CSA ガイダンス version 4.0 を用いた
クラウドセキュリティリファレンス
(OSS マッピング 2019)

一般社団法人 日本クラウドセキュリティアライアンス（CSA ジャパン）
クラウドセキュリティワーキンググループ

目次

- 本書執筆編集メンバー

氏名	所属
井上 淳	NTT テクノクロス株式会社
釜山 公徳	日本電気株式会社
福田 貢士	（個人会員）
森田 翔	（個人会員）

※五十音順

- 変更履歴

日付	版数	変更内容
2019 年 2 月 26 日	1.0	初版発行

- 著作権についての留意事項

1. はじめに

昨今、パブリッククラウドの利用について、業界に関わらず移行が進んでおり、銀行の勘定系システムであっても移行する動きが見られる。このような動きでクラウドジャーニーが激化し、従来のオンプレミス環境とのハイブリッド環境、場合によっては全てのシステムをクラウドへ移行、さらには新規事業においては最初からクラウドのみを利用する等、クラウドネイティブ化が進んでいる。

この理由として、初期投資の不要やスケーリング技術によるビジネスアジリティ、ハードウェア保守の不要や従量課金制によるコスト最適化、さらには各クラウドベンダーが提供する利便性の高いマネージドサービスによる設計・構築負荷の低減などが挙げられる。

まさにクラウドの時代である。

ただし、メリットばかりではなく、デメリットや注意点も存在し、デメリットとしてはクラウドを効果的に利用しないことによるコスト増、ベンダー責任範囲のブラックボックス化、注意点についてはクラウドベンダーによるベンダーロックイン、セキュリティ施策等が挙げられる。

ここでセキュリティに着目すると、各クラウドベンダーでは自身の責任範囲内においては各団体の第三者認証を得ていることから強固なセキュリティを実装しているといえるものの、利用者側の範囲ではその限りではないのではないかと考えられる。利用者側でどのような施策を行えばよいのかについては、まとまった情報は散見されるものの、ほとんどが各セキュリティベンダーからの情報であり、これらは中立性に欠けると言える。

そこで、本書においては、中立性と参照性を鑑みた参考情報として、2017 年にクラウドセキュリティアライアンスにて公開した”クラウドコンピューティングのためのセキュリティガイダンス バージョン 4”と、Open Source Software(OSS)によるセキュリティマッピングを行うこととした。

本書は、利用者側の施策を中心に記述しており、クラウドを利用する多くの方々への一助となれば幸いである。

CSA ジャパン クラウドセキュリティ WG リーダー 釜山 公徳

2. 検討指針

2.1. 目的

Cloud Security Alliance (CSA) において、クラウドコンピューティングのためのセキュリティガイダンス(以下、ガイダンス)といったセキュリティリファレンスはあり、大変有用である。しかしながら、実環境における設計や実装といったフェーズで具体的な検討をするにあたり、人によってはイメージがわかないことがある。そこで、Open Source Software(OSS)とガイダンスをマッピングさせることで、具体的な施策のイメージへの一助になると考えた。なぜ、OSS か。それは、CSA はニュートラルな団体であり、ニュートラルなソフトウェアである OSS との親和性が高いため、題材として選択した。

2.2. 前提

以下 3 事項を前提とする。

① 記載の方針

ガイドラインの要約、ならびにワーキンググループにて独自に解釈したものとする。

② マッピングの正確性

現状、知り得る範囲で執筆しているため、正確性の保証はない。

③ 対象ドメイン

OSS にマッピング可能な技術要素を対象とし、1 章から 5 章、9 章、13 章は対象外とする。

表 2.2-1 本書におけるガイダンスの対象ドメイン

本書の対象	ドメイン No.	ドメインタイトル
	ドメイン 1	クラウドコンピューティングのコンセプトとアーキテクチャ
	ドメイン 2	ガバナンスとエンタープライズリスクマネジメント
	ドメイン 3	法的課題：契約と電子証拠開示
	ドメイン 4	コンプライアンスと監査マネジメント
	ドメイン 5	情報ガバナンス
✔	ドメイン 6	管理画面と事業継続
✔	ドメイン 7	インフラストラクチャ・セキュリティ
✔	ドメイン 8	仮想化とコンテナ技術
	ドメイン 9	インシデント対応、通知、および被害救済
✔	ドメイン 10	アプリケーションセキュリティ
✔	ドメイン 11	データセキュリティと暗号化
✔	ドメイン 12	アイデンティティ、権限付与、アクセスの管理
	ドメイン 13	Security as a Service
	ドメイン 14	関連技術

3. DOMAIN6 管理画面と事業継続

3.1. 概要

クラウドでは従来のインフラストラクチャとは異なり、管理インターフェイスへのアクセス方法が多岐に渡ることから、不正アクセス防止のために、誰が、いつ、どこから、どのようにアクセスしたのかを適切に管理する必要がある。そのため、本ドメインでは、管理画面や管理インターフェイス（Web コンソール、API 含む）は、クラウドコンピューティングでの重要な要素であり、適切なアクセス制限、利用者認証、アクセス監視などの施策が必要と述べている。

3.2. 解説

本章では、管理画面と管理インターフェイス（Web コンソール、API 含む）を利用する上で必要とされるセキュリティ施策の項目（要素）を記載する。

表 3.2-1 必要なセキュリティ施策の項目

主項目	要素
管理画面、管理インターフェイスへのアクセス管理	1. 適切なアクセス制限・分離
	2. 特権ユーザカウントの管理
	3. 境界での防御
	4. 利用者認証
	5. 適切な利用者認証
	6. アクセスの記録・定期的な監視

(1) 管理画面、管理インターフェイスへのアクセス管理

① 適切なアクセス制限・分離

アクセス権限は、利用用途に即してアカウントを作成し、業務レベル、役割に応じて適切に付与する。昨今 ICT 環境が多様化していることから、テレワークによるリモート拠点からのアクセス、BYOD によるアクセス、API 経由でのアクセスといった様々なシチュエーションへの考慮が求められている。その上で、誰が何のリソースにアクセス出来るかを定義し、アカウントの分離やアクセス権限の付与が望ましい。

② 特権ユーザカウントの管理

特権ユーザは、常時利用可能な状況を無くし、必要な時に必要な範囲だけ利用可能にすることが望ましい。そのためには、申請によって利用可能とするワークフローを策定し、利用時間の制限をした上で特権ユーザの払い出しや管理が必要である。また、特権ユーザの管理において、定期的な棚卸しといった運用を取り入れることで、利用していないアカウントの洗い出しを行い、不正利用によるリスクの削減が図れる。

③ 境界での防御

外部ネットワークから管理インターフェイスへの脆弱性攻撃、DDoS攻撃といった不正なアクセスに対

し、各境界で防御手段の実装が必要である。

境界の定義は環境やその利用用途によって異なり、防御手段はそれぞれに応じた適切なレベルでの実装が求められ、例えば、ネットワークファイアウォール、IDS/IPS、WAF、リバースプロキシなどが挙げられる。

④　適切な利用者認証

利用者が管理ダッシュボードへの認証を行う上で、次のような安全な仕組みを利用する必要がある。

- 暗号化
- 電子署名の利用
- 規格化された認証技術の利用（OpenID Connectなど）

また、利用者認証において、特に特権ユーザや権限レベルの高いユーザでは、単一の認証方式ではなく多要素認証（MFA）の実装が重要である。

⑤　アクセスの記録・定期的な監視

誰が、いつ、どのように管理ダッシュボードを利用し、何のリソースにアクセスしているかといった情報について、定常的にモニタリングするために、ロギングの仕組みを整備しておく必要がある。

ロギングの実装は、インシデント発生時といった有事の際、迅速かつ適切に調査可能にするため、前述の情報を整理し、アラートシステムやチケットシステムとの連携も考慮しておくことが望ましい。

3.3. 対応するOSS

本章では、3.2で記載した施策事項に応じてOSSを整理し、一覧化した。

尚、記載するOSSは一例であり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要である。

表 3.3-1 Domain6 目的別OSS適合表

目的	施策/機能	対応OSS	ライセンス	OSSの説明
適切なアクセス管理	認可	OpenAM	CDDL	元々商用の製品がオープンソース化された経緯を持ち、SAML2.0、OAuth2.0、OIDC1.0をはじめ多くの標準プロトコルに対応している。 SAMLに対応しているため、Google Appsや Salesforceなどと連携するといった事例が多数存在する。
特権ユーザカウントの管理	ID管理	OpenIDM	CDDL	アイデンティティ情報のプロビジョニングとライフサイクル管理を実現。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	ID 管理	LISM	LGPL	オープンソースの LDAP サーバ OpenLDAP の Perl バックエンドとして動作する。 LISM のほか、OpenSSO／OpenAM、OpenLDAP といったオープンソースの製品を組み合わせることによって、シングルサインオンと ID 管理を実現。
境界での防御	FW	pfSense	BSD	FreeBSD を基にしたファイアウォールとルーター、VPN、プロキシ機能を保有するソフトウェア。 pfSense のパッケージリポジトリからプラグインを入手することで、様々な機能を拡張することもできる。
	IPS	Suricata	GPLv2	高速で堅牢な脅威検出エンジン。リアルタイムで侵入検知や防御ができ、ルールと署名を利用したネットワークトラフィック検査や、Lua スクリプティングを利用した複雑な脅威の検出といった機能を持つ。 非営利団体 Open Information Security Foundation(OISF)がオープンソースプロジェクトとして開発している。
	IPS	Snort	GPLv2、Proprietary Snort	ネットワーク型 IDS/IPS(不正侵入検知システム)。商用製品にも利用されるなどの実績がある。
	WAF	ModSecurity	GPLv2	Trustwave 社がオープンソースとして提供する ModSecurity は WAF のひとつで、Apache のモジュールとして動作する。 「Core Rule Set」という攻撃パターンルールセットと組み合わせて利用することで効果を発揮する。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	Reverse Proxy	Nginx	BSD-like	処理性能/並行処理/メモリ使用量削減にフォーカスして開発されている Web サーバ/リバースプロキシです。ロードバランサーや HTTP キャッシュのなどの機能を備える。 Apache の代替えとしても利用されている。
適切な利用者認証	認証	OpenAM	**表 3.3-1** 適切なアクセス管理参照	
	MFA	Google Authenticator	Apache License 2.0	Google が開発している多要素認証用トークンソフトウェア。
	MFA	mfa	MIT	Python 製のオープンソース・ソフトウェア。 コマンドラインでのマルチファクタ認証を実現する。
アクセスの記録・定期的な監視	収集 分析 可視化	ELK Stack Elasticsearch logstash kibana	Apache License 2.0	Elasticsearch(蓄積) + Logstash(収集) + Kibana(可視化)の３つの製品を利用したログの分析システム。 総称して「ELK Stack」とも呼ばれる。

4. DOMAIN7 インフラストラクチャ・セキュリティ

4.1. 概要

ガイダンスの第 7 章では、クラウドを利用することにより、これまでのインフラストラクチャ・セキュリティにおける実装方法が大きく変わることを示唆している。

インフラストラクチャは、以下の表に記載の通り、クラウド利用者が管理する区分とクラウド事業者が管理する区分とで分類でき、本書ではクラウド利用者に焦点を当てた。

表 4.1-1 インフラストラクチャの種類

種類	概要
クラウド利用者が管理するインフラストラクチャ	リソースプールから使用するリソース （例）抽象化されたプロセッサ、メモリ、ストレージ、ネットワークなど
クラウド事業者が管理するインフラストラクチャ	クラウドサービスを構築するリソース （例）物理的なプロセッサ、メモリ、ストレージ、ネットワークなど

当ガイダンスでは、特に「ネットワークの仮想化」、「SDP（Software Defined Perimeter）」と「ワークロード」に焦点を当てている。

ネットワークの仮想化については当ガイダンスのドメイン 7、ワークロードセキュリティ（コンテナ）についてはドメイン 8 仮想化とコンテナ技術を参照願いたい。

「SDP」はデバイス認証およびユーザ認証を統合し、リソースへのアクセスを動的にプロビジョニングする概念である。本書では SDP として OSS をマッピングせず、デバイス認証とユーザ認証を分けてマッピングした。

また、実際にクラウド上でセキュリティ施策を実装することを含めて検討することとし、クラウドサービスにおける一般的な脅威についても検討した対策を追加した。

4.2. 解説

(1) ネットワークの仮想化

仮想ネットワークは、クラウドを構築する技術の一つであり、クラウド利用者がクラウドを利用する際、技術的な詳細設定については、特に意識しなくてもよい。

VLAN はクラウド規模の仮想化やセキュリティ用途に策定されていないが、プライベートクラウドを自身で構築する際や、既存システムとクラウドサービスをハイブリッドとして利用する際には、VLAN も考慮すべきであるが、本書では関連性が低いことから対象外とした。

また、SDN については、本書では利用者が実装する場合の OSS について記載している。

(2) ワークロード

ワークロードは処理の単位を示す。クラウド環境において、ワークロードを分離、隔離することで、他

のワークロードや実行環境からの影響を受けないようにし、自身と他のワークロードを保護する。

表 4.2-1 ワークロード

目的	概要
ワークロードの分離、隔離	仮想マシンやコンテナ技術などにより他のワークロードからの影響を受けないようにすること。
変更無用(immutable)のワークロード	ワークロードに対するパッチや変更の影響を受けつけないワークロードのこと。 パッチの適用などは、新しくイメージを作成し新しいインスタンスと入れ替え、ワークロードを更新する。
ロギング/モニタリング	ワークロードで利用するファイルなどの完全性が保たれていることや、ワークロードのサービスインや停止のモニタリングを行い、不正な利用を検出する。
脆弱性診断	利用するワークロードに脆弱性が存在していないか診断を行うこと。

③一般的なインフラストラクチャ・セキュリティ

メール、Web 対策などの経路での多層防御で行う対策について OSS を紹介する。

表 4.2-2 ワークロード

目的	概要
ネットワーク脅威の検出と防御	クラウドサービスへの外側、内側からの攻撃を防御。
安全な接続環境の確保	クラウドサービスへ接続する経路の安全を確保。
資産に対する可用性の確保	クラウドで利用するリソースの管理、障害監視。
パッチ(シグネチャ)マネジメント	使用しているソフトウェアの管理。
ロギングとモニタリング	利用者、操作のログ収集。
セキュリティ診断	クラウドインフラ基盤に対する定期的な診断。

4.3. 対応 OSS

本章では、4.2 で記載した施策事項に応じて OSS を整理し、一覧化した。

尚、記載する OSS は一例となり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要である。

表 4.3-1 Domain7 目的別 OSS 適合表

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
ネットワーク仮想化	SDN	OpenDaylight	EPL	SDN においてネットワーク構成や機能を集中的に制御するソフトウェア「SDN コントローラ」機能を提供。
		Ryu SDN Framework	Apache License 2.0	
		Open vSwitch	Apache License 2.0	
ワークロード	ワークロードの分離	Jenkins + Ansible + Packer		マシンイメージを管理する Packer とソフトウェアのビルド等の作業を自動化する Jenkins、構成管理ツール Ansible の組み合わせにより、マシンイメージを自動生成。
		Jenkins	MIT	
		Ansible	GPLv3	
		Packer	MPLv2	
	変更無用(immutable)のワークロード	―	―	―
	ロギング/モニタリング	―	―	―
	プラットフォーム脆弱性診断	OpenSCAP	GPLv2.1	SCAP（NIST によって規定された情報セキュリティ施策の自動化と標準化のための規格）を用いて OS のシステム設定や脆弱性への対応状況などを検査するツール群。 該当するシステムがどのようなセキュリティ設定になっているか、どこまで脆弱性対応を行っているかなどをファイルとして出力できる。
	プラットフォーム脆弱性診断	OpanVAS	GPL	脆弱性のスキャニングと脆弱性の管理の機能を提供するサービスおよびツール群からなるソフトウェアフレームワーク。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	プラットフォーム脆弱性診断	Vuls	GPLv3	Linux/FreeBSD 向けの脆弱性スキャンツール。 エージェントレスのアーキテクチャを採用しているため導入が非常に簡単である。OS としては Ubuntu、Debian、CentOS、Amazon Linux、RHEL に対応。
ネットワーク脅威の検出と防御	DDoS 対策	go-dots	Apache License 2.0	DOTS プロトコルを用いた DDoS 対策ソフトウェア。
	WAF	ModSecurity	Apache License 2.0	Trustwave 社がオープンソースとして提供する ModSecurity は Web アプリケーションファイアウォール(WAF)のひとつで、Apache のモジュールとして動作する。 「Core Rule Set」という攻撃パターンルールセットと組み合わせて利用することで効果を発揮する。
	WAF	WebKnight	GPL	AQTRONiX 社がオープンソースで開発・提供している IIS 向けのホスト型 WAF 。
	不正アクセス及び不正侵入の検出(防御)	Snort + ACID		パケットスニファ/パケットロガー及びプリプロセッサ機能を提供する Snort と分析する ACID(Analysis Console for Intrusion Databases)の組み合わせ。
		Snort	GPLv2 SNORT LICENSE	
		ACID	GPLv2	

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	不正アクセス及び不正侵入の検出(防御)	Suricata	GPLv2	高速で堅牢な脅威検出エンジン。リアルタイムで侵入検知や防御ができ、ルールと署名を利用したネットワークトラフィック検査や、Lua スクリプティングを利用した複雑な脅威の検出といった機能を持つ。 非営利団体 Open Information Security Foundation(OISF)がオープンソースプロジェクトとして開発している。
	マルウェア対策(メールゲートウェイ)	Proxmox Mail Gateway	AGPLv3	受信メールと送信メールを制御し、スパム、ウイルス、フィッシング、トロイの木馬からユーザーを保護 Postfix Mail Transport Agent (MTA)、ClamAV アンチウィルスエンジン、Apache SpamAssassin プロジェクトなどのメールフィルタリング用のさまざまなサービスが適用されている。
		cuckoo	GPLv3	マルウェアを解析するサンドボックスツール。
		SpamAssassin	Apache License 2.0	スパムメールフィルタツール。
		Rspamd	Apache License 2.0	スパムメールフィルタツール。
	マルウェア対策(Web ゲートウェイ)	―	―	

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	ネットワーク監視(パケットスニッフィング)	WireShark	GPLv2	ネットワークプロトコルアナライザ(パケット取得/プロトコル解析ツール) 多くのプロトコルに対応し、ネットワークインターフェース上を通過するネットワークパケットをキャプチャして収集しリアルタイムで調査/解析することができる。 また多くにプラットフォームに対応している。
	ネットワーク監視(パケットスニッフィング)	tcpdump	BSD	ネットワークトラフィックをキャプチャするためのパケットアナライザ。 UNIX/Linux 環境で利用できる。
安全な接続環境の確保(ハイブリットクラウド利用を想定)	VPN の利用	OpenVPN	GPLv2	暗号化されたトンネルを作成するソフトウェア。
	専用線の利用	―	―	
	踏み台経由のアクセス(要塞仮想ネットワーク)	iptables + OpenSSH		ネットワーク制御を行う iptables と通信を暗号化するプロトコル ssh の組み合わせで実装。 iptables は Linux に実装されたコマンドのひとつ。
		iptables	GPLv2	
		OpenSSH	BSD	
資産に対する可用性の確保	オートスケーリング	OpenStack	Apache License 2.0	クラウド基盤ソフトウェア。 Red Hat、IBM、ヒューレットパッカード（HP）の他、サービスプロバイダーとしては RackSpace が、システムインテグレーターである Mirantis が積極的にコントリビューションしている。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	オートスケーリング	CloudStack	Apache License 2.0	クラウド基盤ソフトウェア。 Citrix、システムインテグレーターである Clogeny、サービスプロバイダー Shubergphilis、Leaseweb などが積極的にコントリビュージョンしている。
	耐障害監視(ヘルスチェック)	Hinemos	GPLv2	システム稼働状況を監視。一般的に監視機能やジョブ管理機能などは機能単位のソフトウェアとして提供されるケースが多くあるが、Hinemos は、「収集・蓄積」、「監視・性能」、「自動化」の機能をすべてワンパッケージで提供している。
	耐障害監視(ヘルスチェック)	Zabbix	GPLv2	Zabbix 社が開発している、Web インタフェースを備えたネットワーク監視ツール。データは RDB に格納される。商用サービスも提供されている。
	耐障害監視(ヘルスチェック)	Nagios	GPLv2	監視をプラグインによって行うのが特徴のファイルベースのネットワーク監視ツール。コミュニティ活動が活発で、多数のプラグインが公開されている。
パッチ(シグネチャ)マネジメント	適切なソフトウェアバージョンの管理	Puppet	Apache License 2.0	システム構成管理ツール。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	適切なソフトウェアバージョンの管理	Ansible	GPLv3	レッドハットが開発するオープンソースの構成管理ツール。ファイルに記述した設定内容に応じて自動的にユーザの作成やパッケージのインストール、設定ファイルの編集などを行うツール。構成管理に加え、オーケストレーションやソフトウェアデプロイメントの機能を持つ。
ロギングとモニタリング	セキュリティイベントに対するロギングとモニタリング	OSSIM ＋ 各センサ	GPLv2	SIEM 機能を提供する OSSIM とシステム稼働情報を収集するセンサの組み合わせ。
セキュリティ診断	プラットフォーム診断	**表 4.3-1** Domain7 目的別 OSS 適合表 “ワークロード”参照		
	ペネトレーションテスト(侵入テスト)	OWASP Zed Attack Proxy	Apache License 2.0	Web アプリケーション脆弱性診断ツールであり、主に次の 3 つのチェック方法で Web アプリケーションの脆弱性を確認する。 1. 簡易スキャン 2. 静的スキャン 3. 動的スキャン
	ペネトレーションテスト(侵入テスト)	BeEF	GPLv2	The Browser Exploitation Framework の略称。 ブラウザのペネトレーションテスト。

5. DOMAIN8 仮想化とコンテナ技術

5.1. 概要

本ドメインでは、クラウドコンピューティングのコアテクノロジーである仮想化のセキュリティを対象としている。ガイダンスでは、仮想化環境におけるセキュリティの観点としては、基盤となるインフラの物理的な制御に加えて、次の 2 つのレイヤーについて新たに考慮が必要であると述べている。

- ハイパーバイザー等の仮想化技術についてのセキュリティ
- 仮想ファイアウォール等による仮想資産のセキュリティ制御

クラウド事業者が物理インフラや仮想プラットフォームのセキュリティについて責任を負う一方で、クラウド利用者は用意された環境適切に実装する必要がある。例えば、仮想化ストレージを暗号化するタイミングや、仮想ネットワークと仮想ファイアウォールの設定、専用ホストと共有ホストのどちらを利用するかなどの決定である。

本ドメインでは、クラウドコンピューティングにおける仮想化技術の柱となる以下の 4 つのカテゴリについて、それぞれのセキュリティ施策を述べている。

- コンピュート
- ネットワーク
- ストレージ
- コンテナ

5.2. 解説

(1) コンピュート

従来は仮想マシンによって行われてきたが、コンテナ型やサーバレス型のコンピュートの抽象化も進んでいる。ガイダンスでは、クラウド利用者の責任範囲として、仮想インフラを管理するためのセキュリティ制御手段として一般的に以下のようなものがあると述べている。

- クラウドリソースに関するアイデンティティ管理（ドメイン 6 の内容）
- 監視とロギング（ドメイン 7 の内容に加え、仮想マシンのステータス・管理イベント・パフォーマンス）
- マスタイメージ（リポジトリ）管理
- 必要に応じた専用ホスティングの使用

さらにガイダンスでは、以下のような仮想化されたリソース内のセキュリティ制御もクラウド利用者の責任範囲であると述べている。

- 仮想マシン、コンテナ、アプリケーションコードなどの（仮想化環境に限らない）すべての標準的セキュリティ

- パッチが適用された最新の仮想マシンイメージなどの安全な構成設定の配備
- サーバレス型配備の場合のホストレベルの監視／ロギングに代わる代替手段（例：通常より堅牢なアプリケーションログ）

(2) ネットワーク

ネットワークの仮想化技術は複数あるが、多くのクラウドコンピューティングでは SDN を使用している。

SDN 環境では、ネットワーク上に組み込まれた監視やフィルタリングのツールではトラフィックを検知できないため、仮想ネットワーク用のツールが必要となる。ガイダンスでは、仮想ネットワークの管理について、クラウド利用者の責任として、仮想ファイアウォールの配備を適切に設定する責任があると述べている。

(3) ストレージ

仮想化ストレージについて、ガイダンスでは、物理ストレージの暗号化でもある程度カバーできるが、データをクラウド事業者に見られないよう保護するためには仮想化レイヤーでの暗号化が必要であると述べられている。これらの追加的施策はドメイン 11 にて記載されている。

(4) コンテナ

コンテナのセキュリティについて、ガイダンスでは以下が必要であると述べられている。

- 基盤となる物理インフラのセキュリティ確保
- 管理ダッシュボードのセキュリティ確保
- イメージリポジトリの適切な保管
- コンテナ内で実行されているタスク／コードのセキュリティ
- イメージやコンテナの構成設定

5.3. 対応 OSS

本章では、5.2 で記載した対策事項に対応する OSS を整理したものである。

尚、記載する OSS は一例となり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要。

表 5.3-1 Domain8 目的別 OSS 適合表

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
ハイパーバイザの安全性確保	アクセス制御	―	―	KVM や Zen などの仮想化基盤向けの OSS は存在するが IaaS 向けの OSS は存在しない。
	脅威検出	―	―	VMWare 向けの商用の製品は存在するが OSS では存在しない。

<table>
<tr><th>目的</th><th>施策/機能</th><th>対応 OSS</th><th>ライセンス</th><th>OSS の説明</th></tr>
<tr><td rowspan="6">仮想マシンの適切な管理</td><td>アイデンティティ管理</td><td>OpenIDM
OpenAM</td><td colspan="2">表 5.3-2 Domain6 目的別 OSS 適合表 参照</td></tr>
<tr><td>監視とロギング</td><td>Zabbix</td><td>GPLv2</td><td>Zabbix 社が開発している、Web インタフェースを備えたネットワーク監視ツール。データは RDB に格納される。商用サービスも提供されている。</td></tr>
<tr><td rowspan="4">マスタイメージ（リポジトリ）管理</td><td colspan="2">Jenkins + Ansible + Packer</td><td>AMI 作成の自動化
Ansible：構成管理
Packer：マシンイメージ自動作成
Jenkins：CI ツール</td></tr>
<tr><td>Jenkins</td><td>MIT</td><td rowspan="3"></td></tr>
<tr><td>Ansible</td><td>GPLv3</td></tr>
<tr><td>Packer</td><td>MPL2.0</td></tr>
<tr><td>専用ホスティングの使用</td><td>－</td><td>－</td><td></td></tr>
<tr><td rowspan="3">仮想リソース内のセキュリティ制御</td><td>仮想マシン、コンテナ、アプリケーションコードの標準的セキュリティ</td><td>－</td><td>－</td><td>具体策についてはガイダンスで省略されているため割愛</td></tr>
<tr><td>安全な構成設定の配備</td><td colspan="3">表 5.3-3 Domain8 目的別 OSS 適合表 “マスタイメージ（リポジトリ）管理”参照</td></tr>
<tr><td>サーバレス型配備の場合のホストレベルの監視／ロギング</td><td>－</td><td>－</td><td>アプリケーションにて実装する必要がある。</td></tr>
<tr><td>仮想ネットワークのセキュリティ</td><td>仮想ネットワーク用のトラフィック監視</td><td>Zabbix</td><td colspan="2">表 5.3-4 Domain8 目的別 OSS 適合表“監視とロギング”参照</td></tr>
</table>

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	仮想ファイアウォールの適切な設定	VyOS	GPLv2	ソフトウェアルータ。日本では VyOS も有名だが、グローバルでは pfSence のほうがメジャーである。
		pfSense	Apache License 2.0	FreeBSD を基にしたファイアウォールとルーター、VPN、プロキシ機能を保有するソフトウェア。 pfSense のパッケージリポジトリからプラグインを入手することで、様々な機能を拡張することもできる。
コンテナのセキュリティ	物理インフラのセキュリティ確保	－	－	具体策についてはガイダンスで省略されているため割愛。
	管理ダッシュボードのセキュリティ確保	OpenIDM OpenAM	**表 5.3-5** Domain6 目的別 OSS 適合表 参照	
	イメージリポジトリの保護	Rancher + Harbor		Docker イメージ管理のためのプライベートレジストリ。
		Rancher Harbor	Apache License 2.0	
	コンテナ内で動作するタスク・コードのセキュリティ	－	－	
	コンテナ構成設定のセキュリティ	－	－	仮想化インフラにより提供される。
	コンテナ環境における認証	－	－	仮想化インフラにより提供される。

6. DOMAIN10 アプリケーションセキュリティ

6.1. 概要

本ドメインでは、クラウドコンピューティングにおいて特に IaaS と PaaS で、アプリケーションをセキュアな開発配備しようと考えているソフトウェア開発と IT のチームを対象としており、開発、配備プロセスにおける主たるものに焦点を当てた対策が記載されている。

6.2. 解説

本章では、アプリケーションの開発~運用を行う上で必要とされるセキュリティ施策の項目（要素）を記載する。

表 6.2-1 Domain10 必要なセキュリティ施策の項目

主項目	要素
利用状況の可視化	アプリケーション操作のモニタリング・ロギング
管理画面・管理インターフェイスのアクセス監視	アクセスの記録・定期的な監視
透明性の制約	外部サービスとの連携管理、アプリケーション間のアクセス制御（API など）
セキュアな設計と開発の管理	開発プロセスの管理
セキュアな配備計画	コードレビュー
	機能テスト
	静的アプリケーション セキュリティ テスト（SAST）
	動的アプリケーション セキュリティ テスト（DAST）
	侵入テスト
	アプリケーション CI/CD パイプラインのセキュリティ
セキュアな運用管理	脆弱性評価（継続的実施）
	開発コード変更管理

(1) 利用状況の可視化

① アプリケーション操作のモニタリング、ロギング

アプリケーションに対する操作の記録を一般公開されている「セキュリティログ管理ガイドライン」を参考に行い、適切な保管を行なう必要がある。ログのモニタリングについては、不審な操作に関するアラートの設定および、定常的なモニタリングによる監視を行なうことで、適切な利用が成されているかを確認することができる。

(2) 管理画面、管理インターフェイスへのアクセス監視

ドメイン 6 の施策（① 適切なアクセス制限・分離）と同様の内容。

(3) 透明性の制約

① 外部サービスとの連携管理、アプリケーション間のアクセス制御（API など）

対象アプリケーションが外部サービスと連携している場合など、適切な分散配置及び、外部アプリケーション間の適切な通信制御や認証/認可を行う必要がある。

連携対象となる外部サービスのセキュリティコントロールを把握することは難しいが、まずは自組織内のアプリケーション管理を行い、透明性を確保していくことが重要であると考える。

(4) セキュア設計と開発の管理

① 開発プロセスの管理

アプリケーションの開発ステップ（教育、定義、設計、開発、テスト）は、各担当者、各プロジェクトでの品質の差異をなくすためにも、標準化が行われていることが望ましい。

また、要件に応じた適切な設計、設計通りの開発、開発されたアプリケーションのテストなど、各ステップのインターフェイスを適切に結合し管理することで、効率的かつ漏れのないセキュアアプリケーションの開発が行なえると考える。

(5) セキュアな配備計画

① コードレビュー

アプリケーションのソースコードは、配備する前にコードのレビュー（チェック）が必要である。いつ誰がどのような変更をおこなったか把握し、管理/承認外の変更があった場合には、それがどのような変更なのか管理できる、またセキュリティ担当者に通知するような仕組みが必要となる。

② ユニットテスト、回帰テスト、機能テスト

これらは、開発プロセスにおける標準的なテストとなるが、このテストにはセキュリティ品質を確保するための観点も含まれていることが望ましい。

- 考慮すべきセキュリティ施策の漏れの確認
- プログラム動作時の脆弱性の確認

③ 静的アプリケーション セキュリティ テスト（SAST）

アプリケーションのソースコード自体を検査して脆弱性となりうる設計やコードを見つけ出すことを目的に実施する。

④ 動的アプリケーション セキュリティ テスト（DAST）

アプリケーションを動作させながら、ペネトレーション・テストを使用し、スキルと動機を合わせ持つ攻撃者による攻撃をシミュレーションすることで、セキュリティ上の脆弱性を見つけ出すことを目的に

実施する。

⑤　侵入テスト

外部からの侵入を想定し、自動的な定型スキャンと手動によるテストを組み合わせて脆弱性を見つけ出すことを目的に実施する。

⑥　アプリケーション CI/CD パイプラインのセキュリティ

アプリケーションを本番環境へ配備する際に、各種機能をテンプレート化し、自動構成を可能にすることで、迅速な開発が可能となるが、この自動化されたパイプラインにおいて、変更履歴の確認（ログ監視）や、API コールの制限、レポジトリの管理を適切に行う必要がある。

(6) セキュアな運用管理

①　継続的な脆弱性評価

本番環境にに配置したアプリケーションに対して、定期的（適宜）な脆弱性診断を行い、脅威の多様化に対応していく必要がある。

②　開発コード変更管理

アプリケーションのソースコードに対して、いつ誰がどのような変更をおこなったか管理し、管理/承認外の変更があった場合には、セキュリティ担当者に通知するような仕組みが必要となる。

6.3. 対応 OSS

本章では、6.2 で記載した対策事項に対応する OSS を整理したものである。

尚、記載する OSS は一例となり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要。

表 6.3-1 Domain10 目的別 OSS 適合表

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
モニタリング、ロギング	**表 3.3-1** Domain6 目的別 OSS 適合表 ”アクセスの記録・定期的な監視”参照			
管理ダッシュボードへのアクセス監視	**表 3.3-1** Domain6 目的別 OSS 適合表 ”適切なアクセス制限・分離”参照			
外部連携サービスとの連携制御、アプリケーション間のアクセス制御（API など）	API 用リバースプロキシ	Kong	Apache License 2.0	プラグインベースの Kong は拡張が容易で、多くの人気サービスのプラグインを用意する。例えば Amazon Web Services のイベント駆動型コード実行サービス「AWS Lambda」の関数の管理と呼び出しが可能で、「Datadog」などのシス
		Nginx	BSD-like	

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
		Apache Cassandra	Apache License 2.0	テム監視ツールや「Loggly」のようなログ管理ツールへ監視データを送信できる。
開発プロセスの管理	リポジトリ管理	GitLab	MIT	「GitHub」のようなサービスを社内などのクローズド環境に独自で構築できる Git リポジトリマネージャー。Git ベースのソースコード管理機能、マージリクエスト、レビュー機能なども備えている。
	プロジェクト管理	Redmine	GPLv2	Ruby on Rails で開発されており、タスク管理、進捗管理、情報共有などを行えるプロジェクト管理ソフトウェア。
	バージョン管理	Apache Subversion	Apache License 2.0	さまざまなソフトウェアの開発現場において広く使われているソースコード管理システム。 ソースコードやそこに加えられた変更点などの履歴はすべて中央リポジトリに記録され、各開発者はネットワーク経由で中央リポジトリにアクセスすることでソースコードを取り出したり、変更点を記録するという中央集権型のバージョン管理機能を備えている。
	バージョン管理	Git (Github)	GPLv3	オープンソース分散型バージョン管理システム。プログラムソースコード/設定ファイル/サイトコンテンツなどの変更履歴を記録/追跡管理する。
コードレビュー	コードレビュー	Rietveld	Apache License 2.0	Mondrian のオープンソース版ソフトウェア。 Google App Engine で動作しており、構成管理ツールと併せて利用する。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
		ReviewBoard	MIT	Python で作成されたコードレビューツール。 ソフトウェア開発におけるソースコードの変更評価を管理するツールで、構成管理ツールなどと併せて利用する。
ユニットテスト、回帰テスト、機能テスト	機能テスト	Selenium	Apache License 2.0	Web アプリケーションのテスト自動化を実現するブラウザ駆動型テストツール群。ブラウザ操作からテストスクリプトを作成でき、Web ベース管理タスクの自動化も行える。
静的アプリケーション セキュリティテスト（SAST）	静的解析	FindBugs	LGPL	Java プログラムコードにあるバグを解析する。
		Serverspec	MIT	Ruby で実装されているサーバ状態のテスト自動化フレームワーク。 Serverspec のテスト実行は、「テスト対象サーバに SSH ログインして、コマンド実行結果を確認する」というシンプルなアーキテクチャとなっている。
動的アプリケーション セキュリティテスト（DAST）	脆弱性検査	OWASP Zed Attack Proxy	Apache License 2.0	Web アプリケーション脆弱性診断ツールであり、主に次の 3 つのチェック方法で Web アプリケーションの脆弱性を確認する。 ・簡易スキャン ・静的スキャン ・動的スキャン

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	ブラックボックステスト	Infrataster	MIT	「サーバ外部からのテスト」(ブラックボックステスト)を実施する Infrataster は、外部から(実際のクライアントから)テストを行う。Infrataster でのテストの場合、サーバ内部で動作しているミドルウェア種類などは関係なく、外部から見てどのような振る舞いをするかを検証する。
	脆弱性検査	OpenVAS	GPLv2	「Nessus(ネサス)」から派生したオープンソース版脆弱性スキャナ。 Linux でも Windows でも、詳細なスキャンとなる。
侵入テスト	侵入検査	OWASP Zed Attack Proxy	**表 6.2-1** Domain10 目的別 OSS 適合表”動的アプリケーション セキュリティ テスト(DAST)”参照	
		Metasploit Framework (Metasploit Community)	BSD	exploit コードの作成や実行を行うためのフレームワーク。 現在、Metasploit プロジェクトの運営は Rapid7 によって行われている。
アプリケーション CI/CD パイプラインのセキュリティ	バージョン管理	Git	**表 6.2-1** Domain10 目的別 OSS 適合表”開発プロセスの管理”参照	
	構成管理	Chef	Apache License 2.0	ファイルに記述した設定内容に応じて自動的にユーザの作成やパッケージのインストール、設定ファイルの編集などを行うツール。
		Ansible	GPLv3	レッドハットが開発するオープンソースの構成管理ツール。ファイルに記述した設定内容に応じて自動的にユーザの作成やパッケージのインストール、設定ファイルの編集などを行うツール。構成管理に加え、オーケストレーションやソフトウェアデプロイメントの機能を持つ。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	DevOps ダッシュボード	Hygieia	Apache License 2.0	様々な CI/CD ツールからデータを収集し、単一のダッシュボードで管理することができる。 管理者向けへのプロジェクト集計メトリックの提供や、リリース管理における監査機能を提供する。
	セキュリティチェック自動化	InSpec	Apache License 2.0	DevOps でのポリシーやコンプライアンス遵守のチェック自動化を行うテストフレームワーク。
	テストの実行、ビルド、デプロイ	Jenkins	MIT	Java で作られているオープンソースの CI（継続的インテグレーション）ツール。
	コンテナ型の仮想化環境	Docker	Apache License 2.0	オープンソースのコンテナ型仮想化ソフトウェア。 1 つの OS にコンテナといわれる「独立したサーバと同様の振る舞いをする区画」を複数作り、それを個別のユーザ/サービスに割り当てる。 コンテナには、個別に CPU/メモリ/ストレージなどを割り当てる必要がないため、システムリソースのオーバーヘッドが少なくて済み、同じ性能のハードウェアならば、より多くのコンテナを作ることが可能となる。
	自動テスト	Selenium	**表 6.2-1** Domain10 目的別 OSS 適合表”ユニットテスト、回帰テスト、機能テスト”参照	
	CI/CD	Concourse CI	Apache License 2.0	Cloud Foundry やアジャイル開発のコンサルティングなどでソフトウェア開発をリードする Pivotal のエンジニアが、Go 言語で開発したパイプラインベースの CI/CD ツール。
継続的な脆弱性評価	**表 6.2-1** Domain10 目的別 OSS 適合表”静的アプリケーション セキュリティ テスト（SAST）”、”動的アプリケーション セキュリティ テスト（DAST）”、“侵入テスト”参照			

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
コードレビュー（開発コード変更管理）	バージョン管理	Apache Subversion	**表 6.2-1　開発プロセスの管理　参照**	
	バージョン管理	Git	**表 6.2-1　開発プロセスの管理　参照**	

7. DOMAIN11 データセキュリティと暗号化

7.1. 概要

本ドメインでは、暗号化をはじめとした、データそのもののセキュリティに関わる管理策に焦点を当てている。

オンプレミスで守られていたデータは、クラウドに移行することによって新たなリスクに晒される可能性があり、クラウドのデータはオンプレミスとは異なる視点でセキュリティ施策を講じる必要がある。さらに、データセキュリティに関しては、事業者と利用者の責任分界という観点で、IaaS/PaaS/SaaS のどのサービス形態であっても最終的には利用者の責任範囲であるという考えが一般的であり、クラウド利用者が最もセキュリティ施策を考慮すべき領域のひとつである。

ガイダンスでは、データセキュリティの管理策は概ね 3 つの領域に分類できると述べられている。

- クラウドへ移行するデータ（およびその所在場所）に対する管理
- クラウド上にあるデータの保護と管理
- 情報ライフサイクル管理に対応したセキュリティの適用

それぞれについて利用者側の理解や対策が必要な点を以下に整理する。

7.2. 解説

(1) データストレージの選択

ほとんどのクラウドプラットフォームは、冗長化した耐久性のあるストレージメカニズムを持っており、多くの場合はデータ分散配置の仕組みを利用している。

(2) クラウドへのデータ移行

まず、どんなデータをどの場所に配置するかのポリシーを定め、それをセキュリティ要件として、クラウドの利用状況やデータ転送をモニタリングし、実際のデータの移動を検知する必要がある。ガイダンスでは、その手段として CASB、URL フィルタ、DLP といったツールを示している。

また、データ転送のセキュリティとして、クライアントサイドでの暗号化や暗号化プロキシといった手段も示しているが、ほとんどのクラウド事業者は TLS をサポートしており、TLS をサポートしていない事業者は選択すべきではないと述べている。

(3) クラウド上のデータに対するアクセス制御

ガイダンスでは、アクセス制御は最低限 3 つのレイヤーにおいて実装されなければならないとしている。

- 管理用ダッシュボード
- 外部向けと内部向けの共有の制御
- アプリケーションレベルコントロール

(4) ストレージの暗号化とトークナイゼーション

暗号化には、IaaS、PaaS、SaaS といったサービスモデルに応じて様々な手段が存在する。暗号化は、復号鍵によってデータを復元することを前提に行うが、トークナイゼーションは、データをランダムな

文字列に置き換えて保存する方式で、長期保存が必要なデータなど、クラウド上に保持した場合に（たとえ暗号化したとしても）暗号危殆化のリスクが懸念される場合に有効な手段となる。

(5) 鍵管理

ガイダンスでは、鍵管理を実施する方法としては下記の 4 つの選択肢があるとしている。

- HSM／アプライアンス
- 仮想アプライアンス／ソフトウェア
- クラウド事業者のサービス
- ハイブリッド

事業者が暗号鍵を管理する場合、国によっては、政府の要求等に基づいてデータが開示されるリスクがあるため、利用者によって鍵管理を行うことで事業者から機微なデータを守るという選択肢が存在する。

(6) モニタリング、監査、警報

ドメイン 11 の内容としては、機微なデータに関する権限付与の変更や外部からのアクセスの把握、および警報発信といった領域を指し、API とストレージアクセスの両方を監視する必要がある。

(7) データマスキング

データマスキングは、開発及びテスト環境におけるデータを守るため、もしくはアプリケーションの中にある機微なデータを隠蔽するために用いられる。

7.3. 対応 OSS

本章では、7.2 で記載した対策事項に対応する OSS を整理したものである。

尚、記載する OSS は一例となり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要。

表 7.3-1 Domain11 目的別 OSS 適合表

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
データ分散配置	分散ファイルシステム	Hadoop Distributed File System (HDFS)	Apache License 2.0	Google 社の分散ファイルシステムである GoogleFS の技術をもとに、Apache Hadoop プロジェクトで開発されたソフトウェア。
	分散オブジェクトストレージ	Swift	Apache License 2.0	OpenStack の一部でオブジェクトストレージを担う。多数のサーバにオブジェクトを分散させることが可能。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	分散ストレージ	Ceph	GPLv2 LGPLv2.1	分散ストレージソフトウェアでオブジェクト単位/ブロック単位/ファイル単位のアクセスが可能。SDS（Software Defined Storage）の一種であり、RedHat 社が商用版を提供している。
データ移動、操作の監視	DLP	MyDLP	GPLv3	2014 年 Comodo Group に買収された後、更新されていない
	DLP	OpenDLP	GPLv3	Google Code で公開。2012 年 8 月の v0.5.1 以降更新されていない。
	URL フィルタ	SafeSquid	フリー	コンテンツフィルタリング機能を持つ Squid ベースのプロキシ。20 ユーザまで利用可能な無償版が存在するが、厳密には OSS ではない。
	URL フィルタ	DansGuardian	GPLv2	Linux などで動作する Web コンテンツフィルタ。
	ログ可視化	Logstash ＋ Elasticsearch ＋ Kibana	Apache License 2.0	Elastic Stack の組み合わせによるデータ可視化。 ・logstash：収集 ・ES：集積 ・Kibana：可視化
	ログ可視化	Fluentd＋ Elasticsearch ＋ Kibana	Apache License 2.0	可視化ツールとして Kibana の代わりに Fluentd を用いるパターン。日本で人気があるため、日本語の情報が多い。
安全なデータ転送	暗号化	OpenSSL	Apache1.0＋四条項 BSD	SSL プロトコル・TLS プロトコルのオープンソース実装で、多くのソフトウェアに組み込まれている。
	暗号化	OpenSSH	BSD	OpenBSD プロジェクトにより開発が行われた、デファクトスタンダードの SSH プロトコル実装ソフトウェア。
共有の制御	オンラインストレージ	ownCloud	AGPLv3	オンラインストレージを構築することができるソフトウェア。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	オンラインストレージ	Nextcloud	AGPLv3	OwnCloud を元に開発された後発ソフト。
データベースの保護	データベースファイアウォール	GreenSQL	GPLv2	MySQL と PostgreSQL に対応したプロキシ型 DBFW。 2008 年以降更新されていない。
	データベースファイアウォール	sql_firewall	PostgreSQL	PostgreSQL 上で実行可能な SQL を制限することで SQL インジェクションを防ぐツール。 PostgreSQL ライセンスで提供されている。
著作権管理	ERM/DRM	OpenIPMP	MPL1.1	オープンソースの DRM として開発されたが、2006 年の v2.0.2 以降更新されていない。
保管データの暗号化	DB 暗号化	MyDiamo	GPLv2	MySQL,MariaDB,Percona, PostgreSQL に対応。個人の非営利的利用に限り無償利用可能。
	DB 暗号化	SQLCipher	BSD Style	SQLite データベースに対して透過的な暗号化を提供するツール。
	ファイル暗号化	VeraCrypt	Apache License 2.0 ＋ TrueCrypt3.0	開発が終了した TrueCrypt を元に開発された暗号化ツール。 ファイルやパーティションの暗号化や、仮想暗号化ディスクの作成が可能。Windows、macOS、Linux に対応している。
	トークナイゼーション	Deeplearning4j	Apache License 2.0	Deeplearning4j は深層学習用の Java ライブラリであり、トークナイゼーション用のクラスが含まれている。
	トークナイゼーション	OpenNMT	MIT	OpenNMT は機械翻訳など、自然言語処理の用途で使われる深層学習フレームワークであり、トークナイゼーションの機能が含まれている。
	その他暗号化	OpenSSL	表 7.3-2 暗号化 参照	
鍵の管理	KMS	KMIP4J	三条項 BSD	KMIP1.をオープンソースで実装した Java ライブラリ。

目的	施策/機能	対応 OSS	ライセンス	OSS の説明
	HSM	CrypTech	BSD	オープンソースの HSM エンジンの作成を目標に活動しているプロジェクト。
監視と警報	運用監視	Nagios	GPLv2	監視をプラグインによって行うのが特徴のファイルベースのネットワーク監視ツール。コミュニティ活動が活発で、多数のプラグインが公開されている。
	運用監視	Zabbix	GPLv2	Zabbix 社が開発している、Web インタフェースを備えたネットワーク監視ツール。データは RDB に格納される。商用サービスも提供されている。
データマスキング	匿名化	ARX	Apache License 2.0	GUI を備えたデータ匿名化ツール。現在も開発が続けられている。
	匿名化	UTD Anonymization ToolBox	GPL	教育機関で開発された匿名化ツール。6 つの匿名化方式に対応している。2012 年以降更新されていない様子。

8. DOMAIN12 アイデンティティ管理、権限付与管理、アクセス管理（IAM）

8.1. 概要

アイデンティティ管理、権限付与管理、アクセス管理を包含して IAM（Identity and Access Management）と当ガイダンスでは呼んでいる。

IAM における主要な用語や IAM の標準規格について当ガイダンスにて説明しているため、ぜひ参照願いたい。

IAM には、ID フェデレーション（ID 連携）や ID プロビジョニング（統合 ID 管理）などのいくつかの標準規格があり、クラウドサービス同士の連携やクラウドサービスと既存システムの ID 管理（Active Directory や LDAP など）との連携に利用されている。

サービス間を独自方式によって連携させる場合は、方式の安全性について自ら担保しなければならないというリスクを負うことになるため、サービス間の連携にはできる限り標準規格を使用するべきである。もし、既存のポリシーをそのままにして、ID フェデレーションや ID プロビジョニングを行うことが困難であれば、IAM の標準規格に準拠するようにポリシーの見直しを検討すべきと当 WG では考えている。

表 8.1-1 ID フェデレーションと ID プロビジョニング

目的	概要
ID フェデレーション（ID 連携）	ユーザ ID をリンクさせる。2 つのドメインで ID フェデレーションを実現すると、一方のドメインで認証を受けたエンドユーザーは、他方のドメインでもログインしないでそのリソースにアクセスさせること
ID プロビジョニング（統合 ID 管理）	ユーザに対し必要に応じてリソース（ID）を割り当てること

なお、技術的な対策ではないが IAM における棚卸は必要な対策であると考え、追加している。

8.2. 解説

(1) アイデンティティ管理

アイデンティティ管理をクラウド個別に行うのではなく、組織のアイデンティティ管理と連携することを考えるべきである。また、不要なアイデンティティが放置されないように管理されないアイデンティティを棚卸する業務を組み入れるべきである。

(2) アクセス管理

クラウドサービスへの接続先は限定されていないため、不正アクセスの脅威にさらされることになる。そのため、多要素認証のような認証が必要である。アクセス権の管理として、保有しているアクセス権の棚卸を行い、不正アクセスを防ぐことも大事であると考える。

「ハードウェアトークン」を用いた認証を実現する OSS は、現時点では発表されていないが、「ハードウェア（スマートカードなど）」を用いて認証する方法は確認されている。

(3) 権限付与管理・アクセス制御

ネットワークや API などのリソースの利用権限を与える認可を行う制御と必要な権限を必要なアイデンティティに与える権限付与についても管理が必要である。

OSS ではそれらと認証も含め提供しているため、認証の OSS と重複している。権限付与についてはマトリクスなどで整理し、維持していく必要がある。当ガイダンスでは権限付与を検討するにあたり 2 つのモデルを紹介している。

表 8.2-1 アクセス制御のモデル

モデル名	概要
ロールベースアクセス制御 (Role Based Access Control)	役割(ロール)を定義し、役割に基づいてアクセス制御を行い、実行できる機能を限定する。
属性ベースアクセス制御 (Attribute Based Access Control)	属性により実行できる機能を限定する。役割と異なり、属性として複数の属性を定義でき、例として役割、端末(接続場所)などがある。

(4) 特権ユーザ管理

特権ユーザは、文字通り強力なシステム権限を所有しており、「いつ」、「だれが」、「何を」したのか、接続するシステムに関連する全ての操作をログに記録わかるようにすべきである。

多要素認証だけでなく、場合によっては物理的な対策を講じた上で接続先を限定するなどの手段も必要であると考えた。もちろん、特権ユーザを利用できる権限保有者も定期的に棚卸を行うことも必要であると考えた。

8.3. 対応 OSS

本章では、8.2 で記載した対策事項に対応する OSS を整理したものである。

尚、記載する OSS は一例となり、利用用途/環境に応じて他のソフトウェアと組み合わせるなどして利用することが必要。

表 8.3-1 Domain12 目的別 OSS 適合表

目的	施策/機能/方法	対応 OSS	ライセンス	OSS の説明
アイデンティティ管理	ID プロビジョニング(統合 ID 管理)	OpenIDM	CDDL	アイデンティティ情報のプロビジョニングとライフサイクル管理を実現。
	ID フェデレーション (アイデンティティブローカ/SSO)	Shibboleth	Apache License 2.0	Shibboleth プロジェクトにより開発され、学術系の認証基盤として広く利用されている SAML ベースのオープンソース実装。

目的	施策/機能/方法	対応 OSS	ライセンス	OSS の説明
	ID フェデレーション (アイデンティティブローカ/SSO)	OpenAM	CDDL1.0	元々商用の製品がオープンソース化された経緯を持ち、SAML2.0、OAuth2.0、OIDC1.0 をはじめ多くの標準プロトコルに対応している。 SAML に対応しているため、Google Apps や Salesforce などのアプリケーションと連携させる事例も多く存在する。
	ID フェデレーション (アイデンティティブローカ/SSO)	Open IG	CDDL1.1	SAML2.0 SP や OAuth2.0 クライアント、代理認証機能などを実装しており、ID 連携に対応していない Web アプリケーションを OpenAM と連携して動作させるためのソフトウェア。
	ID フェデレーション (アイデンティティブローカ/SSO)	Keycloak	Apache License 2.0	OpenAM と比べて後発のプロジェクトだが、近年評価を上げているソフトウェア。 SAML2.0、OAuth2.0、OIDC1.0 に対応している。
アクセス管理	多要素認証			
	ハードウェアトークン	ー	ー	OpenAM + OpenSC + ハードウェア OpenAM と OpenSC を利用することでスマートカードなどによる認証を実装。
	ソフトウェアトークン	FreeOTP	Apache License 2.0	ワンタイムパスワードを用いた多要素認証のためのソフトウェアトークン実装の一。FreeOTP は Google Authenticator の代替実装として Red Hat によって開発が行なわれている。

目的	施策/機能/方法	対応 OSS	ライセンス	OSS の説明
	ソフトウェアトークン	Google Authenticator	Apache License 2.0	Google が開発している多要素認証用トークンソフトウェア。
	OOB 認証（アウトオブバンド）	―	―	―
	生体認証	OpenAM	CDDL1.0	OpenAM と生体認証装置（ハードウェア）を組み合わせることで実現。
	アクセス管理＆権限付与管理	OpenAM	CDDL1.0	オープンソースのアクセス管理。 「認証」「認可」「フェデレーション」などの機能を提供。
		Keycloak	Apache License 2.0	
特権ユーザ管理	アカウントとセッションの記録	OpenAM	CDDL1.0	OpenAM 上で発生した操作は Audit ログとして出力。
	多要素認証	表 8.3-2 **Domain12** 目的別 OSS 適合表”アクセス管理”参照		
	接続環境の隔離 資格情報制御 デジタル証明書 論理的分離 (物理的分離)	―	―	―

9. 参考 URL

本章では、各章における OSS 一覧を作成するために参考とした URL を記載する。

表 9-1 参考 URL 一覧

OSS	URL
ACID	https://sourceforge.net/projects/acidlab/
Ansible	https://github.com/ansible/ansible
Apache Cassandra	http://cassandra.apache.org/
Apache SpamAssassin	https://spamassassin.apache.org/
Apache Subversion	https://subversion.apache.org/download.cgi
ApacheCloudStack	https://cloudstack.apache.org/
ARX	https://arx.deidentifier.org/
BeEF	https://github.com/beefproject/beef
Ceph	https://github.com/ceph/ceph
Chef	https://downloads.chef.io/
Concourse CI	https://concourse-ci.org/download.html
CrypTech	https://trac.cryptech.is/wiki/GitRepositories
cuckoo	https://github.com/cuckoosandbox/cuckoo
DanaGuadian	https://wiki.archlinux.jp/index.php/DansGuardian
Deeplearning4j	https://deeplearning4j.org/
Docker	https://www.docker.com/get-started
Elasticsearch	https://www.elastic.co/jp/products
FindBugs	http://findbugs.sourceforge.net/downloads.html
Fluentd	https://github.com/fluent/fluentd
FreeOTP	https://freeotp.github.io/
Git	https://git-scm.com/downloads
GitLab	https://docs.gitlab.com/omnibus/manual_install.html
Glimpse	https://github.com/Glimpse/Glimpse
go-dots	https://github.com/nttdots/go-dots
Google Authenticator	https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=ja
Hadoop(HDFS)	http://hadoop.apache.org/
Harbor	https://github.com/goharbor/harbor/
Hertbeat	https://www.ossnews.jp/oss_info/Heartbeat
Hinemos	https://ja.osdn.net/projects/hinemos/
Hygieia	https://github.com/Hygieia/Hygieia

OSS	URL
Infrataster	https://github.com/ryotarai/infrataster
InSpec	https://www.inspec.io/downloads/
iptables	https://netfilter.org/projects/iptables/index.html
Jenkins	https://jenkins.io/
Keycloak	https://openstandia.jp/oss_info/keycloak/
Kibana	https://www.elastic.co/jp/products
KMIP4J	http://kmip4j.sourceforge.net/
Kong	https://github.com/Kong/kong/releases
LISM	https://ja.osdn.net/projects/lism/
Logatash	https://www.elastic.co/jp/products
Metasploit Framework	https://www.metasploit.com/download
mfa	https://github.com/broamski/aws-mfa
ModSecurity	https://www.modsecurity.org/download.html
ModSecurity	https://github.com/SpiderLabs/ModSecurity
MyDiamo	https://www.mydiamo.com/ja
MyDLP	https://github.com/mydlp
Nagios	https://github.com/NagiosEnterprises/nagioscore
Nagios	https://www.nagios.com/
NextCloud	https://github.com/nextcloud/server
Nginx	http://nginx.org/en/download.html
OpanVAS	http://www.openvas.org/about.html
Open IG (Identity Gateway)	https://github.com/OpenIdentityPlatform/OpenIG
Open vSwitch	https://www.openvswitch.org/
OpenAM/OpenIDM	https://stash.forgerock.org/projects/OPENAM
OPENDAYLIGHT	https://www.opendaylight.org/
OpenDLP	https://code.google.com/archive/p/opendlp/
OpenIPMP	https://sourceforge.net/projects/openipmp/
OpenNMT	http://opennmt.net/OpenNMT/tools/tokenization/
OpenSCAP	https://github.com/OpenSCAP/openscap
OpenSSH	https://anongit.mindrot.org/openssh
OpenSSL	https://github.com/openssl/openssl
OpenStack	https://governance.openstack.org/tc/reference/licensing.html
OpenVAS	http://openvas.org/download.html
OpenVPN	https://github.com/OpenVPN/openvpn

OSS	URL
OSSIM	https://ja.osdn.net/projects/sfnet_os-sim/ https://github.com/jpalanco/alienvault-ossim/tree/master/os-sim
OWASP ZAP	https://github.com/zaproxy/zaproxy
OWASP Zed Attack Proxy	https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
ownCloud	https://github.com/owncloud
Packer	https://github.com/hashicorp/packer
pfSence	https://www.pfsense.org/
Proxmox Mail Gateway	https://www.proxmox.com/en/proxmox-mail-gateway
Puppet	https://www.ossnews.jp/oss_info/Puppet https://github.com/puppetlabs/puppet/blob/master/LICENSE
Rancher	https://rancher.com/products/rancher/
Redmine	http://redmine.jp/download/
ReviewBoard	https://www.reviewboard.org/downloads/
Rietveld	https://github.com/rietveld-codereview/rietveld
Rspamd	https://github.com/rspamd/rspamd
Ryu SDN Framework	https://www.ossnews.jp/oss_info/Ryu_SDN_Framework https://ryu.readthedocs.io/en/latest/search.html?q=license&check_keywords=yes&area=default
SafeSquid	https://www.safesquid.com/content-filtering/downloads
Selenium	https://www.seleniumhq.org/download/
Serverspec	https://serverspec.org/
Shibboleth	https://www.ossnews.jp/oss_info/Shibboleth
Snort	https://www.snort.org/downloads
sql_firewall	https://github.com/uptimejp/sql_firewall/
SQLCipher	https://www.zetetic.net/sqlcipher/
sqlmap	https://github.com/sqlmapproject/sqlmap
Suricata	https://suricata-ids.org/download/
Swift	https://www.openstack.org/
tcpdump	https://www.tcpdump.org/index.html#
UTD Anonymization ToolBox	http://www.cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php
VeraCrypt	https://www.veracrypt.fr/code/VeraCrypt/
Vuls	https://github.com/future-architect/vuls
VyOS	https://vyos.io/
WebKnight	http://www.aqtronix.com/?PageID=99

OSS	URL
WireShark	https://www.wireshark.org/
Zabbix	https://www.zabbix.com/jp/