Modern Identity Providers Under Attack: Tactics, Techniques, and Mitigations

When the gatekeeper is the target

Anurag Khanna

#whoami

- Director - Incident Response @CrowdStrike
- Advising organizations while they are responding to incidents
- SANS Certified Instructor, GSE #97
- Past speaker at Blackhat, RSA, BSides, SANS Summits, AISA Con etc

A more detailed version of this deck is available on

rudrasec.io/talks

Disclaimer

- The views expressed in this talk represent my own views and not those of my employer
- We are going to talk about known threat actor techniques, published by several organizations

Initial Access to Identity Providers

- Initial access to IDPs
  - Valid Cloud Accounts
    - Password Spraying
    - Credential Stuffing
    - Phishing for Credentials
    - MFA bypass techniques like MFA Fatigue attacks
  - Exploiting Public Facing Applications

This is too basic and boring. We wanna see something more interesting
Attacker
Defender

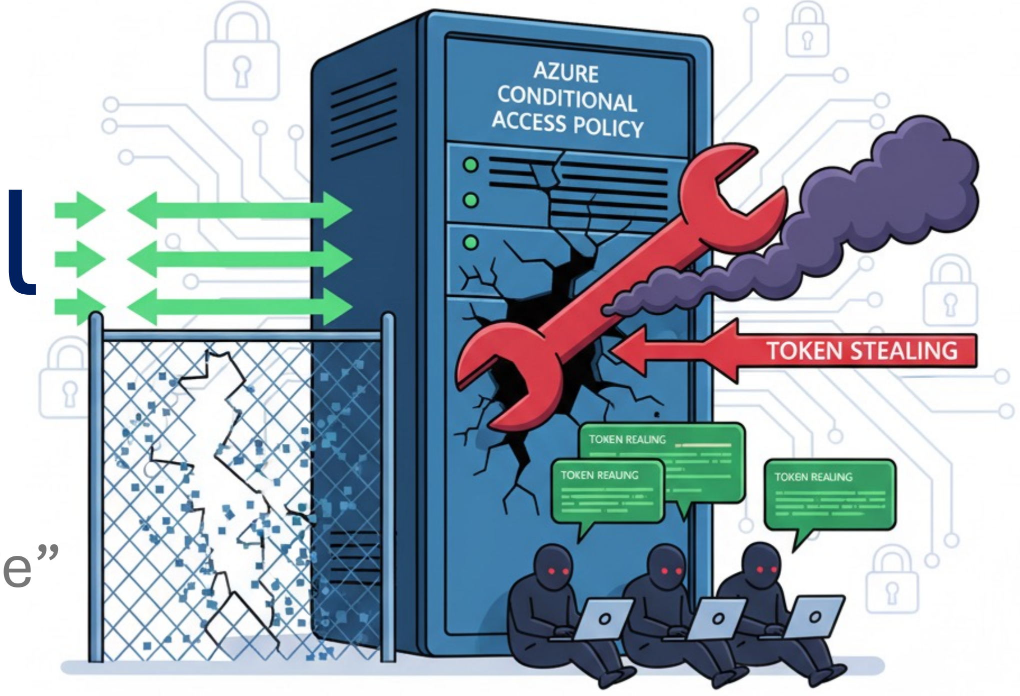
But you know this, we are not here for this.

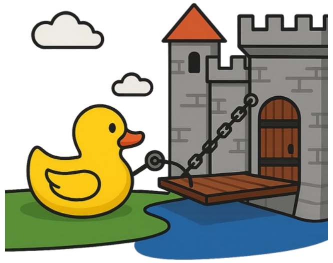
What we are going to talk about

- Attacking Conditional Access Policies
- Attacking SAML Authentication
- Targeting OAuth 2.0
- Trusted Relationship Compromise
  - Attacking OAuth Applications
- Attacking Delegated Admin Permissions
- Cross Tenant Synchronisation Abuse
- Abusing Temporary Access Pass

Attacking Conditional Access Policy

“Rubber Ducks Can Open the Castle's Drawbridge”





Conditional Access Policies (CAP)

- Enforce conditions and access restrictions based on pre-defined conditions
- CAPs only apply at authentication/token issuance, they are typically not checked once tokens are issued
- Attacks often find ways to weaken, bypass and edit CAPs

Exploit CAP Gaps

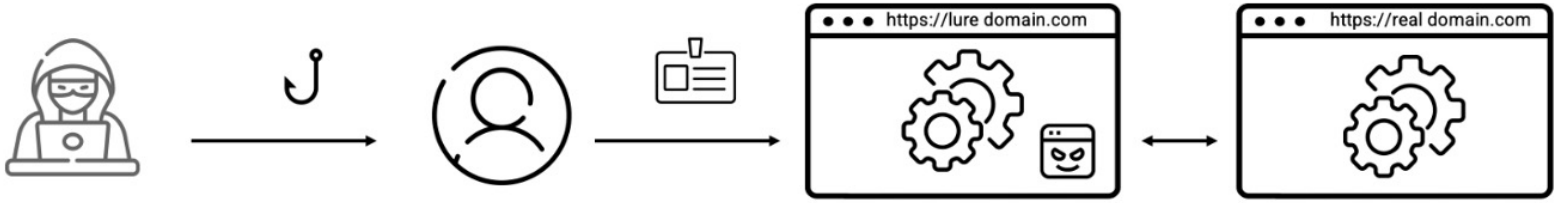
- Conditional Access Policies often have gaps that can be bypassed
  - Excluded Accounts (think, break glass accounts)
  - Legacy Authentication Protocols
  - Exempted platforms or user agents (e.g., legacy apps, mobile apps, Intune-compliant devices

Trusted IP Locations

- Trusted IP addresses in CAPs (office locations) etc
- Attackers can add IP address to trusted IP address pool, to bypass MFAs
- Compromise a system in the Trusted IP Location

Post Token Issuance

- Conditional Access Policies by design are only evaluated at the time of the authentication
  - A Threat Actor can steal an already issued refresh token/cookie



The adversary sets up a fake domain and reverse proxy, and sends a phishing link to the victim

The victim clicks on the link, and authenticates with MFA (as needed)

- The adversary intercepts the requests and forwards to the real server.
- The adversary intercepts the real server's responses and forwards them back to the victim's client, while capturing the valid session cookie

The Adversary uses the captured cookie to authenticate into the real domain

Defend/Response

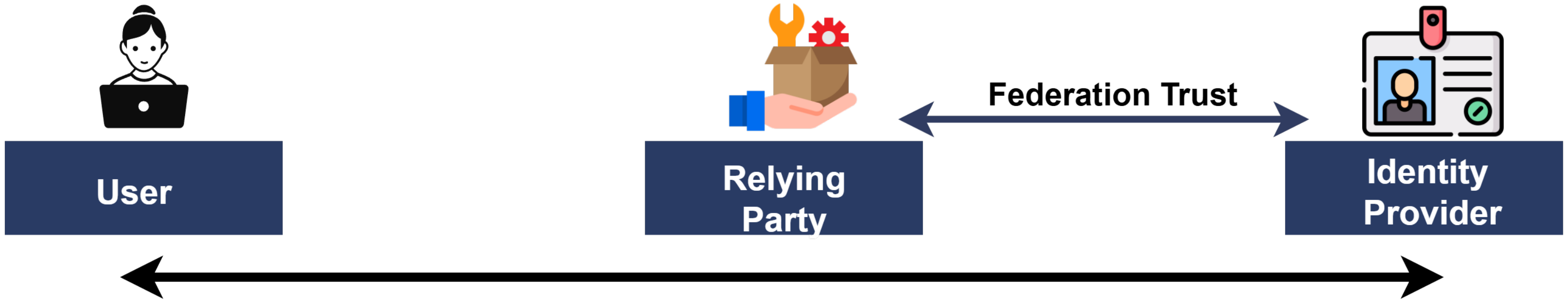
- Review Conditional Access Policies, plug gaps
- Monitor Entra ID audit logs for CAP modifications

  - Update policy and Add policy
- Remove Trusted Locations
  - If you need it review it often and monitor changes
- Protect against Token attacks
  - Enforce a short refresh token lifetime for high-risk accounts
  - Implement Continuous Access Evaluation (CAE)

Attacking SAML Authentication

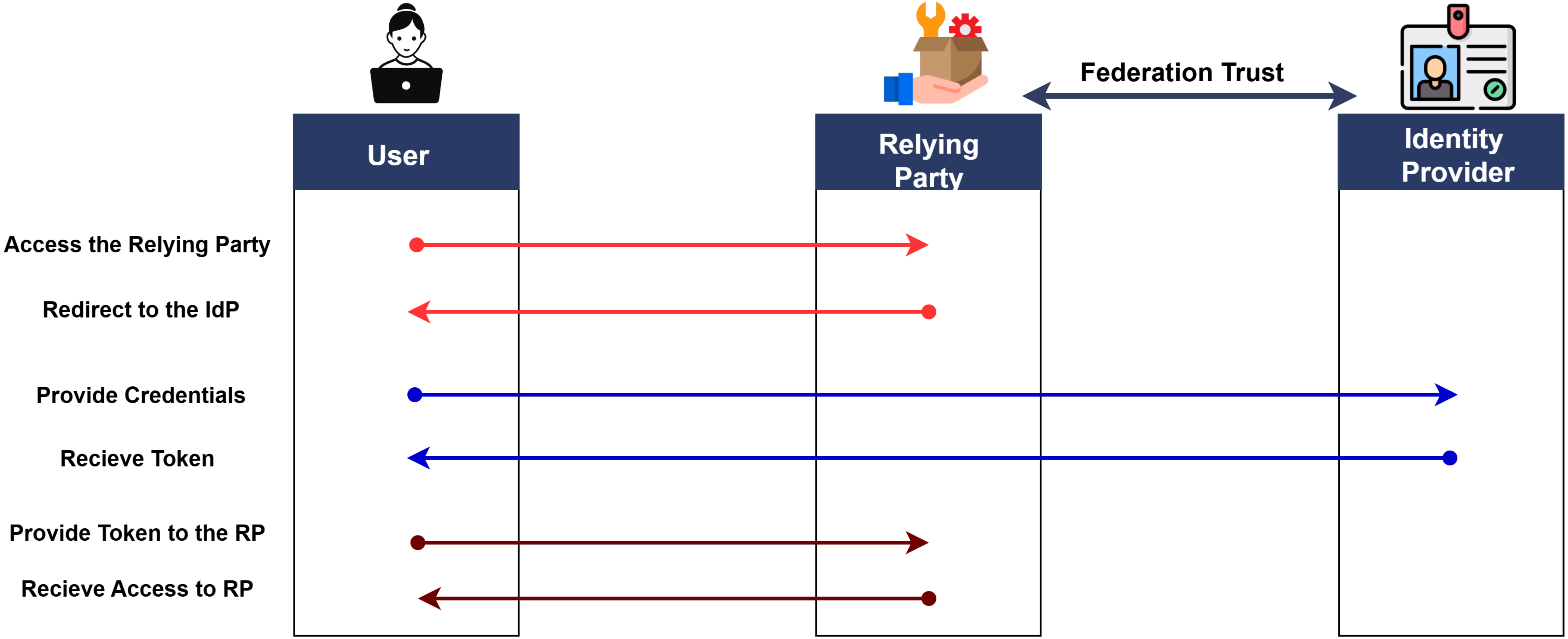
"The Dark Kitchen of SAML: Cooking Trust"

SAML Authentication simplified



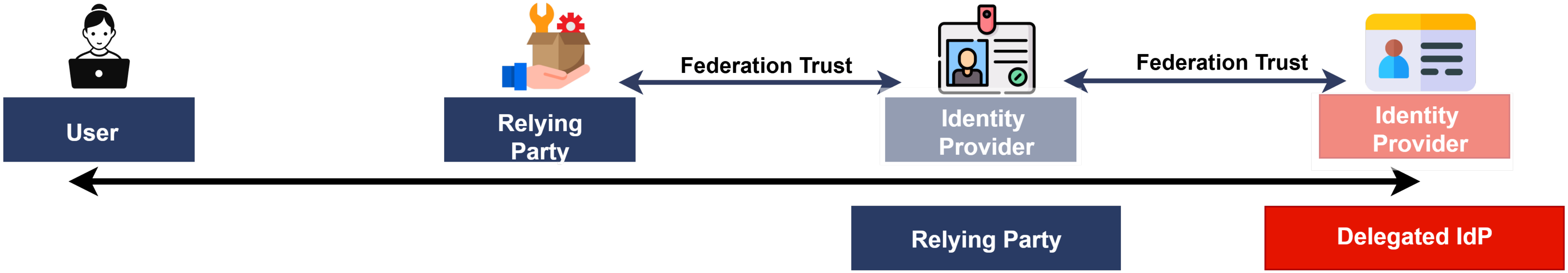
IDP being used as Authentication

SAML Authentication Normal Flow



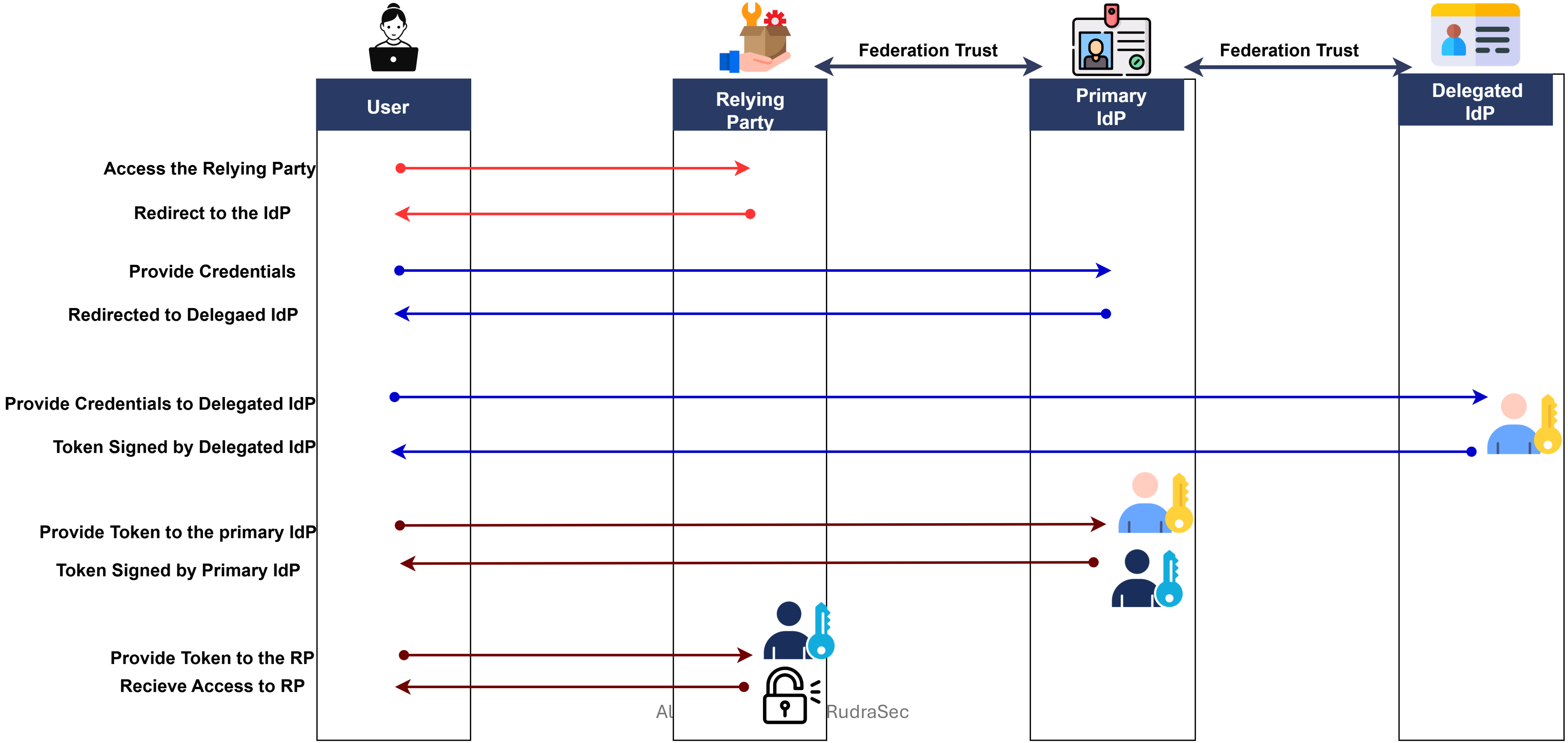
Using SAML to delegate authentication to a third party IdP

Chained SAML Authentication simplified



Delegating your Authentication to another IdP

Chained SAML Authentication Normal Flow



User

Relying Party

Primary IdP

Delegated IdP

Federation Trust

Federation Trust

Access the Relying Party

Redirect to the IdP

Provide Credentials

Redirected to Delegaed IdP

Provide Credentials to Delegated IdP

Token Signed by Delegated IdP

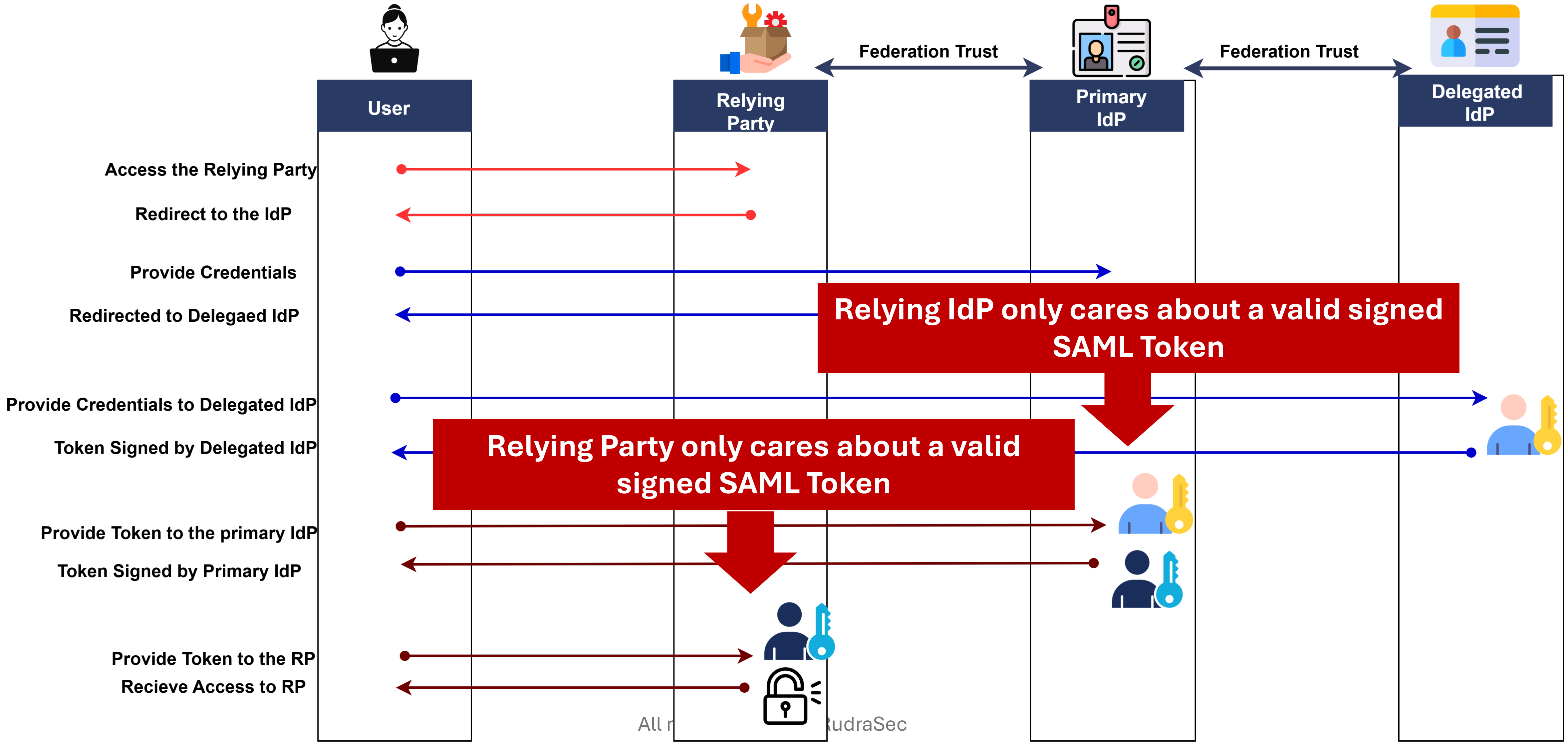
Provide Token to the primary IdP

Token Signed by Primary IdP

Provide Token to the RP

Recieve Access to RP

Chained SAML Authentication



User

Relying Party

Primary IdP

Delegated IdP

Federation Trust

Federation Trust

Access the Relying Party

Redirect to the IdP

Provide Credentials

Redirected to Delegaed IdP

Provide Credentials to Delegated IdP

Token Signed by Delegated IdP

Provide Token to the primary IdP

Token Signed by Primary IdP

Provide Token to the RP

Recieve Access to RP

Relying IdP only cares about a valid signed SAML Token

Relying Party only cares about a valid signed SAML Token

The SAML Vulnerability: Trusting the Signature

Relying Party only cares about a valid signed SAML Token

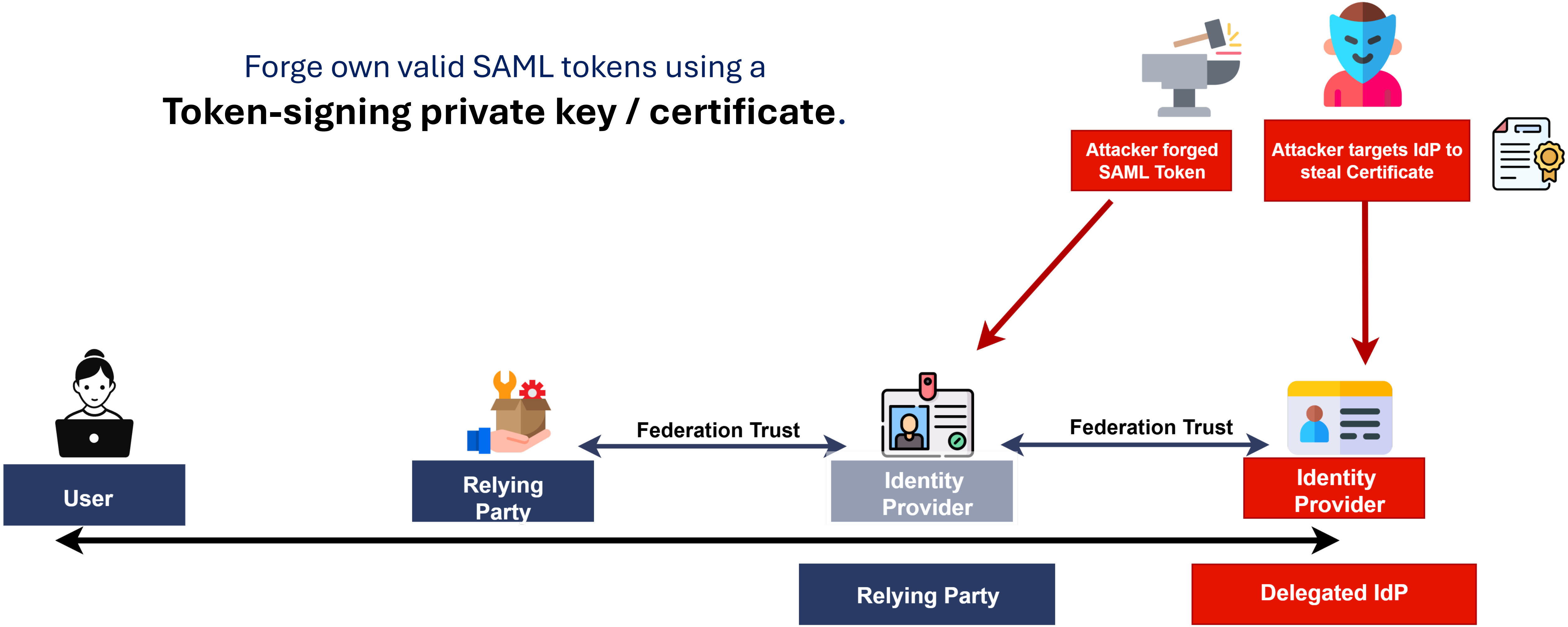
If an attacker can find a way to <u>Forge the Token</u> they can successfully authenticate to the Relying Party or the Relying IdP

Let's look at *how* an attacker can get the ability to forge that token

Attack I - Golden SAML Attack

Forge own valid SAML tokens using a

Token-signing private key / certificate.



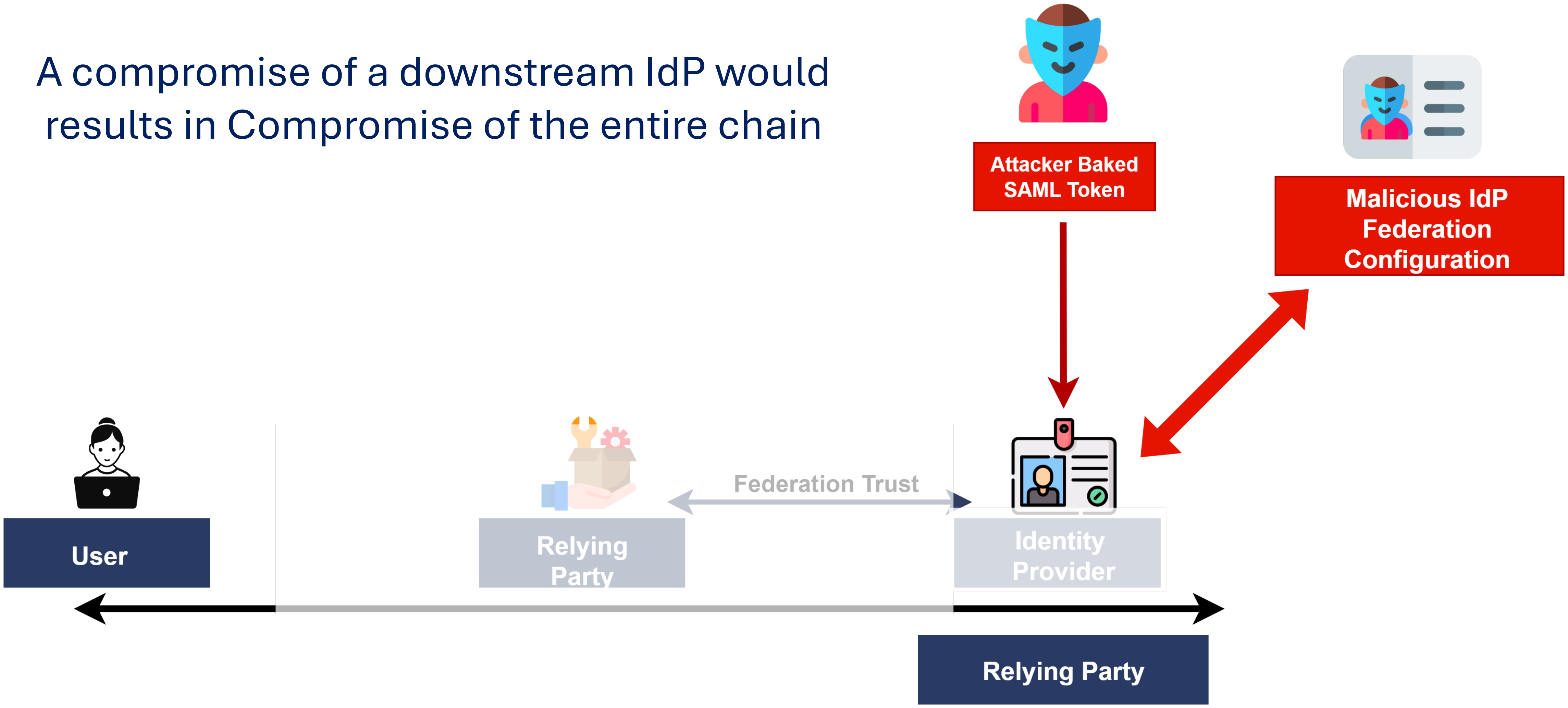
Attack II - Adding a Federation Trust

- Identity Providers can be federated to an external IdP like AD FS
- An attacker with privileged access can add their own Federation configuration in the IdP
- Once done, tokens can be generated allowing the attacker to login as any user

This is like **Golden SAML Attack**, instead of stealing a certificate, the Attacker added their Federated domain for which TA controls certificate.

Attack II - Adding a Federation Trust

A compromise of a downstream IdP would results in Compromise of the entire chain


Attacker Baked SAML Token
Malicious IdP Federation Configuration
Federation Trust
User
Relying Party
Identity Provider
Relying Party

Detection & Prevention

- Protect the private keys used for signing tokens (e.g., using HSMs)
- Monitor for suspicious changes to federation trust settings
- Look for activity logs showing a new federated domain being added to the IdP
- Review Audit Logs in Microsoft Entra ID
  - "Create cross-tenant synchronization setting"
  - “Update cross-tenant access settings”

Targeting OAuth 2.0

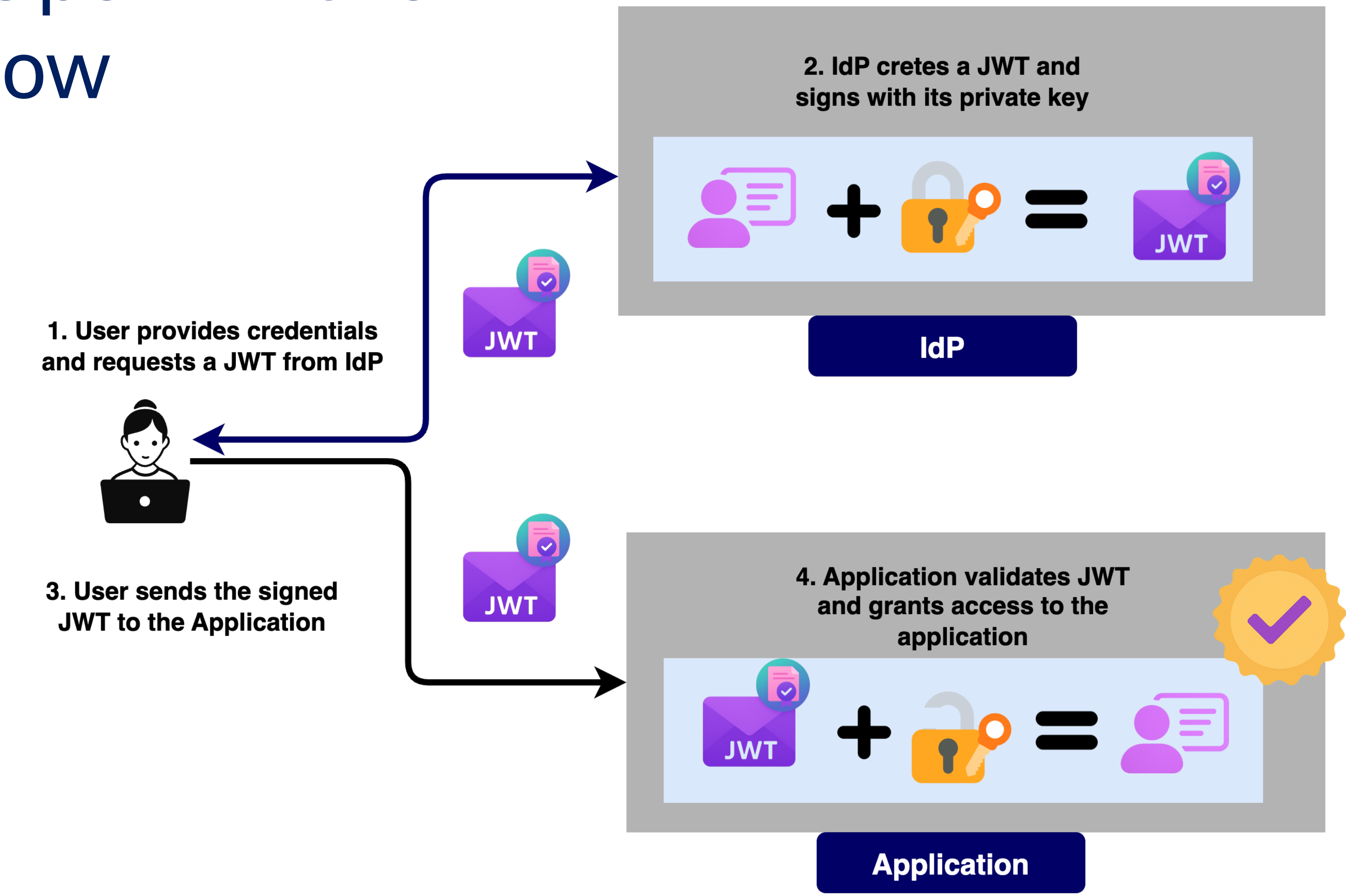
Trouble at the Gate

OAuth 2.0

- **OAuth is for Authorization:** A third-party app getting permission to access a user's resources
- **OpenID Connect (OIDC) is for Authentication:** "Sign in with Google," that's OIDC in action

OAuth uses JSON Web Tokens (JWTs) instead of the XML-based SAML assertions.

OAuth/OpenID Token Grant Flow



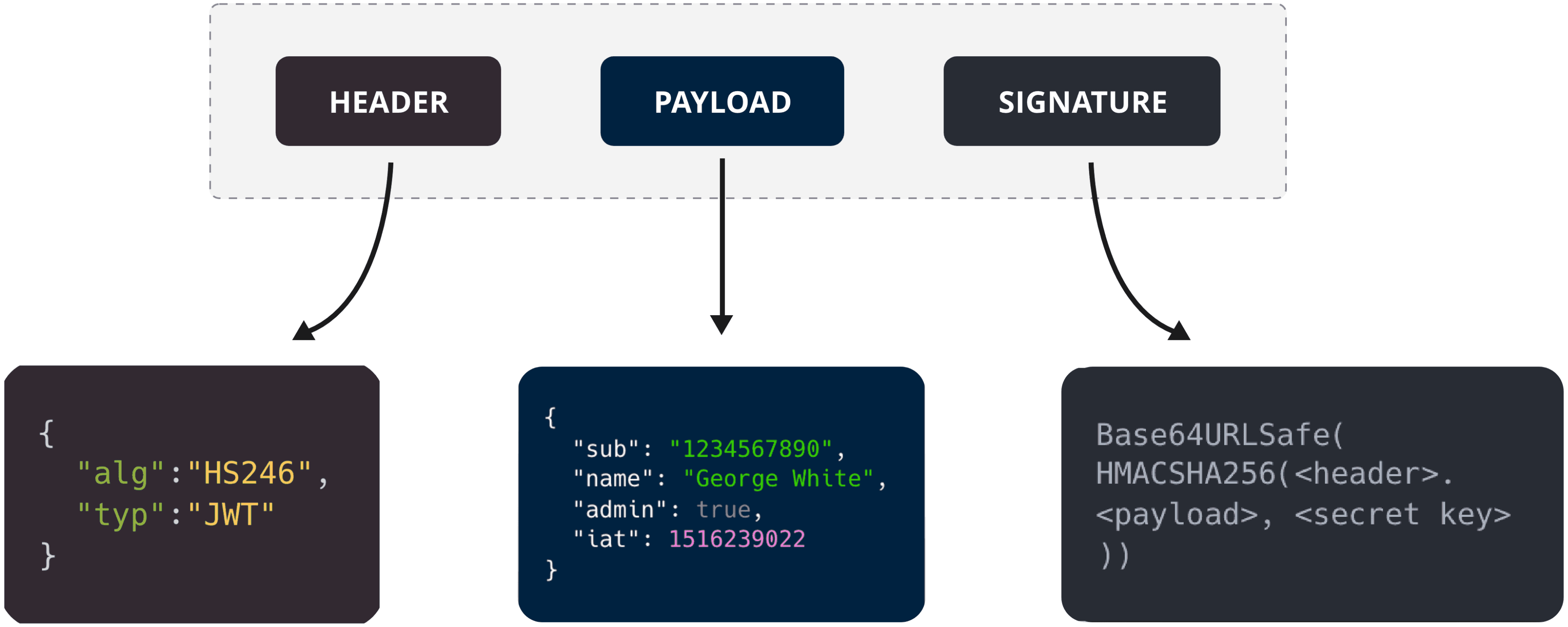
2. IdP cretes a JWT and signs with its private key

1. User provides credentials and requests a JWT from IdP

3. User sends the signed JWT to the Application

4. Application validates JWT and grants access to the application

Structure of a JSON Web Token (JWT)



HEADER

PAYLOAD

SIGNATURE

```
{
  "alg":"HS246",
  "typ":"JWT"
}
```

```
{
  "sub": "1234567890",
  "name": "George White",
  "admin": true,
  "iat": 1516239022
}
```

```
Base64URLSafe(
HMACSHA256(<header>.
<payload>, <secret key>
))
```

The JWT Vulnerability: Trust in the JWT Signature

Entire system rests on the integrity of the private key

If an attacker can get the private key, they can forge a valid token for any user in the organization.

Forging JWT for Application access

Storm in the Mailbox

Storm-0558 targeting Microsoft 365

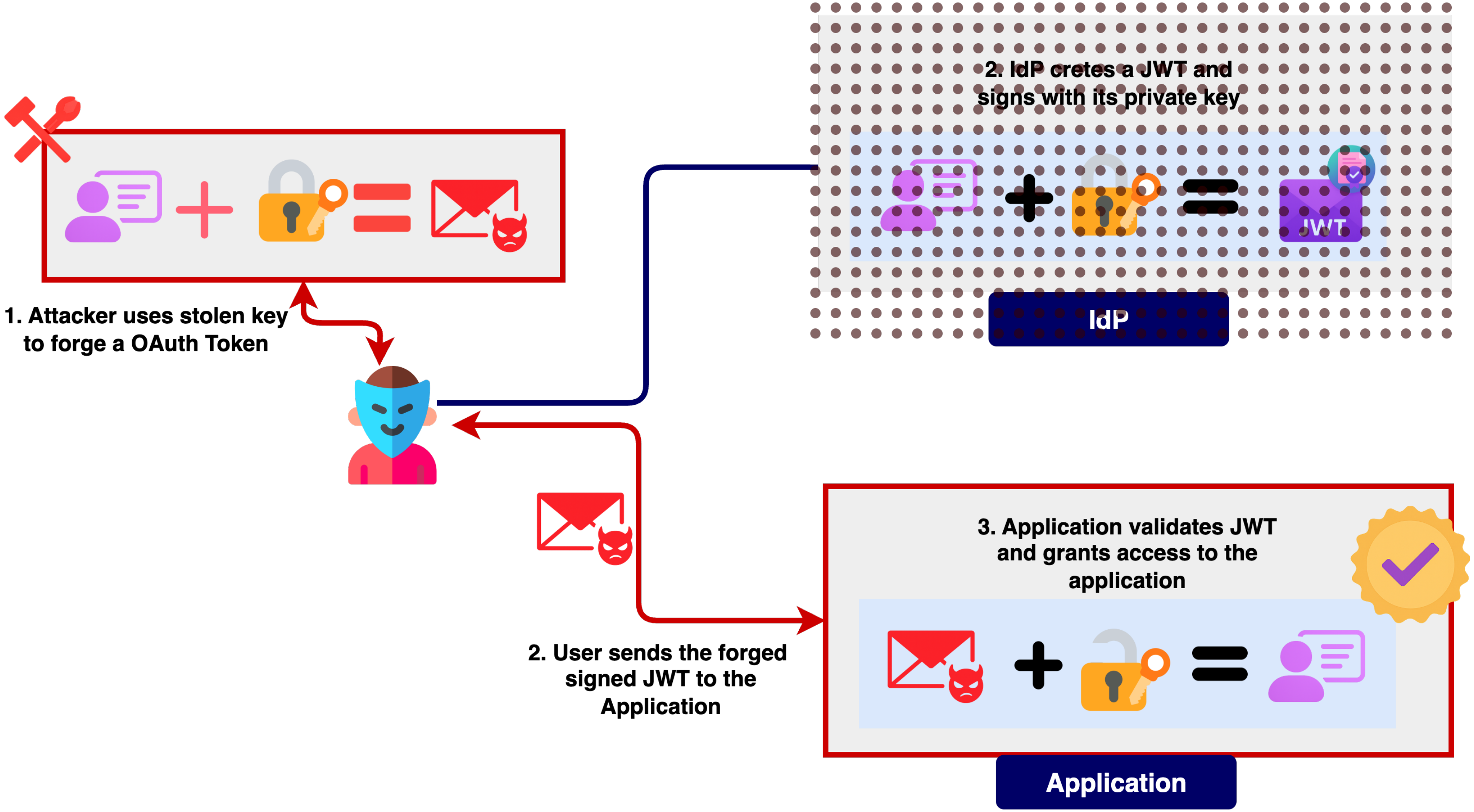
- An attacker was able to get the private key and forge a valid token
- The forged tokens were able to pass validation for both the OIDC and the OAuth 2.0
- Such an attack can happen on custom applications with keys in vaults

A consumer-level signing key was stolen and incorrectly trusted by the enterprise-level Azure AD system.

CYBER SAFETY REVIEW BOARD

Review of the Summer 2023 Microsoft Exchange Online Intrusion

OAuth/OIDC Token Forgery



Detection & Prevention

- Protect your private keys, monitor for suspicious key activity, such as key rotation, new keys being created, or keys being used in unexpected contexts
- Implement a strong key management strategy using Hardware Security Modules (HSMs)
- Use a SIEM to alert on authentication failures or unexpected token issuance from untrusted sources
  - U.S. State Department identified Storm-0558 activity through a custom SIEM detection rule (“Big Yellow Taxi”)

Trusted Relationship Compromises

Turning Partnerships into Persistence

What are Trusted Relationships?

- Pre-existing technical trusted connections between different tenants of different organizations
- Attackers can target trusted relationships, to access multiple down stream environments

Attacking OAuth Applications

Attacking Trusted Partners: Turning Partnerships into Persistence

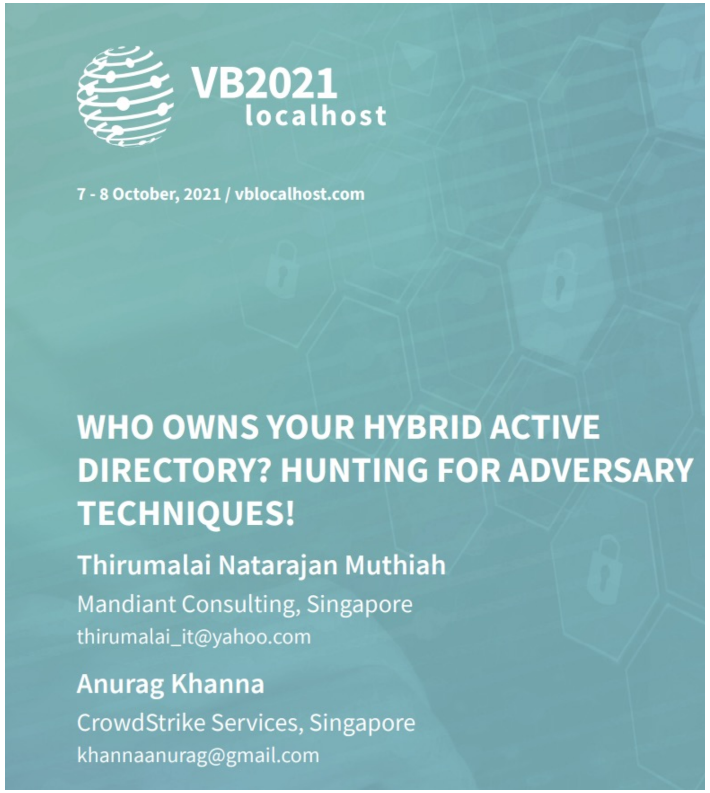
Application Registrations

- Azure Application (App Registration)
  - Is the global, multi-tenant representation (identified by a GUID – App ID) of an application across all tenants like a blueprint of the application
  - Exists in the home tenant where the application is created
  - Define Application permission and scope

Does need an identity to work with called **Service Principal (SP)**

Service Principals

- Service Principal
  - Is the local representation of an application in a specific client tenant where the application is used
  - Exists in every tenant, where the application is to be used
  - Actual identity for the app inside the tenant and holds the credentials
  - Has a unique **ObjectId** in each tenant, which is different from the shared **AppId** of its parent Application Object.
  - SP configuration has details of the AppID



VB2021
localhost

7 - 8 October, 2021 / vblocalhost.com

WHO OWNS YOUR HYBRID ACTIVE DIRECTORY? HUNTING FOR ADVERSARY TECHNIQUES!

Thirumalai Natarajan Muthiah
Mandiant Consulting, Singapore
thirumalai_it@yahoo.com

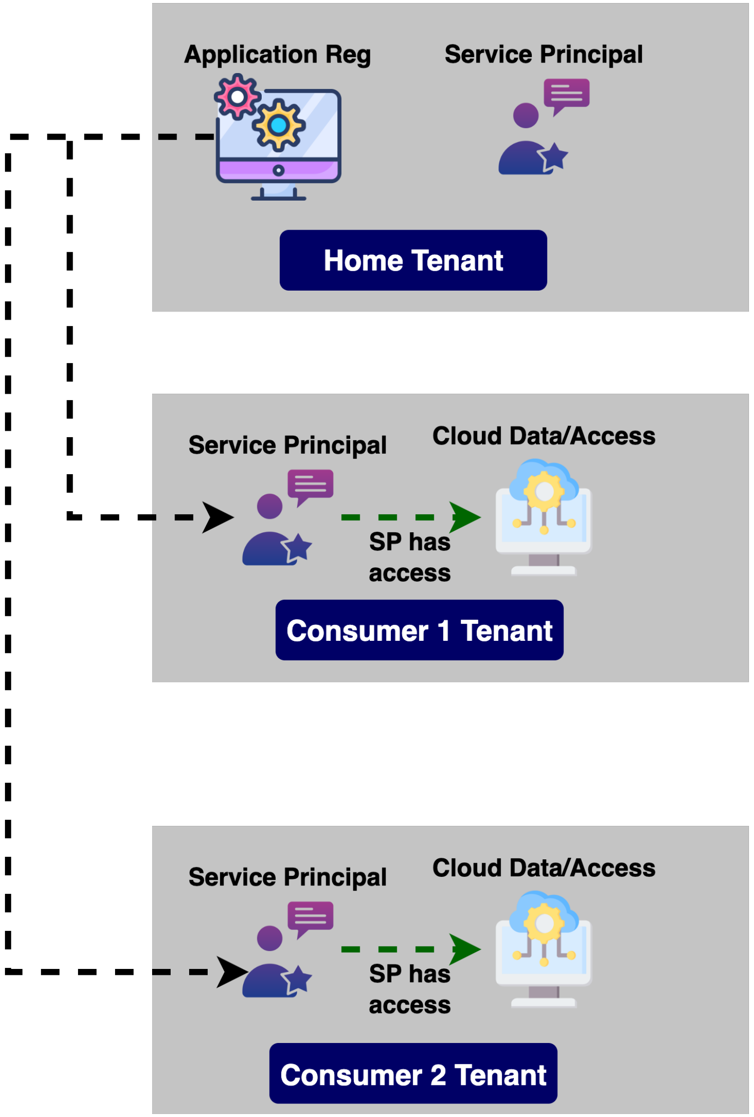
Anurag Khanna
CrowdStrike Services, Singapore
khannaanurag@gmail.com

https://vblocalhost.com/uploads/VB2021-Thirumalai-Khanna.pdf

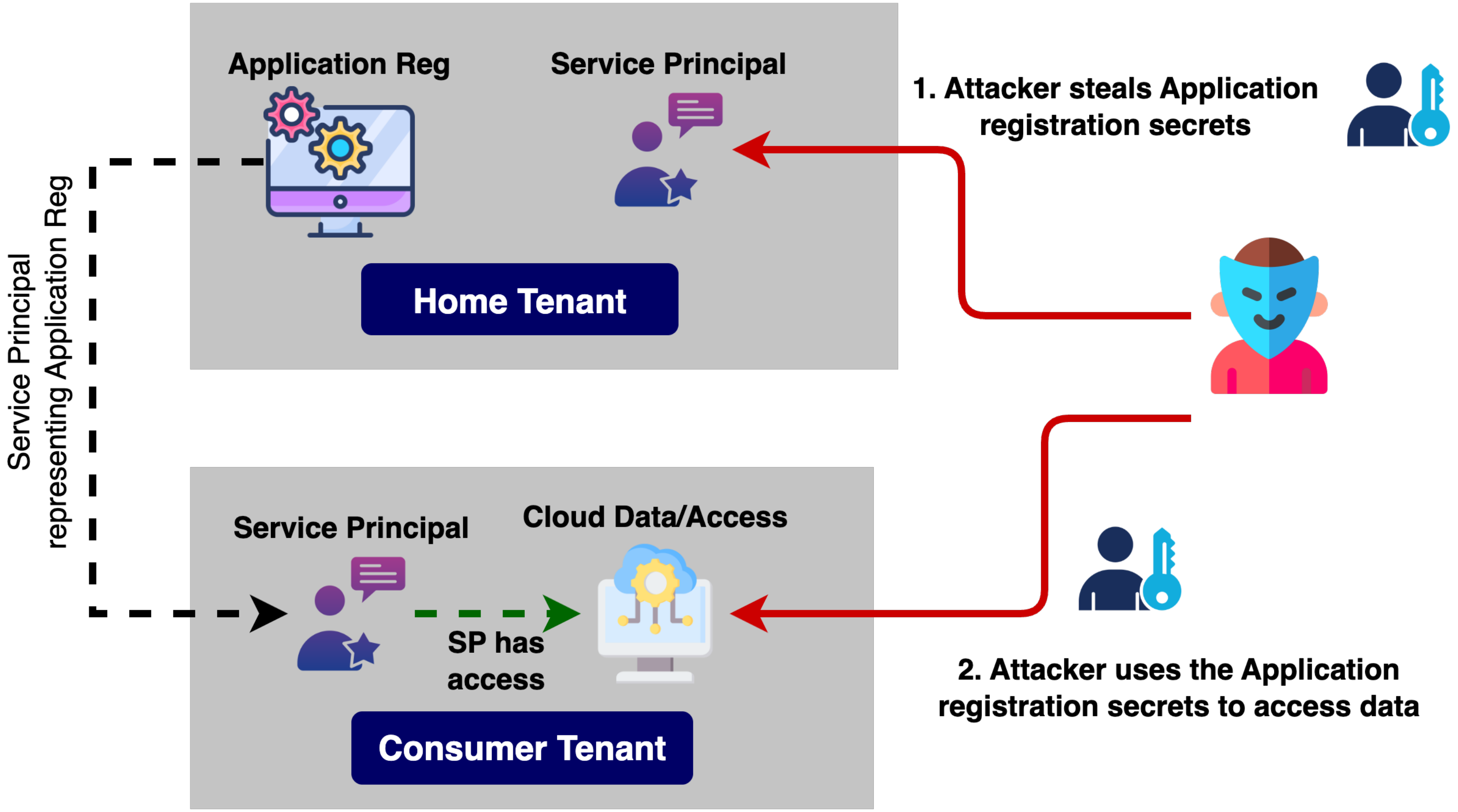
Not only EntraID thing

Microsoft Entra ID	Salesforce	Function
Application Registration	Connected App	The global definition of the application. It holds the shared credentials (Consumer Key and Consumer Secret) and defines the OAuth scopes. A Connected App is a single entity in the developer's org.
Service Principal	The OAuth session / The App's Authorization	The instance of the app in a specific Salesforce customer's org. It is what holds the granted permissions, and it's to this instance that the OAuth token is issued.

Applications & Service Principals in Multi Tenant Apps

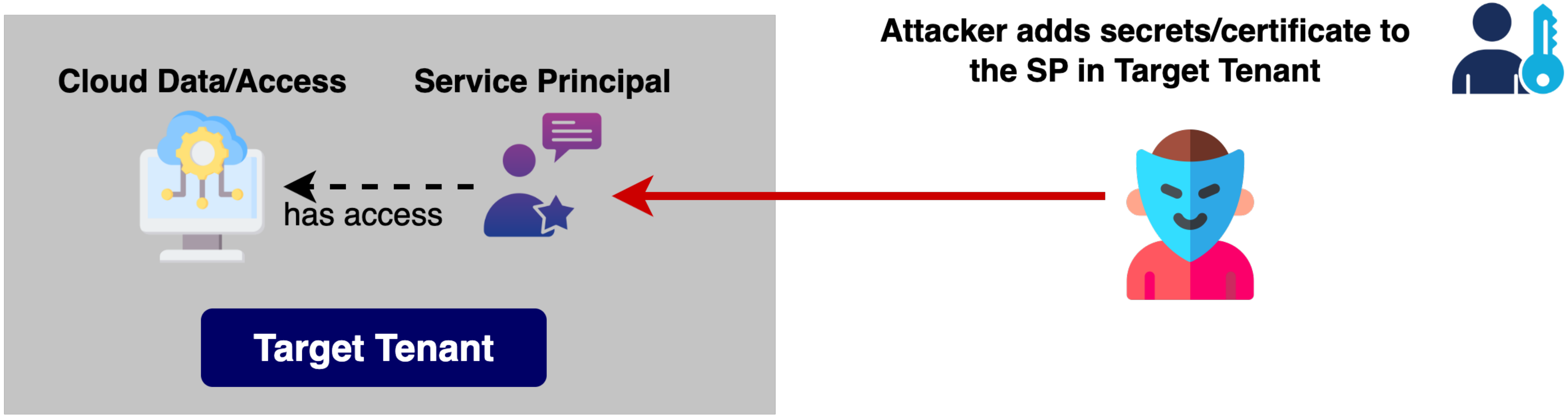


Attack I – Stealing App Secrets


Application Reg
Service Principal
1. Attacker steals Application registration secrets
Home Tenant
Service Principal representing Application Reg
Service Principal
Cloud Data/Access
SP has access
Consumer Tenant
2. Attacker uses the Application registration secrets to access data

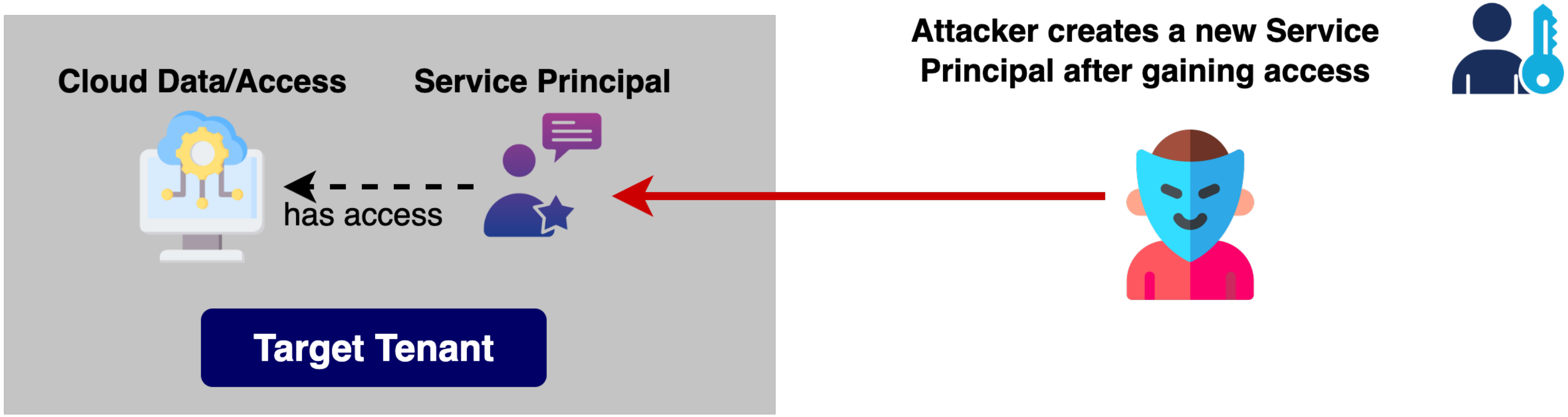
CrowdStrike. (2025, August 21). *MURKY PANDA: A trusted-relationship threat in the cloud*. CrowdStrike. https://www.crowdstrike.com/en-us/blog/murky-panda-trusted-relationship-threat-in-cloud/

Attack II – Adding Application/Service Principal Secrets



Attacker adds secrets/certificate to the SP in Target Tenant

Attack III - Creating a new Service Principal

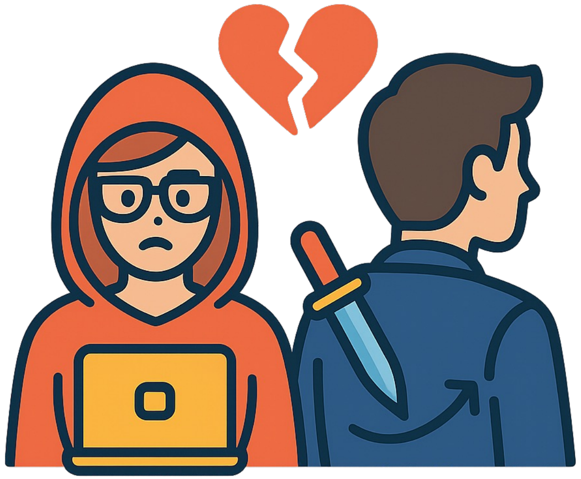

Cloud Data/Access
Service Principal
has access
Target Tenant
Attacker creates a new Service Principal after gaining access

Detection & Prevention

- Protect Application Credentials
  - Never hard-code or store application secrets in plain text. Use a secure vault like Azure Key Vault to manage and rotate secrets automatically
- Enforce Least Privilege
  - Grant applications and Service Principals only the minimum permissions they need to function
- Monitor for Secret Changes
  - Actively monitor Entra ID audit logs for activities like "Add application password credential” and “Add service principal credentials”

Attacking Delegated Admin Permissions

When your partner betrays you



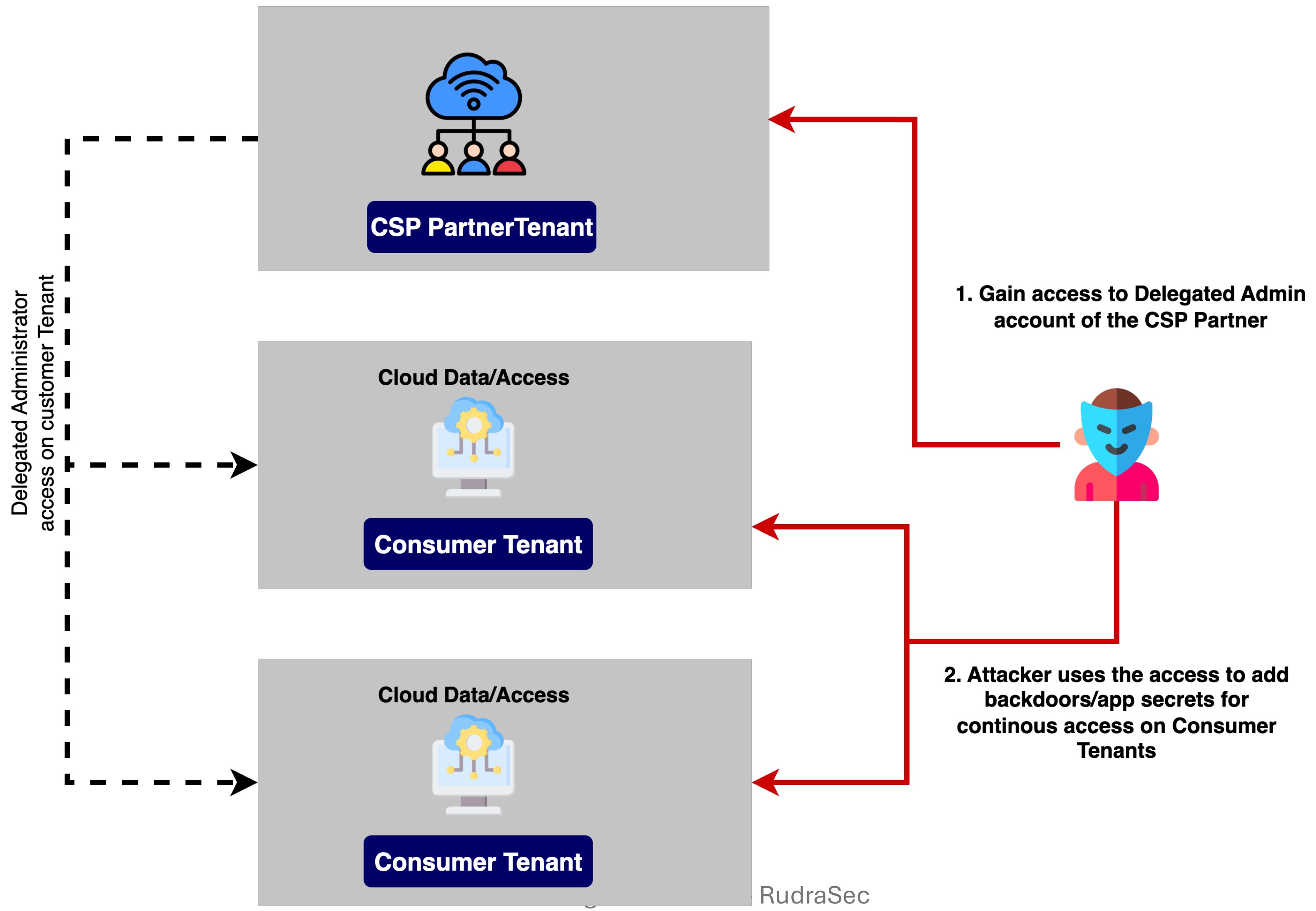
Delegated Admin Privileges(DAP)

- DAP permission model used in the Microsoft Cloud Solution Provider (CSP) program designed for CSPs to support customers
- **Default DAP Roles:**
  - Admin Agents security group was Global Administrator
  - Helpdesk Agents group was typically granted the Helpdesk Administrator role
- Microsoft has moved to Granular Delegated Admin Privileges (GDAP)
  - Still often has permissions that can be misused like, Privileged Role Administrator, User Administrator, Application Administrator etc

Targeting Delegated Admin Privileges

- CSP partners have delegated administrative privileges to manage customer tenants.
- CSP partner tenant’s users can act as global admins or privileged role admins in the customer's tenant without appearing in the customer directory.

Attack I – Abusing DAP from a CSP partner

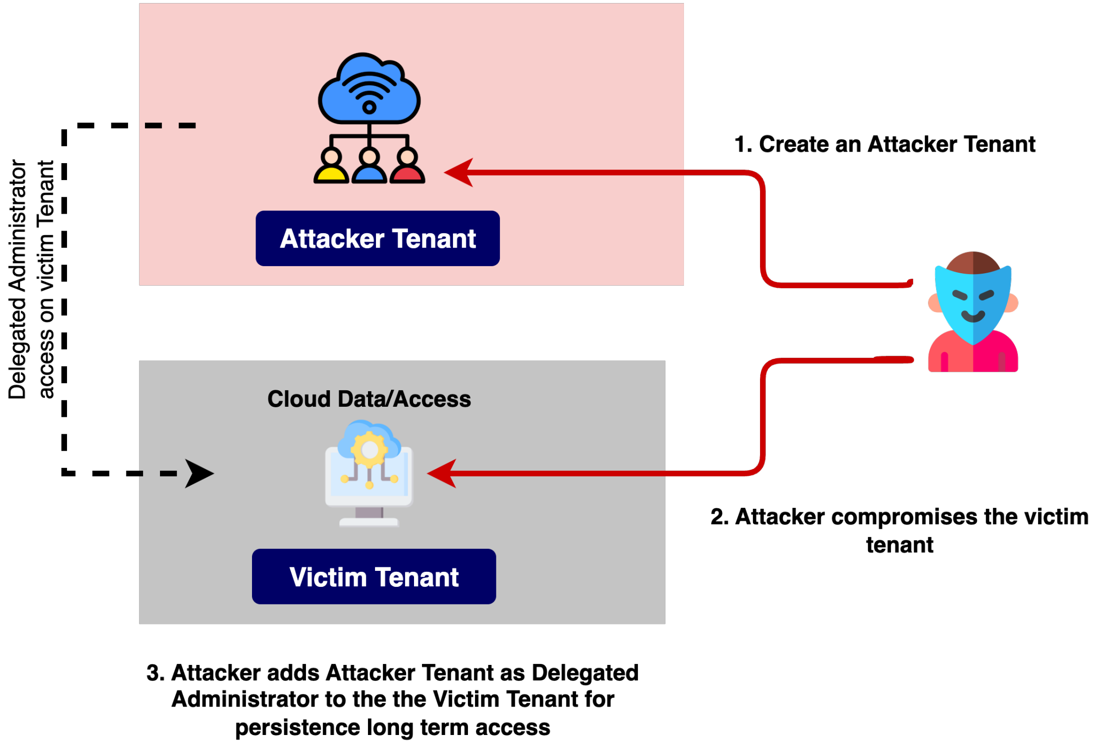


1. Gain access to Delegated Admin account of the CSP Partner

2. Attacker uses the access to add backdoors/app secrets for continous access on Consumer Tenants

Attack II – Add a CSP Partner for access

- Theoretically an attacker can register their tenant as a CSP partner account/ or use a partner tenant they control to add as DAP to target systems
- DAP cannot be configured in non CSP tenants
- There is no publicy documented cases of an attack that has happened, possibly because CSP Partner accounts are tightly controlled by Cloud Companies


Delegated Administrator access on victim Tenant
Attacker Tenant
1. Create an Attacker Tenant
Cloud Data/Access
Victim Tenant
2. Attacker compromises the victim tenant
3. Attacker adds Attacker Tenant as Delegated Administrator to the the Victim Tenant for persistence long term access

Why this works?

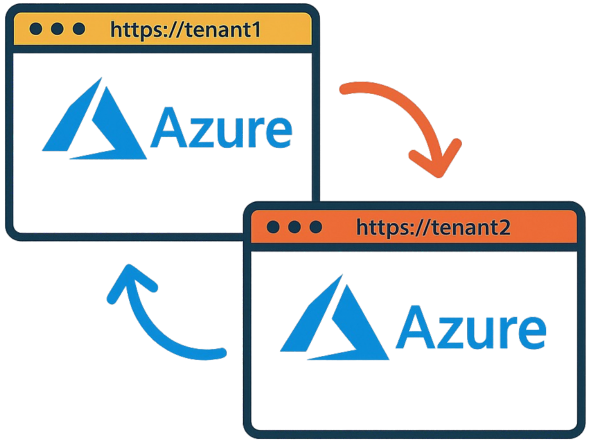
- Trusted Party attack, that can result in multiple tenant access
- Many customers never review or limit DAP assignments, DAP assignments need to be explicitly removed, which often are never removed

Detection & Prevention
Trusted Relationship Compromise

- For CSP
  - Regularly review all DAP and GDAP assignments. Remove access if it is no longer required or if the partnership has ended.
  - Follow Partner security requirements , MFA for all accounts
- Review Partner relationships and DAP access
  - Ensure all partner accounts with administrative privileges have Multi-Factor Authentication (MFA) enabled, regardless of their location
  - Continuously audit and monitor partner activity in your tenant. Pay attention to activities that go beyond the expected scope of their role "Cross-tenant access type"

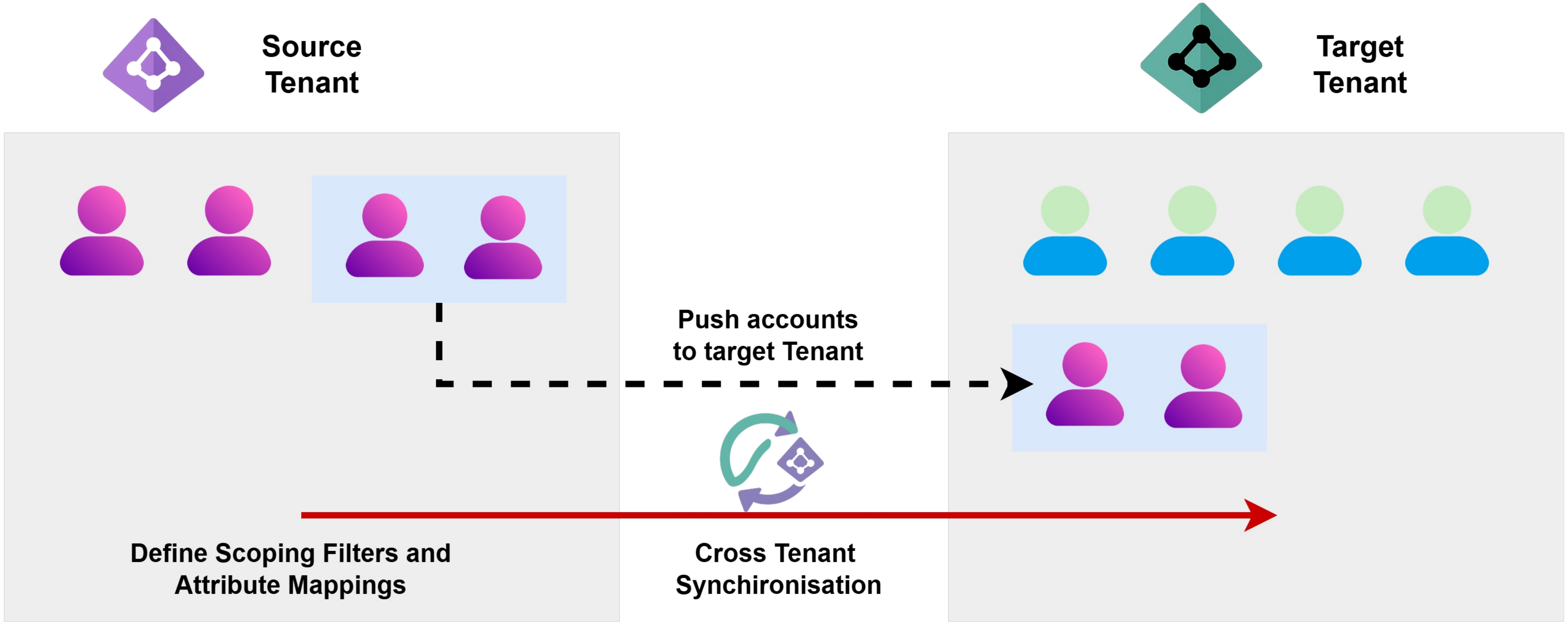
Cross Tenant Synchronisation Abuse

B2B Sync abuse

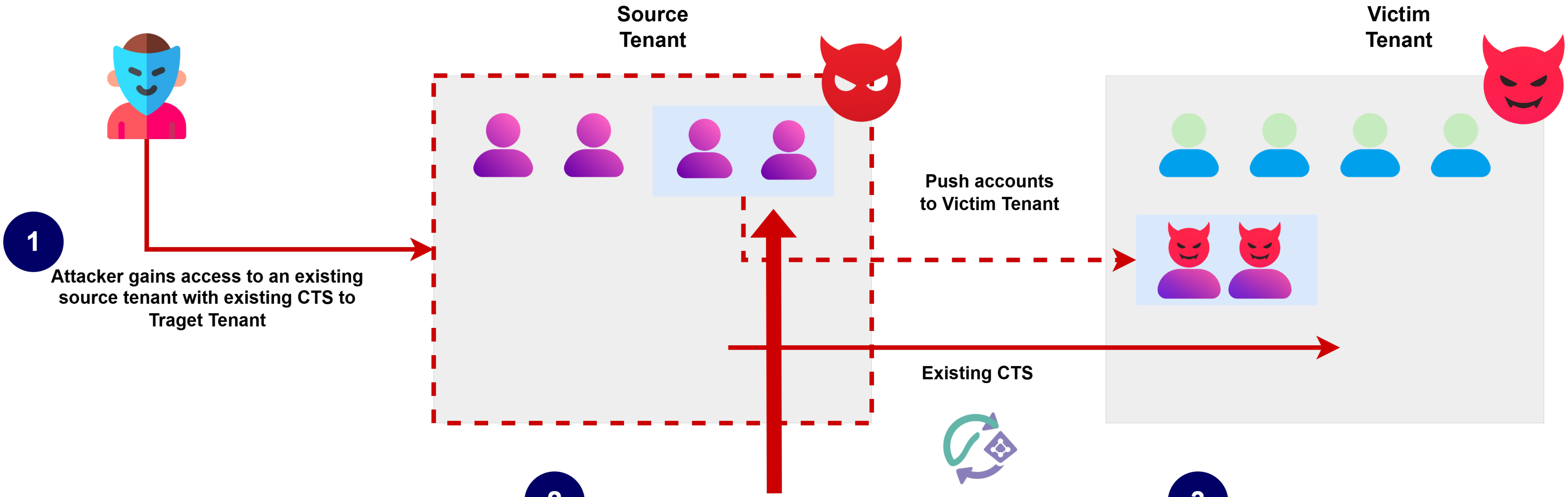


Cross Tenant Synchronisation

- CTS is EntraID feature that allows two separate EntraID tenants to synchronize users between each other


Source Tenant
Target Tenant
Push accounts to target Tenant
Define Scoping Filters and Attribute Mappings
Cross Tenant Synchironisation

Attack I – Lateral Movement using existing CTS



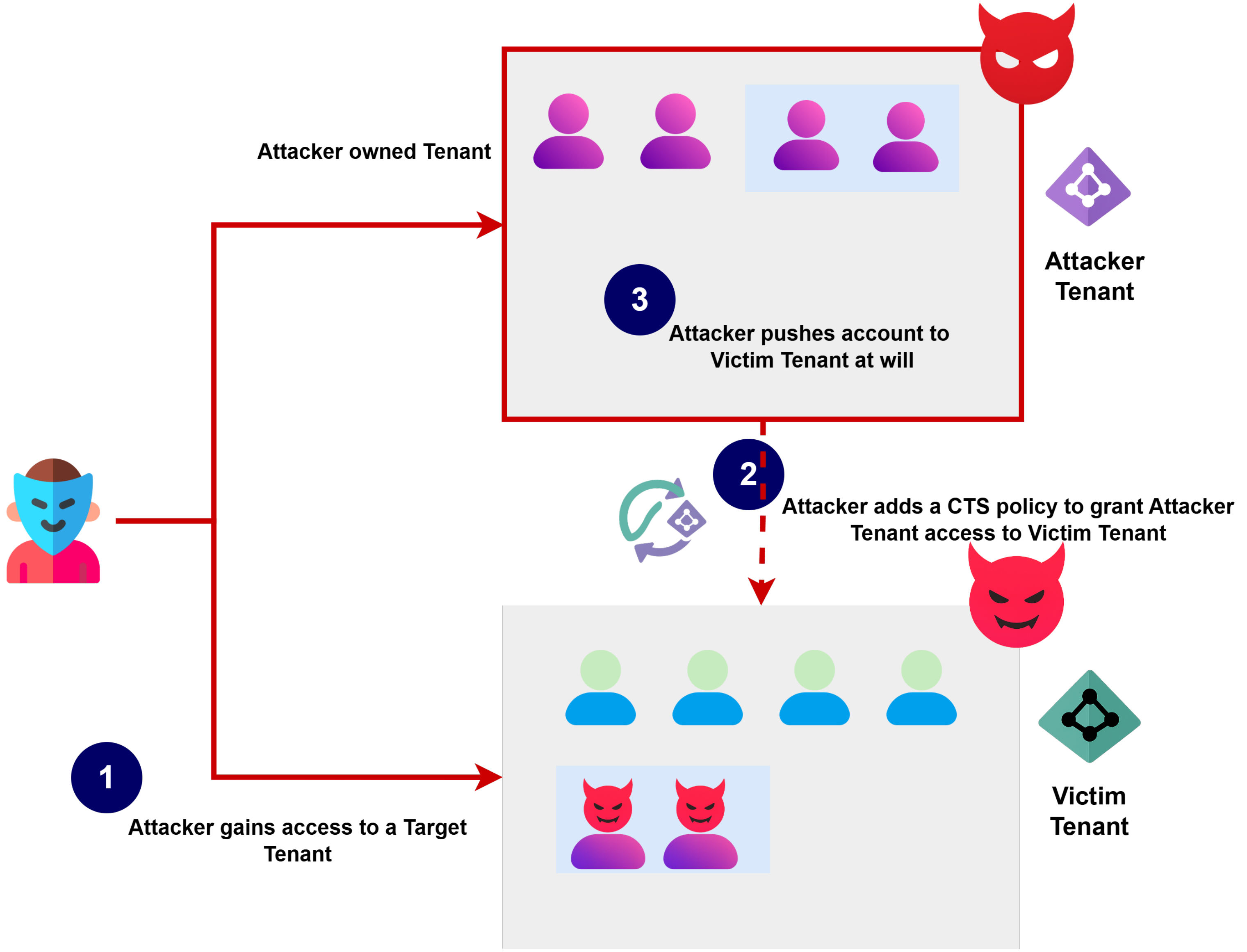
Attacker gains access to an existing source tenant with existing CTS to Traget Tenant

Attacker compromises an exisitng Synchornized user account and uses it to access Victim Tenant

Attacker adds an account that they control and sync it to the Victim Tenant

Attacker gains access to the Victim Tenant

Attack II – Backdoor in Victim Tenant



Attacker owned Tenant
Attacker Tenant
3
Attacker pushes account to Victim Tenant at will
2
Attacker adds a CTS policy to grant Attacker Tenant access to Victim Tenant
1
Attacker gains access to a Target Tenant
Victim Tenant

Abusing
Temporary Access Pass

Temporary Access Pass

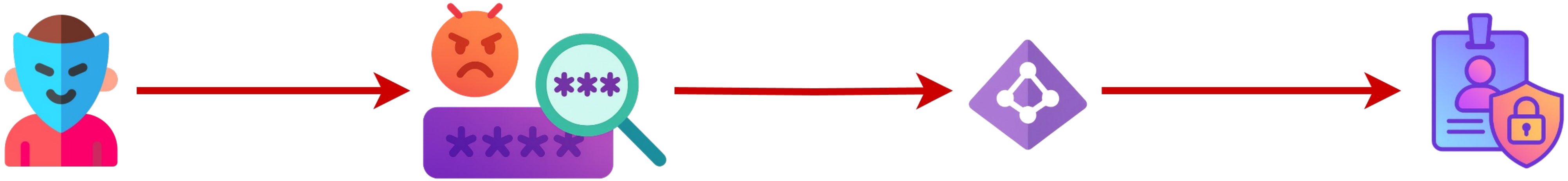
- Temporary Access Pass (TAP) is a time-limited passcode that can be configured for single use or multiple sign-ins
- Does not require changing the password of the user
- TAP can be used to bypass any Strong Authentication/MFA requirements, not enabled by default

An attacker who has gained access to the environment, could enable and generate TAP and use them as non-destructive alternate credentials, meaning attackers can use them without triggering password resets or MFA disruptions

Detection/Monitoring

- Monitor for EntraID Audit Logs for
  - OperationName =~ "Admin registered security info"
  - ResultDescription has "temporary access pass”
- Limit who can create a TAP and where it can be used, and enforcing a very short lifespan

Adding MFA to a dormant account



1. Attacker performs password spraying attack

2. Identify valid credentials

3. Account can login in EntraID and requests MFA registration

4. Attacker registers MFA for persistent access

Detection/Prevention Dormant Account

- **Disable/Remove Dormant Accounts:** Have a process to regularly review and disable accounts that haven't been used for a specified period
- **Conditional Access Policies:** Block sign-ins from unexpected locations or anonymous IP addresses.

SSPR – Self Service Password Reset

Abusing SSPR using SIM Swapping

- Transferring the user phone numbers to adversary controlled device

Registering Malicious Recovery Factors

- After initial compromise, attackers add **their own email or phone** as a recovery method.
- Even if defenders rotate passwords or reset MFA, attackers can regain access by using the new recovery method they enrolled.

Similar Attacks in AWS

Category	AWS Technique	Abuse Example
Cross-Account Role Assumption	IAM roles trusted across accounts	Attacker compromises Account A → assumes role into Account B via overly broad trust policy
AWS Organizations Delegated Admin	Delegated admin model across org	Compromise of delegated admin = control over all member accounts (policies, SCPs, role creation)
Third-Party SaaS Integrations	External partners assume roles	Compromised SaaS provider uses trust to access customer environments
Overly Broad Trust Policies	Wildcard principals or excessive permissions	"Principal": "*" allows unintended role assumption by attackers
Persistence via External Accounts	Attacker's own AWS account trusted	Malicious partner account retains long-term access even if local creds rotated

Thanks for listening!

Presentation deck available at rudrasec.io/talks

Anurag Khanna

X @khannaanurag

in www.linkedin.com/in/khannaanurag