デジタル社会推進実践ガイドブック　DS-212

ゼロトラストアーキテクチャ適用方針における属性ベースアクセス制御に関する技術レポート

2023（令和 5）年 3 月 31 日

デジタル庁

〔ドキュメントの位置付け〕
Informative
参考とするドキュメント

〔キーワード〕
アクセス制御、ゼロトラストアーキテクチャ

〔概要〕
クラウド・バイ・デフォルト原則に従い、今後の政府情報システムにおける多くの業務がクラウドサービスを通じて処理される。従来の業務処理環境においても堅牢性を維持・向上するには、サイバーセキュリティを新しい環境に適応させる「ゼロトラストアーキテクチャ」の考え方を組み込むことが重要になる。ゼロトラストアーキテクチャは、業務プロセスに必要な各リソース間のアクセスを、様々な情報から制御することを核としている。本文書ではアクセス制御モデルの１つであり、リソースに付与された属性や環境の情報等を活用した属性ベースアクセス制御に関する俯瞰的な技術的内容を記載する。

改定履歴

改定年月日	改定箇所	改定内容
2023 年 3 月 31 日	－	・初版決定

目次

1. はじめに

2022 年 6 月に刊行された「ゼロトラストアーキテクチャ適用方針」では、ゼロトラストアーキテクチャを「特定の業務フロー内で、あるリソースから別のリソースへのアクセスが最小権限の原則を満たすよう、業務フローを取り巻く環境の情報を活用し、事前に定められたアクセス制御のルールによって評価され、その結果に従うアクセス制御が施行される」考え方であると説明した。その中核はアクセス制御であることは明らかだが、具体的な方法や実装例は示されていない。そこで、本レポートでは、特定のリソースに関する属性を複数組み合わせたアクセス制御「属性ベースアクセス制御（Attribute-Based Access Control: ABAC)」について解説する。

一般的に普及したアクセス制御の方法には、任意アクセス制御（DAC)、強制アクセス制御（MAC)、役割ベースのアクセス制御（RBAC）がある。しかし、これらは「ゼロトラストアーキテクチャ適用方針」が想定する業務環境・業務システムに対して十分かつ継続的な安全性を提供できない恐れがある。クラウド・バイ・デフォルト原則により利活用が拡大されると想定されるクラウドサービスでは、安全性が必ずしも確保されていないインターネット上を通じて、日常の業務と関連するデータが処理されるためである。長期間にわたるインターネット上の業務や、多様な経路を経由する処理では、より侵害される恐れが高まる。サイバーセキュリティの基本的な考え方の一つである多層防御に則り、アクセス制御にもより多くの多面的な情報を活用することが重要になると予想される。例えば、OS に関連する脆弱性は継続的に発見されるため、業務端末における残存脆弱性を突かれる蓋然性もやはり高くなる。そうであれば、リスクに適切に対応する構成情報やパッチ適用状況がシステム保護における重要な情報になる。影響範囲が広く重要な業務であれば、それを評価した上で処理やアクセスを許可する要件が想定される。
また、多くのセキュリティインシデントの起点となるフィッシングサイトへの対策として通常より強固な認証を強制したい場合は、各種リソースの情報に加え、認証手法に関連する情報も取り込む必要がある。

変化する業務環境やリスクに対して、アクセス制御は最小権限の原則を継続的に実現しなければならない。不正アクセスから業務環境を保護するには、識別子（ID）や役割といった単一の情報は、アクセス制御において十分ではない。

ABAC はリソースに付与される属性情報に加え、業務をとりまく環境情報を活用することで、より効果的なアクセス制御を実現できる。

1.1.目的とスコープ

本文書は「ゼロトラストアーキテクチャ適用方針」における補助文書として位置づけられる。属性ベースアクセス制御の説明に加え、ゼロトラストアーキテクチャにおいて採用する際の効果や考慮事項について言及する。情報システ

ム管理者やアクセス管理者等は、本文書を参考にすることで、「ゼロトラストアーキテクチャ適用方針」の実現に対して ABAC がどれほど寄与するか理解できる。

1.2. 適用対象

本文書は、政府情報システムにおけるセキュリティ対策を適用の対象とする。

1.3. 位置づけ

本文書は、標準ガイドライン群の Informative（情報提供）のレベルの参考文書である。

1.4. 本書の構成

第２章ではゼロトラストアーキテクチャにおける ABAC の意義を示すとともに、必要な用語の整理と考慮事項について説明する。ゼロトラストアーキテクチャにおける ABAC に関する知見がない読者は、本章を理解することで、実際にシステムを構築する際の必要事項や運用に関して理解を深めることができる。なお、内容の根拠についての詳しい解説は第３章に譲り、本章ではあくまでも「ゼロトラストアーキテクチャ適用方針」の文脈での解説をする。

第３章では各要素を国際標準や先行研究の事例から ABAC の構成要素を解説する。具体的には ISO/IEC 29416「A Framework for Access Management」や NIST SP 800-162「Guide to ABAC Definition and Considerations」を取り上げる。本章の記載内容より一般的な ABAC の外観を把握することで、ベンダーや組織・チーム外とのコミュニケーションにおける前提や用語を揃えることが容易となる。

1.5. 用語

本文書において使用する用語は、表１-1 及び本文書に別段の定めがある場合を除くほか、標準ガイドライン群用語集の例による。その他専門的な用語については、民間の用語定義を参照すること。

表 １-1　用語の定義

用語	意味
属性（Attribute）	リソースの特性や特質を表す情報名。
属性情報（Attribute Value）	属性に設定された値。

用語	意味
リソース（Resource）	ユーザーオブジェクト、デバイスオブジェクト、アプリケーションなど、アクセス管理の対象となる論理的な主体。一つ以上の属性を付与された主体。
サブジェクト（Subject）	オブジェクトに対する処理を試みるリソース[1]。
オブジェクト（Object）	サブジェクトの処理対象であるリソース。
最小権限の原則（Principle of least privilege）	事前に定義した職種・役職・属性等に応じて、付与するアクセス権限を必要最小限にする原則。
処理（Transaction）	業務の処理や、業務に関連するデータに対する一連の操作。
アクセス制御（Access Control）	サブジェクトからオブジェクトに対する処理やアクセスを、特定のルールやサブジェクトに付与された権限をもとに許可または拒否する機能。
アクセス管理（Access Management）	アクセス制御を管理する一連のプロセス。
アクセス制御ポリシー ポリシー（Access Policy）	特定範囲の管理下において、許容される処理やアクセスを、サブジェクトの権限とオブジェクトを保護するための要件に基づいて規定するルール。
アクセス認証、認証（Authentication）	実在性が確認されている主体とリソースの紐づきを確認する処理。例）従業員（主体）が、情報システム管理者によって作成されたユーザーアカウント（リソース）へのログイン（行為）処理。
アクセス認可、認可	ポリシーとの照合による評価に基づいてアクセスを許可するか決定し、その結果を発行・通知する処理。

[1] NIST SP800-162 にならい、本文書において subject をアクセス元として定義する

用語	意味
(Authorization)	
アクセス施行、施行 (Enforcement)	アクセス認可の結果に基づいて、実際のリソースへのアクセスに適用される処理。

2.アクセス制御と ABAC の概要

2.1. アクセス制御の一般論

ゼロトラストアーキテクチャの適用に関わらず、管理者は業務やセキュリティ要件に基づき、リソースを不正アクセスから保護しなければならない。アクセス管理のうち、リソース間で実行される操作を決定・施行する一連の処理を持つ機能がアクセス制御である。これは主に認証、認可、認可結果の施行の順に実行される。このプロセスに必要となる、主なコンポーネントを次の図に示し、解説する。

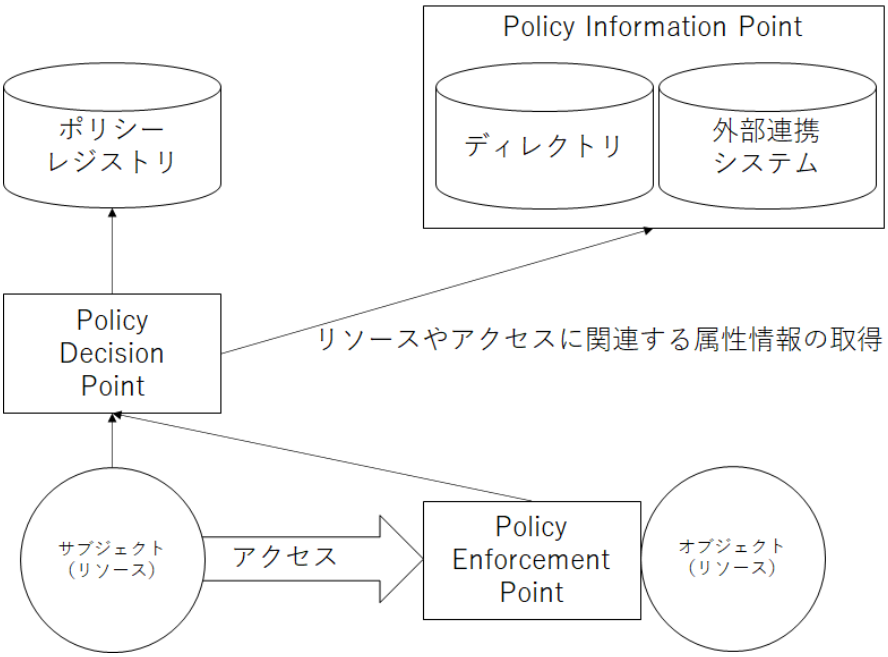

図 1: アクセス制御の概要図

2.1.1 Policy Decision Point(PDP)と Policy Information Point(PIP)

サブジェクトからオブジェクトへのアクセスを確立する、あるいは有効なセッション中にアクセスの維持を試みる際、管理者や管理機能は正当なアクセスであるか検証・評価し、その結果を反映させねばならない。このうちアクセスを評価するのが PDP である。アクセスにかかわるリソースの情報が PDP の入力値となり、その内容を PDP のポリシーが評価する。

リソースに関する情報源はリソース自体であることもあれば、外部システムからの提供、あるいは外部システムへの取得であることもある。このコンポーネントが PIP である。PIP はディレクトリサービスなどの外部システムを指すこともある。PIP からの情報で PDP はアクセス可否を決定する。PDP は更に別のアクセス制御を適用することも可能である。PDP は一連のアクセス制御の結果を何らかのデータに整形して発行する役割も負う。

2.1.2 Policy Enforcement Point(PEP)

PDP の結果を、実際の処理に対して施行するコンポーネントが Policy Enforcement Point (PEP) である。PEP はゲートウェイ的なトポロジー構成をとることもあれば、サブジェクトのエージェントとして動作することもありうる。

なお、PDP も PEP も概念的なコンポーネントである。そのため、導入するサービスや実際に動作するソフトウェアでは PDP と PEP を兼ねるようなものや、それぞれが独立しているものなど、多種多様である。

2.2. ABAC を含む既存のアクセス制御モデル

アクセス制御モデルで特に普及しているものに、任意アクセス制御（Discretionary Access Control, DAC)、強制アクセス制御（Mandatory Access Control、MAC)、役割ベースアクセス制御（Role-Based Access Control、RBAC）があげられる。それらと本ガイドラインの焦点である属性ベースアクセス制御（Attribute-Based Access Control, ABAC）を本章で取り上げる。

2.2.1 Discretionary Access Control (DAC)

DAC は情報システム内のオブジェクトに設定されるアクセス制御ポリシーで、オブジェクトの所有者がサブジェクトの操作可能な範囲を指定する。例えば、UNIX/Linux 上のファイルやフォルダの所有者は、各ファイルやフォルダに対する権限（読み出し、書き込み、実行）を組み合わせて各ユーザーやグループに付与している[2]。

2.2.2 Mandatory Access Control (MAC)

MAC は、情報システム内にあるすべてのサブジェクトとオブジェクトに対して一律に適用されるポリシーによる制御モデルである。オブジェクトに機密性を指定し、サブジェクトに許可された権限をもって、操作可能な範囲を指定する。例えば SELinux では、サブジェクトおよびオブジェクトに sensitivity と category からなるセキュリティレベルを付与し、それとポリシーを用いて、アクセス制御が適用される。DAC 単体と大きく異なる点としては、リソースに名前や識別子以外にレベルという属性を一元管理的に付与し、その属性をベースにしたポリシーもさらに一元管理化したことである。

2.2.3 Role-Based Access Control (RBAC)

RBAC は、サブジェクトあるいはオブジェクトに付与するロール（役割）による制御モデルである。ここでのロールは、組織図を反映した階層構造上で表現される職種や役職のような情報を指す。これらの情報がリソースのロールとい

[2] ファイルやフォルダの識別子ベースでのアクセス制御になることから IBAC（Identifier Based Access Control）の一種であるともいえる。

う属性に設定される。PDP がロールの情報を参照する方法は、処理要求時のリクエストにアクセストークンのような形で含まれる push 型や、ディレクトリ等の PIP から取得する pull 型といったケースに大別できる。

ロールと認可される処理の紐づけは、事前にポリシーとして定義されており、処理の要求が発生した際に PDP によって照合される。ロールを使うことでリソースそのものの情報をポリシーに記述する必要がなくなり、管理が効率化される。一方、特定の情報を定数としてポリシーに記述することから、RBAC は DAC や MAC と同様に静的なアクセス制御モデルとなる。この場合、異動や命名変更が頻繁におこらない比較的不変性の強い組織に有効なモデルとなる。

2.2.4 Attribute-Based Access Control (ABAC)

ABAC は、サブジェクトやオブジェクトの属性に設定された情報や周辺の環境情報をもとにアクセス制御をするモデルである。属性情報を入手する経路としては、RBAC と同様に、リソース自体が提示する場合と PIP から取得する場合が考えられる。RBAC のようにロールのような属性の値を基にする点でも類似しているが、ロール以外の複数種別の属性情報や環境情報を使った複雑なポリシーを定義できる点が最大の特徴になる。

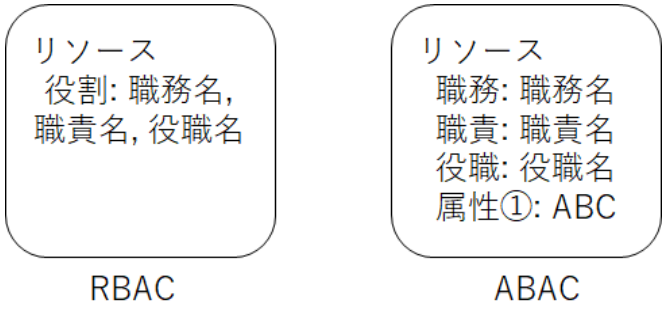

図 2：RBAC と ABAC の違い

2.3. ABAC の特性

DAC, MAC, RBAC はリソースの識別子、機密性の分類、ロールといった情報を用いたアクセス制御モデルである。これらの情報はリソースに属する情報、つまり属性情報である。したがって、これらアクセス制御モデルは簡易的な ABAC といえる。しかし、先述したとおり ABAC 最大の性質は、複数の属性および属性情報、そして処理をとりまく環境情報など、より広範囲で多様な値を使ったアクセス制御にある。その特性を次のように応用することで、柔軟な運用が可能になる。

2.3.1 属性の加工・変換

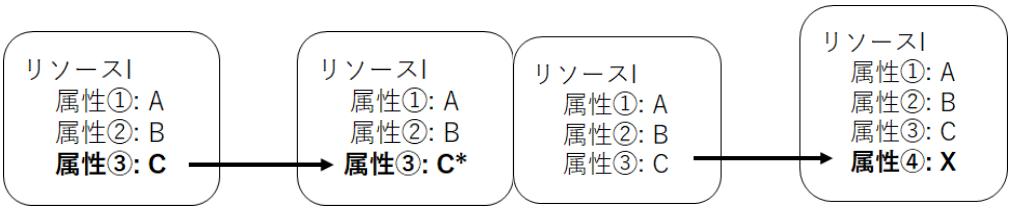

図 3: 属性の変換・加工

ABAC は特定の属性を組み合わせる、グルーピングする、変換するなど行うことで、新たな属性を導き出し、それをアクセス制御に活用できる。例えば、特定の業務契約で日中時間帯以外のアクセスを想定しないユーザーには、ログイン時間のような属性の値を日中時間帯と比較し、true/false のような値をもつ属性を付与できる。また、利用者の情報、通信環境やデータ保護の構成情報などからリスクスコアを算出する常時リスク診断・対処（CRSA）も１つの例である。
ロールのような単一の属性に依拠する制御モデルの場合、運用が進むにつれ属性情報の複雑性も増し、同時に管理も困難になる。ABAC は柔軟な属性の定義のみならず、運用の複雑性への対処にも寄与する。

2.3.2 外部情報の活用

ABAC はリソースの属性情報だけではなく、外部システムから取得した情報を制御に活用できる。

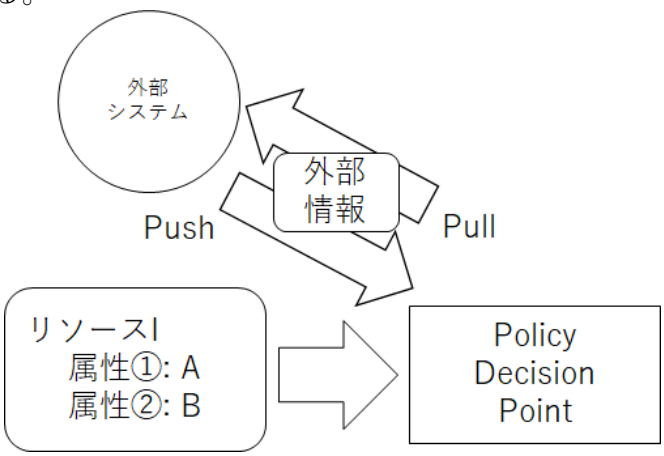

図 4: 属性としての外部情報取り込み

境界防御の観点では外部通信先のドメインや IP アドレスの妥当性を確認し、アクセス可否を決定したいことがしばしばある。ABAC であれば、アクセス先すなわちオブジェクトのドメインや IP アドレスを属性情報とみなし、それらに関する追加的な情報を脅威インテリジェンスサービスから得るといったモデルを構築できる。

3. ABAC とゼロトラストアーキテクチャ適用方針

3.1. ゼロトラストに ABAC を実装する意義

リスクが比較的変わりやすい業務環境を想定する「ゼロトラストアーキテクチャ適用方針」の各適用方針を実現するにあたり、ABAC は次のようなメリットを提供する。

3.1.1 「リソースを識別し、特定できる状態にする」

ゼロトラストアーキテクチャが対象とするリソースの対象はユーザーだけでなく、デバイス、アプリケーション（サービス）も含まれる。つまり管理すべきリソースの量は膨大であることが予想される。そのため、識別や特定作業の効率化が求められる。

DAC や MAC では、事前にリソースの識別子を把握しなければ特定できない。RBAC はロールから絞り込みができるものの、運用が長くなるにつれ様々な値がつくことから、絞り込みもより複雑になる。ABAC であれば属性とその値（情報）を組み合わせて絞り込みができるので、識別と特定作業を効率化できる。

3.1.2 「主体の身元確認・当人認証を実施する」

フィッシング攻撃や不正送金など、その攻撃手法を更新し続ける高い脅威を考慮しなければならない環境において、主体の身元確認・当人認証は重要な処理である。ABAC はその処理の結果を関連する付加情報やメタデータを属性情報に取り込むことで、より強固なアクセス制御を実現できる。

身元確認は従業員と称する個人が実在するか確認する処理である。この際、法規制や標準化された手続きに則ることが、業務処理上、重要になることがある。ABAC では、その手続きやあるいはエビデンス元を属性情報としてとりこむことで、実際の処理におけるアクセス制御に身元確認手続きを統合できる。

また、当人認証においても、認証方法を強固なものに限定すべき要件があることもある。昨今のフィッシング攻撃では単純な二要素認証で十分でない事例も少なくないことから、ハードウェアキーを使った認証を適用する要件も考えられる。そういった認証コンテクストも ABAC では付加的な情報として活用する[3]。

3.1.3 「ネットワークを保護する」

業務処理の安全性を確保する場合は、安全な通信経路を使うことが重要であり、そうでない経路であればアクセスは拒否すべきである。ネットワークの通信経路の安全性は、暗号技術を用いて保護されることが多く、それは暗号化す

---

[3] 3.2 ABAC 適用に関連する留意事項にも記述するが、同時に個人データの機微性についても留意する必要がある。

る経路を確立する前のやりとりから入手可能な情報である。ABAC では、この情報を環境情報やトランザクションにおける属性情報として扱い、安全な通信経路に限定したアクセス制御を実現可能とする。

境界型ネットワークにおいても、各リソースのネットワーク情報を基にネットワーク制御ルールを作成できることは一般的に知られている。それぞれのリソースが参加するネットワークもまた、属性としてみなせるため、境界型セキュリティのような環境であってもABACを適用できる。

3.1.4 「リソースの状態を確認する」

ゼロトラストアーキテクチャが想定する業務環境は潜在的に脆弱である。そういった環境において、リソースや処理環境が正常な状態であるか、意図通りであるかを確実にしなければ安全性を確保できない。従来とは異なる環境からのサインインなどの正常である確証を持てない処理に対して、システム管理者は適切なリスク対応をとる必要がある。検知と事後的な調査なども１つの手段ではあるが、ABAC の特性を活用すれば、過去のサインイン状態やアクセス元のレピュテーションを考慮したアクセス制御により、より防御的な対応が可能になる。

このリソースの正常な状態に対する確証を上げる ABAC の特性は、誤検知の低減にも有効である。単一の情報から判定するのではなく、多角的な追加の情報を求めることができるからである。例えば正当性を判断できないサインインについては、追加の認証を要求することで、正当なユーザーによる行為である確証を高められる。あるいは、サインイン元の端末を、資産情報と照合し実在性の確認や、構成情報から最新のパッチ適用状態を確認することで、多層防御をより強化できる。ABAC の特性はこのようにトランザクションをとりまく各種リソースの属性情報を統合して判定することで、ゼロトラストアーキテクチャをより安全性が高く効率的なものに実装できる。

3.1.5 「アクセス制御ポリシーで、アクセスを管理する」

ゼロトラストアーキテクチャは特定の業務内で複数のアクセス制御を要する一連の手続きを踏む考え方である（図 4)。その中でアクセス制御モデルはそれぞれの手続きにおいて排他的である。例えば、図4におけるリソースA～B間でABAC を実装し、リソース Y～Z 間では MAC を実装することも可能である。既存のアクセス制御モデルをABACに置き換えたとしても、全体のリプレイスにはならないため、影響範囲のコントロールが容易になる。

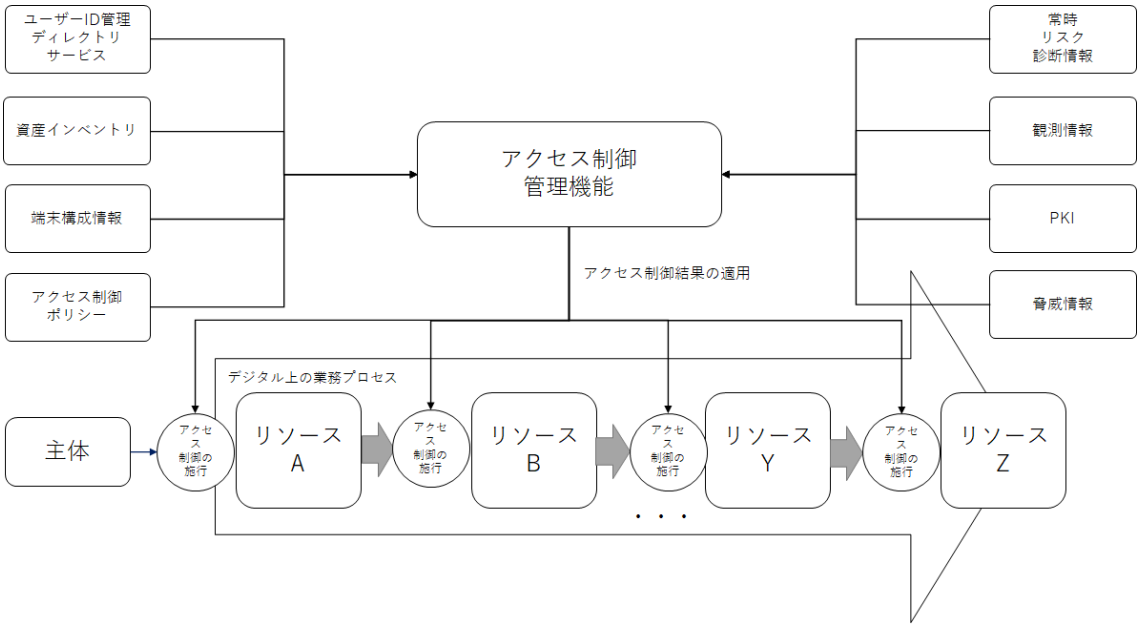

図 5: ゼロトラストアーキテクチャ概念図[4]

3.1.6 「リソースとアクセスを観測する」

本項目については、観測に直接的な ABAC 適用のメリットがあるとは考えられない。管理画面や管理サービスなど、観測に関連するリソースに対するアクセス制御にも ABAC を適用できる。

3.2. ABAC 適用に関連する留意事項

ABAC をそのアクセス制御モデルとして採用した際の特有の留意事項が考えられるので、次にあげる。

3.2.1 個人データへの留意

アクセス制御に関連するシステムを運用するうえで、個人の特定が可能な属性、あるいは個人データそのものが値になる属性がある場合、個人の権利・利益保護のために、個人情報保護法を遵守しなければならない。

3.2.2 運用・保守体制の確保

属性情報、処理に関連する環境情報など ABAC 特有のコンポーネントが存在する。これらは、アクセス制御機構におけるシステム管理者のみで決定できない。そのため、各リソースを管理する担当者や外部システムの管理者など多岐にわたるステークホルダーを識別・特定し、調整をしなければならない。

---

[4] ゼロトラストアーキテクチャ適用方針: ゼロトラストアーキテクチャ　適用方針（digital.go.jp）　図 1-1 参照

3.2.3 属性情報を包有するデータの構造化と標準化

属性および属性情報をアクセス制御に用いる以上、それらを含む情報は機械可読形式なデータ構造を持っていることが望ましい。そのため、属性名や属性情報のデータ型、属性情報として設定可能な値といったものを事前に定め、標準化する必要がある。また、標準化したものを各種ステークホルダーに採用されやすくするため、普及活動や運用支援などの支援活動を行うことが望ましい。

3.2.4 属性情報の送受信にける安全性確保

アクセス制御の入力値として活用する以上、属性の値である属性情報の正確性や信頼性を確保しなければならない。そのためには前述した通り、標準化した構造を運用することに加え、受理したデータを検証する仕組みが必要である。

この際に、不正なエンティティによる改ざん、あるいは不正なエンティティからのなりすましなどの脅威を考慮しなければならない。

また、異なる管理下にあるシステムへのデータ伝送方法についても安全性を確保すべきである。データ構造の標準化に加え、送受信するデータ処理についてポリシーや技術を確立しなければならない。

3.2.5 データ種別や量の考慮

各種リソースの属性や関連する環境情報を活用するため、全体的なデータ量の増加が想定される。そのため、アクセス制御そのものにおける処理速度への影響が考えられるため、適切に容量を見積った設計をしつつ、パフォーマンスの監視などの観測による運用をすべきである。

又、複数の属性や属性情報を組み合わせることで様々な分析が可能となるが、本来の管理業務外における用途に繋がらないよう管理するべきである。具体例として、運用目的にそぐわない個人の特定などが挙げられる。

4.参考資料

ゼロトラストアーキテクチャ適用方針: ゼロトラストアーキテクチャ　適用方針 (digital.go.jp)

DS-211 常時リスク診断・対処 (CRSA) アーキテクチャ: 常時リスク診断・対処 (CRSA) システムアーキテクチャ (digital.go.jp)

[FK92] Ferraiolo, D.F. & Kuhn, D.R. (October 1992). "Role-Based Access Control" (PDF). 15th National Computer Security Conference: 554-563.

SP 800-162, Guide to ABAC Definition and Considerations | CSRC (nist.gov)

SP 800-205, Attribute Considerations for Access Control Systems | CSRC (nist.gov)

SP 800-207, Zero Trust Architecture | CSRC (nist.gov)

ISO/IEC 29146 Information technology - Security techniques - A framework for access management

OpenID Connect: Core Final: OpenID Connect Core 1.0 incorporating errata set 1

OpenID Connect for Identity Assurance: OpenID Connect for Identity Assurance 1.0

OpenID Connect: Shared Signals Framework: Software Grant and Contribution License Agreement | OpenID