変動標的防御の紹介

小出　洋

九州大学

情報基盤研究開発センター

大分：テーマ宇宙産業

小出　洋（こいで　ひろし）

JavaOne2009 SunSPOT BOFにて

Twitter　@hirosk
Facebook　Hiroshi Koide

経歴

日本原子力研究所計算科学技術推進センター
↓
九州工業大学工学部（戸畑）
↓
九州工業大学情報工学部（飯塚）
↓
九州大学　情報基盤研究開発センター　副センター長
サイバーセキュリティセンター（2017.4～）
システム情報科学府（2017.6～）

社会貢献

- ☆ **SECCON実行委員**
- ☆ **セキュリティ・キャンプ in 福岡**
- ☆ **福岡県警サイバー犯罪テクニカルアドバイザ**

得意分野

- ☆ **プログラミング**
- ☆ **サイバーセキュリティ**

（Moving Target Defense, 脅威トレース…）

Awards

- ☆ **IPA スーパークリエータ**
- ☆ **JavaOne 2005 Duke’s Choice Award**
- ☆ **Sun Microsystems Center of Excellence**

One of my hobbies (Making something)

夏向けの真空管アンプ

★6 ◉3598

74HCU04パラレルアンプ

★5 ◉4981

Arduino Mega でポケコン

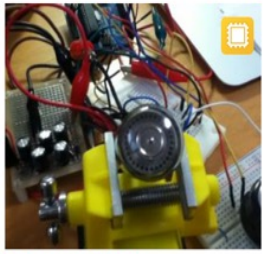
★5 ◉3343

Raspberry Pi ロボットカー（カメラ付）

★1 ◉2403

RasPi+USB-DAC ミュージックサーバ...

★1 ◉2559



Dekatron 遊び

★1 ◉1752

駄球 5AT8 シングルアンプの試作

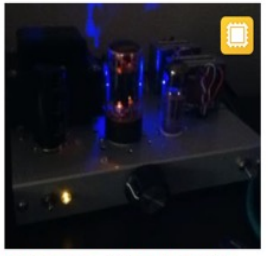
★2 ◉2184

6MJ8 シングルアンプ

★2 ◉1913

バーボンな音がするスピーカをつくってみた

★2 ◉2144



USB DAC を積んだ7号機

★1 ◉1723

真空管ドライブ 6A6 B級PPアンプ

★1 ◉2441

6AT8 シングルアンプ改

★0 ◉2353

【安価】5AT8 シングルアンプ

★2 ◉1937

ユーハイムな音がする4号機

★2 ◉1142

SunSPOT インベーダ

★2 ◉1303

6AT8シングルアンプ（ニッケル水素充電池バイ...

★0 ◉2520

直熱管☆33シングルアンプ☆スタイリッシュ！

★0 ◉1826

Sun SPOT ガイガーカウンター

★0 ◉1118

これまでの話題（振り返り）と今回の話題

第0弾

究極のサイバーセキュリティ対策

～人材育成：九州大学での社会人サイバーセキュリティ教育の取り組みと地域・コミュニティの連携～

2021/2/3 14:15～14:45

地域セキュニティを盛り上げていこう！

第1弾

地域セキュニティにおける産官学の連携について考える

2021/9/28 15:10～15:40

産官学のそれぞれのステークフォルダの連携が肝要

第2弾

ビジネスに必要なサイバーセキュリティの要素

2021/10/28 14:20～14:50

サイバーセキュリティが持つ科学的特徴の性質からビジネスに必要なサイバーセキュリティの要素について考える

今回の話題！

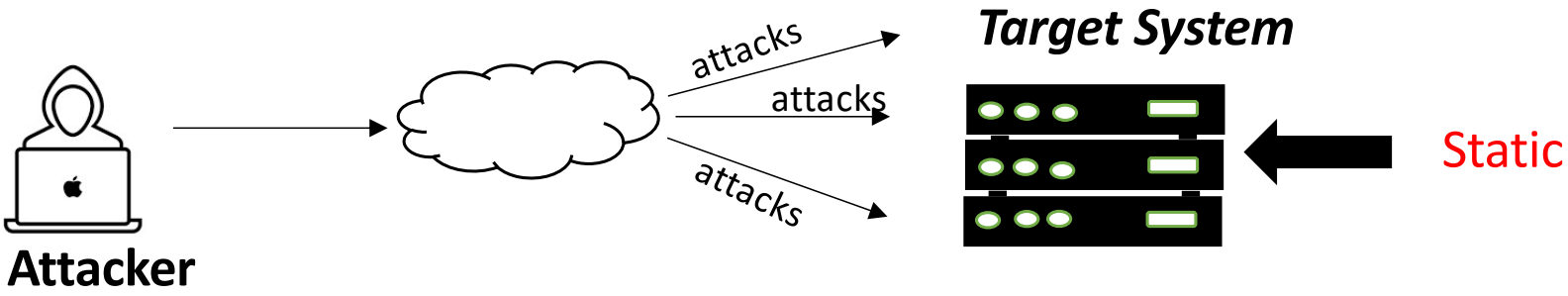
第3弾

変動標的防御の紹介

2021/11/10　14:10～14:40

小出が研究しているMTD（Moving Target Defense）について紹介

サイバー空間の状況
Situation in Cyberspace



攻撃者に有利な状況

- 短時間に攻撃を実行可能
- ひとつ攻撃可能な脆弱性を見つければ良い
- 偵察と準備に多くの時間を掛けることが可能
- 継続的，適用的，目的主導的（最小攻撃コストで最大利益）

いままでの防御技術では不十分

Attackers have advantages.

- can execute attack in a short time
- need to find only a single vulnerable entry point
- unlimited time for reconnaissance / preparation
- They are persistent, adaptive and incentive-driven (minimum attack cost with maximum dangerous outcome)

Traditional defense techniques are not sufficient

解決したい課題
Problem statements

- どうすれば攻撃者側の負荷と攻撃に要する時間，コストを増やすことができるか
- どうすれば攻撃成功確率を減らすことができるか
- どのように新しい防御技術追加することによりシステムの防御能力を拡張することができるか
- 複数の多様化を行う方法が最小限の防御コストでセキュアな防御システムとなりうるかどうか

- How can we increase attackers' workload and attack time, costs?
- How can we reduce the *probability of attack success* over time?
- How can we enhance a system's resiliency with additional defense techniques?
- Whether a hybrid diversification approaches can introduce a secure defense system with minimum defense cost or not?

Moving Target Defense Approach

MTD (Moving Target Defense)の基本的な考え方は，複数のシステムの変更を制御することにより，保護したいシステムの情報に関する不確実性を高めて攻撃側を複雑にすること

これにより攻撃者の調査と攻撃にかかるコストを増やし攻撃の機会を減らすことができる

The concept of controlling the alteration of *multiple systems* with the aim to *increase uncertainty* about a protected system's information and give *complexity* for attackers.

By doing so, we can reduce the attack window of opportunity and increase the costs of attackers' probing and attack efforts.

停まっている標的を射撃

Shoot a stationary target

動いている標的を射撃

Shoot a moving target

Goals of This Technique

- 攻撃の不確実性を高める
- 攻撃に必要な努力とコストを増加させる
- 保護したいシステムは探査しにくく予測しにくい標的となる
- 攻撃成功確率を低減する

MTD技術によりランダム性が加わるため，特定の瞬間にシステムがどうような構成となっているかわからない．このランダム性のため攻撃者がゼロディ攻撃などを成功させるためのコストが増加する可能性を持つ

MTDフレームワークを設計するときの3つの考慮するべき事項：*(**what, when, how**)* が変更可能なパラメータ

- To increase the uncertainty for the attacker
- To increase *attacker's effort and cost* in making attacks
- A protected system will be *hard to be exploited and unpredictable* destination
- To reduce the *probability of attack success* over time

MTD techniques will have randomness built, so this randomness can increase the cost for an attacker to succeed in using e.g, ***zero-day attacks*** because it does not necessarily know which configuration of the system is in place at any particular moment.

Three consideration when to design MTD framework: *(what, when, how) to move the elements*

What to Move ?

攻撃者を混乱させるために動的に変更できるシステム構成，属性，構成要素（すなわち attack surface のこと）

例）

- 命令セット
- アドレス空間のレイアウト
- IPアドレス，ポート番号
- プロキシ
- 仮想マシン
- OS
- ミドルウェア
- フレームワーク
- ソフトウェア

what system configuration attribute or elements (components) (i.e., attack surface) can be dynamically changed to confuse attackers.

For example,

- ❖Instruction sets
- ❖address space layouts
- ❖IP addresses, port numbers
- ❖proxies, virtual machines
- ❖operating systems
- ❖software programs

When to Move ?

MTDシステムのある状態から別の状態に変更するのに適切なタイミングを決定し，攻撃者が取得した現在の状態，または攻撃の状況を無効化

- 反応的な適応
- 積極的な適応
- ハイブリッド適応

deciding the optimal time to change from the current state of an MTD system to a new state, invalidating information or progress gained by an attacker in the current state.

- Reactive adaptation
- Proactive adaptation
- Hybrid adaptation

How to Move ?

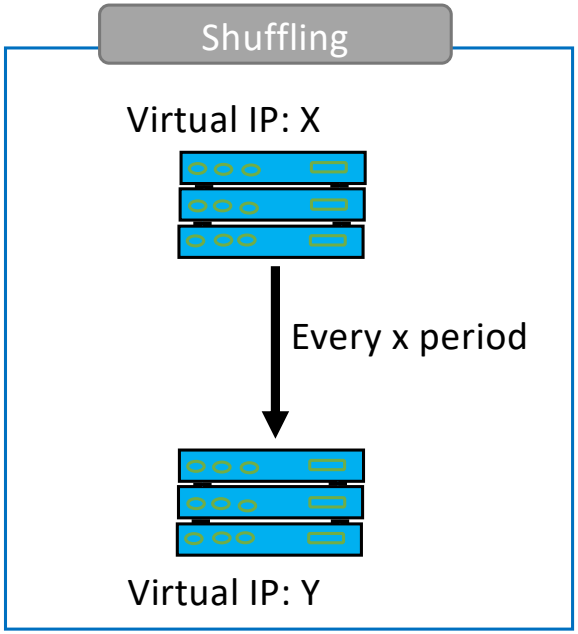
- どのようにシステムの属性や構成要素（すなわち標的）を変化させて，攻撃者側の予測不可能性や不確実性を増加させ，攻撃者を混乱させるか

❖シャッフリング，
❖多様性，
❖冗長性

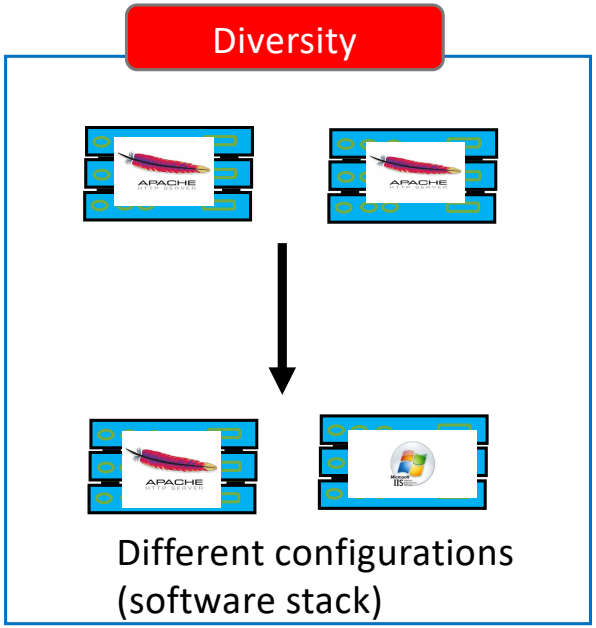
- *how to change the moving attributes or elements (components) (i.e., targets) to increase unpredictability and/or uncertainty, leading to an attacker's high confusion.*

❖ shuffling,
❖ diversity,
❖ Redundancy

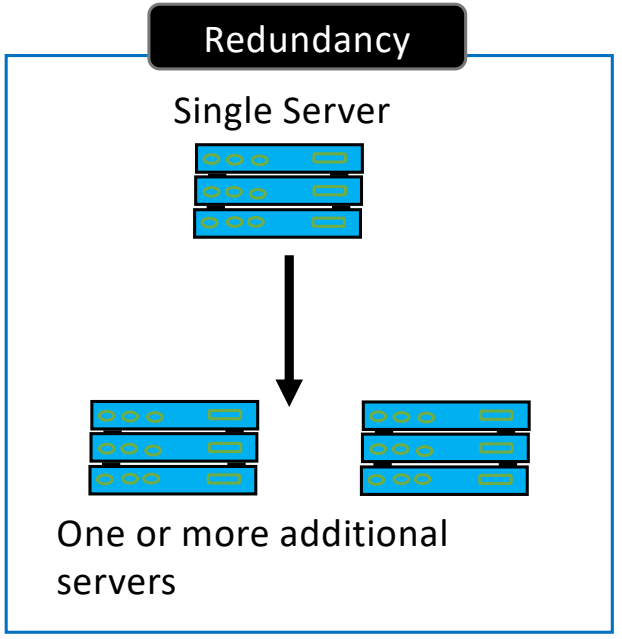
Examples of MTD Techniques



Performance and Efficiency
正規の利用者に対する
Availabilityの確保



Resilience and Robustness



Reliability and Availability

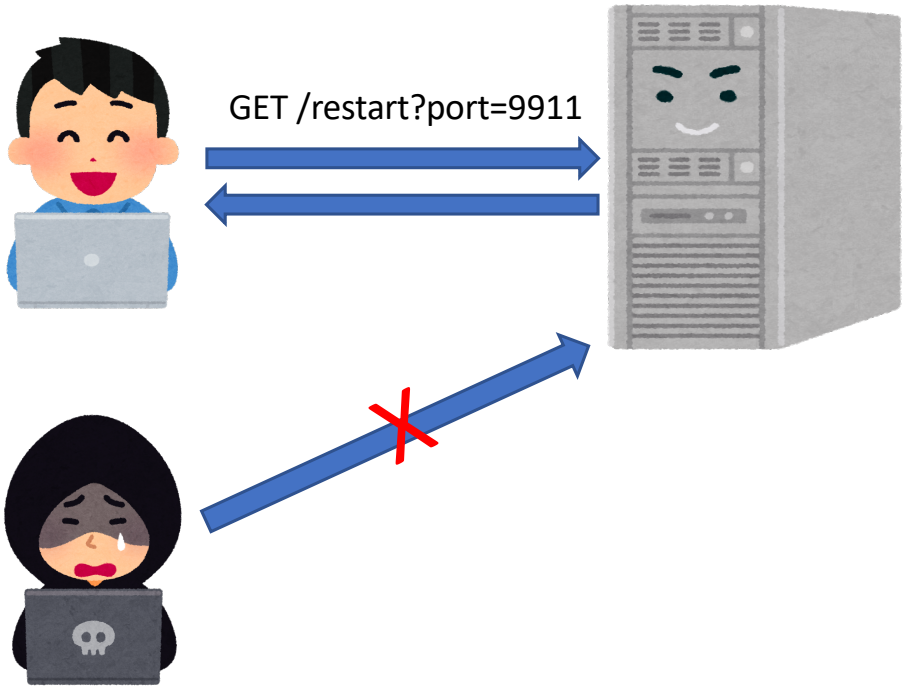
異なる階層におけるMTD技術の変更要素
Elements of MTD Techniques at Different Layers

Layers	MTD Techniques		
	Shuffling	Diversity	Redundancy
Application	TCP/UDP port numbers	Web: Apache, IIS, etc. App: Java, PHP, etc. DB: SQL, MySQL, Oracle, etc. Others: mail-server, proxy-server, etc.	Web service replica Application replica Database replica Other service replica
OS-host VM-instance	IP address, System Call Numbering Virtual IP address	Windows server various versions Linux various versions Other Unix various versions	Host OS and VM replica
Virtual Machine Manager	Failover, switchover	VMware, ESXi Other kernel-based VM, Vbox, etc.	Hypervisor's replica

MTD機能を持つWebアプリケーション

Webアプリケーションにクライアント側からポート番号を指定できる次の機能を追加

1. 最初ユーザは指定されたポート番号（初期値）でアクセス可能
2. 「GET /restart?port=新しいポート番号」により次のアクセス時の新しいポート番号に変更可能
3. それ以外は普通のWebアプリケーションの機能を持つ



MTD機能を持つ簡単なWebアプリケーションを作成

```python
import threading
from urllib.parse import urlparse,parse_qs
from http.server import BaseHTTPRequestHandler,HTTPServer

class MyHandler(BaseHTTPRequestHandler):

    def set_port(self, port = 8123):
        self.port = port

    def set_server(self, server):
        self.server = server

    def contents(self):
        # Write content here.
        body = b'<<This is a response.>>\n'
        self.send_response(200)
        self.send_header('Content-type', 'text/html; charset=utf-8')
        self.send_header('Content-length', len(body))
        self.end_headers()
        self.wfile.write(body)

    def do_GET(self):
        parsed_path = urlparse(self.path)
        queries = parse_qs(parsed_path.query)
        self.contents()
        if parsed_path.path == '/restart' :
            next_port = int(queries.get('port')[0])
            self.server.running = False
            m = MainServer(next_port)
            m.start()
        elif parsed_path.path == '/shutdown':
            self.server.running = False
```

```python
class MainServer:

    def __init__(self, port = 8123):
        handler = MyHandler
        handler.set_port(self)
        self.server = HTTPServer(('0.0.0.0', port), handler)
        handler.set_server(self, self.server)
        self.thread = threading.Thread(target=self.run)
        self.thread.deamon = True

    def set_port(self, port = 8123):
        self.port = port

    def get_port(self):
        return self.port

    def run(self):
        self.server.running = True
        while self.server.running:
            self.server.handle_request()

    def start(self):
        self.thread.start()

    def shut_down(self):
        self.server.shutdown()

m = MainServer()
m.start()
```

サンプル実装　https://bit.ly/3Aufwtr https://gist.github.com/koide55/09198d833d0e9c6444c7d1d73cd126c3

MTD機能を持つ簡単なWebアプリケーションを作成

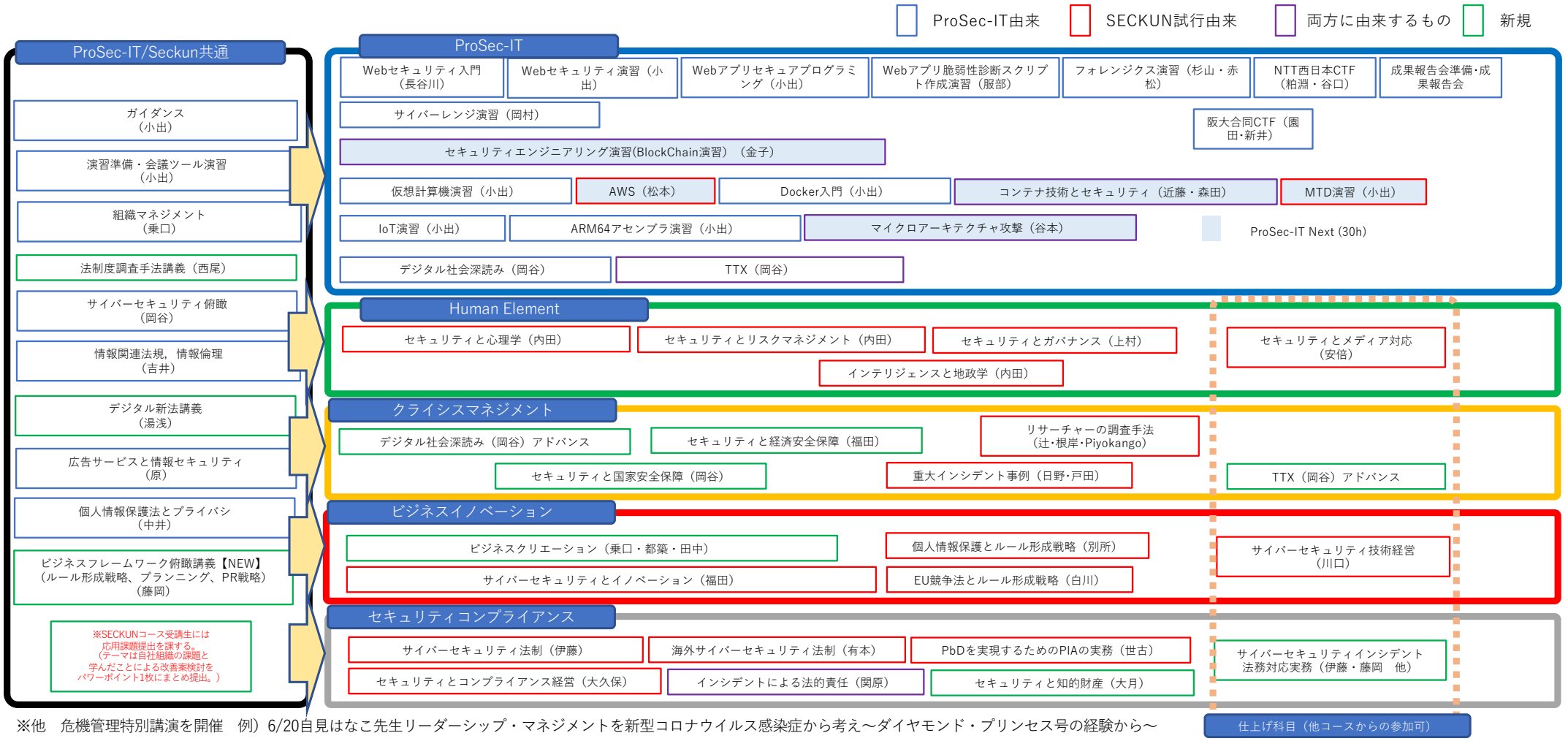
実行例

```
$ curl localhost:8123 ↓ ← 当初はポート番号8123でアクセス可能
<<This is a response.>>
$ curl localhost:8123/restart?port=9999 ↓  ← ポート番号9999に変更
<<This is a response.>>
$ curl localhost:8123 ↓ ← ポート番号8123ではアクセスできない
^C
$ curl localhost:9999 ↓ ← ポート番号9999でアクセス可能
<<This is a response.>>
$ curl localhost:9999/restart?port=8111 ↓   ← ポート番号8111に変更
<<This is a response.>>
$ curl localhost:8111 ↓ ← ポート番号8111でアクセス可能
<<This is a response.>>
```

サンプル実装 **https://bit.ly/2YJLAfv**

2021年度SECKUN+ProSec-ITカリキュラム

九州大学



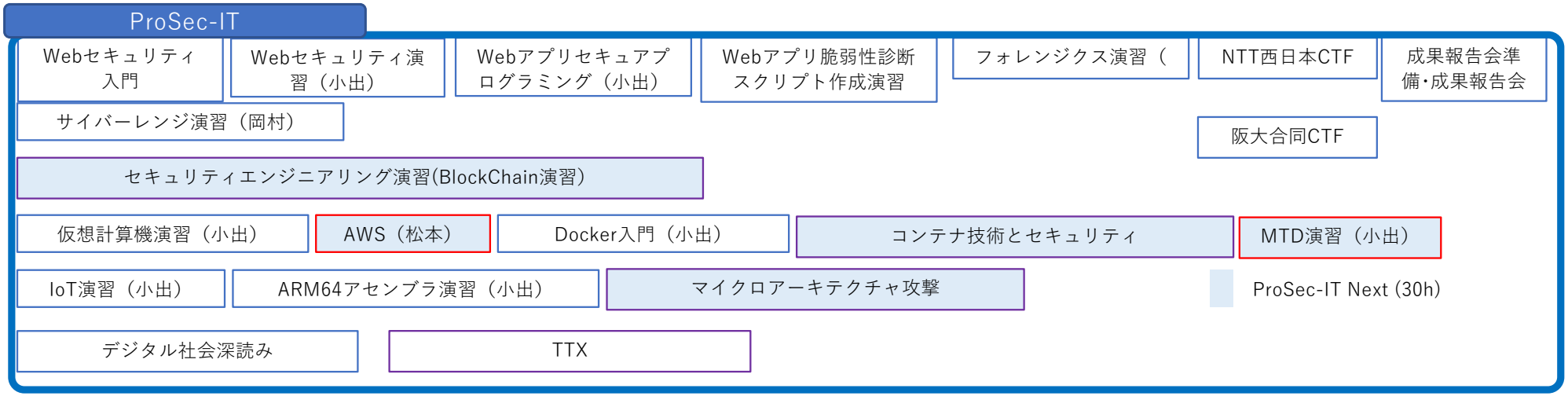
※他　危機管理特別講演を開催　例）6/20目見はなこ先生リーダーシップ・マネジメントを新型コロナウイルス感染症から考え～ダイヤモンド・プリンセス号の経験から～

サイバーセキュリティの要素（テクノロジ）

- テクノロジ（IT）
  - Webセキュティ（セキュアプログラミング，脆弱性診断）
  - フォレンジクス
  - 仮想環境（AWS, Docker，コンテナ技術）
  - IoT
  - アセンブラ，機械語
  - マイクロアーキテクチャ
  - CTF
  - Moving Target Defense
  - 最新技術について調査し，応用する技術

- 決して無視はできない
- ビジネスに（情報システム，インターネット，セキュリティ）は必要
- 情報システムについての基本的な理解は必要

小出のMoving Target Defense（Hybrid R2.9.13）



まとめ

- MTD (Moving Target Defense) という技術を紹介
- MTDの例としてネットワークレベルのMTDのデモを実施
- （可用性などとのトレードオフになるが）攻撃者側の負担を大きくするという意味でMTDは有効に利用可能
- /bin/sh を別の名前に変えたら多くの shell code は失敗