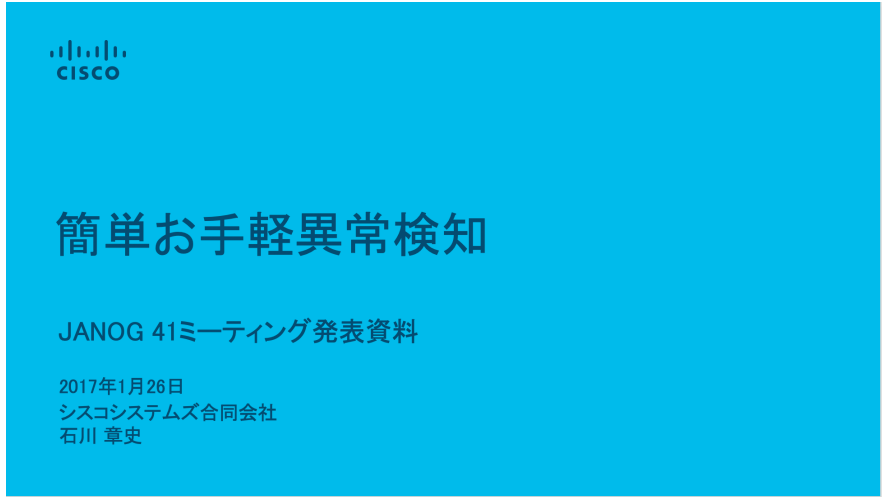
まあまあ簡単まあまあお手軽異常検知

JANOG45ミーティング発表資料

シスコシステムズ合同会社
石川 章史
Akifumi Ishikawa

JANOG41 in 広島にて

- 簡単な異常検知方法を紹介
  - 目的
    - 「ネットワークやホストの振る舞いにおいて，何かインシデントの予兆となるようないつもと異なるイベントを見つけたい」
    - DNSトンネリングの検知を例に
  - まとめ
    - 「環境に応じたホワイトリストが作成できれば非常に高精度なルールになる」
    - 「近い将来 AIが誤検知無しで見つけてくれるはずなので将来的にはそれに期待」



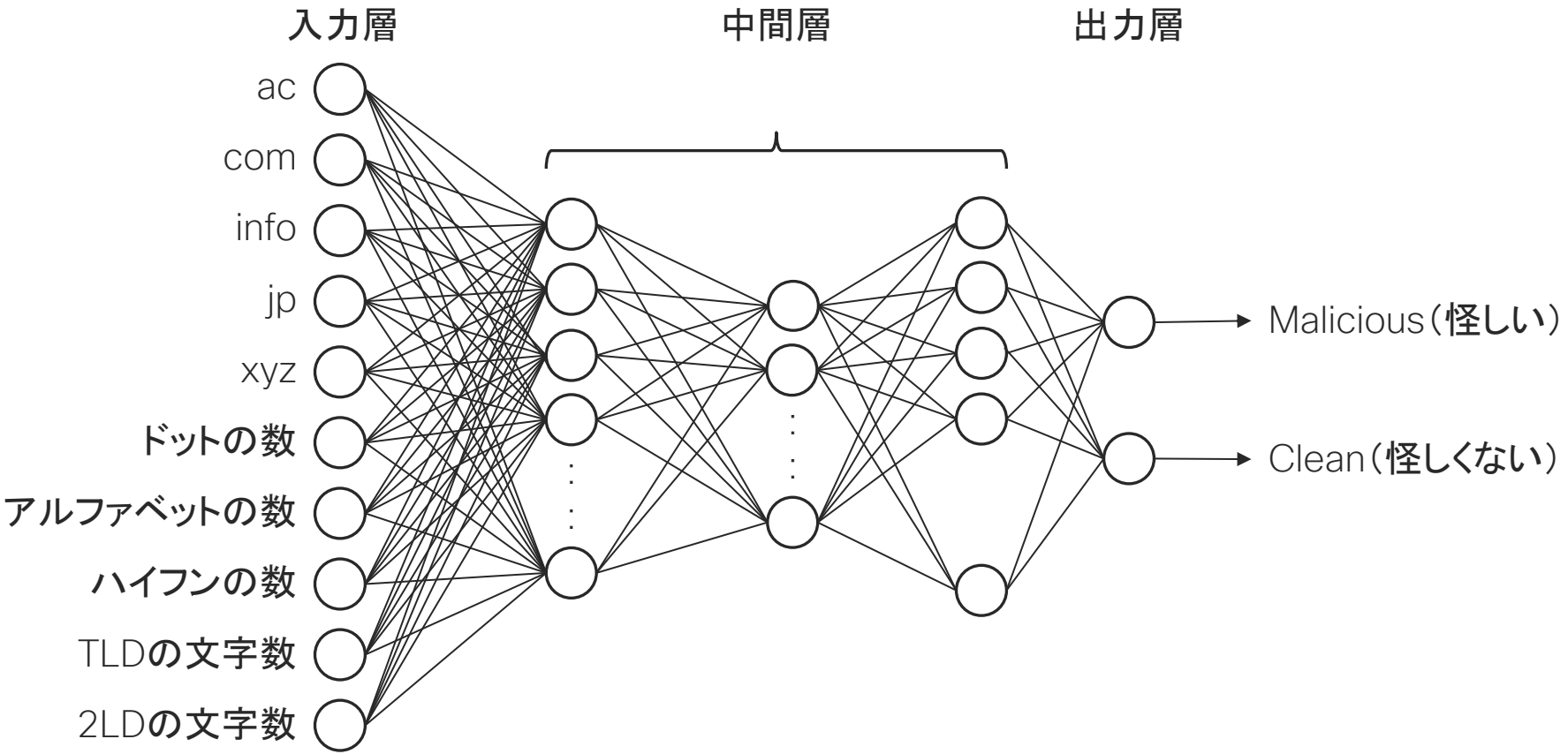
自己紹介

- 石川 章史(Akifumi Ishiakwa)
- シスコ14年目(弊社でこの勤続年数は**なかなかな**ベテランと言われる)でセキュアなネットワーク設計やセキュリティのコンサル業務を経て現在はSOCで働いております。
- ちなみに「シスコさんのSOCって北米にしかないんでしょう?」とよく尋ねられますが，実際にはアメリカ・ポーランド・日本の3拠点にSOCがあり，それぞれのSOCで分析官がちゃんと働いております。
- ルーティングは RIPv2 VLSMぐらいまでの知識で止まってますので，ルーティングのことは聞かないでください。
- 最近は少々マネジメント的な仕事をするはめになり，主戦場をサイバースペースから他 SOCの SOCマネージャとの予算取り合戦の場へと移しております。

目的など

- 前回の発表で今後の課題としてあげた AIによる実装および有効性の検証
- 今回はFQDNやドメイン名を見た時，直感的に「なんかこれ怪しい!?」と感じる SOCの人の感覚をモデル化し，膨大な DNSクエリの中から怪しい DNSクエリを検知できないか挑戦
- 具体的にはディープラーニングにて DNSクエリを「分類」
  - www.cisco.com → 怪しい? 怪しくない?
  - luqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com → 怪しい? 怪しくない?
- まあまあ簡単まあまあお手軽に実装できそうなので，これからディープラーニングを始める方の参考になれば

異常検知モデル(FQDN分類)の定義



教師データ

- 様々な脅威インテリジェンスをベースに収集
  - IPA, JPCERT/CC, JC3など関連団体の提供する情報
  - 検査サイト(Virus Total)
  - MISP(Malware Information Sharing Platform)上に流通する脅威情報
  - ベンダのデータベースやWebサイトにて公開されている脅威情報 (弊社のTalosなど)
- 学習に使用した教師データ
  - 約 5,000件 = Malicious FQDN 2,500件 + Clean FQDN 2,500件
  - Maliciousデータをがんばって収集し同等数の Cleanデータをランダムに選択収集 -----

使用したフレームワーク/ツール/モジュールなど

- ディープラーニング
  - TensorFlow
  - Keras
  - Chainer
  - ReNom
- その他。データの前処理などに利用したツール/モジュール等
  - Numpy
  - Pandas
  - scikit-learn
    - 教師データの学習データとテストデータへの分割など ----- 

怪しいと感じる FQDN例 1

- 特定のトップレベルドメイン
- アクセス頻度の低い国

.info

.life

.sk

.zw

怪しいと感じる FQDN例 2

- やたらと “-”が多い
- 最近のフィッシングサイトによく見られる

warnning-accounts-recovery-appleid-apple.com

resetting-accounts-recovery-supports-amazon.info

protection-account-recovery-support-rakuten.com

怪しいと感じる FQDN例 3

- JPドメイン風

www.cr-mufg-co-jp.mobi

www.amacojp.com

怪しいと感じる FQDN例 4

- 何かの略語的な文字列
- ドメイン生成アルゴリズム(DGA)ドメイン

tamnhindoanhnhan.com

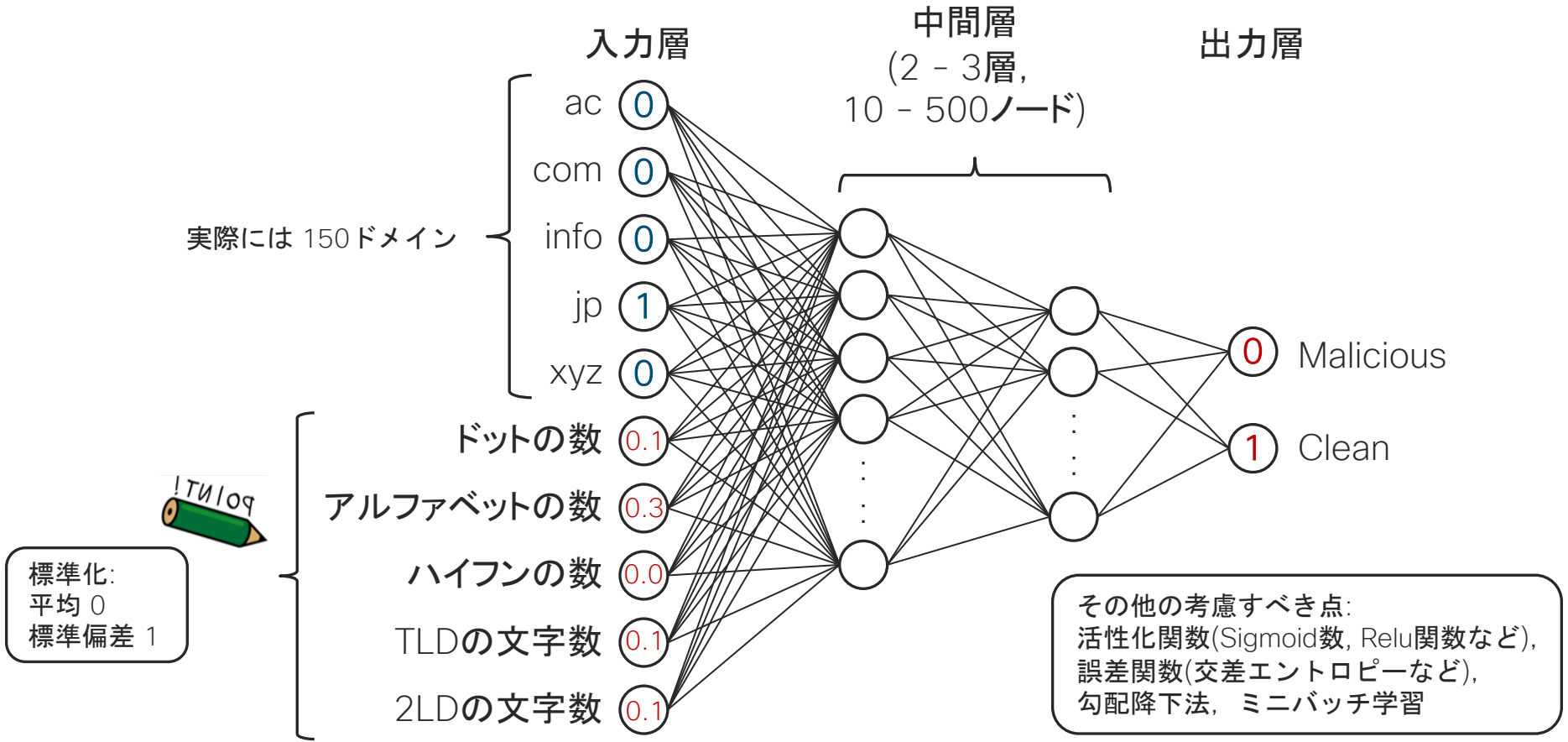
fiyvusa.com

sljjjupfgagolpg.ru

uftfesnodnjflwta.info

検知モデル Version 1.0の定義

例. www.example.co.jp



入力データ Version 1.0

- FQDNに使用されているアルファベット，数字，”-”，”.”の数
- トップレベルドメイン種別。one-hotエンコーディング ----
- まずは思いつくままに入力データにしてみる

FQDN	TLD (150 TLDs)									Total					TLD	2LD	Malicious	Clean
	ac	...	com	...	info	...	jp	...	xyz	Dots	Alphabets	Hyphens	Digit	Chars	Chars	Chars		
www.example.co.jp	0	...	0	...	0	...	1	...	0	3	14	0	0	17	2	2	0	1
account-problem-resetting-support-amazn.com	0	...	1	...	0	...	0	...	0	1	38	4	0	43	3	39	1	0
grp01.id.rakuten.co.jp.c6eb5aa856708ef01e34ee398fdc7de75b63656d.info	0	...	0	...	1	...	0	...	0	6	37	0	25	68	4	40	1	0

テストデータの分類結果

・テストデータとして教師データの10%程度を予め確保

混同行列

		予測	
		Malicious (怪しい)	Clean (怪しくない)
実測	Malicious (怪しい)	237 (True Positive, TP)	24 (False Negative, FN)
	Clean (怪しくない)	34 (False Positive, FP)	268 (True Negative, TN)

正解率: 0.896 = (TP+TN) /(TP+FP+FN+TN)
適合率: 0.874 = TP/(TP+FP) ----------
再現率: 0.908 = TP/(TP+FN)
F**値**: 0.890 = 2x((**適合率**+**再現率**)/(**適合率**x**再現率**))

怪しいと感じる FQDN例 4

- 何かの略語的な文字列
- ドメイン生成アルゴリズム(DGA)ドメイン

tamnhindoanhnhan.com

fiyvusa.com

sljjjupfgagolpg.ru

uftfesnodnjflwta.info

wxphewjnfhlyyjj.net

検知モデル Vertion1.1

- 例 4により対応できるようアルファベットおよび数字を細かく分類して入力してみる

FQDN	TLD (150 TLDs)								
	ac	...	com	...	info	...	jp	...	xyz
www.example.co.jp	0	...	0	...	0	...	1	...	0
account-problem-resetting-support-amazn.com	0	...	1	...	0	...	0	...	0
grp01.id.rakuten.co.jp. c6eb5aa856708ef01e34ee398fdc7de75b63656d.info	0	...	0	...	1	...	0	...	0

Total														TLD	2LD		
Dots	Alphabets	Hyphens	Digit	Chars	Digit			Upper Case			Lower Case			Chars	Chars	Malicious	Clean
					0	...	9	A	...	Z	a	...	z				
3	14	0	0	17	0	...	0	0	...	0	1	...	0	2	2	0	1
1	38	4	0	43	0	...	0	0	...	0	3	...	1	3	39	1	0
6	37	0	25	68	3	...	1	0	...	0	3	...	0	4	40	1	0

テストデータの分類結果

・テストデータとして教師データの10%程度を予め確保

混同行列

		予測	
		Malicious (怪しい)	Clean (怪しくない)
実測	Malicious (怪しい)	262 (True Positive, TP)	19 (False Negative, FN)
	Clean (怪しくない)	22 (False Positive, FP)	260 (True Negative, TN)

正解率: 0.927 = (TP+TN) /(TP+FP+FN+TN)
適合率: 0.922 = TP/(TP+FP) ----------
再現率: 0.932 = TP/(TP+FN)
F**値**: 0.926 = 2x((**適合率**+**再現率**)/(**適合率**x**再現率**))

検知モデル Vertion1.3

- せっかくなのでさらに細かく分解して入力

FQDN	TLD (150 TLDs)									Total															
	ac	...	com	...	info	...	jp	...	xyz	Dots	Alphabets	Hyphens	Digit	Chars	Digit			Upper Case			Lower Case			
															0	...	9	A	...	Z	a	...	z	
www.example.co.jp	0	...	0	...	0	...	1	...	0	3	14	0	0	17	0	...	0	0	...	0	1	...	0	
account-problem-resetting-support-amazn.com	0	...	1	...	0	...	0	...	0	1	38	4	0	43	0	...	0	0	...	0	3	...	1	
grp01.id.rakuten.co.jp.c6eb5aa856708ef01e34ee398fdc7de75b63656d.info	0	...	0	...	1	...	0	...	0	6	37	0	25	68	3	...	1	0	...	0	3	...	0	

TLD	2LD				3LD				Hostname				Malicious	Clean
Chars	Alphabets	Hyphens	Digit	Chars	Alphabets	Hyphens	Digit	Chars	Alphabets	Hyphens	Digit	Chars		
2	2	0	0	2	7	0	0	7	3	0	0	3	0	1
3	35	4	0	39	0	0	0	0	35	4	0	39	1	0
4	17	0	23	40	2	0	0	2	3	0	2	5	1	0

テストデータの分類結果

・テストデータとして教師データの10%程度を予め確保

混同行列

		予測	
		Malicious (怪しい)	Clean (怪しくない)
実測	Malicious (怪しい)	252 (True Positive, TP)	19 (False Negative, FN)
	Clean (怪しくない)	17 (False Positive, FP)	275 (True Negative, TN)

正解率: 0.936 = (TP+TN) /(TP+FP+FN+TN)
適合率: 0.937 = TP/(TP+FP) ----------
再現率: 0.930 = TP/(TP+FN)
F**値**: 0.933 = 2x((**適合率**+**再現率**)/(**適合率**x**再現率**))

検知モデル，インターネットの荒波へ (1/2)

- 実際に発生した DNSクエリの分類を実施
  - 最近のフィッシングサイトによく見られるドメイン
    - 例: cert-recovery-support-account-amazon.com
      [Malicious度: 4.653965, Clean度: -3.5990973]
    - 例: account-problem-resetting-support-amazn.com
      [Malicious度: 4.727064, Clean度: -2.733167]
      → 十分な教師データの取得により分類可能
  - DGAドメイン? (ドメイン管理者の方，誤検知だったらすみません)
    - 例: ws1xr1u2b4.top
      [Malicious度: 16.889648, Clean度: -10.467191]
      → 脅威インテリジェンス上 Malicious判定はないもののDGAに似ていると判断
      → ナイスキャッチ!?

検知モデル，インターネットの荒波へ (2/2)

- 実際に発生した DNSクエリの判定を実施
  - アクセス頻度の低い国
    例: www.google.co.zw
    [Malicious度: 23.844645, Clean度: -11.416595]
    → ランダムに選択した Cleanな教師データに“.zw”ドメインが存在しないため
    → 教師データに Cleanな “.zw”ドメインの情報を追加することで誤検知を軽減
    → TLD情報を削除したバージョン (v1.2)も実装したが正解率は低下

  - 以下も同様
    www.google.mk : [Malicious度: 30.812378, Clean度: -16.79775]
    www.google.com.np : [Malicious度: 17.737768, Clean度: -16.856146]
    www.google.ge : [Malicious度: 23.431396, Clean度: -35.038956]

まとめ

- 検知の有効性
  - 一部の特徴的な文字列の FQDNの検知ができた一方で誤検知も多い
    - ”-”が多く使用される最近よくみるフィシングサイト
    - DGAドメイン
  - インシデント調査のきっかけとして利用 (Hunt活動)
- 教師データ
  - Maliciousな教師データの収集はもちろん難しい
  - 一方でMaliciousな教師データに見合う Cleanなデータが量・質ともに必要
  - 教師データにはきちんと精査が必要そう
    - Maliciousな教師データは TLD種別が多く，都度 Cleanな情報を取得する必要有り
    - 実は Maliciousなデータの方に ”www”というホスト名を持つ FQDNが多かった
  - データの偏りは偏った結果を生じさせる

今後の課題

- 入力データ
  - 英文スペース無し形態素解析 -----
    - ihaveabadreputation.com → i have a bad reputation
  - ドメイン登録情報
    - ドメインの登録日，ドメイン登録者の国名 -----
  - FQDNの出現頻度。最近こういう感じの文字列のドメインが増加したなど
  - 侵入履歴のある正当なサイトの検知もしくは除外
- 教師データ
  - 有用な Malicious/Clean教師データをどう効率的に収集するか
- その他の検知への応用
  - C2サーバとのコールバック通信の検知
    - hxxp://www.example.com/MyWS.asmx/GetUpdate?val=H7ddew3rfJid97fer374887sdnJDgsdte
  - フィッシングメールの検知 (Subuject, サイズ，時間などを入力)

ツールをご存知の方いませんか?

APIをご存知の方いませんか?


CISCO