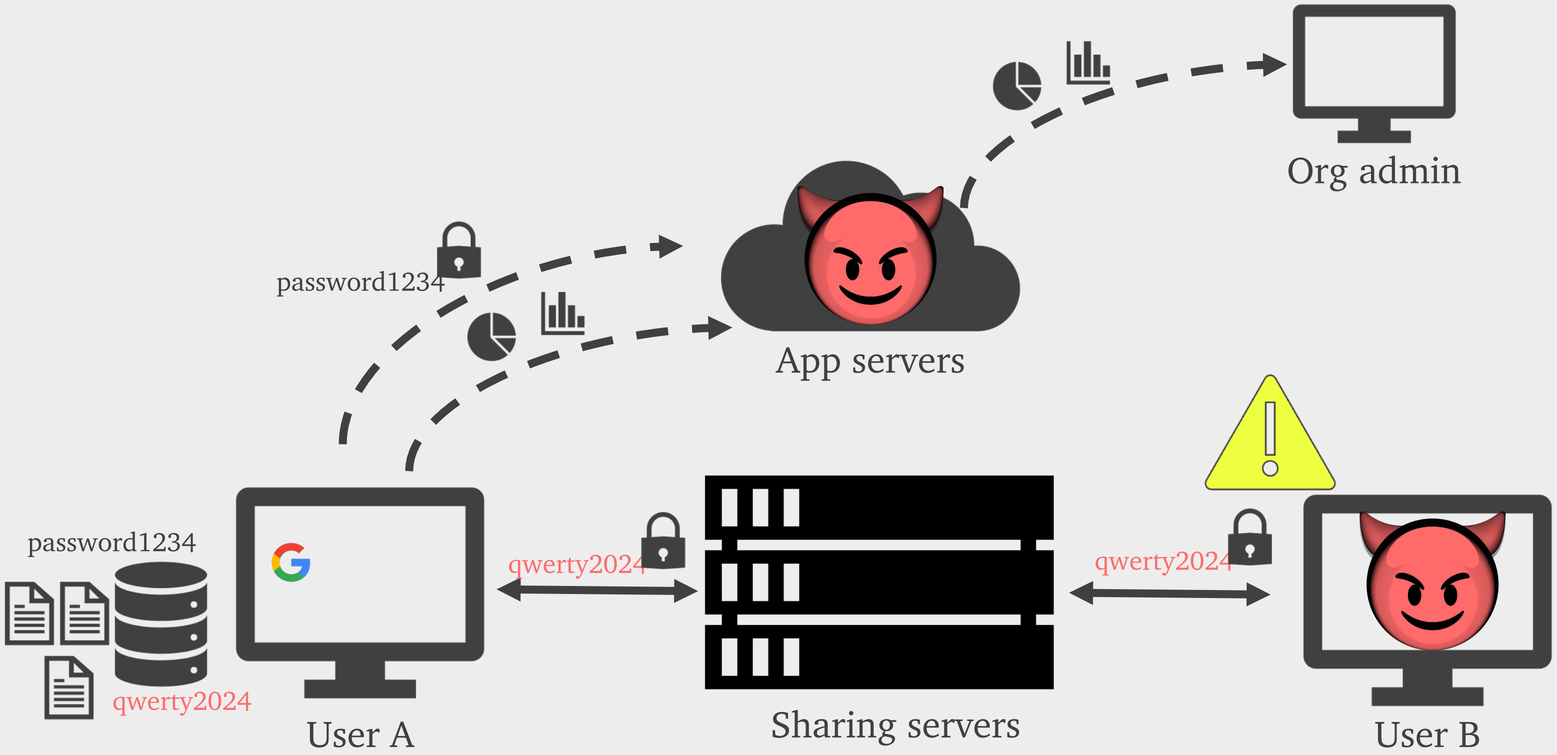
New threat model for password managers that exploits app complexity

Security analysis of 10 password managers

Uncovered four general design patterns that can lead to vulnerabilities



PASSWORD MANAGERS BACKGROUND



What attacks arise from interaction with adversarial clients?

INJECTION ATTACKS

Two key ingredients of injection attacks:

1. observe some form of protected application state
   - Eavesdropper: Encrypted credentials and plaintext metadata
   - Network adversary: HTTPS traffic
2. "inject" payloads into victim's vault from an adversarial client
   - For example, via credential sharing

Idea: application logic can result in cross-user data interactions, which may lead to side-channel leakage

INCLUSION-EXCLUSION CRITERIA

Criteria #1: support for cross-user credential sharing

- ❌Browser-integrated password managers

Criteria #2: cryptographic access control for shared credentials

- ❌Bitwarden

Final list: LastPass, Dashlane, Zoho Vault, 1Password, Enpass, Roboform, Keeper, NordPass, Proton Pass, and KeePassXC

- Over 30% of all password manager users [1]

[1] https://security.org/digital-safety/password-manager-annual- report/

SUMMARY OF FINDINGS

Pattern #1: vault-health metrics

➔ Credential spoofing attack

5/10 applications vulnerable

Pattern #2: URL icon caching

➔Dictionary attack on URLs

6/10 applications vulnerable

Patterns #3 - #4: file deduplication and vault compression

➔ Dictionary attack on attachment contents, URLs, usernames

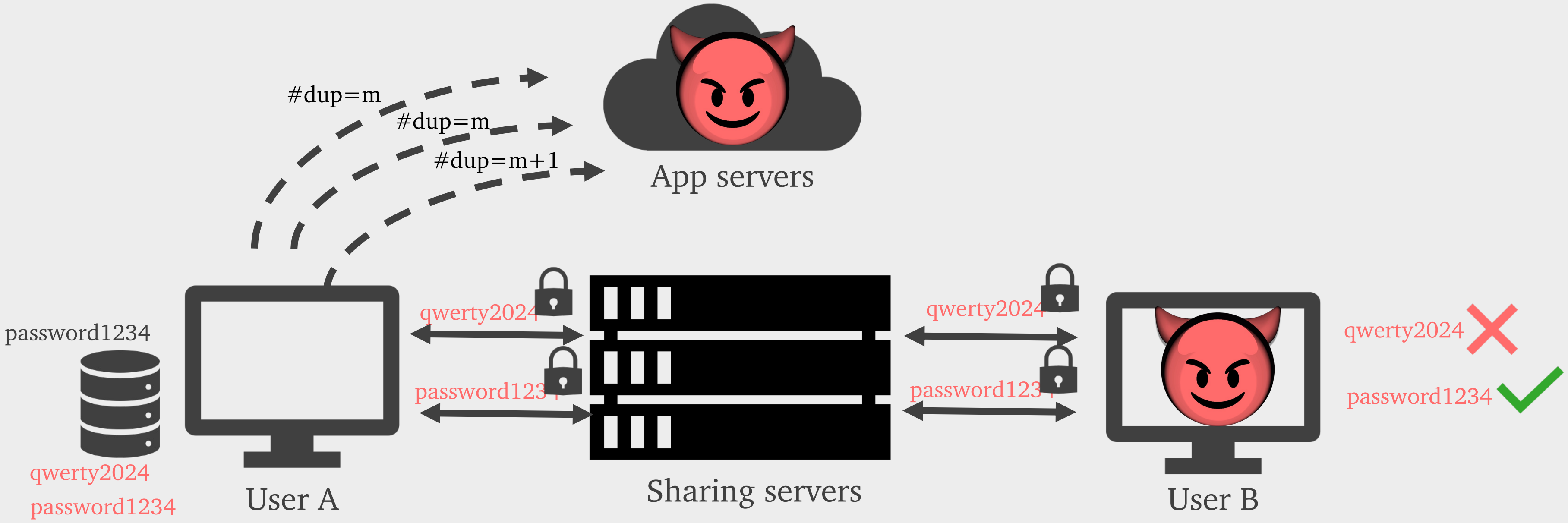
1*/10 application vulnerable

PATTERN#1: VAULT-HEALTH METRICS

Common feature: metrics about the "health" of a user's credentials, such as the number of reused passwords in their vault

- Computed across both personal *and* shared passwords
- Logged outside device, e.g., the application servers

➔ Side-channel that reveals whether a password is in the victim's vault or not!



CREDENTIAL SPOOFING ATTACK

Adversary has an "oracle" to test whether a candidate password is in the victim's vault or not!

➔ Efficient credential spoofing attack via binary search

1. Let $D = (p_1, \ldots, p_n)$ be the list of candidate passwords
2. Share all of $D_{n/2} := (p_1, \ldots, p_{n/2})$ at once
3. If # dup increases, recurse into $D_{n/2}$.
   - Else, recurse into $(p_{n/2+1}, \ldots, p_n)$

Can be modified to work with encrypted metrics

Relevant adversarial goal, even for password manager users [LSFBB18][PZBNC19]

CREDENTIAL SPOOFING ATTACK

Affected applications: LastPass, Dashlane, Zoho Vault, Keeper, and NordPass

Pre-conditions for attack:

1. Application has duplicate password reports
2. Number of duplicates computed across all credentials
3. Number of duplicates logged outside the device

OTHER ATTACKS

Pattern #2: URL icon caching

- Most password managers display icons identifying the domain of credentials
- In many cases, icons are cached on the client, and reused across all credentials
  - Side channel: icon fetched ←→ domain is not in vault
- Leads to dictionary attack on domains in vault
- Network adversary is sufficient*
- Vulnerable applications: Dashlane, 1Password, Enpass, Roboform, NordPass, and Proton Pass

Patters #3 and #4: vault compression and file deduplication

- Lead to vulnerability in other contexts, but first to show for password managers
- Vulnerable applications: KDBX4 (KeePassXC)

MITIGATIONS

Pattern #1: vault-health metrics

- Compute metrics separately

Pattern #2: URL icons

- Retrieve icons every time
- PIR

Patterns #3 - #4: vault compression and file deduplication

- Disable dedup/compression
- Namespace
- Padding or noise

General mitigations for injection attacks?

RESPONSIBLE DISCLOSURE

Application	Attack vector(s)	Mitigations
LastPass	Dup. metrics	Yes
DashLane	Dup. metrics	Yes
	URL icons	Yes
NordPass	Dup. metrics	Yes
	URL icons	Yes
Zoho Vault	Dup. metrics	Yes
Enpass	URL icons	Yes
KeePassXC	File dedup.	Yes
	Vault compression	Yes
Keeper	Dup. metrics	TBD
1Password	URL icons	No
Proton Pass	URL icons	No
Roboform	URL icons	No

TL;DR of RESPONSIBLE DISCLOSURE

- 13 vulnerabilities across 10 applications
- 9/10 vendors acknowledged vulnerabilities
- 6/9 vendors deployed mitigations (partial or full)
- 9/13 vulnerabilities have mitigations deployed for

TAKEAWAYS

- Interaction with adversarial clients may lead to attacks
    - Broader trend in E2EE application security
- New attack vectors suggest that we need new frameworks for auditing E2EE applications
    - Need to reason about interaction with adversarial clients in audit of password managers / E2EE applications
    - How do we detect injection attacks in an automated way?
- How do we navigate security-performance/usability tradeoffs?