cloud security alliance SM

クラウド利用者視点のクラウドセキュリティ (OSSマッピング編)

釜山 公徳　-Masanori KAMAYAMA-

日本クラウドセキュリティアライアンス
クラウドセキュリティWG リーダー

CSAJC cloud security alliance SM Japan Chapter

Self-introduction

略歴：

ICTインフラのオールフェーズ(開発、設計、構築、運用)、テクニカルサポート、メモリダンプ解析、セキュリティコンサルティング、クラウドアーキテクティング等を経て、現在セキュリティやクラウド領域のコンサルティングに従事。

主な団体活動：

- CompTIA Subject Matter Expert
- 日本セキュリティ監査協会(JASA) FinTech WG / 金融WG

著作：

- Software Design 2017年10月号
  - ネットワークエンジニア　その技術の極め方
- NISC (内閣サイバーセキュリティセンター) : サイバーセキュリティ ひとこと言いたい！
  - あなたの守りたい「モノ」は何ですか？どうやって守りますか？

本WGの目的と課題

本WGの目的と課題

これまで、CSAの根幹である
クラウドセキュリティ自体について
言及していなかったんちゃうか？

CSAガイダンスがあるやん

まとまった情報で参考にはなるけど、
実装イメージがわかへんのや

確かにそうやけど、実装例を出すと商用製品の紹介になって、ニュートラルちゃうやん

そうや、OSS やろう

クラウドセキュリティの概要

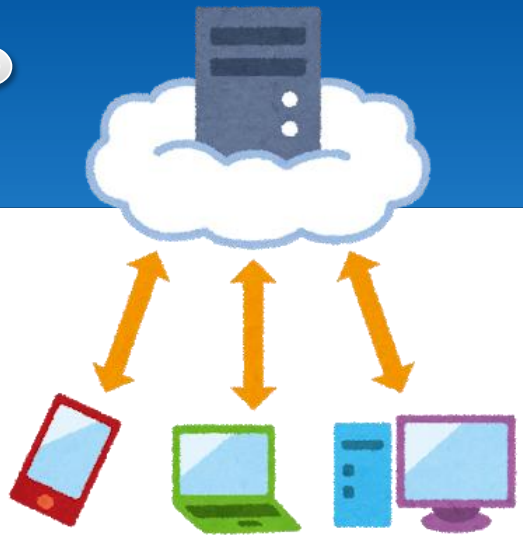
クラウドへの不安

- 年々減少しているものの、まだまだクラウド利用への不安が強い

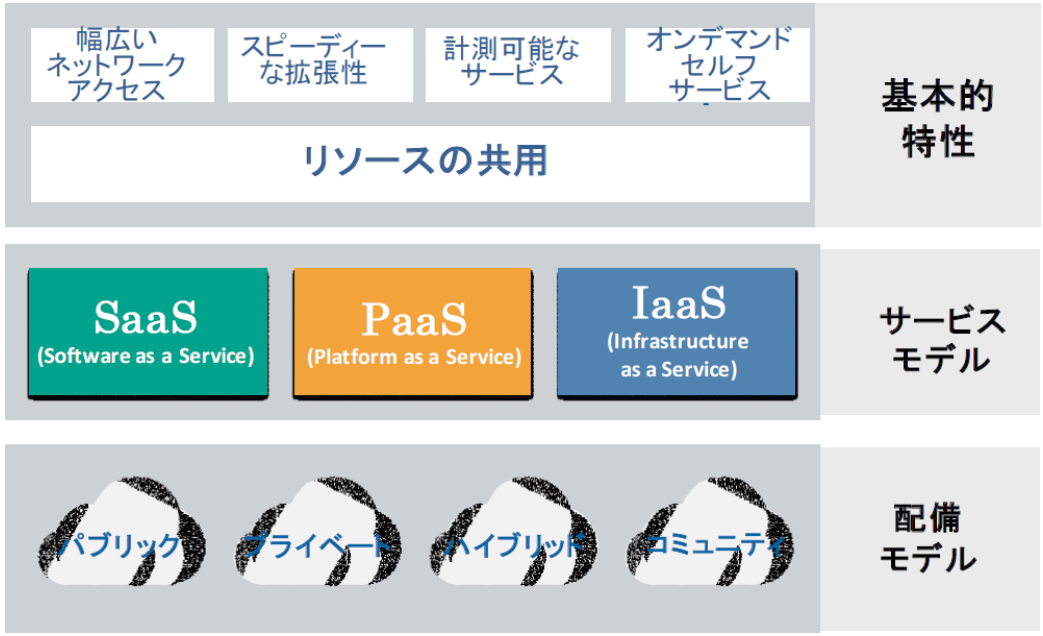
ほとんどが未知に対する恐怖

心理障壁

そもそもクラウドって？



- NIST SP800-145 で3つを定義
  1. 基本特性 (Essential Characteristics)
  2. サービスモデル (Service Models)
  3. 配置モデル (Deployment Models)



パブリッククラウド事業者のセキュリティ

- データ保護の徹底で高セキュリティレベルを実現
- 第三者によるコンプライアンス認証

利用者側でのセキュリティ施策

クラウド事業者側で強固なセキュリティ施策をやっとるから、利用者側は何もいらんの？

やらなあかん

何でや？

クラウド事業者と利用者とでは、責任の範囲が分かれとるからな

クラウド事業者	クラウド利用者
内部のセキュリティ施策と利用者向けのセキュリティ機能をはっきりと文書で示し、クラウド利用者が情報に基づく意思決定をできるようにすること。 事業者はまた、それらのセキュリティ施策を適切に設計し実装しなければならない。	クラウドプロジェクトに際し、責任分担表を作成して誰がどの対策をどのように実装する必要があるのかを文書で示さなければならない。 これは同時に、必要な準拠すべき基準に対応していなければならない。

責任共有モデル

- サービスモデルによって異なる責任範囲

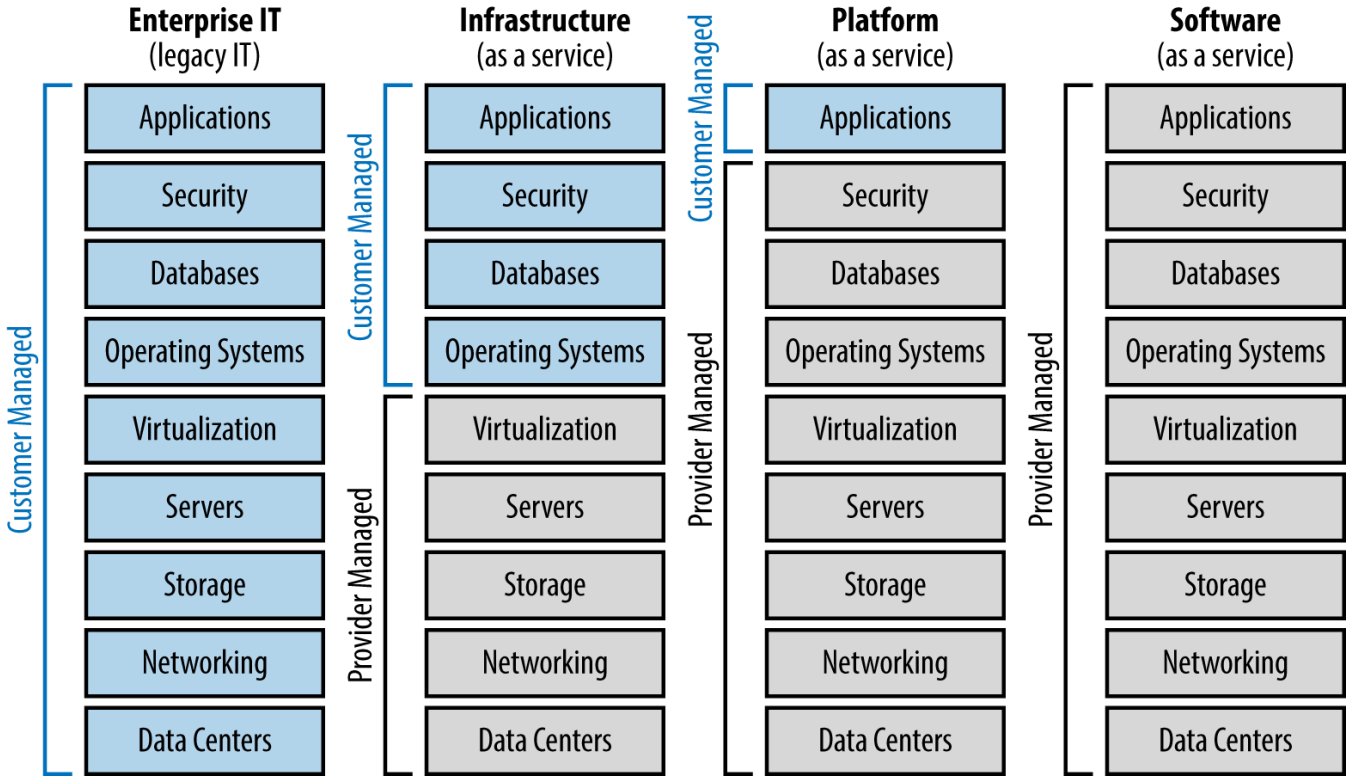


Figure 1-5. Cloud provider versus customer roles for managing cloud services
<https://www.oreilly.com/library/view/the-enterprise-cloud/9781491907832/ch01.html>

クラウド利用者の課題と解決

セキュリティ施策を
どうすればいいか
わからん

オンプレミスとの
違いは？

クラウドでの責任範囲が
いまいちわからん

CSA ガイダンス V4.0 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）

本書の前提と範囲

- OSSにマッピング可能な技術要素が対象

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

ドメイン6
管理画面と事業継続

1. 適切なアクセス制限・分離
   - ✓ 利用用途に応じたアカウントの作成とアクセス権の設定
2. 特権ユーザーアカウントの管理
   - ✓ 常時利用を禁止 (利用が必要な場合は最小限の範囲で許可)
3. 境界での防御
   - ✓ 不正アクセス、サイバー攻撃に備えた境界での防御を実装
4. 適切な利用者認証
   - ✓ 認証技術を利用し、セキュアなアクセスを実現
5. アクセスの記録・定期的な監視
   - ✓ トレーサビリティを担保する

Sample

目的	施策/機能/方法	対応OSS
適切なアクセス管理	認可	OpenAM
境界での防御	IDS/IPS	snort
	WAF	ModSecurity
	Reverse Proxy	Nginx
適切な利用者認証	MFA	Google Authenticator

ドメイン7
インフラストラクチャセキュリティ

1. ネットワークの仮想化
   - ✓ クラウドではVLANやSDNといった技術を意識しなくてもよい
2. ワークロード (処理の単位)
   - ✓ ワークロードを分離・隔離し、他環境による影響を回避
3. 一般的なインフラストラクチャ・セキュリティ
   - ✓ 入口、内部、出口の各レイヤーで実装するセキュリティ

目的	施策/機能/方法	対応OSS
ネットワーク仮想化	SDN	OpenDaylight
ワークロード	ワークロードの分離	Jenkins + Ansible + Packer
	プラットフォーム脆弱性診断	OpenSCAP
ネットワーク脅威の検出と防御	DDoS対策	go-dots(DDOS)
	WAF	ModSecirty
	不正アクセス及び不正侵入の検出(防御)	Snort ＋ ACID
	メールゲートウェイ	SpamAssassin
	ネットワーク監視	WireShark
		Zabbix
安全な接続環境の確保	VPNの利用	OpenVPN
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
セキュリティ診断	ペネトレーションテスト(侵入テスト)	Sqlmap
		BeEF

Sample

ドメイン8
仮想化とコンテナ技術

1. コンピュート

   利用者に求められるセキュリティ実装

   - ✓ クラウドリソースに関するアイデンティティ管理
   - ✓ 監視とロギング
   - ✓ マスタイメージ・リポジトリ管理
   - ✓ 専用ホスティングの使用 (必要な場合)
   - ✓ 全レイヤーへの標準的セキュリティ(仮想マシン、コンテナ、コード等)
   - ✓ 最新パッチ適用済み仮想マシンイメージなどの安全な配備
   - ✓ サーバレス型利用の際のホストレベルの監視・ロギングに代わる手段

2. ネットワーク
   - ✓ 仮想ファイアウォールの適切な配備・設定

3. ストレージ
   - ✓ 利用者側レイヤーでの暗号化により、不正アクセス侵入者等から保護

仮想化とコンテナ技術

4. コンテナ

コンテナで要するセキュリティ実装

- ✓ (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

Sample

目的	施策/機能/方法	対応OSS
仮想マシンの適切な管理	アイデンティティ管理	OpenIDM/OpenAM
	監視とロギング	Zabbix
	マスタイメージ（リポジトリ）管理	Jenkins+Ansible+Packer
仮想リソース内のセキュリティ制御	安全な構成設定の配備	Jenkins+Ansible+Packer
仮想ネットワークのセキュリティ	仮想ネットワーク用のトラフィック監視	Zabbix
	仮想ファイアウォールの適切な設定	Vyatta
		pfSense
コンテナのセキュリティ	管理ダッシュボードのセキュリティ確保	OpenIAM
		OpenIDM/OpenAM
	イメージリポジトリの保護	Rancher+Harbor

ドメイン10
アプリケーションセキュリティ

1. 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

   - ✓ 不審な操作のアラート設定と継続的なモニタリング

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の対策と同様)

3. 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

   - ✓ 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、アプリケーション間の適切な通信制御や認証/認可を実装

4. セキュア設計と開発の管理

   開発プロセスの管理

   - ✓ 開発ステップで標準化し、インターフェイスの適切な結合と管理により、効率的かつ漏れのないセキュアアプリケーション開発を実現

ドメイン10
アプリケーションセキュリティ

5. セキュアな配備計画

① コードレビュー

- ✓ コードのレビューと変更管理、変更後の通知をシステマティックに実装

② ユニットテスト、回帰テスト、機能テスト

- ✓ セキュリティ品質を確保されたテスト
  - 実装すべきセキュリティ施策の漏れの有無
  - プログラム動作時の脆弱性確認

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より脅威の有無をテスト
- ✓ 網羅的に潜在的な脅威を検出可能

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実際に動作させ脅威の有無をテスト
- ✓ 実際に発生し得る脅威を効率的に検出可能

アプリケーションセキュリティ

⑤　侵入テスト

- ✓ 外部からの侵入を想定し、動的手法と静的手法とを組み合わせたテスト

⑥　アプリケーションCI/CDパイプラインのセキュリティ

- ✓ 各種機能をテンプレート化により自動構成を可能にし、迅速な開発とセキュリティホールの極小化が可能

6.　セキュアな運用管理

①　継続的な脆弱性評価

- ✓ 定期的な脆弱性評価により脅威の多様化に対応

②　開発コード変更管理

- ✓ いつ誰が何の変更を行ったか記録・管理
- ✓ 管理・承認外作業の検出と通知の仕組み

Sample

目的	施策/機能/方法	対応OSS
開発プロセスの管理	リポジトリ管理	GitLab
	プロジェクト管理	Redmine
	バージョン管理	Subversion
	バージョン管理	Git
コードレビュー	コードレビュー	rietveld
	コードレビュー	ReviewBoard
ユニットテスト、回帰テスト 機能テスト	機能テスト	Selenium
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
	静的解析	Serverspec
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 ZAP
	ブラックボックステスト	Infrataster
	脆弱性検査	OpenVAS

ドメイン11
データセキュリティと暗号化

1. データストレージの選択
   - ✓ 多くのクラウド事業者にて可用性と耐久性を有し、データ分散配置を採用
2. クラウドへのデータ移行
   - ✓ データ移動を検知するため、クラウドの利用状況やデータ転送をモニタリング
   - ✓ セキュアなデータ転送のためCASB、URLフィルタ、DLP等を利用
3. クラウド上のデータに対するアクセス制御
   - ✓ アクセス制御は最低3つのレイヤーにおいて実装が必要
     - ① 管理用ダッシュボード
     - ② 外部向けと内部向けの共有の制御
     - ③ 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
   - ✓ クラウドサービスモデルに適した手段を採用
   - ✓ 暗号化：復号鍵によってデータを復元することが前提
   - ✓ トークナイゼーション：データをランダムな文字列に置き換えて保存する方式
     - ➢ 長期保存が必要なデータをクラウドに保持した場合等、暗号危殆化のリスクが懸念される場合に有効

ドメイン11
データセキュリティと暗号化

1. 鍵管理
   - ✓ 4つの選択肢
     - ① HSM・アプライアンス
     - ② 仮想アプライアンス・ソフトウェア
     - ③ クラウド事業者のサービス
     - ④ ハイブリッド
   - ✓ カントリーリスク(事業者が暗号鍵を管理する場合)
     - ➢ 利用者側で鍵管理を行うといった選択肢も
2. モニタリング、監査、警報
   - ✓ 機微なデータに関する権限付与の変更
   - ✓ 外部からのアクセスの把握とアラート発信
   - ✓ APIとストレージアクセスを監視
3. データマスキング
   - ✓ 開発及びテスト環境におけるデータを保護
   - ✓ アプリケーションの中にある機微なデータの隠蔽

Sample

目的	施策/機能/方法	対応OSS
データ分散配置	分散ファイルシステム	Hadoop(HDFS)
	分散ストレージ	Ceph
データ移動、操作の監視	URLフィルタ	SafeSquid
	URLフィルタ	DansGuardian
共有の制御	オンラインストレージ	ownCloud
	オンラインストレージ	NextCloud
データベースの保護	DBファイアウォール	GreenSQL
保管データの暗号化	DB暗号化	SQLCipher
	ファイル暗号化	VeraCrypt
	トークナイゼーション	DL4j
鍵の管理	KMS	KMIP4J
監視と警報	運用監視	Nagios
	運用監視	Zabbix

ドメイン12
アイデンティティ、権限付与、アクセスの管理

1. アイデンティティ管理
   - ✓ 不正使用を防止するための定期的な棚卸
   - ✓ ハイブリッドクラウドの場合、クラウドに閉じた管理ではなく、オンプレミスと連携した一元管理を推奨
2. アクセス管理
   - ✓ 不正アクセス防止のため多要素認証を推奨
3. 権限付与管理・アクセス制御
   - ✓ 2つの方法で検討
     - ① ロールベースアクセス制御 (RBAC)
       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
     - ② 属性ベースアクセス制御 (ABAC)
       属性により実行できる機能を限定する。RBAC と異なり、複数を定義可能

Sample

目的	施策/機能/方法	対応OSS
アイデンティティ管理	統合ID管理	OpenAM
	ID連携	Keycloak
アクセス管理	多要素認証	FreeOTP
		Google Authenticator
	アクセス管理 権限付与管理	OpenAM
		Keycloak

まとめ

- クラウドセキュリティは全レイヤーで！
- クラウド事業者だけでなく、利用者もセキュリティ実装が必要
- 網羅性を考慮するならガイダンスを読み込む
- フルOSSでそれなりにセキュア化可能

Thanks!