Pre-hijacked accounts: An Empirical Study of Security Failures in User Account Creation on the Web

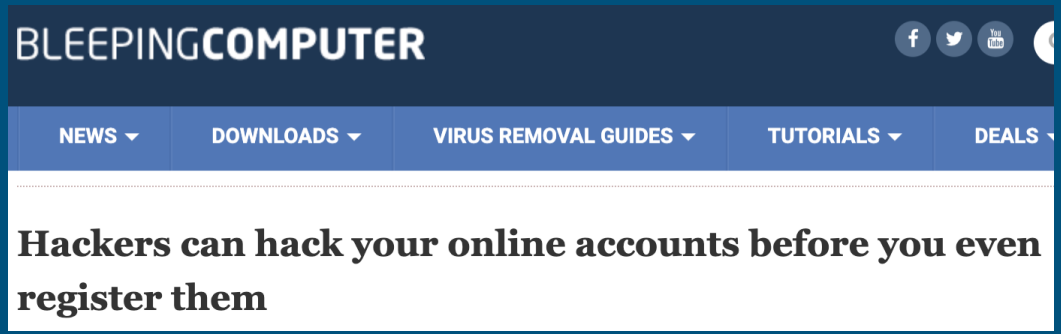
Avinash Sudhodanan
Independent Researcher

Andrew Paverd
Microsoft Security Response Center

USENIX Security, 2022


SIGN IN
The Register
{* RESEARCH *}
About half of popular websites tested found vulnerable to account pre-hijacking
In detail: Ocean's Eleven-grade ruse in which victims' profiles are rigged from the start
Thomas Claburn in San Francisco
Wed 25 Ma


The Daily Swig
Cybersecurity news and views
Dozens of high-traffic websites vulnerable to 'account pre-hijacking', study finds


BLEEPINGCOMPUTER
NEWS
DOWNLOADS
VIRUS REMOVAL GUIDES
TUTORIALS
DEALS
Hackers can hack your online accounts before you even register them


SECURITYWEEK
CYBERSECURITY NEWS, INSIGHTS & ANALYSIS
Subscribe | 2022 CIS
Malware & Threats
Cybercrime
Mobile & Wireless
Risk & Compliance
Security Architecture
Vulnerabilities
Email Security
Virus & Malware
IoT Security
Threat Intelligence
Endpoint Securi
Home › Identity & Access
Hackers Can 'Pre-Hijack' Online Accounts Before They Are Created by Users


The Hacker News
Home
Newsletter
Offers
Learn How Hackers Can Hijack Your Online Accounts Even Before You Create Them

Account Hijacking

Victim 

- owns an account at the service

Victim **Service** **Attacker**

Action: *login*
Inputs: *{victim@gmail.com, victim_passwd}*

Welcome, *Victim*
{victim_cookie}

Account Hijacking

Victim 

- owns an account at the service

Attacker 

- **Goal:** obtain access to the victim's account at the service
- leverages different adversarial tactics to achieve their goal

Victim

Service

Attacker

Phishing Attack

{victim@gmail.com, victim_passwd}

Exploit vulnerabilities

{victim_cookie}

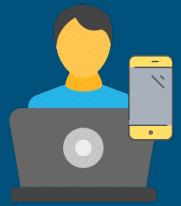
Account **Pre-**Hijacking

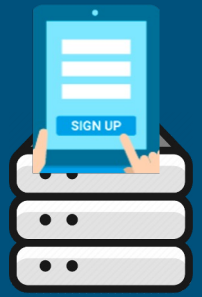
An attacker hacks the online account of the victim even **before** it is created

Assumptions:

- the victim does **not** have an account at the service (at the beginning of the attack)
- victim **ignores** the notifications sent by the services where they don't have an account
- attacker **knows** the identifier (e.g., email address) and basic details (e.g., name) of the victim
- if necessary, the attacker can make the victim **visit** a URL (e.g., through click-bait)

Account Pre-Hijacking





Victim | **Service** | **Attacker**

Attack Timeline

1. attacker creates an account with the victim's identifier

Action: *create_account*
Inputs: *{victim@gmail.com, attacker_passwd, victim_name}*

Welcome, *Victim*
{attacker_cookie}

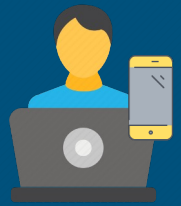
:

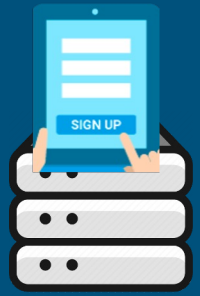
Attacker waits..

Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tries to create an account with the same identifier
3. victim initiates password reset





Victim **Service** **Attacker**

Action: *create_account*
Inputs: *{victim@gmail.com, victim_passwd}*

Account already exists

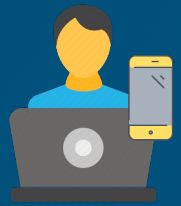
Action: *reset_password*
Inputs: *{victim@gmail.com}*

Password reset link sent

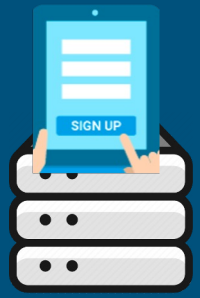
Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tries to create an account with the same identifier
3. victim initiates password reset
4. victim changes password
5. logs in and starts using the account



Victim



Service

Attacker

Action: *set_new_password*
Inputs: *{victim_passwd, reset_token}*

Password Changed

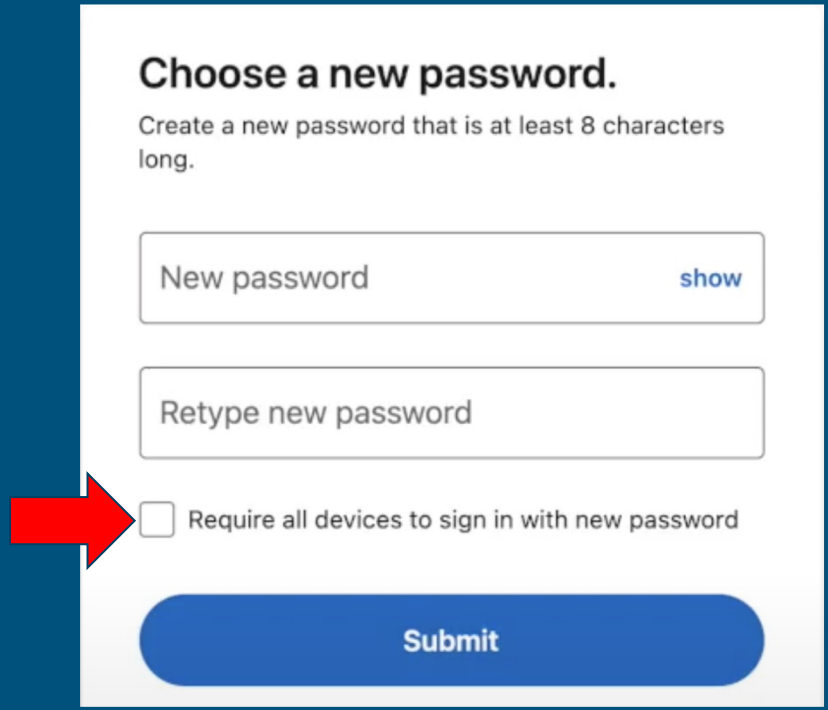
Action: *login*
Inputs: *{victim@gmail.com, victim_passwd}*

Welcome, *Victim*
{victim_cookie}

What can go wrong?

Case Study: LinkedIn

- If the victim does not invalidate old sessions, the attacker can maintain access to the recovered account

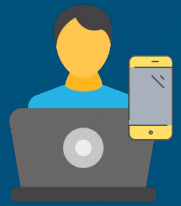


LinkedIn’s Password-Change Form

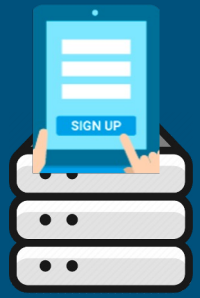
Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tries to create an account with the same identifier
3. victim initiates password reset
4. victim changes password
5. logs in and starts using the account
6. maintains access to the account



Victim



Service

Attacker

Action: *create_account*
Inputs: *{victim@gmail.com, attacker_passwd, victim_name}*

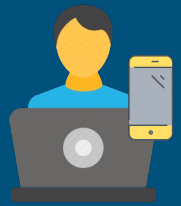
Welcome, *Victim*
{attacker_cookie}

:

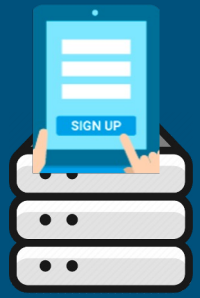
Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tries to create an account with the same identifier
3. victim initiates password reset
4. victim changes password
5. logs in and starts using the account
6. maintains access to the account



Victim



Service

Attacker

Action: *upload_file*
Inputs: *{victim_cookie, secret.txt}*

Uploaded!

:

Action: *view_saved_files*
Inputs: *{attacker_cookie}*

secret.txt

Account Pre-Hijacking

Attacker

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tri… same identi…
3. victim initiates password reset
4. victim changes password
5. logs in and starts using the account
6. attacker maintains access to the account

Unexpired Session Attack

Inputs: *{victim_cookie, secret.txt}*

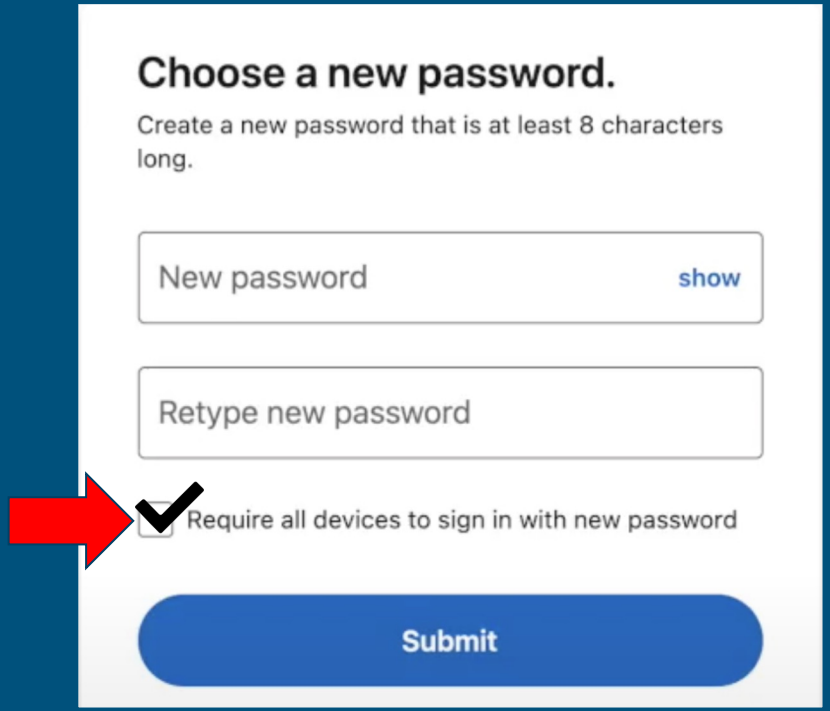
Uploaded!

:

Action: *view_saved_files*
Inputs: *{attacker_cookie}*

secret.txt

How did LinkedIn fix this?

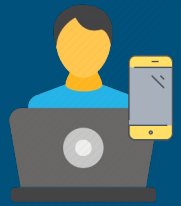


LinkedIn’s Password-Change Form

Account Pre-Hijacking

Attack Timeline

1.  attacker creates an account with the victim's identifier
2.  victim tries to create an account with the same identifier (through Single Sign-On)



Victim

Service

Attacker

Action: *create_account*
Inputs: *{victim@gmail.com, attacker_passwd, victim_name}*

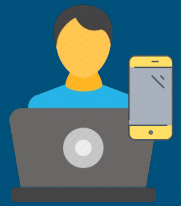
Welcome, *Victim*
{attacker_cookie}

Attacker waits..

Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim tries to create an account with the same identifier (through Single Sign-On)



Victim

Service

IdP Service

Attacker

Action: *sso_signup*
Inputs: *{victim_access_token}*

Action: *get_userinfo*
Inputs: *{victim_access_token}*

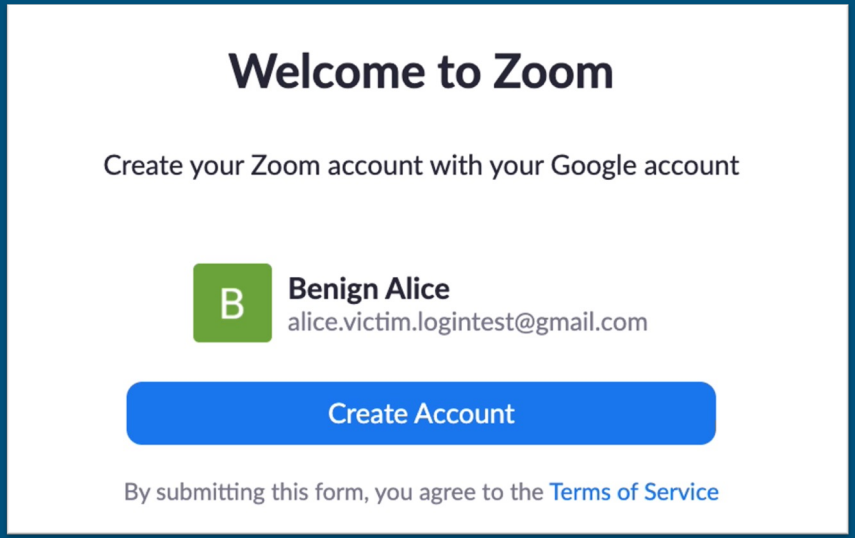
{Victim, victim@gmail.com, ..}

Account already exists

What can go wrong?

Case Study: Zoom

- silently merged the attacker-created account and the victim's account
- the victim may think that they are creating a new account



UI shown to the victim during Account Merging

Account Pre-Hijacking

Attack Timeline

1. attacker creates an account with the victim's identifier
2. victim
same ide
3. attacker maintains access to the account

vice

Attacker

Classic-Federated Merge Attack

Inputs: *{victim_access_token}*

Action: *get_userinfo*
Inputs: *{victim_access_token}*

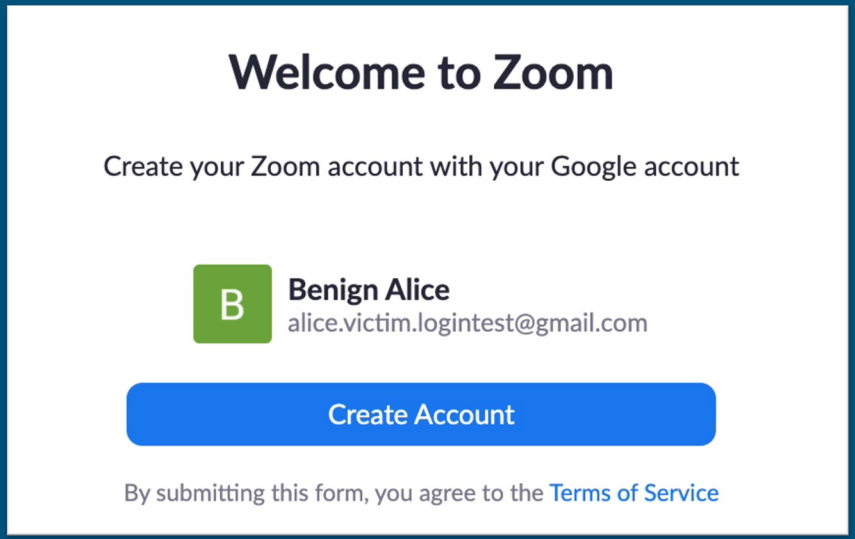
{Victim, victim@gmail.com, ..}

Account already exists

Account Created

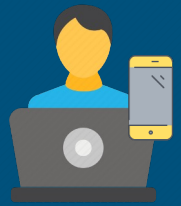
How did Zoom fix this?

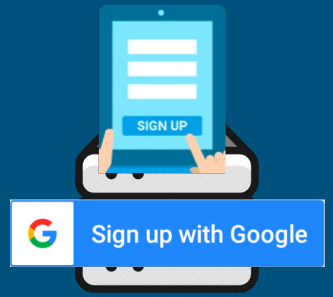
- Implemented strict identifier verification



User Interface of Zoom during Account Merging

Account Pre-Hijacking





Victim | Service | Attacker

Attack Timeline

1. attacker creates an account with the victim's identifier

Action: *create_account*
Inputs: *{victim@gmail.com, attacker_passwd, victim_name}*

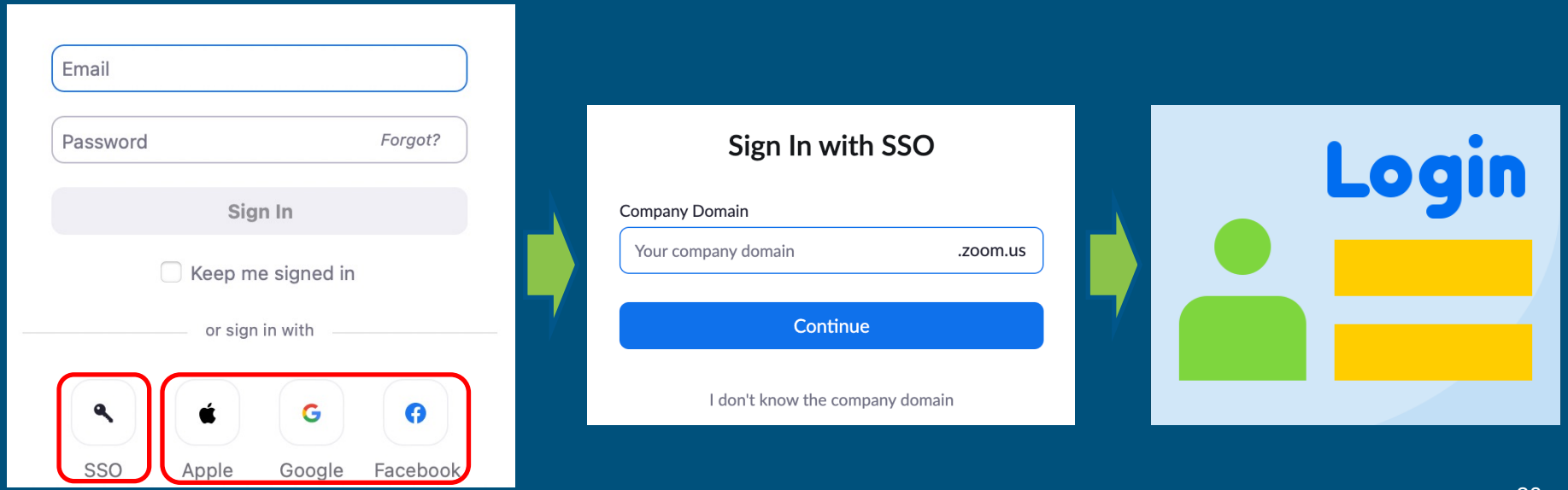
27TH USENIX SECURITY SYMPOSIUM

O Single Sign-Off, Where Art Thou? An Empirical Analysis of Single Sign-On Account Hijacking and Session Management on the Web

Authors:
Mohammad Ghasemisharif, Amrutha Ramesh, Stephen Checkoway, Chris Kanich, and Jason Polakis, *University of Illinois at Chicago*

Custom Single Sign-On

Case Study: Zoom



Account Pre-Hijacking

Attack Timeline

1. attacker creates an IdP and **configures** it with the online service
2. attacker creates an account with the **victim's identifier** at the **attacker's IdP**
3. attacker logs into the service through the attacker's IdP

Victim

Service

Attacker's IdP

Attacker

configure

Action: *create_account*
Inputs: *{victim@gmail.com, attacker_passwd}*

Action: *custom_sso_login*
Inputs: *{attacker_token}*

{victim@gmail.com, ..}

Account created

Account Pre-Hijacking

Attacker's IdP

Attacker

Non-Verifying IdP Attack

Attack Timeline

1. attacker creates an IdP and **configures** it with the online se
2. attacker create **victim's identifier** at the **attacker's IdP**
3. attacker logs into the service through the attacker's IdP
4. victim tries to create an account with the same identifier
5. attacker maintains access to the account

Action: *create_account*
Inputs: *{victim@gmail.com, victim_passwd}*

Accounts merged

Account Created

Types of Pre-Hijacking Attacks

1. Unexpired Session Attack

2. Classic-Federated Merge Attack

3. Non-Verifying IdP Attack

4. Unexpired Email Change Attack

5. Trojan Identifier Attack

Paper: https://www.usenix.org/conference/usenixsecurity22/presentation/sudhodanan

MSRC Blog: https://msrc-blog.microsoft.com/2022/05/23/pre-hijacking-attacks/

Experiments

Measure prevalence of account pre-hijacking attacks

Focused on **150** most-popular websites (Alexa global top websites)

- **136** websites supported account creation
- tested **75** websites (skipped duplicates, unreachable, high-requirements,..)

Results

- **35** websites were vulnerable (Zoom, LinkedIn, Dropbox, Instagram, ..)
- responsibly disclosed the vulnerabilities to the affected vendors

Countermeasures

- Strict identifier verification
  - verify ownership of claimed identifier before account creation
- Defense-in-depth
  - Password-Reset: Sign-out all existing sessions
  - Merging account: Ask consent before merging
  - Unverified account pruning: Delete unverified accounts after a certain period
  - Multi-factor authentication: Notify the victim of new login attempts
  - …

Conclusion

- Focused on the **account creation** process
- Identified **five** different account pre-hijacking attacks
- Tested **75** websites for account pre-hijacking attacks
- **35** websites were vulnerable (including prominent ones)
- Responsibly **disclosed** all the vulnerabilities
- Presented different **countermeasures**

Paper: https://www.usenix.org/conference/usenixsecurity22/presentation/sudhodanan

MSRC Blog: https://msrc-blog.microsoft.com/2022/05/23/pre-hijacking-attacks/