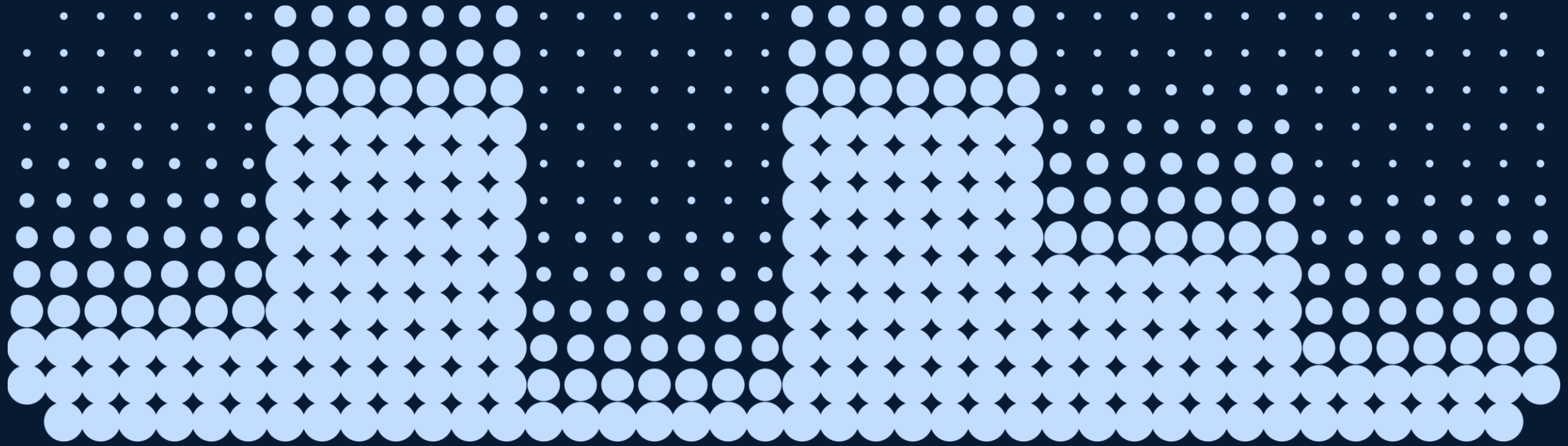


Review and clean up Applications with Excessive Permissions

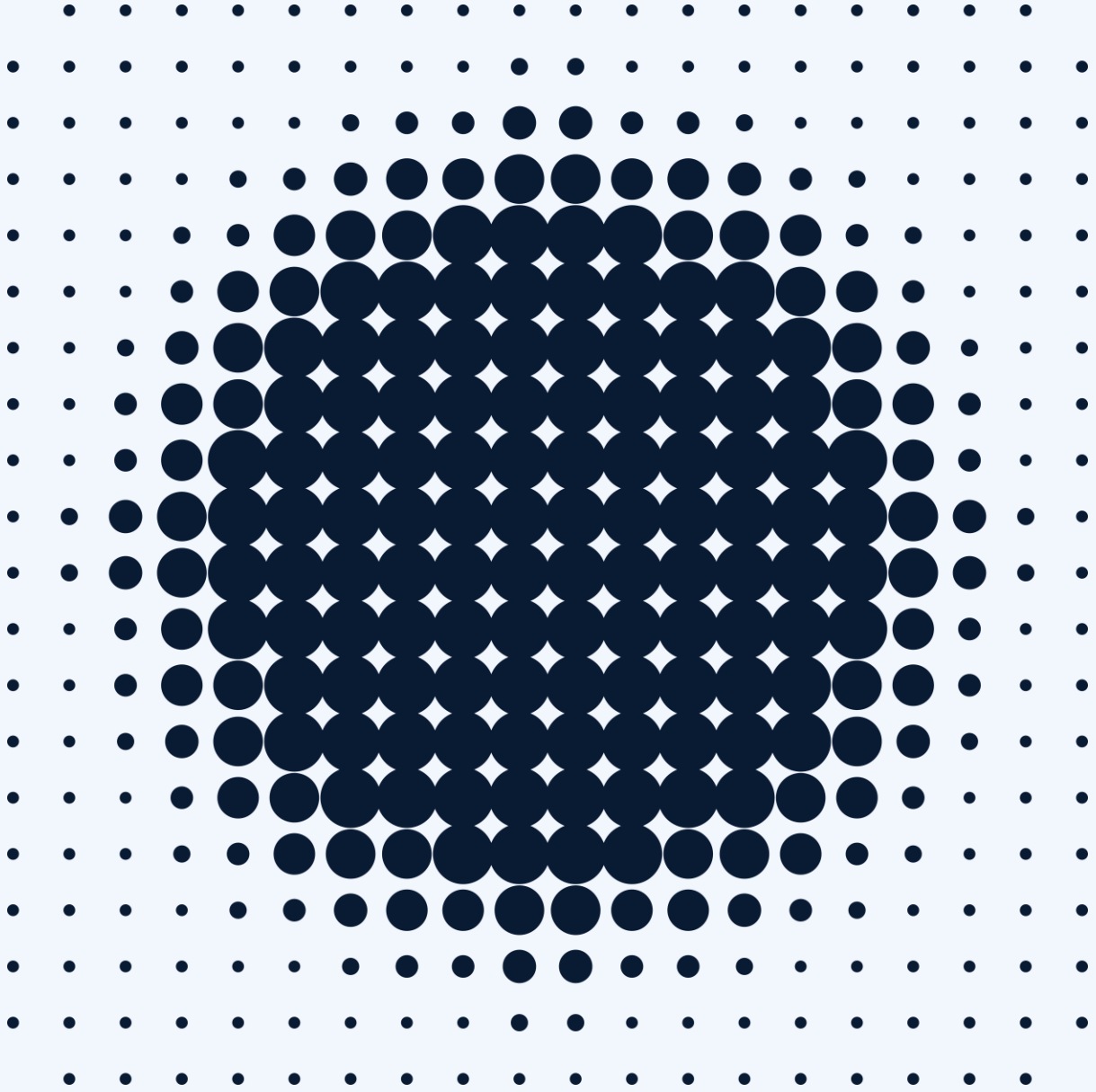
Recommended Actions

bsure.

October 31, 2025 Olav Helland

Agenda

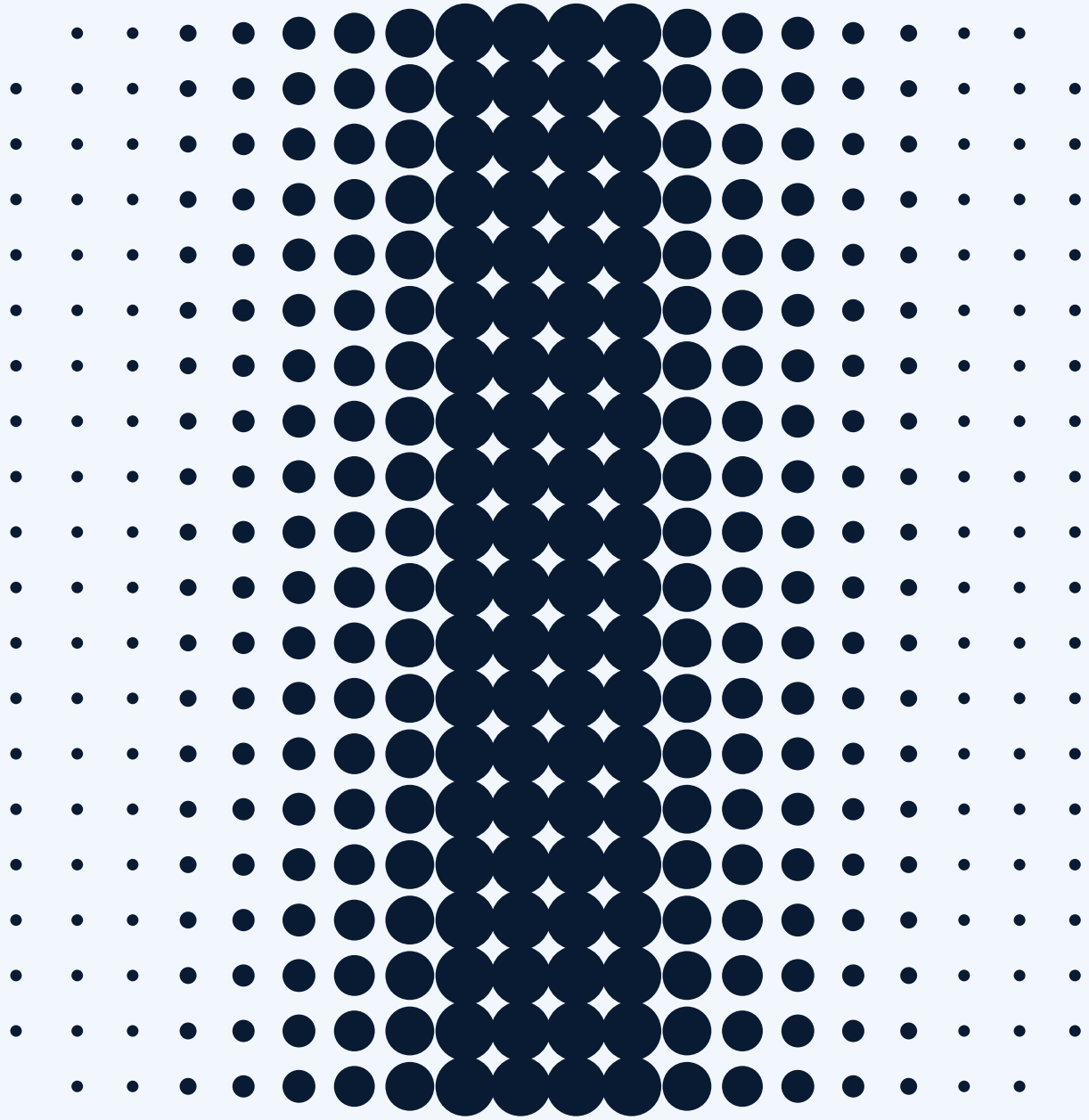
- Enterprise Apps with SSO
- Non-human identitites
- Permissions
- Threat landscape
- Demo: Bsure Applications reports
- Q&A



Enterprise Apps with SSO

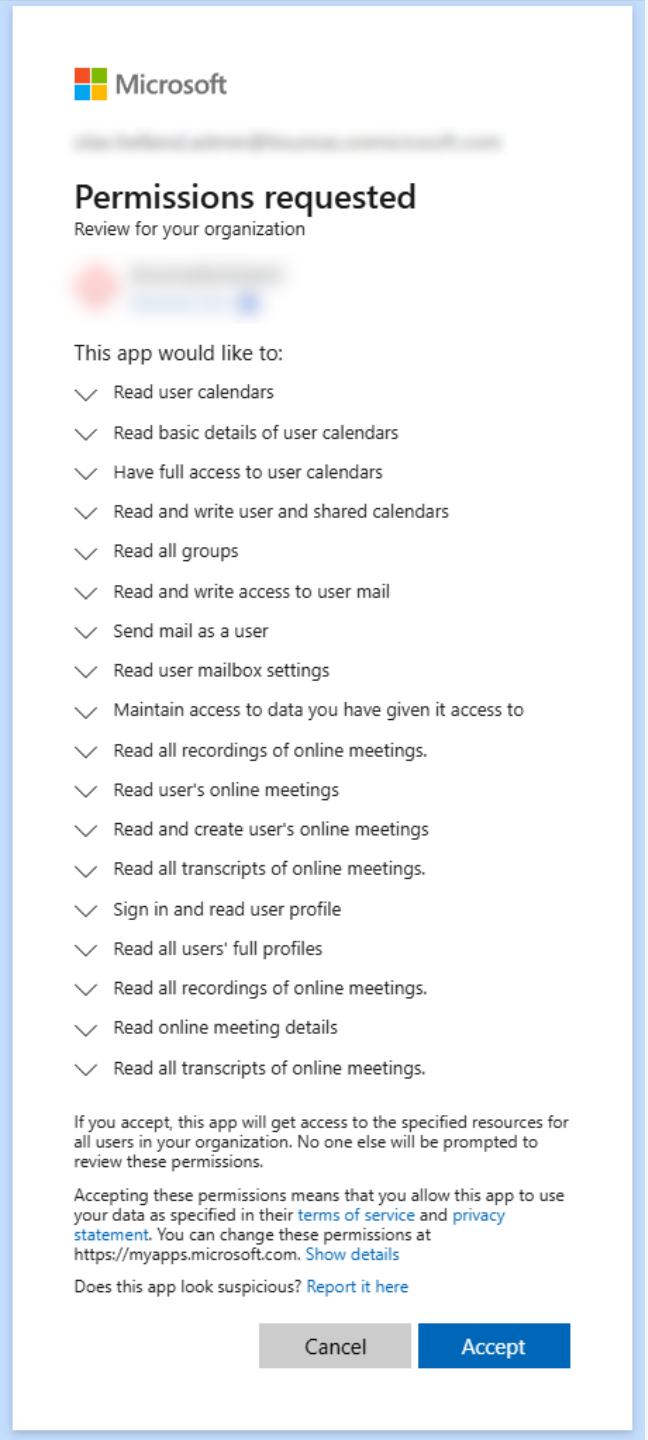
The benefits

- Centralized access and authentication
- Easier user management and offboarding
- Visibility and reporting
- Stronger security with Conditional Access and MFA
- Simplifies compliance and license control
- Seamless experience for users – one sign-in for all apps



Non-human identities

- Workload Identities
  - An umbrella term used by Microsoft for all non-human identities in Entra ID, including applications, services, and automated processes.
- App Registrations
  - The definition of an application, describing how it signs in and what permissions it needs.
- Enterprise Applications
  - The Service Principal instance of an App Registration, representing the actual app as it runs in your tenant.
- Service Principals
  - The identity that an application uses to sign in and access resources, like a user account for an app.
- Managed Identities
  - Special types of Service Principals automatically created and managed by Azure for resources such as VMs, Container Apps, and Functions, removing the need for secrets or credentials.



Permissions

Consent

- Do not allow user consent! Always require admin consent.

Delegated permissions

- The app acts on behalf of a signed-in user.
- The user's own permissions limit what the app can do.

Application permissons

- The app acts as itself, without a user signed in.

API name	Claim value	Permission	Type	Granted through
Microsoft Graph (22)				
Microsoft Graph	OnlineMeetings.Read.All	Read online meeting details	Application	Admin consent
Microsoft Graph	OnlineMeetingRecording.R...	Read all recordings of online meetings.	Application	Admin consent
Microsoft Graph	OnlineMeetingTranscript.R...	Read all transcripts of online meetings.	Application	Admin consent
Microsoft Graph	Calendars.Read	Read user calendars	Delegated	Admin consent
Microsoft Graph	Calendars.ReadBasic	Read basic details of user calendars	Delegated	Admin consent
Microsoft Graph	Calendars.ReadWrite	Have full access to user calendars	Delegated	Admin consent
Microsoft Graph	Calendars.ReadWrite.Shared	Read and write user and shared calendars	Delegated	Admin consent
Microsoft Graph	email	View users' email address	Delegated	Admin consent

Read and create user's online meetings

Allows the app to read and create online meetings on behalf of the signed-in user.

Read all transcripts of online meetings.

Allows the app to read all transcripts of all online meetings, without a signed-in user.

bsure.

From end users to workloads: The new horizon in identity threats

As phishing-resistant MFA and conditional access strengthen user defenses, attackers are pivoting to workload identities—apps, services, and scripts that access cloud resources. These non-human identities often hold elevated privileges but lack sufficient security controls, resulting in a growing blind spot that attackers are exploiting.

App consent phishing tricks users into granting malicious apps OAuth permissions, bypassing MFA and persisting beyond password resets. Key Vault pivoting involves compromising apps with access to secrets, enabling lateral movement and privilege escalation, often undetected. Microsoft has observed layered attacks that combine device code phishing and OAuth consent phishing, sometimes redirecting users to AiTM sites. Compromised identities are also used for internal phishing and lateral movement.

Identity protection must extend to every identity—including non-human identities—by verifying explicitly, enforcing least privilege, and assuming breach.

Learn more

https://aka.ms/identity-attack-techniques

Configure cryptographic key auto-rotation in Azure Key Vault | Microsoft learn (May 2025)

*

In the first half of 2025, identity-based attacks rose by 32%. This escalation may reflect adversaries' increasing use of AI to craft highly convincing social engineering lures—posing new challenges for detection and response at scale.

User impersonation tactics

User impersonation

As organizations move to technologies like phishing-resistant MFA which make the hacking or phishing of passwords exponentially more difficult, adversaries are being forced to use more sophisticated methods to compromise user accounts. These include:

- **Token theft**. Stealing a user's token after they've authenticated, meaning no password compromise is necessary.
- **Slow password spray.** Trying multiple passwords over an extended period to avoid detection.
- **Location proximity emulation.** Mimicking a legitimate user's location to bypass policies with geographical restrictions.
- **One-time code (OTC) intercept.** Tricking a user into generating an OTC and then intercepting it to authenticate.

Secret store compromise

A secret store is a secure, local vault that protects sensitive information—including API keys, passwords, tokens, and certificates—from unauthorized access, allowing only approved systems to retrieve them as necessary. While platforms like Microsoft Azure Key Vault, AWS Secret Manager, and HashiCorp Vault offer significant improvements over patchwork solutions of the 2010s, they've also become highly valuable targets.

Application impersonation and malicious applications abuse

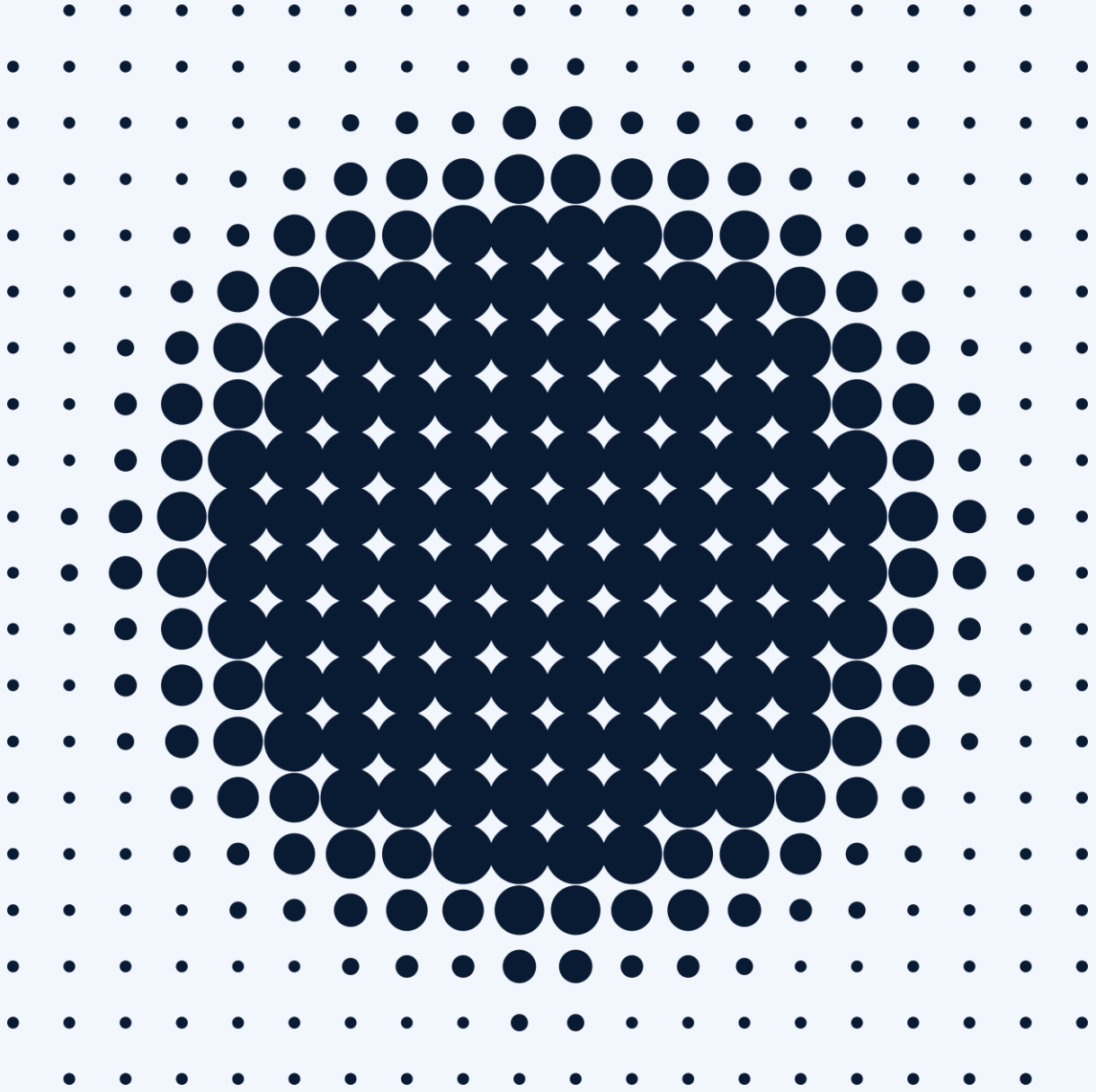
Attackers compromise applications and users with the same toolbox. Apps often have more permissions than they need—the exact elevated permissions that attackers seek. Another attack vector lures users into installing malicious apps and granting them broad permissions that the attacker can use until the user or the administrator explicitly revokes them. Application consent screens look legitimate and seem benign because they don't ask for credentials.

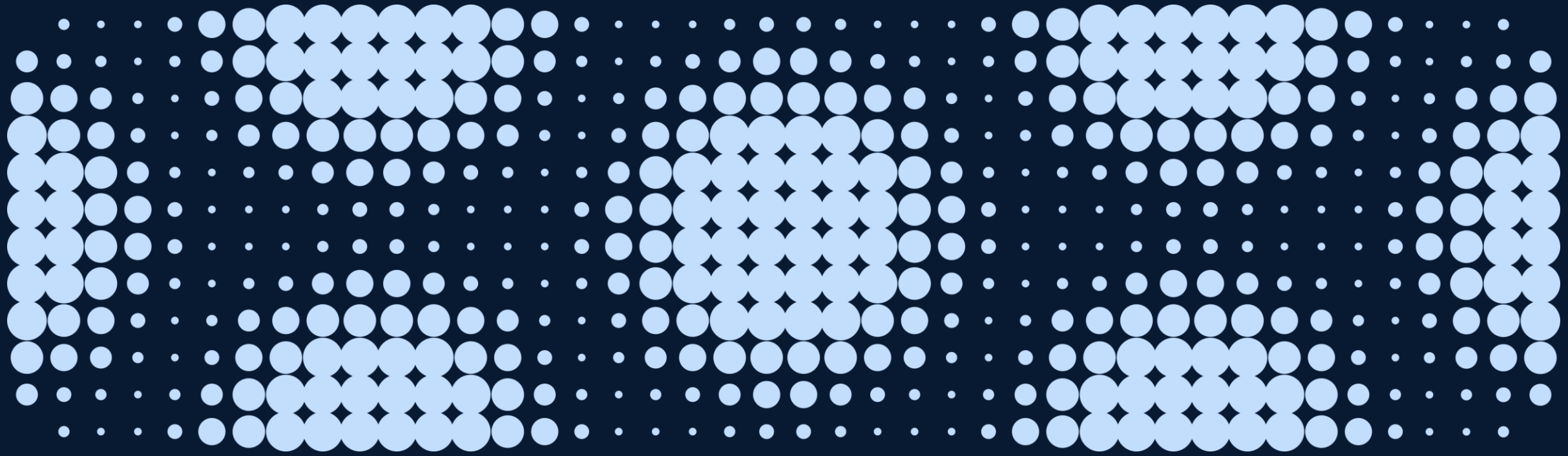
Authentication system impersonation

The most catastrophic scenario in identity security is the theft of a signing key, which compromises the trust and integrity of entire identity systems. A signing key is the private half of a public-private cryptography key pair used to encrypt and decrypt data. It signs messages so that systems can verify their authenticity using the pair's public key. With a captured signing key, attackers can impersonate the authentication system itself, forging credentials to gain access to protected resources and high-value data.

Demo

Bsure Insights Applications





Questions?

bsure.

support@bsure.io +47 64 80 82 22 bsure.io

Follow us