Connexion et Authentification NetSuite : Guide pour les Administrateurs et DAF

Publié le 19 mai 2025 50 min de lecture

Guide des méthodes de connexion et d'authentification NetSuite (Pour les Directeurs Financiers et les Administrateurs NetSuite)

Introduction

NetSuite offre une variété de méthodes d'authentification pour les utilisateurs et les systèmes, chacune équilibrant sécurité et commodité. En tant que Directeur Financier ou Administrateur NetSuite, il est important de comprendre comment les employés et les intégrations peuvent se connecter à NetSuite – de la connexion standard par nom d'utilisateur/mot de passe et l'authentification à deux facteurs, à l'authentification unique (SSO) avec des fournisseurs d'identité d'entreprise, jusqu'à l'accès basé sur des

jetons pour les API. Ce guide fournit un aperçu éducatif de toutes les méthodes de connexion NetSuite actuelles (y compris 2FA, SSO via SAML et OIDC, OAuth 2.0, authentification basée sur des jetons, etc.), discute des méthodes héritées (comme SuiteSignOn) et de leur statut de dépréciation, explique comment les intégrations (services web, SuiteTalk, RESTlets) s'authentifient, et décrit les meilleures pratiques pour une gestion sécurisée des accès. Des exemples concrets sont inclus pour illustrer comment les entreprises configurent l'authentification NetSuite en pratique. L'objectif est de donner aux Directeurs Financiers et aux administrateurs une compréhension claire et pratique des options d'authentification NetSuite – en évitant le jargon technique inutile tout en couvrant les détails essentiels.

Aperçu des méthodes de connexion NetSuite actuelles

NetSuite prend en charge plusieurs méthodes d'authentification pour accéder à l'interface utilisateur et aux API du système. Voici un aperçu des principaux mécanismes de connexion disponibles aujourd'hui :

- **Connexion standard par e-mail/mot de passe** – Les utilisateurs se connectent avec leur ID de compte NetSuite, leur adresse e-mail et leur mot de passe via la page de connexion NetSuite (ou l'application mobile) en utilisant des identifiants définis dans NetSuite (Source: docs.oracle.com). C'est la méthode de connexion traditionnelle, désormais souvent complétée par des mesures de sécurité supplémentaires comme la 2FA.

- **Authentification à deux facteurs (2FA)** – Une étape de vérification secondaire pour les connexions à l'interface utilisateur. Les utilisateurs doivent entrer un code à usage unique (par exemple, depuis une application d'authentification mobile) après avoir entré leur mot de passe (Source: docs.oracle.com). NetSuite **exige** la 2FA pour tous les rôles d'administrateur et autres rôles à privilèges élevés par défaut (Source: docs.oracle.com), et la 2FA peut être activée pour d'autres rôles si nécessaire afin de renforcer la sécurité.

- **Authentification unique (SSO) via SAML 2.0** – Intégration avec des fournisseurs d'identité externes (comme Okta, Azure AD, etc.) utilisant la norme SAML 2.0. Les employés peuvent s'authentifier via l'IdP central de l'entreprise et accéder à NetSuite sans mot de passe NetSuite séparé (Source: docs.oracle.com). NetSuite agit comme le fournisseur de services et fait confiance aux assertions SAML de l'IdP, permettant une expérience de connexion transparente.

- **Authentification unique via OpenID Connect (OIDC)** – Une méthode SSO moderne alternative (introduite dans NetSuite 2019.2) basée sur OAuth 2.0 et les jetons web JSON (Source: docs.oracle.com). OIDC permet à NetSuite d'accepter des jetons d'identité de fournisseurs comme Azure AD B2C, Okta, etc., et offre des avantages tels qu'une commutation de rôle plus facile entre plusieurs comptes NetSuite lorsque la même configuration OIDC est partagée (Source: docs.oracle.com).

- **Autorisation OAuth 2.0** – OAuth 2.0 est principalement utilisé pour les intégrations (pas pour les connexions interactives à l'interface utilisateur) et permet à une application externe d'obtenir un jeton pour accéder aux services web REST de NetSuite, aux RESTlets ou à SuiteAnalytics Connect au nom d'un utilisateur (Source: docs.oracle.com). Il élimine le besoin de stocker le mot de passe d'un utilisateur dans le code d'intégration et est désormais la méthode **préférée** pour l'authentification API (plus simple que la signature de jetons héritée) (Source: docs.oracle.com).

- **Authentification basée sur des jetons (TBA)** – Accès basé sur des jetons de NetSuite utilisant OAuth 1.0a. Une intégration peut utiliser une paire clé/secret de consommateur et ID/secret de jeton pour authentifier les appels API sans session de connexion utilisateur (Source: info.ennvee.com). La TBA a été introduite pour prendre en charge les intégrations API sécurisées et reste largement utilisée pour les services web SOAP, les RESTlets et les méthodes d'intégration plus anciennes (Source: docs.oracle.com)(Source: docs.oracle.com). (NetSuite recommande désormais OAuth 2.0 pour les nouvelles intégrations REST, mais la TBA est toujours entièrement prise en charge.)

- **Contrôles d'accès basés sur les rôles affectant la connexion** – Le système de rôles de NetSuite influence la manière dont les utilisateurs se connectent et l'accès qu'ils ont. Chaque utilisateur a un ou plusieurs rôles, et lors de la connexion, il opère sous les permissions d'un rôle spécifique. Certains rôles imposent des exigences de connexion supplémentaires (par exemple, les rôles **"2FA requise"** ne peuvent pas être utilisés sans authentification à deux facteurs) (Source: docs.oracle.com). Certains rôles peuvent être marqués comme rôles "**SAML Single Sign-on**" ou rôles "**OIDC SSO**", ce qui signifie que ces utilisateurs doivent se connecter via l'IdP SSO au lieu d'utiliser un mot de passe NetSuite (Source: docs.oracle.com)(Source: docs.oracle.com). De plus, un rôle "**Intégration**" ou "**Services Web Uniquement**" peut être utilisé pour restreindre un utilisateur à l'accès API uniquement (pas de connexion à l'interface utilisateur) pour des raisons de sécurité (Source: docs.oracle.com). Nous discuterons de ces contrôles basés sur les rôles en détail plus tard.

Chacune de ces méthodes répond à des cas d'utilisation différents – par exemple, les employés se connectant via un navigateur utiliseront généralement le SSO ou la 2FA, tandis qu'une intégration automatisée utilisera un jeton ou OAuth. Ensuite, nous approfondirons chaque méthode de connexion actuelle.

Connexion standard par nom d'utilisateur et mot de passe

La manière la plus basique de se connecter à NetSuite est d'utiliser un identifiant utilisateur spécifique au compte (généralement une adresse e-mail) et un mot de passe. Lors de l'accès à l'interface utilisateur de NetSuite via la page de connexion du navigateur, les utilisateurs saisissent leur **e-mail et mot de passe**, ainsi que l'**ID de compte** de l'instance NetSuite si nécessaire (NetSuite utilise des ID de compte uniques pour identifier le compte/l'instance client) (Source: docs.oracle.com)(Source: docs.oracle.com). Cette connexion standard accorde l'accès en fonction des permissions de rôle attribuées à l'utilisateur.

NetSuite applique par défaut des politiques de mot de passe robustes – les administrateurs peuvent configurer la longueur minimale, la complexité et les exigences de rotation des mots de passe conformément aux politiques de sécurité de l'entreprise (Source: netsuite.com). Il est important de noter que depuis 2018-2019, l'**utilisation d'un simple nom d'utilisateur/mot de passe n'est plus considérée comme suffisante pour un accès hautement privilégié**. NetSuite exige désormais la 2FA pour les administrateurs et les rôles similaires, ce qui signifie que même si le mot de passe correct est entré, un second facteur est nécessaire pour ces comptes (Source: docs.oracle.com). (Nous aborderons la 2FA dans la section suivante.)

Du point de vue de l'utilisateur final, le processus de connexion standard est simple mais nécessite la gestion d'un ensemble d'identifiants supplémentaire, à moins que le SSO ne soit implémenté. De nombreuses organisations choisissent d'implémenter l'authentification unique pour éviter que les utilisateurs n'aient besoin d'un mot de passe NetSuite séparé (améliorant la commodité et centralisant le contrôle de l'authentification), ce que nous abordons ci-dessous.

Authentification à deux facteurs (2FA)

L'authentification à deux facteurs ajoute une exigence de code de vérification à usage unique en plus des identifiants de connexion standard pour améliorer considérablement la sécurité. L'implémentation 2FA de NetSuite utilise des codes d'accès à usage unique basés sur le temps (TOTP) générés par une application d'authentification (ou obtenus via SMS/appel vocal en mode hérité) (Source: docs.oracle.com)

 https://technologyblog.rsmus.com/technologies/netsuite/two-factor-authentication-2fa-for-netsuite/

. Lorsque la 2FA est activée, un utilisateur doit entrer son e-mail et son mot de passe **puis saisir un code de vérification à 6 chiffres** depuis son application d'authentification (telle que Google Authenticator ou Microsoft Authenticator) pour chaque connexion (Source: docs.oracle.com). Le code est unique et expire après une courte période, donc même si un mot de passe est volé, un attaquant ne peut pas se connecter sans le second facteur.

NetSuite **impose la 2FA pour tous les rôles d'administrateur et autres rôles à privilèges élevés** dans tous les comptes (production, sandbox, etc.) (Source: docs.oracle.com). Ces rôles sont marqués comme "2FA requise" par défaut et cela ne peut pas être supprimé – c'est une exigence de sécurité intégrée. En fait, certaines permissions sensibles (comme l'accès aux données financières, les déploiements SuiteScript, etc.) marqueront automatiquement tout rôle personnalisé qui les contient comme nécessitant la 2FA (Source: docs.oracle.com). Le système fournit une page d'administration ("Rôles d'authentification à deux facteurs") où vous pouvez voir quels rôles nécessitent la 2FA et éventuellement activer la 2FA pour des rôles supplémentaires comme bonne pratique. Les administrateurs ont la flexibilité d'imposer la 2FA sur d'autres rôles (par exemple, tous les utilisateurs de la comptabilité) via cette page, et de définir la

durée du périphérique de confiance (à quelle fréquence la 2FA est demandée – de chaque connexion à tous les 30 jours sur un appareil) (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com).

Lorsqu'un utilisateur avec un rôle activé pour la 2FA se connecte, NetSuite demandera un second facteur. La **première fois** qu'un utilisateur se connecte avec la 2FA, il peut recevoir un code de vérification par e-mail pour confirmer son identité, puis sera guidé pour configurer sa méthode 2FA préférée (Source: technologyblog.rsmus.com). Les utilisateurs sont encouragés à utiliser une **application d'authentification** (conforme TOTP) comme méthode principale, ce qui est plus sécurisé et est l'option par défaut recommandée lors de la configuration. (Historiquement, NetSuite autorisait également les codes de vérification par SMS ou appel téléphonique comme alternative, mais Oracle a annoncé la fin du support de la 2FA via SMS/voix à partir de début 2024 (Source: withum.com), reflétant un passage aux applications d'authentification comme seule méthode à l'avenir.) Lors de la configuration de la 2FA, les utilisateurs reçoivent un ensemble de **codes de secours** – des codes à usage unique qui peuvent être sauvegardés au cas où le périphérique 2FA principal serait perdu ou indisponible (Source: docs.oracle.com).

Du point de vue de l'administrateur, la 2FA dans NetSuite présente plusieurs avantages : elle est incluse sans coût supplémentaire (aucune licence spéciale ni matériel de jeton requis) et les utilisateurs gèrent eux-mêmes leurs périphériques et codes 2FA via l'interface utilisateur de NetSuite (Source: docs.oracle.com). Une fois la 2FA imposée, les utilisateurs avec ces rôles ne seront même plus invités à répondre aux anciennes questions de sécurité lors de la connexion – le code 2FA remplace cette fonctionnalité par un niveau d'assurance beaucoup plus élevé (Source: docs.oracle.com). (Les questions de sécurité n'interviennent que si un utilisateur 2FA doit réinitialiser son mot de passe et n'a aucun autre moyen de vérifier son identité.) Il est important de noter que la **2FA est uniquement pour la connexion à l'interface utilisateur** – elle n'est **pas** utilisée pour les intégrations/services web en arrière-plan. En fait, si un rôle requiert la 2FA, vous ne pouvez pas utiliser les identifiants de ce rôle pour les services web SOAP ou les appels RESTlet (Source: docs.oracle.com)(Source: docs.oracle.com). Ces scénarios non-UI doivent utiliser l'authentification par jeton ou OAuth (discutés plus tard), c'est pourquoi NetSuite a introduit l'authentification par jeton qui peut coexister avec les politiques 2FA (Source: info.ennvee.com) (Source: docs.oracle.com).

En résumé, la 2FA est une couche de défense critique pour l'accès à NetSuite. Les Directeurs Financiers doivent savoir que NetSuite les oblige probablement déjà à utiliser la 2FA s'ils ont un rôle d'administrateur ou un autre rôle privilégié – ce qui signifie que vous utiliserez une application d'authentification pour obtenir un code à chaque fois (ou au moins tous les 30 jours, selon les paramètres). Cela réduit considérablement le risque de compromission de compte, protégeant les données financières sensibles contre les accès non autorisés (Source: docs.oracle.com). Les administrateurs NetSuite doivent s'assurer

que tous les rôles applicables ont la 2FA imposée et éduquer les utilisateurs sur le processus de configuration simple. La légère étape supplémentaire lors de la connexion vaut bien le gain de sécurité pour la protection de votre ERP.

Authentification unique (SSO) via SAML 2.0

L'authentification unique permet aux utilisateurs de s'authentifier via un fournisseur d'identité externe (IdP) – tel qu'Okta, Microsoft Entra ID (Azure AD), OneLogin, Ping Identity, etc. – puis d'accéder à NetSuite sans saisir d'identifiants NetSuite séparés. NetSuite prend en charge le SSO en utilisant la norme **SAML 2.0**, largement utilisée dans les systèmes d'identité d'entreprise. Dans une intégration SAML SSO, l'IdP gère la vérification de l'utilisateur (par exemple, via la connexion réseau de l'entreprise, les identifiants de domaine AD ou l'authentification multi-facteurs), puis transmet une **assertion** sécurisée à NetSuite indiquant l'identité de l'utilisateur et qu'il est authentifié (Source: docs.oracle.com). NetSuite fait confiance à cette assertion et connecte l'utilisateur, le dirigeant vers sa page d'accueil NetSuite sans demander de mot de passe (Source: docs.oracle.com).

Du point de vue de l'utilisateur, le SAML SSO signifie qu'il peut se connecter à un portail central (par exemple, un tableau de bord Okta ou Microsoft 365) et cliquer sur l'icône NetSuite, ou qu'il visite l'URL NetSuite et est redirigé vers la page de connexion de son entreprise. Après avoir entré ses identifiants d'entreprise habituels (et toute MFA requise à ce niveau), il arrive de manière transparente dans NetSuite. Cela améliore la commodité (un mot de passe de moins à gérer) et donne au service informatique de l'entreprise un contrôle centralisé sur les politiques d'authentification (la complexité des mots de passe, les heures de connexion, la MFA, etc., sont appliquées par l'IdP). Pour les Directeurs Financiers, l'adoption du SSO peut simplifier le processus de connexion et garantir que si, par exemple, un employé quitte l'entreprise, la désactivation de son compte dans l'annuaire central coupe également immédiatement l'accès à NetSuite.

La configuration du SAML SSO dans NetSuite implique l'activation de la fonctionnalité et l'échange de métadonnées entre NetSuite et l'IdP. Les administrateurs doivent activer la fonctionnalité **SAML Single Sign-on** dans NetSuite (sous Configuration > Société > Activer les fonctionnalités > SuiteCloud) puis configurer les paramètres SAML dans NetSuite avec les métadonnées de l'IdP (certificats, URL SSO, ID d'entité, etc.) (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com). Ils doivent également attribuer la permission "SAML Single Sign-on" à tout rôle qui devrait être autorisé à utiliser le SSO (Source: technologyblog.rsmus.com) – seuls les utilisateurs ayant un rôle avec cette permission peuvent se connecter via SAML. Généralement, un administrateur créerait ou personnaliserait des rôles pour l'utilisation du SSO (par exemple, vous pourriez avoir un rôle "Centre Employé (SSO)"). Les utilisateurs se voient ensuite attribuer ces rôles activés pour le SSO. NetSuite permet un contrôle granulaire – vous pourriez exiger que certains rôles utilisent le SAML SSO tandis que d'autres (peut-être des contractuels externes) se connectent toujours via mot de passe.

Une note importante : Si un rôle est marqué pour l'authentification unique SAML, cette exigence **prime sur l'authentification à deux facteurs (2FA)** pour ce rôle (Source: docs.oracle.com). En pratique, cela signifie que NetSuite ne demandera pas de code 2FA pour une connexion basée sur SAML ; il suppose que l'authentification de l'IdP (qui peut elle-même inclure la MFA) est suffisante. Cela évite un conflit entre deux systèmes MFA différents. Les entreprises appliquent souvent la MFA dans leur IdP SAML (par exemple, Okta Verify ou Microsoft Authenticator pour tous les accès aux applications), de sorte que NetSuite fait confiance à l'assertion SAML et ne demande pas deux fois la 2FA dans ces cas (Source: docs.oracle.com).

L'authentification unique via SAML est largement utilisée par les clients de NetSuite. Par exemple, de nombreuses organisations utilisent **Okta comme IdP pour NetSuite** – les employés se connectent à Okta, qui gère ensuite le transfert SAML vers NetSuite (Source: technologyblog.rsmus.com). Le résultat est un flux de connexion simplifié : les utilisateurs cliquent sur une tuile NetSuite et sont dans le système sans avoir à taper un mot de passe distinct (et les politiques de sécurité de l'organisation sur Okta s'appliquent à l'accès à NetSuite). Du point de vue du retour sur investissement, l'authentification unique peut réduire les appels au service d'assistance pour les mots de passe oubliés et améliorer la conformité en centralisant les journaux d'audit de l'activité de connexion des utilisateurs.

Authentification unique via OpenID Connect (OIDC)

OpenID Connect est une autre méthode d'authentification unique prise en charge par NetSuite, offrant des avantages similaires à SAML mais utilisant un protocole plus moderne basé sur OAuth2 et des jetons web JSON. La fonctionnalité **OIDC Single Sign-on** de NetSuite (ajoutée en 2019.2) permet à NetSuite d'agir comme un **client** OAuth2 (partie de confiance) qui fait confiance à un fournisseur d'identité OpenID Connect externe (Source: docs.oracle.com). En termes plus simples, NetSuite peut accepter une connexion utilisateur via des jetons OIDC émis par des fournisseurs comme Azure AD, Okta (mode OIDC), Google, etc.

La configuration de l'authentification unique OIDC est conceptuellement similaire à celle de SAML : l'administrateur active la fonctionnalité d'authentification unique OpenID Connect dans NetSuite, enregistre NetSuite comme application dans l'IdP (vous obtenez un ID client/secret et des URL de redirection), et configure les paramètres OIDC dans NetSuite (URL de l'émetteur, ID client, étendues, etc.) (Source: docs.oracle.com)(Source: docs.oracle.com). Les rôles qui doivent utiliser l'authentification unique OIDC nécessitent l'attribution de la permission "OpenID Connect Single Sign-on" (Source: docs.oracle.com), de manière analogue à la permission SAML. Une fois configurés, les utilisateurs auront la possibilité de se connecter via OIDC. L'expérience utilisateur est à nouveau qu'ils s'authentifient auprès du fournisseur externe, puis sont redirigés vers NetSuite sans utiliser de mot de passe spécifique à NetSuite.

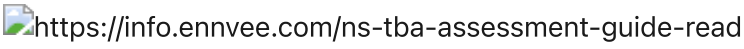
Pourquoi OIDC alors que SAML existe déjà ? OIDC est basé sur OAuth2 et utilise des charges utiles JSON, ce que certains trouvent plus facile à implémenter et plus flexible (par exemple, OIDC peut permettre des flux de **déconnexion basés sur des jetons** et une délégation d'authentification au niveau de l'API). Un avantage pratique noté est que si une entreprise a **plusieurs comptes NetSuite** (par exemple, un compte de production et un sandbox, ou plusieurs filiales avec des comptes séparés), et qu'elle configure le même IdP OIDC pour tous, un utilisateur peut basculer entre les rôles de ces comptes *sans avoir besoin de se reconnecter* à chaque fois (Source: docs.oracle.com). L'interface utilisateur de NetSuite permettra un basculement transparent de rôle/compte tant que la session OIDC est valide. C'est un avantage pour les administrateurs ou les directeurs financiers qui travaillent dans plusieurs environnements.

NetSuite recommande OIDC ou SAML comme alternatives aux méthodes d'authentification unique propriétaires plus anciennes qui ont été dépréciées (plus de détails à ce sujet dans la section héritée). Le choix d'une organisation entre SAML et OIDC dépend souvent des capacités ou des préférences de son IdP – les deux atteignent l'objectif de centralisation de la connexion. Le point clé à retenir est que NetSuite prend entièrement en charge les normes d'authentification unique modernes, vous offrant la flexibilité de l'intégrer à votre infrastructure d'authentification unique existante pour une expérience de connexion utilisateur fluide et sécurisée (Source: docs.oracle.com).

Authentification basée sur les jetons (Jetons d'accès)

Pour les intégrations et les scripts automatisés, NetSuite propose l'**Authentification Basée sur les Jetons (TBA)**, qui émet des jetons d'accès persistants pouvant être utilisés à la place d'un nom d'utilisateur/mot de passe. La TBA est basée sur OAuth 1.0a et utilise un **ID de jeton/secret** (lié à un utilisateur et un rôle NetSuite spécifiques) ainsi qu'une clé/secret de consommateur (provenant d'un enregistrement d'intégration) pour signer les requêtes API (Source: docs.oracle.com)(Source: docs.oracle.com). En pratique, une fois la TBA configurée, une intégration externe (comme un CRM, un site e-commerce ou un middleware) peut appeler les API SOAP ou REST de NetSuite en incluant un en-tête d'autorisation avec les identifiants du jeton (souvent appelé en-tête NLAuth ou OAuth1), au lieu de devoir gérer des connexions interactives ou stocker le mot de passe d'un utilisateur.

Comment configurer la TBA : Un administrateur active d'abord la fonctionnalité "Authentification basée sur les jetons" dans NetSuite (sous Configuration > Société > Activer les fonctionnalités > Gérer l'authentification)



. Ensuite, un **enregistrement d'intégration** est créé (Configuration > Intégration > Gérer les intégrations) pour enregistrer l'application externe – cela génère une clé et un secret de consommateur uniques à cette application. Ensuite, l'administrateur s'assure que le compte utilisateur qui sera utilisé

pour l'intégration a un rôle avec les permissions appropriées (notamment le rôle a besoin de la permission "Jetons d'accès utilisateur" pour permettre la création de jetons, et toutes les permissions de données requises par l'intégration) (Source: info.ennvee.com). Ce rôle peut être un "rôle d'intégration" dédié avec un accès limité. L'administrateur (ou l'utilisateur) génère ensuite un **ID de jeton et un secret de jeton** pour la combinaison utilisateur+rôle et intégration spécifique (cela se fait sous **Jetons d'accès** dans l'interface utilisateur de NetSuite, en sélectionnant l'application, l'utilisateur et le rôle) (Source: docs.oracle.com)(Source: docs.oracle.com). NetSuite produira un ID de jeton et un secret de jeton, chacun étant une longue chaîne, qui doivent être copiés immédiatement – ils ne sont affichés qu'une seule fois (Source: docs.oracle.com). Ces identifiants (clé/secret de consommateur + ID/secret de jeton) sont ensuite configurés dans le code ou la configuration de l'intégration externe. À partir de ce moment, l'intégration peut s'authentifier en signant ses requêtes avec ces jetons.

Un grand avantage de la TBA est qu'elle **découple l'authentification de l'intégration du mot de passe de connexion interactif de tout individu**. L'accès de l'intégration ne se rompt pas si un utilisateur change son mot de passe ou quitte l'entreprise, et vous n'avez pas à coder en dur les mots de passe. Elle est également immunisée contre les exigences de la 2FA – les jetons TBA peuvent être utilisés même si le rôle de l'utilisateur exige la 2FA pour la connexion à l'interface utilisateur (Source: info.ennvee.com). En fait, la TBA a été introduite en partie pour **permettre aux intégrations API de continuer à fonctionner dans un environnement où la 2FA et l'authentification unique sont appliquées pour l'interface utilisateur**(Source: info.ennvee.com)(Source: info.ennvee.com). Par exemple, à partir de la version 2018.2, NetSuite a imposé la 2FA pour tous les rôles d'administrateur, ce qui aurait rompu toute intégration utilisant les identifiants de l'administrateur – la solution a été d'utiliser des jetons TBA pour ce rôle d'administrateur (pris en charge depuis 2018.1) afin que l'intégration puisse s'authentifier via un jeton malgré la 2FA (Source: info.ennvee.com). La TBA **contourne** essentiellement **les politiques centrées sur l'interface utilisateur** mais de manière contrôlée, puisque les jetons sont accordés par un administrateur et peuvent être révoqués à tout moment sans affecter la connexion normale de l'utilisateur.

Autre avantage : les jetons TBA n'expirent pas à moins d'être révoqués, et ne sont pas soumis aux politiques de rotation des mots de passe (Source: info.ennvee.com). Cela réduit les frais de maintenance (pas besoin de mettre à jour régulièrement les identifiants stockés de l'intégration). Cependant, les administrateurs doivent examiner périodiquement les jetons actifs et révoquer ceux qui ne sont plus nécessaires, comme mesure d'hygiène de sécurité.

L'authentification basée sur les jetons de NetSuite est considérée comme un **mécanisme d'authentification unique entrante pour les intégrations**(Source: info.ennvee.com). C'est effectivement une authentification unique au niveau de l'API. Elle est bien prise en charge par l'API NetSuite SuiteTalk (SOAP) et par les RESTlets. À partir de 2020+, NetSuite a même permis à certaines

technologies d'intégration plus récentes (comme les services web REST et SuiteAnalytics Connect ODBC) de s'authentifier via des jetons ou OAuth2 au lieu de mots de passe (Source: docs.oracle.com) (Source: docs.oracle.com).

En résumé, les **TBA (Jetons d'accès)** sont la méthode privilégiée pour authentifier les systèmes externes se connectant à NetSuite. Elles améliorent la sécurité (pas de partage de mot de passe, fonctionne avec la 2FA) (Source: info.ennvee.com) et sont désormais requises dans de nombreux cas (NetSuite a déprécié l'ancienne pratique consistant à utiliser des identifiants utilisateur bruts pour les connexions API, décrite dans la section héritée). Les directeurs financiers pourraient entendre parler d'"intégration basée sur les jetons" lors de discussions avec l'informatique – cela signifie simplement que les intégrations utilisent ces jetons sécurisés plutôt que le mot de passe de quelqu'un. Il est fortement recommandé d'utiliser la TBA ou OAuth pour toute intégration personnalisée à NetSuite.

OAuth 2.0 pour les intégrations REST

NetSuite a introduit la prise en charge d'**OAuth 2.0** pour offrir un moyen plus standard et simplifié aux intégrations de s'authentifier, en particulier avec les services basés sur REST. Avec OAuth 2.0, les applications tierces peuvent obtenir un **jeton d'accès** de NetSuite via un flux d'autorisation, puis utiliser ce jeton (un jeton porteur) pour effectuer des appels API aux services web REST ou aux RESTlets de NetSuite (Source: docs.oracle.com). Contrairement à OAuth1/TBA, il n'est pas nécessaire de signer chaque requête avec une signature de jeton ; le jeton OAuth2 lui-même est présenté pour autoriser les requêtes, ce qui facilite l'implémentation pour les développeurs.

Points clés concernant l'implémentation d'OAuth 2.0 par NetSuite :

- **Types d'intégration pris en charge :** OAuth 2.0 peut être utilisé pour les services web REST SuiteTalk (l'API REST Records), pour les appels RESTlet, et même pour l'accès SuiteAnalytics Connect (ODBC/JDBC) (Source: docs.oracle.com). Il **n'est pas pris en charge pour les services web SOAP** (SOAP utilise toujours la TBA) (Source: docs.oracle.com)(Source: docs.oracle.com).
- **Flux :** NetSuite prend en charge à la fois le **flux d'octroi de code d'autorisation** (pour l'accès délégué par l'utilisateur via un consentement interactif) et le **flux d'octroi d'identifiants client** (pour les intégrations directes de serveur à serveur) (Source: docs.oracle.com)(Source: docs.oracle.com). Le flux de code d'autorisation est utile si une application a besoin de la permission d'un utilisateur NetSuite spécifique (il redirigera l'utilisateur vers la page de connexion/consentement de NetSuite). Le flux d'identifiants client permet à une intégration serveur d'échanger directement des identifiants (ID client/secret) contre un jeton sans interaction utilisateur – adapté aux intégrations en arrière-plan fonctionnant sous un utilisateur d'intégration fixe (Source: docs.oracle.com).

- **Configuration :** Pour utiliser OAuth2, un administrateur doit créer un **enregistrement d'application d'intégration** dans NetSuite et y activer l'option "OAuth 2.0" (similaire à la façon dont la TBA nécessite l'activation sur l'enregistrement d'intégration). Cela fournit un ID client et un secret. L'administrateur doit également attribuer des **permissions d'intégration REST** au rôle utilisé, telles que "Services web REST" et/ou les permissions RESTlet appropriées, et surtout le rôle doit avoir la permission "Jetons d'accès utilisateur" s'il utilise le flux de code d'autorisation (pour permettre à cet utilisateur d'autoriser le jeton). Pour le flux d'identifiants client, l'enregistrement d'intégration lui-même doit être configuré pour l'autoriser et est généralement lié à un utilisateur d'intégration.

- **Sécurité et commodité :** OAuth 2.0 est désormais considéré comme la **"méthode d'authentification préférée"** de NetSuite pour les intégrations REST (Source: docs.oracle.com). La documentation d'Oracle indique explicitement que vous devez utiliser OAuth 2.0 au lieu de la TBA chaque fois que possible (Source: docs.oracle.com). Cela est en partie dû au fait qu'OAuth2 est une approche standard de l'industrie que de nombreux développeurs et outils prennent en charge nativement (par exemple, si vous créez une application mobile ou utilisez une plateforme d'intégration qui prend en charge le rafraîchissement de jetons OAuth2, etc.), et cela élimine la complexité de la génération et de la gestion des signatures HMAC-SHA1 pour chaque requête, comme l'exige OAuth1. De plus, les jetons OAuth2 peuvent être délimités et avoir une durée de vie limitée, ce qui renforce la sécurité (les jetons TBA, en revanche, n'expirent pas à moins d'être révoqués manuellement).

- **Pas de connexion à l'interface utilisateur nécessaire :** Avec OAuth 2.0, un système externe n'a pas du tout besoin d'un nom d'utilisateur/mot de passe NetSuite – il a juste besoin du jeton. Les utilisateurs peuvent explicitement accorder ou révoquer des applications OAuth dans NetSuite (il existe une interface de gestion pour les applications autorisées OAuth 2.0) (Source: docs.oracle.com), donnant aux administrateurs une visibilité et un contrôle sur les applications externes qui ont accès.

Pour illustrer, imaginez une application tierce de gestion des dépenses qui doit extraire des données de NetSuite via l'API REST. En utilisant OAuth 2.0, l'application peut soit inviter un administrateur une fois à autoriser l'accès (flux de code d'autorisation, où l'administrateur se connecte et approuve la connexion, générant un jeton lié à son rôle), soit utiliser un utilisateur d'intégration préconfiguré et des identifiants client pour obtenir un jeton. L'application stocke ensuite le jeton et l'utilise pour récupérer ou envoyer des données à NetSuite, sans jamais manipuler un mot de passe NetSuite. Le jeton OAuth peut être révoqué par un administrateur à tout moment via l'interface utilisateur de NetSuite, et il a probablement une expiration fixe nécessitant un rafraîchissement périodique – toutes de bonnes pratiques de sécurité.

Pour les directeurs financiers et les administrateurs, le point principal est que **OAuth 2.0 et l'authentification basée sur les jetons sont les méthodes à utiliser pour les intégrations** – elles protègent et séparent vos identifiants NetSuite. Si une intégration ou un fournisseur demande un nom

d'utilisateur et un mot de passe NetSuite, c'est un signal d'alarme de nos jours ; l'intégration devrait plutôt utiliser un jeton ou une autorisation d'application OAuth, que vous pouvez provisionner en toute sécurité. Le passage de NetSuite à OAuth 2.0 (et l'élimination progressive de l'authentification de base) s'aligne sur les tendances plus larges de l'industrie visant à renforcer la sécurité (Source: docs.oracle.com)(Source: docs.oracle.com).

Contrôle d'accès basé sur les rôles et implications de connexion

Le modèle de permissions basé sur les rôles de NetSuite ne régit pas seulement les données/actions qu'un utilisateur peut effectuer une fois connecté, mais il affecte également *comment* les utilisateurs peuvent se connecter. Voici quelques considérations de connexion liées aux rôles à prendre en compte :

- **Exigences de 2FA par rôle :** Comme discuté, certains rôles sont désignés "Authentification à deux facteurs requise" par défaut (Administrateur et autres rôles très sensibles) (Source: docs.oracle.com). Si un rôle est marqué comme nécessitant la 2FA, toute connexion à NetSuite avec ce rôle doit passer par la 2FA. Si un utilisateur tente d'utiliser les identifiants de ce rôle pour une intégration (par exemple, des services web SOAP utilisant uniquement l'e-mail/mot de passe), l'authentification sera **bloquée** – NetSuite n'autorisera pas une connexion API pour un rôle nécessitant la 2FA en utilisant uniquement les identifiants utilisateur (Source: docs.oracle.com). C'est pourquoi l'authentification par jeton est obligatoire pour les intégrations impliquant de tels rôles. Les administrateurs ne peuvent pas supprimer l'indicateur 2FA des rôles par défaut, mais ils peuvent étendre la 2FA à des rôles supplémentaires en tant que décision de politique (par exemple, exiger la 2FA pour tous les rôles pouvant approuver des paiements, etc.). La page "Rôles d'authentification à deux facteurs" dans NetSuite est l'endroit où résident ces paramètres (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com).

- **Rôles réservés au SSO :** De même, lors de l'utilisation du SSO, vous contrôlez les rôles qui peuvent utiliser le SSO SAML ou OIDC en accordant la permission appropriée. Les utilisateurs se connectant via le SSO n'auront accès qu'aux rôles pour lesquels le SSO a été activé. Si un rôle ne dispose pas de la permission SSO, l'utilisateur ne pourra pas accéder à NetSuite avec ce rôle via la connexion SSO (il devrait se connecter en utilisant les identifiants NetSuite pour ce rôle, si autorisé). Cela peut être utilisé pour s'assurer que certains rôles ne sont utilisés que dans certains contextes. Par exemple, vous pourriez *exiger* que vos employés utilisent le SSO pour leurs rôles principaux (afin que toutes les connexions passent par votre IdP), mais vous pourriez conserver un rôle d'« Administrateur d'urgence » basé sur un mot de passe au cas où votre IdP serait hors service – ce rôle n'aurait intentionnellement pas la permission SAML. Notez que si un rôle exige à la fois le SSO et la 2FA, la connexion SSO contournera l'invite 2FA dans NetSuite (Source: docs.oracle.com) (l'hypothèse étant que l'IdP gère la MFA).

- **Rôles multiples et sélection de rôle :** NetSuite permet d'attribuer plusieurs rôles aux utilisateurs. Lors d'une connexion interactive, si un utilisateur a plus d'un rôle, il peut basculer entre les rôles *après* s'être connecté (via une liste déroulante de sélection de rôle dans l'interface utilisateur). Cependant, il est important de comprendre que l'authentification de connexion est liée à un seul rôle à la fois. Par exemple, si un utilisateur a un rôle « Employé » et un rôle « Administrateur », et que seul le rôle Administrateur exige la 2FA, lorsqu'il se connecte en tant qu'Administrateur, il effectuera la 2FA, mais s'il passe au rôle Employé, la session est toujours considérée comme authentifiée (il ne se reconnecte pas, il change juste de contexte). Avec le SSO SAML, un IdP peut en fait spécifier le rôle dans lequel l'utilisateur doit se connecter par défaut via les attributs d'assertion SAML (Source: docs.oracle.com). En pratique, les entreprises utilisant le SSO gèrent souvent l'attribution des rôles avec soin – en attribuant éventuellement à chaque utilisateur un seul rôle principal pour éviter toute confusion, ou en les guidant sur la manière de changer de rôle dans NetSuite si nécessaire. L'interface affichera « Actuellement connecté en tant que [Rôle] » et permettra de changer de rôle si d'autres sont disponibles (sous réserve de restrictions telles que « Services Web uniquement »). **Figure :** La capture d'écran ci-dessous montre un exemple d'utilisateur invité à saisir un code de vérification 2FA lors de la connexion, avec la possibilité de choisir un rôle alternatif si disponible. Notez que le rôle de l'utilisateur (MFG – Ventes) est indiqué lors de la connexion, et d'autres rôles seraient listés comme options le cas échéant

   https://technologyblog.rsmus.com/technologies/netsuite/two-factor-authentication-2fa-for-netsuite/

  .

- **Rôles réservés aux services Web :** NetSuite propose une case à cocher sur les rôles appelée « **Rôle réservé aux services Web** ». Marquer un rôle comme étant réservé aux services Web signifie que ce rôle **ne peut pas être utilisé pour se connecter via l'interface utilisateur de NetSuite** – il est exclusivement destiné à l'accès API (SuiteTalk) (Source: docs.oracle.com)(Source: docs.oracle.com). Si un utilisateur tente de se connecter de manière interactive avec ce rôle, NetSuite générera une erreur de validation et l'en empêchera (Source: docs.oracle.com). Il s'agit d'une fonctionnalité de sécurité : vous pourriez créer un rôle avec des permissions élevées dont une intégration a besoin, mais vous ne voulez *jamais* qu'un humain utilise ce rôle dans l'interface web (car il pourrait alors utiliser ces permissions de manière non intentionnelle). Par exemple, supposons qu'un rôle d'intégration ait la permission de modifier des enregistrements financiers (parce qu'une intégration automatisée doit le faire), mais que vous ne voulez pas qu'un employé ordinaire se connecte via l'interface utilisateur et effectue directement ces modifications. Rendre le rôle réservé aux services Web résout ce problème – l'intégration peut se connecter via l'API SOAP/REST en utilisant le jeton de ce rôle, mais l'utilisateur ne peut pas sélectionner ce rôle dans l'interface utilisateur (Source: docs.oracle.com). Cela confine essentiellement le rôle à une utilisation

programmatique. Bonne pratique : pour chaque intégration, attribuez un rôle unique réservé aux services Web avec les permissions minimales nécessaires à un utilisateur d'intégration dédié. Cela réduit considérablement les risques.

- **Impact des permissions de rôle sur l'authentification d'intégration :** Les permissions au sein d'un rôle régissent également ce qu'une intégration peut faire en utilisant ce rôle via TBA/OAuth. Il est à noter que pour utiliser certaines méthodes d'authentification, des permissions spécifiques sont nécessaires. Par exemple, pour utiliser l'authentification par jeton, le rôle doit avoir la permission « Jetons d'accès utilisateur » (pour permettre à l'utilisateur de créer des jetons) et/ou « Gestion des jetons d'accès » si vous voulez qu'il gère les jetons pour l'ensemble du compte (Source: info.ennvee.com)(Source: info.ennvee.com). Pour utiliser les services web REST, le rôle a besoin de la permission « Services web REST » ; pour utiliser les RESTlets, le rôle a besoin de la permission « RESTlets », etc. Si celles-ci sont manquantes, les appels d'intégration recevront des erreurs de permission. NetSuite exige également souvent la permission « Services Web » (pour SOAP) sur tout rôle utilisé pour les intégrations SOAP. Ce sont des détails techniques, mais l'administrateur configurant les rôles pour l'intégration doit veiller à inclure toutes les permissions nécessaires, mais rien de plus.

Globalement, le contrôle d'accès basé sur les rôles dans NetSuite est un outil puissant pour appliquer le principe du moindre privilège et l'accès basé sur le contexte. Les DAF devraient s'assurer que la bonne séparation des tâches est mise en œuvre via les rôles – par exemple, votre connexion personnelle pourrait avoir un rôle de DAF qui ne peut pas, disons, administrer le système (pour réduire les risques), et une connexion administrateur distincte existe pour les changements de système. Et pour les intégrations, utilisez toujours des rôles spéciaux plutôt que de réutiliser un rôle existant à haut niveau de privilège. Une configuration appropriée des rôles et de leurs exigences de connexion (2FA, SSO, ou restreint à l'API) renforcera considérablement votre posture de sécurité NetSuite.

Méthodes de connexion héritées et fonctionnalités obsolètes

Les capacités d'authentification de NetSuite ont évolué au fil des ans. Ce faisant, quelques anciennes méthodes de connexion ont été dépréciées ou remplacées par des approches modernes plus sécurisées. En tant qu'administrateur ou partie prenante de NetSuite, vous devez être conscient de ces méthodes héritées, surtout si vous avez d'anciennes intégrations qui pourraient encore en dépendre :

- **SuiteSignOn (authentification unique sortante) – Obsolète :** « SuiteSignOn » était le mécanisme antérieur de NetSuite pour l'authentification unique **sortante**, ce qui signifie qu'il permettait à une application externe de s'authentifier auprès de NetSuite via un échange de jetons (souvent utilisé avec des intégrations de services web héritées). Malgré son nom, SuiteSignOn ne concerne pas la connexion des utilisateurs à NetSuite via le SSO (ce qui est le SSO entrant), mais plutôt d'autres

systèmes utilisant NetSuite comme fournisseur d'identité ou courtier SSO pour les appels de services web (Source: docs.oracle.com)(Source: docs.oracle.com). SuiteSignOn est en cours de retrait complet. Oracle a annoncé la fin du support pour SuiteSignOn : dans NetSuite 2024.1, il a été désactivé dans les comptes de non-production, et dans 2024.2, il a été désactivé dans les comptes de production (Source: community.oracle.com)(Source: community.oracle.com). À partir de la version 2025.1, SuiteSignOn n'est plus du tout supporté (Source: docs.oracle.com). Toute intégration utilisant SuiteSignOn cessera de fonctionner si elle n'a pas déjà été migrée. Le remplacement recommandé est d'utiliser la fonctionnalité **« NetSuite en tant que fournisseur OIDC »** pour les besoins de SSO sortant (Source: community.oracle.com) – essentiellement, si un système externe a besoin de se connecter via SSO à NetSuite, il doit maintenant utiliser un flux OpenID Connect. À des fins pratiques, la plupart des clients ont remplacé les intégrations SuiteSignOn par l'authentification basée sur des jetons ou OAuth il y a longtemps, car SuiteSignOn n'était pas largement utilisé en dehors de scénarios SuiteCloud spécifiques. Si votre organisation avait une intégration héritée qui effectuait un étrange échange de connexion avec NetSuite, il est probable que ce soit cela, et elle doit être refactorisée maintenant qu'elle est obsolète.

- **Authentification unique entrante (héritée) – Remplacée par SAML/OIDC :** Avant de prendre en charge SAML 2.0 et OIDC, NetSuite avait ses propres méthodes SSO entrantes propriétaires (à un moment donné, une fonctionnalité pour se connecter via un OpenID Oracle ou un OpenID Google, etc.). Ces approches plus anciennes sont maintenant entièrement obsolètes. En fait, la fonctionnalité **SSO entrant de NetSuite a été officiellement dépréciée en 2021.1** – toute solution utilisant l'ancien SSO entrant a cessé de fonctionner après cette date (Source: docs.oracle.com)(Source: docs.oracle.com). Oracle a explicitement conseillé aux clients de migrer vers la nouvelle **fonctionnalité d'authentification unique OIDC (introduite en 2019.2) ou vers le SSO SAML 2.0**(Source: docs.oracle.com). En substance, si vous utilisiez un SSO personnalisé antérieur (peut-être une connexion OpenID Google héritée ou un outil tiers non conforme à SAML), il n'est plus pris en charge. Aujourd'hui, SAML et OIDC couvrent tous les besoins en matière de SSO utilisateur entrant vers NetSuite. Par exemple, il y a quelques années, certaines entreprises autorisaient la connexion à NetSuite avec des comptes Google Apps via une ancienne méthode OpenID – celles-ci auraient été migrées vers SAML avec Google ou vers OIDC à présent. En résumé : toute méthode SSO *non*-SAML, *non*-OIDC est héritée et a disparu.

- **Identifiants utilisateur directs pour l'API (NLAuth) – Suppression progressive :** Dans le passé, les intégrations pouvaient s'authentifier auprès de NetSuite en fournissant un nom d'utilisateur et un mot de passe (ainsi qu'un ID de compte et un rôle) soit via une opération de connexion SOAP spéciale, soit via des en-têtes HTTP (souvent appelée NLAuth). C'était une approche courante avant 2015 et elle a été supportée jusqu'en 2020 sous diverses formes. Cependant, NetSuite s'est fortement éloigné de l'autorisation des identifiants utilisateur directs pour les intégrations. À partir du point de terminaison API 2020.2, les services web SOAP n'acceptent plus la connexion `Passport`

(identifiants utilisateur) pour l'authentification – toute tentative d'utilisation d'un nom d'utilisateur/mot de passe sur les points de terminaison 2020.2+ entraîne une erreur (Source: docs.oracle.com). De même, à partir de 2021, les appels RESTlet n'acceptent plus l'autorisation de base avec les identifiants utilisateur pour les rôles hautement privilégiés (Source: docs.oracle.com) (Source: docs.oracle.com). Et comme les rôles d'administrateur doivent avoir la 2FA, vous ne pouvez effectivement *pas* utiliser le mot de passe d'un administrateur via l'API du tout (Source: docs.oracle.com). Tous ces changements signifient que l'**ancien modèle d'intégration consistant à « stocker un nom d'utilisateur et un mot de passe d'administrateur dans mon script pour appeler NetSuite » est officiellement obsolète** et échouera dans les versions actuelles. Le message de NetSuite est clair : utilisez TBA ou OAuth 2.0 pour toute intégration API (Source: docs.oracle.com)(Source: docs.oracle.com). La seule exception mineure pourrait être certains scénarios de non-production ou scripts internes, mais même ceux-ci devraient faire la transition. Si vous avez des scripts ou des connecteurs utilisant encore NLAuth (authentification de base) avec un mot de passe enregistré, prévoyez de les mettre à jour – non seulement pour la conformité (ils pourraient déjà être cassés) mais aussi pour la sécurité (personne ne veut de identifiants en clair stockés dans le code).

- **Opération « Login » de SuiteTalk SOAP – Héritée :** SuiteTalk est le nom de l'API SOAP de NetSuite. Historiquement, un client pouvait appeler une opération `login` avec l'e-mail, le mot de passe, le rôle et le compte d'un utilisateur pour démarrer une session. Cette authentification basée sur la session est également essentiellement retirée dans les versions WSDL plus récentes. La recommandation depuis l'introduction de TBA est de ne pas utiliser l'opération de connexion mais d'authentifier chaque requête avec des identifiants basés sur des jetons à la place (Source: docs.oracle.com). En fait, mélanger l'ancienne connexion de session avec l'authentification par jeton dans la même requête SOAP n'est pas autorisé (Source: docs.oracle.com). Les nouvelles intégrations devraient éviter d'utiliser la connexion SOAP du tout – c'est une étape inutile et stateful alors que vous pouvez simplement envoyer un jeton dans chaque requête. La dépréciation par Oracle des identifiants utilisateur dans SOAP en 2020 signifie que si vous mettez à jour vers la dernière version de l'API, vous ne pourriez de toute façon pas vous connecter de cette manière (Source: docs.oracle.com).

- **Code PIN mobile et autres méthodes mineures :** Les anciennes versions de l'application mobile de NetSuite permettaient un code PIN pour une connexion rapide après l'authentification initiale, et l'application mobile actuelle de NetSuite prend en charge le déverrouillage biométrique (FaceID/TouchID sur les téléphones) (Source: docs.oracle.com)(Source: docs.oracle.com). Ce ne sont pas des méthodes d'authentification distinctes en soi – elles sont basées sur les méthodes primaires (le biométrique ou le code PIN déverrouille simplement un jeton de session stocké localement qui a été obtenu via une connexion normale/SSO). Pour être complet : il existait auparavant un concept de « mots de passe spécifiques à l'application » pour certaines intégrations

(comme l'intégration de messagerie Outlook) qui permettait un mot de passe distinct contournant la 2FA. NetSuite gère désormais également ces cas d'utilisation avec l'authentification par jeton, de sorte qu'ils ont largement disparu ou sont gérés en arrière-plan.

- **2FA par SMS/Voix (dépréciation douce) :** Comme indiqué, NetSuite met fin au support de la 2FA par SMS ou appel vocal (parfois appelée « SMA » dans la documentation) à compter du 1er mars 2024 (Source: withum.com). Il s'agit davantage d'une mise à jour de fonctionnalité que d'une dépréciation de méthode de connexion, mais il est important de le mentionner pour éviter toute confusion. Si un utilisateur utilisait des codes par message texte pour se connecter, il devra passer à une application d'authentification. Cela s'aligne sur les meilleures pratiques de l'industrie, car la 2FA basée sur SMS est moins sécurisée (vulnérable aux échanges de carte SIM, etc.). À l'avenir, toute la 2FA de NetSuite utilisera des applications d'authentification ou des méthodes similaires.

En résumé, les méthodes d'authentification héritées dans NetSuite – SuiteSignOn, l'ancien SSO entrant et l'authentification directe par identifiants – ont toutes été retirées au profit des méthodes plus robustes que nous avons détaillées dans la section précédente. Si votre organisation utilise NetSuite depuis de nombreuses années, il est utile de vérifier qu'aucune de vos intégrations ou personnalisations ne s'accroche à ces approches plus anciennes. Très probablement, toute intégration critique aurait été mise à jour d'ici maintenant pour utiliser des jetons, mais c'est un bon point de maintenance. Les versions NetSuite 2021+ ont imposé ces changements pour garantir que les clients utilisent une authentification moderne et plus sécurisée.

Authentification pour les intégrations (API, services Web et RESTlets)

Au-delà des connexions à l'interface utilisateur, un aspect significatif de l'accès à NetSuite est l'**intégration machine-à-machine**. De nombreuses entreprises ont des systèmes externes (par exemple, des sites web de commerce électronique, des systèmes CRM, des entrepôts de données, etc.) qui doivent communiquer avec NetSuite. NetSuite fournit plusieurs interfaces d'intégration – et chacune a des mécanismes d'authentification spécifiques. Voici comment fonctionne l'authentification pour les principales méthodes d'intégration :

- **Services Web SOAP SuiteTalk :** Il s'agit de l'API classique basée sur SOAP (souvent simplement appelée « Services Web » dans la documentation NetSuite). Comme discuté, les services web SOAP autorisaient historiquement la connexion via des identifiants utilisateur, mais sur les versions actuelles, **la seule authentification prise en charge pour SOAP est l'authentification basée sur des jetons**(Source: docs.oracle.com)(Source: docs.oracle.com). Une intégration appelant l'API SOAP doit inclure le jeton TBA dans l'en-tête SOAP (NetSuite prend en charge une approche de « identifiants au niveau de la requête » où vous fournissez la clé/secret du jeton et la clé/secret du

consommateur dans les en-têtes HTTP, plutôt que d'effectuer une connexion basée sur une session) (Source: docs.oracle.com). Si vous tentez d'utiliser l'opération `login` ou de fournir un nom d'utilisateur/mot de passe dans une requête SOAP 2020.2+, elle sera rejetée (Source: docs.oracle.com). Ainsi, toute intégration SOAP doit utiliser un jeton. La documentation de NetSuite déclare sans ambages : *« Vous ne devriez pas utiliser les identifiants utilisateur pour les services web SOAP. Utilisez plutôt l'authentification basée sur des jetons. »*(Source: docs.oracle.com) (Source: docs.oracle.com). L'API SOAP exige également qu'un **ID d'application** soit inclus (un GUID de votre enregistrement d'intégration) si vous utilisez des identifiants utilisateur sur des versions plus anciennes, mais avec TBA, il est implicitement lié via le jeton. En résumé, pour SOAP : créez un enregistrement d'intégration, attribuez un utilisateur+rôle, générez un jeton, et utilisez-le. Assurez-vous que le rôle dispose des permissions SOAP nécessaires (permission « Services Web ») et de toutes les permissions d'enregistrement requises.

- **Services Web REST (SuiteTalk REST API) :** Ces dernières années, NetSuite a introduit une API RESTful pour l'accès aux enregistrements (souvent appelée SuiteTalk REST API, différente des RESTlets personnalisés). Pour ces **services web REST**, NetSuite prend en charge à la fois l'authentification basée sur les jetons (TBA) et OAuth 2.0 pour l'authentification (Source: docs.oracle.com)(Source: docs.oracle.com). OAuth 2.0 est encouragé pour REST car il est plus simple (pas de calcul manuel de signature) (Source: docs.oracle.com). En fait, le guide officiel stipule : *« OAuth 2.0 est la méthode d'authentification préférée... utilisez OAuth 2.0 au lieu de TBA chaque fois que possible. »*(Source: docs.oracle.com). Cependant, TBA reste une option si nécessaire. Cette API ne prend pas du tout en charge la connexion directe avec les identifiants utilisateur (ce qui n'aurait d'ailleurs pas de sens en REST). Ainsi, toute intégration utilisant l'API REST Records doit être enregistrée en tant qu'intégration et utiliser un jeton OAuth 2.0 ou un jeton TBA dans l'en-tête HTTP Authorization. De nombreuses plateformes d'intégration (comme Boomi, MuleSoft, etc.) disposent désormais de connecteurs qui prennent nativement en charge OAuth 2.0 de NetSuite – simplifiant la vie des administrateurs (il suffit de brancher la clé/secret du consommateur et de laisser le connecteur gérer la récupération/le rafraîchissement du jeton). Si vous développez en interne, les développeurs peuvent utiliser les flux OAuth documentés de NetSuite pour obtenir un jeton pour les appels REST (Source: docs.oracle.com).

- **SuiteScript RESTlets :** Les RESTlets sont des points d'accès RESTful personnalisés que vous pouvez développer avec SuiteScript pour étendre l'API de NetSuite. L'authentification aux RESTlets fonctionne de manière similaire à SOAP ou à l'API REST – vous ne pouvez pas simplement les appeler sans authentification. À l'origine, vous pouviez appeler les RESTlets en utilisant l'authentification de base (en-tête NLAuth avec l'e-mail et le mot de passe de l'utilisateur), mais cela n'est plus autorisé pour les rôles avec 2FA et n'est officiellement plus pris en charge à partir de 2021 (Source: docs.oracle.com)(Source: docs.oracle.com). Au lieu de cela, vous devriez appeler les RESTlets avec un jeton TBA ou un jeton OAuth 2.0 dans l'en-tête (Source: docs.oracle.com)(Source:

docs.oracle.com). La *Matrice d'Authentification* dans l'aide d'Oracle indique clairement pour les RESTlets : « Méthodes prises en charge : Authentification basée sur les jetons ou OAuth 2.0 (les identifiants utilisateur ne sont plus pris en charge à partir de 2021) » (Source: docs.oracle.com) (Source: docs.oracle.com). Cela signifie que si vous avez des intégrations RESTlet (par exemple, une page web personnalisée soumettant des données à un RESTlet NetSuite), vous devez implémenter l'authentification par jeton. Le processus est le même : traitez-le comme n'importe quelle autre intégration – créez un enregistrement d'intégration, obtenez des jetons ou configurez OAuth, et utilisez-les.

- **SuiteAnalytics Connect (ODBC/JDBC) :** Il s'agit du pilote ODBC / JDBC de NetSuite pour la connexion à l'entrepôt de données analytiques (permettant essentiellement des requêtes SQL sur les données NetSuite). L'authentification pour SuiteAnalytics Connect nécessitait historiquement un *compte ODBC* distinct (vous utilisiez votre e-mail NetSuite et un mot de passe ODBC spécial, car il ne pouvait pas demander la 2FA). Cependant, NetSuite prend désormais également en charge OAuth 2.0 pour SuiteAnalytics Connect (Source: docs.oracle.com). L'utilisation d'OAuth est préférée car elle contourne le problème d'incompatibilité de la 2FA – comme mentionné précédemment, les rôles 2FA ne peuvent pas utiliser directement Connect avec des identifiants de base (Source: docs.oracle.com). Les administrateurs peuvent configurer un client OAuth 2.0 pour Connect, ou utiliser un jeton si le pilote le prend en charge. Si quelqu'un au sein du service financier utilise Excel avec une connexion ODBC à NetSuite, assurez-vous qu'il met à jour son pilote et utilise OAuth si possible, sinon il pourrait avoir besoin d'une exemption (comme un rôle sans 2FA, ce qui n'est pas idéal). L'objectif de NetSuite est d'éliminer tout accès basé sur un mot de passe pour ces connexions externes.

- **Authentification par ID d'appareil (SuiteCommerce InStore) :** Un cas unique est l'authentification utilisée par SuiteCommerce InStore (SCIS), qui est le système de point de vente de NetSuite. SCIS utilise un concept d'**authentification par ID d'appareil** – il s'agit essentiellement d'enregistrer un appareil (comme une caisse enregistreuse iPad) avec un ID d'appareil et une clé afin qu'il puisse s'authentifier automatiquement auprès de NetSuite sans qu'un utilisateur n'ait à saisir ses identifiants à chaque fois (Source: docs.oracle.com)(Source: docs.oracle.com). Cela a été développé pour rationaliser l'utilisation en magasin, où un vendeur pourrait ne pas se connecter/déconnecter individuellement pour chaque transaction. Les clients NetSuite peuvent potentiellement exploiter la même authentification par ID d'appareil pour des applications personnalisées si nécessaire (Source: docs.oracle.com), mais il s'agit d'un scénario de niche. Il est mentionné ici pour être complet : les ID d'appareil sont configurés dans NetSuite et liés à un certain niveau d'accès (probablement via un rôle). Cette méthode est strictement contrôlée et principalement pertinente pour les déploiements de vente au détail. La plupart des DAF ne la rencontreront pas à moins de traiter avec une branche

de vente au détail utilisant SCIS. Le point clé : c'est une autre façon pour NetSuite d'authentifier une « entité » (appareil) de manière fiable sans connexion interactive. Si vous n'utilisez pas SCIS, vous ne l'utiliserez probablement pas.

- **NetSuite en tant qu'IdP OAuth2 (SSO d'intégration sortante) :** C'est le revers de la médaille de SuiteSignOn. NetSuite peut agir en tant que **fournisseur OpenID Connect** pour les applications externes (Source: docs.oracle.com)(Source: docs.oracle.com). Par exemple, si vous avez une application web personnalisée qui doit permettre aux utilisateurs de se connecter avec leurs identifiants NetSuite, vous pourriez utiliser NetSuite comme source d'identité via OAuth2/OIDC. Il s'agit d'un scénario avancé et relativement rare (car on utiliserait généralement un IdP d'entreprise, et non NetSuite, comme source). Néanmoins, cela existe : NetSuite peut émettre des jetons OAuth2 pour ses utilisateurs vers d'autres applications. Ceci est plus pertinent pour les développeurs qui étendent l'écosystème NetSuite. La majorité des organisations intégreront plutôt NetSuite à leur IdP existant plutôt que l'inverse.

Pour résumer en termes de conseils pratiques : **Toutes les intégrations à NetSuite doivent utiliser soit l'authentification basée sur les jetons (TBA), soit OAuth 2.0** (ou un flux spécialisé comme l'ID d'appareil si applicable). Les connexions directes des utilisateurs dans les intégrations ne sont plus prises en charge. Lors de la planification d'une intégration, décidez si l'API REST ou l'API SOAP convient le mieux, puis configurez le jeton ou les identifiants OAuth appropriés. Si vous travaillez avec une intégration tierce, assurez-vous qu'elle prend en charge l'authentification basée sur les jetons de NetSuite – la plupart des SuiteApps certifiées et des outils d'intégration le font. En tant qu'administrateur, maintenez une liste des enregistrements d'intégration et des jetons, et examinez-les périodiquement (qui utilise quel jeton, ce jeton a-t-il un rôle approprié avec le moindre privilège, etc.). Cela maintiendra votre système sécurisé et conforme aux pratiques recommandées par NetSuite (Source: emergetech.com) (Source: emergetech.com).

Bonnes pratiques pour l'authentification sécurisée et la gestion des accès

Gérer l'authentification NetSuite ne se limite pas aux mécanismes des méthodes de connexion – il s'agit d'implémenter des *politiques* et des *pratiques* pour garantir la sécurité de votre compte. Voici quelques bonnes pratiques que les DAF et les administrateurs NetSuite devraient considérer :

1. Appliquer l'authentification à deux facteurs pour tous les accès sensibles : NetSuite exige déjà la 2FA pour les administrateurs et les rôles à privilèges élevés (Source: docs.oracle.com), mais vous pourriez vouloir aller plus loin et activer la 2FA pour *tous* les rôles d'employés. Les données financières sont sensibles, et même les rôles qui ne font que consulter des rapports pourraient être des cibles de compromission. NetSuite facilite le marquage de tout rôle comme nécessitant la 2FA (Source:

technologyblog.rsmus.com). Envisagez d'exiger la 2FA pour tout rôle pouvant consulter ou modifier des données critiques (comptabilité, opérations de vente, etc.). Les utilisateurs pourraient protester au début, mais avec les applications d'authentification, c'est un processus rapide qui réduit considérablement le risque de prise de contrôle de compte. Assurez-vous également que les utilisateurs disposent de directives pour stocker en toute sécurité leurs codes de sauvegarde 2FA (et que votre processus de support technique pour les réinitialisations 2FA est robuste). De nombreuses entreprises ont désormais une politique : « Si vous accédez à NetSuite, vous devez avoir la 2FA. » C'est judicieux compte tenu des données financières et personnelles contenues dans un ERP.

2. Utiliser l'authentification unique (SSO) pour la commodité et le contrôle des utilisateurs : Si votre organisation dispose d'un fournisseur d'identité ou d'une plateforme SSO, intégrez NetSuite à celle-ci (via SAML ou OIDC). Cela non seulement évite aux utilisateurs de devoir se souvenir d'un autre mot de passe, mais cela permet aussi un contrôle centralisé : lorsqu'un employé quitte l'entreprise, le désactiver dans l'IdP coupe instantanément l'accès à NetSuite. Cela signifie également que vous pouvez appliquer votre politique de mot de passe d'entreprise et votre politique MFA à NetSuite. Par exemple, si votre entreprise utilise Okta avec MFA adaptative, un utilisateur se connectant à NetSuite via Okta respectera automatiquement ces règles. Le SSO améliore également l'expérience utilisateur – accès en un clic et moins de demandes de connexion. Du point de vue de la conformité, le SSO fournit une piste d'audit dans votre IdP de qui a accédé à NetSuite et quand, ce qui peut compléter l'audit de connexion interne de NetSuite. Assurez-vous simplement de conserver un identifiant administrateur d'urgence (avec 2FA) pour les moments où l'IdP pourrait être indisponible (un compte de secours). De plus, testez le provisionnement des rôles – souvent, vous voudrez peut-être utiliser SCIM ou le provisionnement automatisé des utilisateurs afin que les nouvelles recrues obtiennent un compte NetSuite et une attribution de rôle appropriée dès le premier jour, et le SSO rend cela transparent.

3. Sécuriser vos intégrations – Pas de mots de passe partagés : Ne jamais utiliser les identifiants de connexion d'un utilisateur général dans un script d'intégration. Chaque intégration doit avoir son propre **compte utilisateur d'intégration** dans NetSuite, avec un rôle personnalisé dédié n'accordant que les permissions dont cette intégration a besoin (Source: emergetech.com)(Source: emergetech.com). Marquez ce rôle comme « Services Web Uniquement » si une connexion interactive n'est pas nécessaire (Source: docs.oracle.com). Utilisez l'authentification basée sur les jetons ou OAuth 2.0 pour l'authentification de l'intégration – de cette façon, vous n'intégrez pas de mots de passe dans le code et n'êtes pas affecté par les changements de mot de passe ou les problèmes de 2FA (Source: info.ennvee.com)(Source: docs.oracle.com). Si l'intégration est fournie par un fournisseur (SuiteApp ou connecteur), il devrait vous demander de créer un enregistrement d'intégration et de générer des jetons – suivez ces étapes avec diligence et **ne réutilisez pas** le jeton d'un administrateur pour plusieurs intégrations. En isolant les intégrations, vous pouvez en révoquer une individuellement sans impacter les autres si quelque chose de suspect est détecté.

4. Principe du moindre privilège (Conception des rôles) : Lors de la création de rôles pour les utilisateurs ou les intégrations, accordez les permissions minimales requises. Par exemple, si un outil de budgétisation tiers doit extraire des données NetSuite via l'API, n'utilisez pas un jeton d'administrateur complet – créez un rôle « Exportation Budget » qui n'a peut-être qu'un accès en lecture aux enregistrements spécifiques (financiers, transactions) nécessaires à l'intégration (Source: emergetech.com)(Source: emergetech.com). De cette façon, même si le jeton est compromis, les dommages sont limités. La personnalisation des rôles de NetSuite permet un contrôle très granulaire, et vous pouvez avoir plusieurs rôles personnalisés à des fins différentes. Examinez régulièrement qui a quels rôles – les DAF pourraient demander un examen trimestriel des accès utilisateurs pour s'assurer, par exemple, que la personne ayant le rôle « Administrateur NetSuite » est toujours au sein du service informatique et autorisée, etc. Portez une attention particulière aux permissions hautement privilégiées (comme Gérer la facturation, Modifier le GL, etc.) – ces rôles devraient absolument avoir la 2FA et être strictement attribués.

5. Tirer parti des fonctionnalités de restriction d'adresse IP : NetSuite vous permet de restreindre la connexion par adresse IP au niveau du compte ou de l'employé (Source: emergetech.com)(Source: emergetech.com). Pour une sécurité accrue, certaines entreprises choisissent de n'autoriser l'accès à NetSuite qu'à partir de certains réseaux (par exemple, le bureau de l'entreprise ou les IP VPN). Dans les enregistrements d'informations sur l'entreprise ou d'employés de NetSuite, vous pouvez spécifier des plages d'adresses IP autorisées (Source: emergetech.com). Si un attaquant obtient d'une manière ou d'une autre les identifiants d'un utilisateur, mais tente de se connecter depuis une adresse IP qui ne figure pas sur la liste autorisée, il sera bloqué (Source: emergetech.com)(Source: emergetech.com). Cela pourrait ne pas être pratique pour tous (surtout avec le télétravail – collecter les IP domestiques est délicat et elles changent), mais c'est une considération pour des rôles comme Administrateur. Vous pourriez, par exemple, autoriser la connexion administrateur uniquement depuis l'IP publique de votre siège social ou via votre SSO (qui pourrait lui-même appliquer des règles IP). Si vous mettez en œuvre des restrictions IP, ayez un processus pour les mettre à jour lorsque les utilisateurs voyagent ou que les FAI changent, afin d'éviter de bloquer les accès légitimes.

6. Surveiller l'activité de connexion et configurer des alertes : NetSuite fournit une piste d'audit de connexion qui enregistre toutes les tentatives de connexion (réussies et échouées). En tant qu'administrateur, vous devriez examiner cela périodiquement ou configurer des recherches enregistrées pour détecter les anomalies – par exemple, plusieurs tentatives de connexion échouées pour un utilisateur (pourrait indiquer une tentative de force brute), ou des connexions depuis des lieux inhabituels. Étant donné que les DAF sont souvent préoccupés par les risques, il pourrait être judicieux de configurer une alerte si, par exemple, un compte administrateur se connecte en dehors des heures de bureau ou depuis un pays inattendu. NetSuite ne peut pas envoyer d'alertes prêtes à l'emploi pour cela, mais vous pouvez utiliser des alertes e-mail de recherche enregistrée ou SuiteScript pour la

personnalisation. De plus, Oracle propose des services comme Adaptive Access Control dans certaines éditions qui peuvent signaler les connexions suspectes. Même sans cela, un examen proactif des journaux de connexion est une bonne pratique.

7. Éduquer les utilisateurs sur les pratiques de sécurité : Tous les contrôles techniques ne serviront à rien si les utilisateurs sont victimes de phishing. Assurez-vous que les utilisateurs savent que NetSuite ne leur demandera jamais leur mot de passe par e-mail, et qu'ils doivent se méfier de toute page de connexion qui ne semble pas correcte. Étant donné que de nombreux utilisateurs accèdent à NetSuite via un navigateur web, rappelez-leur de vérifier l'URL (elle doit être un domaine *.netsuite.com ou le domaine de votre IdP pour le SSO). L'utilisation du SSO aide ici, car les utilisateurs s'habituent à ne se connecter qu'au niveau de l'IdP. Encouragez également l'utilisation de gestionnaires de mots de passe pour toute connexion directe afin d'éviter les mots de passe faibles ou la réutilisation. Le DAF peut jouer un rôle de leadership dans la promotion d'une culture soucieuse de la sécurité, car la finance est souvent la cible de l'ingénierie sociale.

8. Maintenir NetSuite et les appareils à jour : C'est un conseil informatique général mais pertinent – assurez-vous que toute personne accédant à NetSuite dispose de navigateurs et de systèmes d'exploitation à jour (NetSuite cessera de prendre en charge les anciennes versions de TLS, par exemple). Oracle maintient une sécurité stricte sur son cloud, mais le point d'accès (la machine de l'utilisateur) doit également être sécurisé. Le service informatique de l'entreprise devrait appliquer les dernières mises à jour sur les appareils des employés pour réduire les risques de keyloggers ou de logiciels malveillants qui pourraient compromettre les sessions NetSuite (Source: emergetech.com) (Source: emergetech.com).

En suivant ces bonnes pratiques – application de la 2FA, adoption du SSO, utilisation de jetons pour les intégrations, limitation des privilèges et vigilance constante – vous réduisez considérablement le risque d'accès non autorisé à vos comptes NetSuite. En termes d'audit, ces contrôles contribuent à garantir que seules les bonnes personnes (ou systèmes) ont le bon accès au bon moment, ce qui est exactement ce que vous souhaitez pour un système qui contient des données financières et opérationnelles critiques.

Exemples concrets de configurations de connexion NetSuite

Pour tout relier, examinons quelques **exemples de scénarios** montrant comment les entreprises pourraient configurer l'authentification NetSuite en pratique. Ceux-ci illustrent la combinaison des méthodes décrites ci-dessus dans des contextes commerciaux réels :

Exemple 1 : MidCorp Inc – SSO pour les utilisateurs, TBA pour les intégrations

MidCorp Inc, une entreprise de taille moyenne, utilise **Okta** comme fournisseur d'identité d'entreprise. L'administrateur NetSuite de MidCorp a activé le SSO SAML dans NetSuite et configuré Okta avec les métadonnées SAML de NetSuite (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com). Tous les utilisateurs internes se voient attribuer un rôle SSO NetSuite (leur rôle d'employé standard avec permission SAML). Désormais, les employés se connectent à NetSuite via Okta – soit en se rendant sur le portail d'Okta et en cliquant sur l'application NetSuite, soit en entrant leur URL NetSuite qui les redirige vers Okta. Ils s'authentifient avec leurs identifiants Okta + MFA Okta (application Okta Verify) et sont ensuite automatiquement connectés à NetSuite. Plus personne n'utilise de mot de passe spécifique à NetSuite pour un usage quotidien. Cela facilite l'intégration et le départ des employés (le service informatique gère simplement les groupes Okta), et le DAF constate moins de problèmes de type « J'ai oublié mon mot de passe NetSuite ».

MidCorp dispose également de plusieurs intégrations : son site web e-commerce extrait la disponibilité des stocks de NetSuite et publie les commandes, et une intégration Salesforce distincte synchronise les données clients. Pour chaque intégration, l'administrateur a créé un **rôle d'intégration dédié** dans NetSuite, limité aux permissions nécessaires (par exemple, le rôle e-commerce peut lire les articles et écrire les commandes, rien de plus). Chaque rôle d'intégration est marqué « Services Web Uniquement » pour empêcher quiconque de l'utiliser dans l'interface utilisateur (Source: docs.oracle.com). L'administrateur a créé un « Utilisateur d'intégration Ecom » et un « Utilisateur de synchronisation CRM » dans NetSuite, chacun n'étant affecté qu'à son rôle d'intégration. Ensuite, pour le site e-commerce, elle a créé un enregistrement d'intégration d'authentification basée sur les jetons (TBA) dans NetSuite et a généré un jeton pour l'utilisateur/rôle d'intégration Ecom – fournissant l'ID/secret du jeton et la clé/secret du consommateur aux développeurs web. Le site web authentifie désormais les appels d'API à l'aide de ces jetons (sans mots de passe). De même, le connecteur Salesforce a été configuré à l'aide d'un flux d'identifiants client OAuth 2.0 – l'administrateur a créé un enregistrement d'intégration autorisant OAuth2 et a fourni l'ID/secret client au connecteur, qui les échange contre un jeton d'accès pour appeler l'API REST de NetSuite (Source: docs.oracle.com)(Source: docs.oracle.com). Les deux intégrations fonctionnent de manière fluide et sécurisée : si un problème survient, l'administrateur peut révoquer uniquement ce jeton sans impacter aucune autre partie de NetSuite. Le DAF de MidCorp dort mieux sachant que même si un compte d'intégration était compromis, il n'aurait qu'un accès très limité, et aucun mot de passe utilisateur réel ne serait compromis.

Exemple 2 : SmallCo Ltd – 2FA native et OAuth pour une application tierce SmallCo Ltd est une petite entreprise qui n'utilise pas de fournisseur SSO externe. Elle s'appuie sur la connexion native de NetSuite par nom d'utilisateur/mot de passe avec 2FA. L'administrateur NetSuite (qui est aussi le responsable informatique) a appliqué la 2FA pour tous les rôles du compte, pas seulement pour l'administrateur – ce qui signifie que chaque utilisateur, y compris le DAF, a dû configurer Google Authenticator lors de sa première connexion (Source: docs.oracle.com)(Source: docs.oracle.com). Désormais, chaque fois que les utilisateurs se connectent à NetSuite, après avoir entré leur e-mail et leur mot de passe, un code à 6

chiffres leur est demandé depuis leur téléphone. C'est une étape supplémentaire, mais ils ont l'esprit tranquille, sachant qu'un mot de passe volé seul ne peut pas accorder l'accès. L'administrateur a également réglé la durée de « l'appareil de confiance » pour la 2FA à 30 jours par commodité, afin que les utilisateurs n'aient pas à le faire tous les jours sur leurs appareils principaux (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com).

SmallCo utilise également un logiciel de budgétisation cloud qui doit extraire des données de NetSuite chaque mois. Cette application de budgétisation prend en charge l'API REST de NetSuite via OAuth 2.0. L'administrateur de SmallCo a créé un enregistrement d'intégration dans NetSuite pour « BudgetApp », a activé OAuth 2.0 et a obtenu un ID/secret client. Dans NetSuite, elle a également attribué un rôle spécial « Rapports Budgétaires » à l'utilisateur Contrôleur, avec des permissions pour consulter les transactions et les enregistrements financiers, et a donné à ce rôle la permission Services Web REST nécessaire. Lors de la configuration de BudgetApp, le Contrôleur a suivi un flux de code d'autorisation OAuth 2.0 : BudgetApp a ouvert une fenêtre de navigateur vers la page OAuth de NetSuite, le Contrôleur s'est connecté avec ses identifiants NetSuite + 2FA, et il lui a été demandé « Autorisez-vous BudgetApp à accéder à vos données NetSuite ? ». Il a approuvé, et l'application a reçu un jeton OAuth. Désormais, l'application de budgétisation peut récupérer les données selon les besoins, sous le rôle du Contrôleur, sans demander à nouveau les identifiants. Le jeton se rafraîchira automatiquement. Le DAF de SmallCo apprécie cette configuration car elle était facile (pas de travail informatique complexe au-delà de la configuration initiale) et elle sait exactement quelles données sont partagées. Si l'application de budgétisation est un jour remplacée ou suspectée d'un problème, l'administrateur peut révoquer son accès OAuth dans l'écran de gestion OAuth 2.0 de NetSuite (Source: docs.oracle.com).

Exemple 3 : GlobalCorp – Accès multi-comptes avec OIDC et sécurité spécifique aux appareils
GlobalCorp opère dans plusieurs régions et possède deux comptes NetSuite (un pour l'Amérique du Nord, un pour l'Europe). De nombreux utilisateurs financiers, y compris le DAF, ont besoin d'accéder aux deux. Pour simplifier, GlobalCorp a configuré le **SSO OpenID Connect** de NetSuite en utilisant Azure AD (Microsoft Entra ID) comme IdP. Les deux comptes NetSuite sont configurés pour faire confiance au même tenant Azure AD via OIDC. Cela permet à un utilisateur de se connecter une fois via Azure (avec ses identifiants Office 365 et MFA), puis dans l'interface de NetSuite, il peut basculer entre les rôles NA et EU sans se reconnecter – grâce à la session OIDC partagée (Source: docs.oracle.com). Le DAF apprécie que le changement de filiale ne soit plus qu'un menu déroulant, alors qu'avant il devait gérer deux identifiants et se réauthentifier constamment. Azure AD applique l'accès conditionnel, donc si le DAF se connecte depuis un nouvel appareil ou un nouvel emplacement, une vérification supplémentaire peut être demandée – ces politiques s'étendent également à l'accès NetSuite via OIDC.

De plus, GlobalCorp possède des magasins de détail utilisant SuiteCommerce InStore (SCIS) pour les points de vente. Les iPads des magasins sont configurés avec l'**Authentification par ID d'appareil**. Chaque appareil a été enregistré dans NetSuite avec un ID d'appareil et a reçu un rôle spécifique qui

accorde uniquement les permissions nécessaires pour l'inventaire et les transactions de vente. Lorsqu'un commis de magasin lance l'application SCIS, elle ne demande pas de connexion NetSuite ; au lieu de cela, elle utilise les identifiants de l'appareil pour se connecter (NetSuite sait que l'appareil 123 est autorisé à effectuer certaines opérations) (Source: docs.oracle.com). Les commis s'authentifient sur l'iPad lui-même via un code PIN, et NetSuite fait confiance à l'appareil. Cela accélère le processus de paiement car les commis ne se connectent et ne se déconnectent pas de NetSuite pour chaque vente. Du point de vue du siège social, ils ont une traçabilité complète : les transactions sont enregistrées sous un utilisateur générique « Commis de magasin (Appareil 123) » avec un journal d'audit, et ils peuvent révoquer instantanément l'accès d'un appareil si un iPad est perdu. Ils complètent cela en faisant pivoter les clés d'appareil périodiquement.

Exemple 4 : ServiceCo – Configuration de sécurité renforcée ServiceCo est un cabinet de services professionnels traitant des données client très sensibles dans NetSuite. Ils ont mis en œuvre plusieurs couches de sécurité pour l'accès à NetSuite. Tous les utilisateurs doivent se connecter via le VPN d'entreprise, et les connexions NetSuite sont restreintes aux plages d'adresses IP publiques de l'entreprise (Source: emergetech.com)(Source: emergetech.com). Ils utilisent le SSO SAML avec PingFederate comme IdP, et PingFederate lui-même exige une clé de sécurité physique U2F pour la MFA. Ainsi, un utilisateur a besoin de sa carte à puce ou de sa YubiKey pour même se connecter via SSO à NetSuite. ServiceCo a également configuré NetSuite pour **verrouiller automatiquement tout utilisateur après 3 tentatives de connexion infructueuses** et en informer l'administrateur – une approche à l'ancienne, mais étant donné que presque toutes les connexions se font via SSO, toute tentative de connexion directe est suspecte. Pour les intégrations, ServiceCo n'autorise aucune intégration directe tierce ; au lieu de cela, ils ont construit un middleware qui extrait des données via SuiteTalk à l'aide d'un jeton. Le rôle de cet utilisateur d'intégration est en lecture seule pour des enregistrements spécifiques. Ils examinent régulièrement le journal d'utilisation des jetons d'accès dans NetSuite (qui indique la dernière utilisation des jetons) et désactivent les jetons qui n'ont pas été utilisés depuis un certain temps, dans le cadre de la maintenance.

Bien que cela puisse sembler extrême, pour le DAF de ServiceCo, ces mesures offrent une assurance à leurs clients lors des évaluations de sécurité. Ils peuvent démontrer que l'accès à NetSuite nécessite plusieurs facteurs et une sécurité au niveau du réseau, rendant l'accès non autorisé hautement improbable.

Chacun de ces exemples montre une combinaison différente des outils d'authentification de NetSuite en action. La configuration de votre organisation dépendra de sa taille, de ses exigences de conformité et de son infrastructure informatique. Une petite entreprise pourrait s'en tenir à la 2FA intégrée de NetSuite et à des intégrations de jetons simples, tandis qu'une grande entreprise s'intégrera au SSO et aura des

stratégies de rôles plus complexes. En tant que DAF ou administrateur NetSuite, comprendre ces possibilités vous permet de prendre des décisions éclairées qui équilibrent la facilité d'utilisation pour votre équipe avec le niveau de sécurité nécessaire pour vos données.

Conclusion

NetSuite offre un riche ensemble de méthodes d'authentification – des connexions traditionnelles avec une 2FA robuste, à l'authentification unique transparente avec SAML/OIDC, et à l'accès sécurisé basé sur des jetons pour les intégrations – afin de garantir que seuls les utilisateurs et systèmes autorisés peuvent accéder à vos données financières. Le paysage des options de connexion NetSuite a considérablement mûri, abandonnant les pratiques héritées au profit d'approches sécurisées et conformes aux normes de l'industrie. En tant que garants du système financier de votre organisation, les DAF et les administrateurs NetSuite devraient travailler main dans la main pour mettre en œuvre ces méthodes d'authentification en suivant les meilleures pratiques : appliquer l'authentification à deux facteurs, s'intégrer au SSO d'entreprise, utiliser des rôles et des jetons à privilèges minimaux pour les intégrations, et rester vigilant par l'audit et la politique. Ce faisant, vous protégerez votre environnement NetSuite contre les accès non autorisés et vous conformerez aux exigences de conformité, tout en offrant une expérience utilisateur fluide à votre équipe.

N'oubliez pas que la sécurité est un processus continu. Examinez régulièrement les notes de version de NetSuite pour toute modification des fonctionnalités d'authentification (par exemple, la dépréciation de la 2FA par SMS ou de SuiteSignOn), et maintenez à jour la formation de sensibilisation de vos utilisateurs. Heureusement, Oracle continue d'investir dans les fonctionnalités de sécurité de NetSuite – et grâce aux connaissances de ce guide, vous pouvez utiliser en toute confiance toutes les méthodes de connexion disponibles pour protéger les joyaux de la couronne de votre entreprise. Connexion NetSuite heureuse (et sécurisée) !

Sources :

- Oracle NetSuite Help – *Authentication Overview & Matrix*(Source: docs.oracle.com)(Source: docs.oracle.com); *Mandatory 2FA for NetSuite*(Source: docs.oracle.com)(Source: docs.oracle.com); *SAML Single Sign-on*(Source: docs.oracle.com); *OIDC SSO*(Source: docs.oracle.com); *Token-Based Authentication (TBA)*(Source: info.ennvee.com)(Source: info.ennvee.com); *OAuth 2.0 in NetSuite*(Source: docs.oracle.com)(Source: docs.oracle.com); *Authentication for SOAP Web Services*(Source: docs.oracle.com); *RESTlet Authentication*(Source: docs.oracle.com)(Source: docs.oracle.com); *Outbound SSO (SuiteSignOn) Deprecation Notice*(Source: docs.oracle.com).
- Oracle NetSuite Help – *Roles and Permissions:* Web Services Only Role (Source: docs.oracle.com); Permissions Requiring 2FA (Source: docs.oracle.com).

- RSM Technology Blog – *Configuring SAML SSO 2.0 within NetSuite* (Okta example) (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com); *NetSuite 2FA Guide* (RSM) (Source: technologyblog.rsmus.com)(Source: technologyblog.rsmus.com).
- NetSuite Support Community – *SuiteSignOn End of Support Announcement*(Source: community.oracle.com)(Source: community.oracle.com).
- eMerge Technologies – *NetSuite Login Security Practices 2023* (IP restrictions, etc.) (Source: emergetech.com)(Source: emergetech.com).
- Oracle NetSuite Release Notes – *Deprecation of Inbound SSO and Outbound SSO timing*(Source: docs.oracle.com)(Source: community.oracle.com).
- Techfino Blog – *Dealing with 2FA in NetSuite* (general insights on 2FA requirements and bypass using tokens) (Source: docs.oracle.com).

Étiquettes: netsuite, authentification, methodes-connexion, 2fa, sso, authentification-par-jeton, authentification-api, securite, gestion-acces, administration-netsuite

---

À propos de Houseblend

HouseBlend.io is a specialist NetSuite™ consultancy built for organizations that want ERP and integration projects to accelerate growth—not slow it down. Founded in Montréal in 2019, the firm has become a trusted partner for venture-backed scale-ups and global mid-market enterprises that rely on mission-critical data flows across commerce, finance and operations. HouseBlend's mandate is simple: blend proven business process design with deep technical execution so that clients unlock the full potential of NetSuite while maintaining the agility that first made them successful.

Much of that momentum comes from founder and Managing Partner **Nicolas Bean**, a former Olympic-level athlete and 15-year NetSuite veteran. Bean holds a bachelor's degree in Industrial Engineering from École Polytechnique de Montréal and is triple-certified as a NetSuite ERP Consultant, Administrator and SuiteAnalytics User. His résumé includes four end-to-end corporate turnarounds—two of them M&A exits—giving him a rare ability to translate boardroom strategy into line-of-business realities. Clients frequently cite his direct, "coach-style" leadership for keeping programs on time, on budget and firmly aligned to ROI.

End-to-end NetSuite delivery. HouseBlend's core practice covers the full ERP life-cycle: readiness assessments, Solution Design Documents, agile implementation sprints, remediation of legacy customisations, data migration, user training and post-go-live hyper-care. Integration work is conducted by in-house developers certified on SuiteScript, SuiteTalk and RESTlets, ensuring that Shopify, Amazon, Salesforce, HubSpot and more than 100 other SaaS endpoints exchange data with NetSuite in real time. The goal is a single source of truth that collapses manual reconciliation and unlocks enterprise-wide analytics.

Managed Application Services (MAS). Once live, clients can outsource day-to-day NetSuite and Celigo® administration to HouseBlend's MAS pod. The service delivers proactive monitoring, release-cycle regression testing, dashboard and report tuning, and 24 × 5 functional support—at a predictable monthly rate. By combining fractional architects with on-demand developers, MAS gives CFOs a scalable alternative to hiring an internal team, while guaranteeing that new NetSuite features (e.g., OAuth 2.0, AI-driven insights) are adopted securely and on schedule.

Vertical focus on digital-first brands. Although HouseBlend is platform-agnostic, the firm has carved out a reputation among e-commerce operators who run omnichannel storefronts on Shopify, BigCommerce or Amazon FBA. For these clients, the team frequently layers Celigo's iPaaS connectors onto NetSuite to automate fulfilment, 3PL inventory sync and revenue recognition—removing the swivel-chair work that throttles scale. An in-house R&D group also publishes "blend recipes" via the company blog, sharing optimisation playbooks and KPIs that cut time-to-value for repeatable use-cases.

Methodology and culture. Projects follow a "many touch-points, zero surprises" cadence: weekly executive stand-ups, sprint demos every ten business days, and a living RAID log that keeps risk, assumptions, issues and dependencies transparent to all stakeholders. Internally, consultants pursue ongoing certification tracks and pair with senior architects in a deliberate mentorship model that sustains institutional knowledge. The result is a delivery organisation that can flex from tactical quick-wins to multi-year transformation roadmaps without compromising quality.

Why it matters. In a market where ERP initiatives have historically been synonymous with cost overruns, HouseBlend is reframing NetSuite as a growth asset. Whether preparing a VC-backed retailer for its next funding round or rationalising processes after acquisition, the firm delivers the technical depth, operational discipline and business empathy required to make complex integrations invisible—and powerful—for the people who depend on them every day.
