BitVisor Summit 7

CTFVisor: BitVisorによるCTF作問・出題支援

松原 克弥　　源 啓多　　中田 裕貴
公立はこだて未来大学

2018年11月28日

背景：実践的情報教育

- 大学等への実践的情報教育の導入が進んでいる
  →現実のシステムと直結した内容を学ぶ
  - ex.) プロジェクト学習 (PBL)

両方の実践的教育を通して、
高度IT人材を育成

- 学外では、ハッカソンやハンズオン， LT(Lightning Talk) 大会，**CTF**などの IT 技術の向上を目的とした教育系イベントが開催が増加
  →現実のシステムを用いて学ぶ

Catch The Flag (CTF)

- IT 技術に関する問題に対して適切な形で対処することで，その結果得られる得点で勝敗を決める競技
  - Attack & Defense - 脆弱性のあるシステムを攻撃から防御しつつ，他チームのシステムの脆弱性を突いて情報を読み出す攻防戦形式
  - **Jeopardy** - ファイルや画像，システム入出力データから指定された情報 (フラグ) を読み出す早さを競うクイズ形式

暗号や符号理論，信号処理，画像処理やネットワーク技術，プログラミング言語，データベース，ファイルシステムといった OS 技術などの計算機科学に関する実践的な知識が求められる

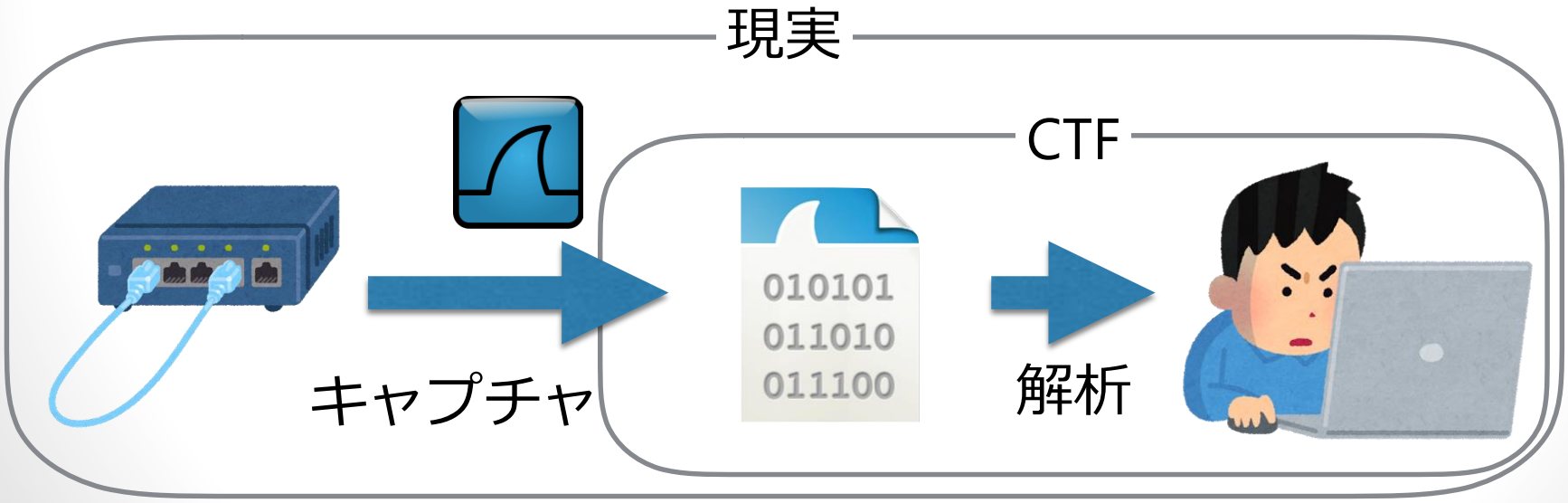
- 高度IT人材の育成手段として**産官学が注目**
  - 特にセキュリティ分野

Jeopardy の出題形式

形式	説明	対応する教育分野の例
Pwn	サーバ上で動くプログラムの脆弱性を攻撃して権限を奪取	セキュリティ，OS, DB, etc.
Reversing	プログラムバイナリを読み解いて動作を理解	OS, コンパイラ, プログラミング
Web	Web サービスの脆弱性を発見	情報セキュリティ, ネットワーク
Crypto	暗号文を解読	暗号処理
Network	ネットワークパケットを読み解く	ネットワーク, OS
Forensics	デバイス入出力 RAW データを解析	計算機アーキテクチャ, OS
Stego	画像データや音声データを解析	画像処理，音声処理
Recon	SNS等のインターネット上のデータを探索	ネットワーク
PPC	競技プログラミングの課題を順に回答	プログラミング

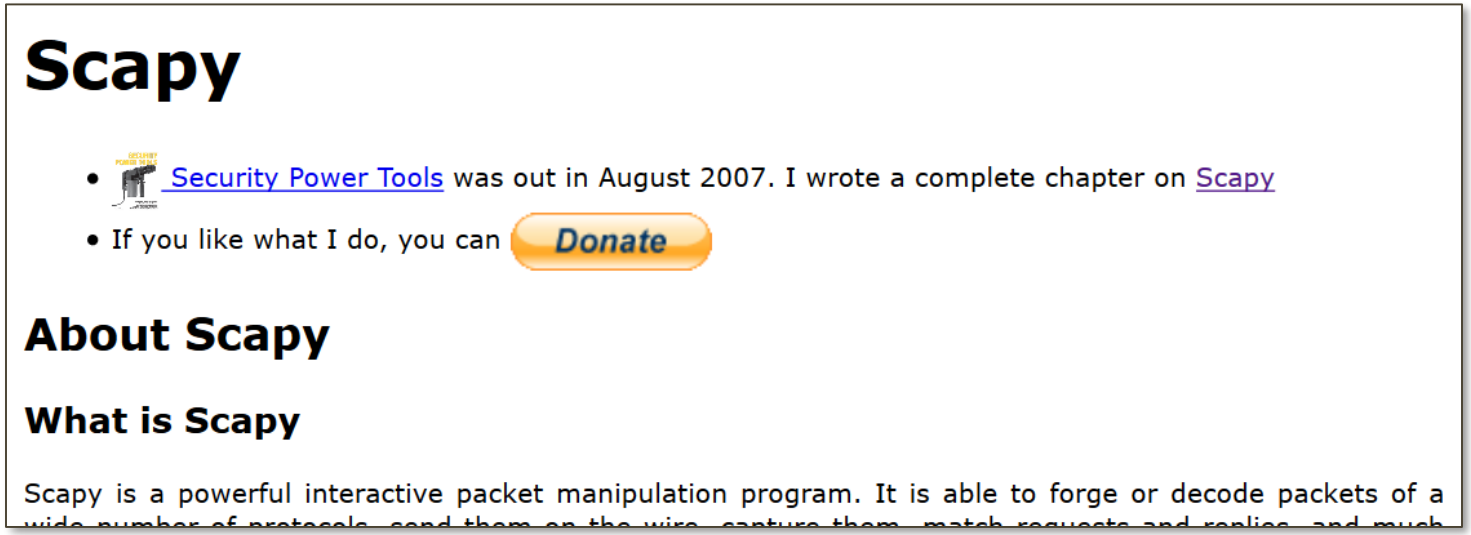
課題

- 教育機関や企業で手軽に開催するのは難しい
  - 特殊なツールへの精通（Scapy, USBPcap…）
- 実践的な問題を作成しにくいジャンルがある
  - ログやダンプを渡して解析をさせるだけになりやすい
    - 例: ネットワークパケットの解析問題



Scapy

- Pythonで書かれた対話型のパケット作成ツール
  - http://www.secdev.org/projects/scapy/

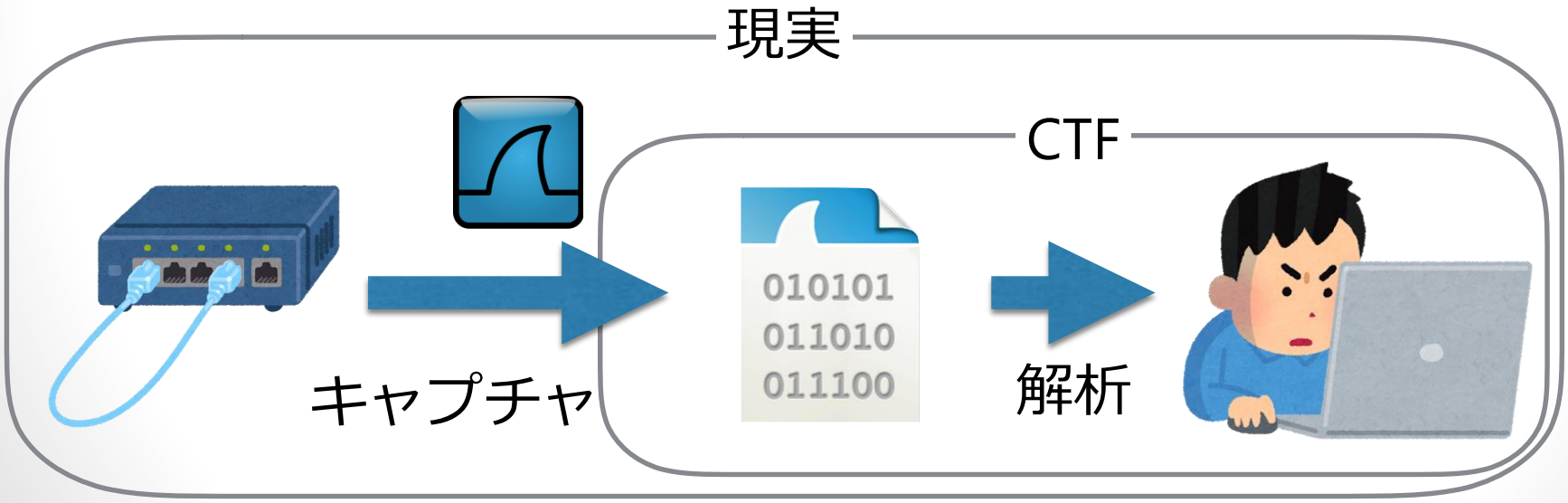



```
$ sudo scapy
Welcome to Scapy (2.2.0)
>>> ip = IP(dst='192.168.0.1', id=1000)
>>> tcp = TCP(dport=888, flags='FSRPAU')
>>> send(ip/tcp)
.
Sent 1 packets.
```

TCPヘッダのコントロールフラグをすべて1にしたパケットの作成

課題

- 教育機関や企業で手軽に開催するのは難しい
  - 特殊なツールへの精通（Scapy, USBPcap…）
- 実践的な問題を作成しにくいジャンルがある
  - ログやダンプを渡して解析をさせるだけになりやすい
    - 例: ネットワークパケットの解析問題



Jeopardy の出題形式

形式	説明	対応する教育分野の例
Pwn	サーバ上で動くプログラムの脆弱性を攻撃して権限を奪取	セキュリティ，OS, DB, etc.
Reversing	プログラムバイナリを読み解いて動作を理解	OS, コンパイラ, プログラミング
Web		イ, ネッ
Crypto	暗号文を解読	暗号処理
Network	ネットワークパケットを読み解く	ネットワーク, OS
Forensics	デバイス入出力 RAW データを解析	計算機アーキテクチャ, OS
Stego	画像データや音声データを解析	画像処理，音声処理
Recor		
PPC		

ログやダンプでの出題が多い

ログやダンプではないが，
画像や音声も静的なファイルとして
渡される事が多い

目的と提案

目的

① 手軽にCTFの作問を行えるようにする

② より実践的な問題の作成

- 特にJeopardyにおけるNetwork, For, Stegoの3ジャンル

提案

① 複数のジャンルを作問できる，統一的なツールの作成

② 作問・出題に仮想化技術を活用して，実環境での体験を提供

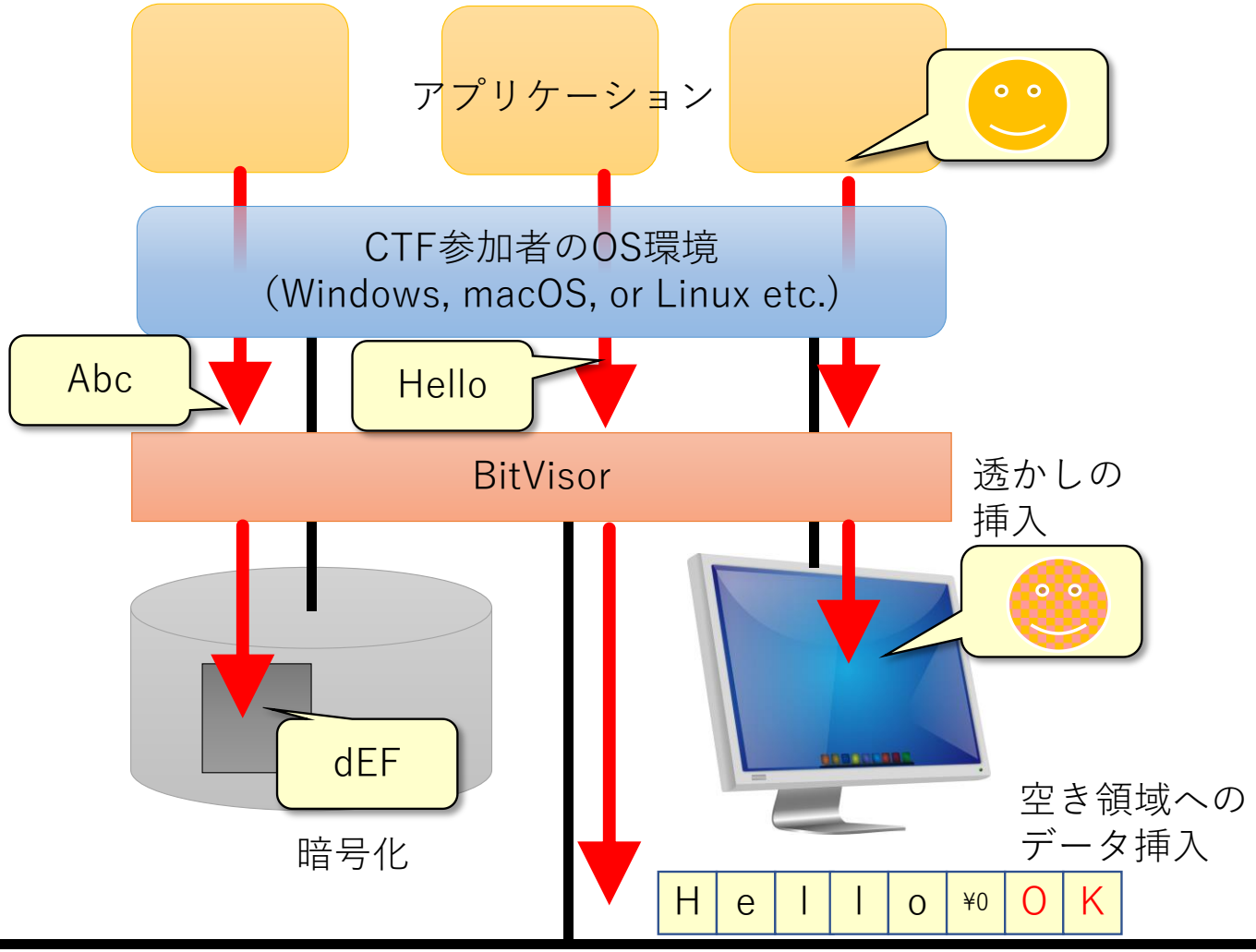
- OSとハードウェアの間で動作し，デバイス入出力を改変
- 競技者の使用しているOSに依存しない

実現手法

- BitVisorをベースにしたCTF作問支援ソフトウェア CTFVisor を開発
  - USBから起動できる形で競技者に配布し，競技者のマシンで直接動作
    - 仮想化技術を用いるため，競技者の使用しているOSに依存しない
  - 特定のデバイス入出力を改変しFlag挿入
    - 競技者は普段と違う挙動を見つけ，調査する

CTFVisor

CTFVisorの機能例



作問例１：
TCPヘッダへのフラグ挿入

（従来の方法）

1. Scapyをインストール
2. TCPパケットを作成
3. 実際に送信
4. WireSharkで送信パケットをキャプチャ
5. pcapファイルを保存

→ **pcapファイルをCTF参加者へ配布**

作問例１：CTFVisorによるTCPヘッダへのフラグ挿入

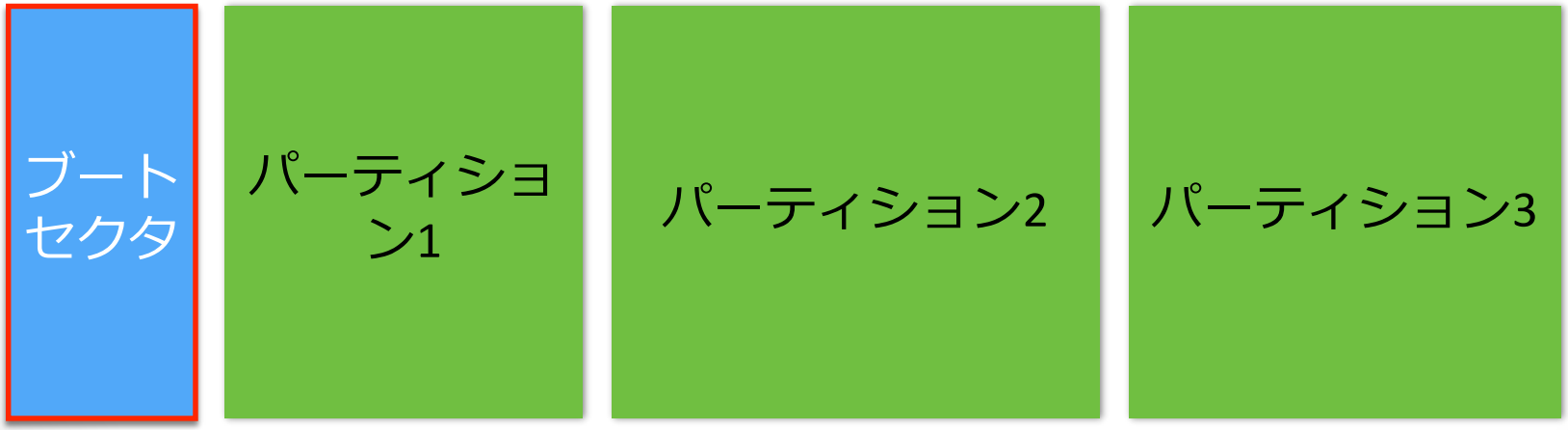
- NIC（ネットワークデバイス）へのDMA転送をフックし，特定のパケットにFlagを挿入
- TCPヘッダにある3bitの予約領域を利用する

```
59990 → 80 [SYN, Reserved] Seq=157145027 Win=29200 Len=0 MSS=1460 SACK_…
59990 → 80 [ACK, Reserved] Seq=157145028 Ack=1632986782 Win=29312 Len=0…
GET / HTTP/1.1
59990 → 80 [ACK, Reserved] Seq=157145103 Ack=1632988358 Win=32384 Len=0…
59990 → 80 [FIN, ACK, Reserved] Seq=157145103 Ack=1632988358 Win=32384 …
59990 → 80 [ACK, Reserved] Seq=157145104 Ack=1632988359 Win=32384 Len=0…
```

```
▼ Flags: 0xe02 (SYN, Reserved)
    111. .... .... = Reserved: Set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
```

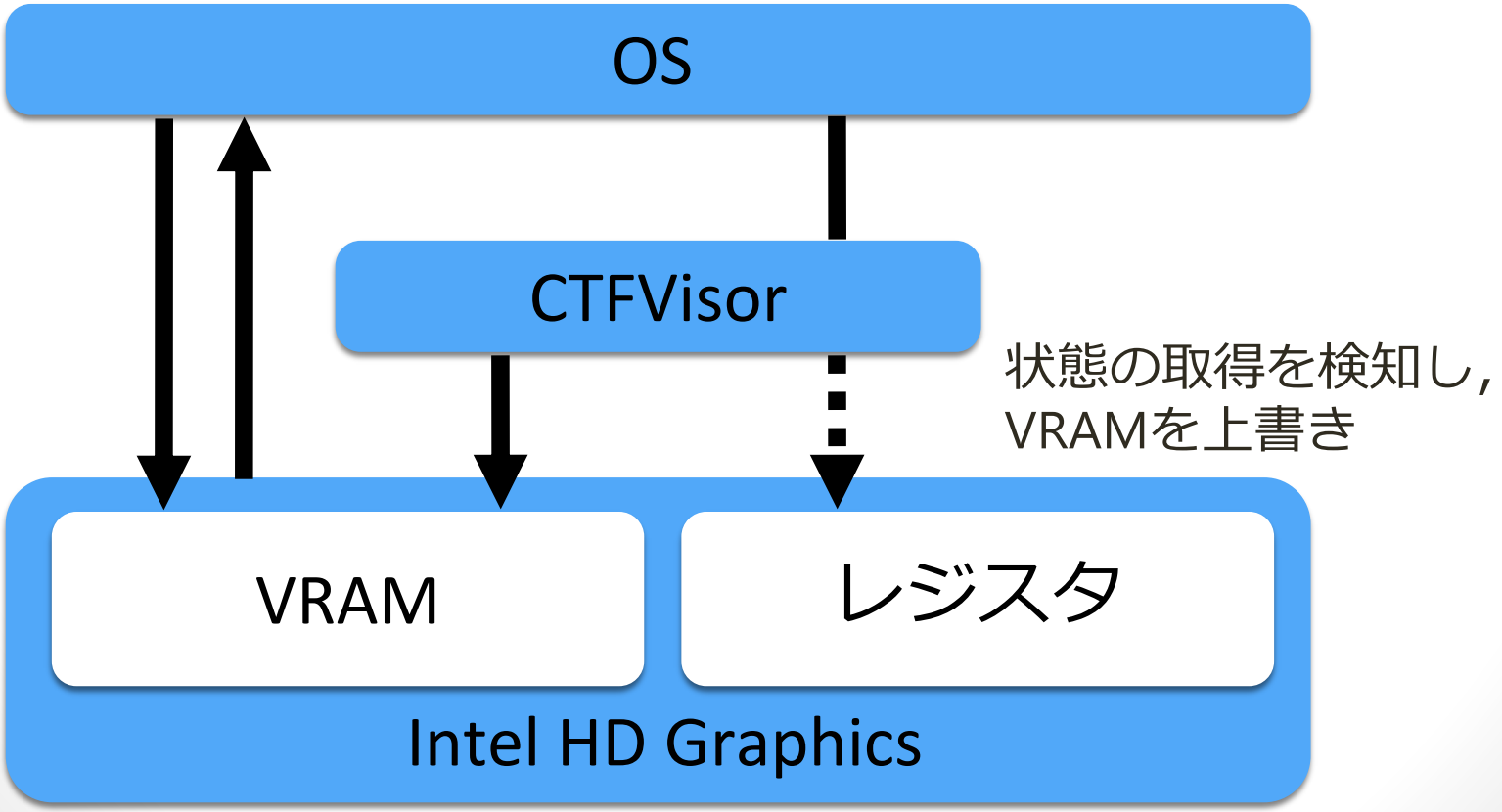
作問例２：ディスクI/Oへのフラグ挿入

- OSからアクセスされないようなディスク領域への読み出しをフックして，偽のデータとしてFlagを返す
- ブートセクタなどのOSから読み出されない領域を利用



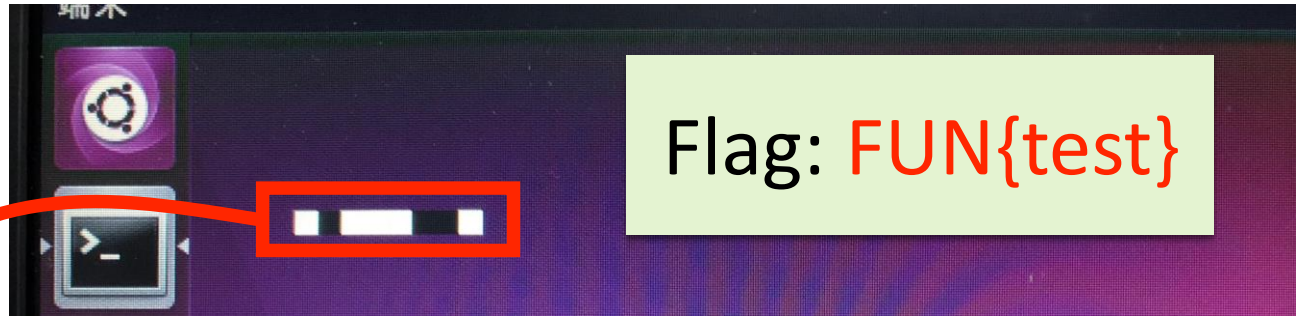
作問例３：
画面描画へのフラグ挿入

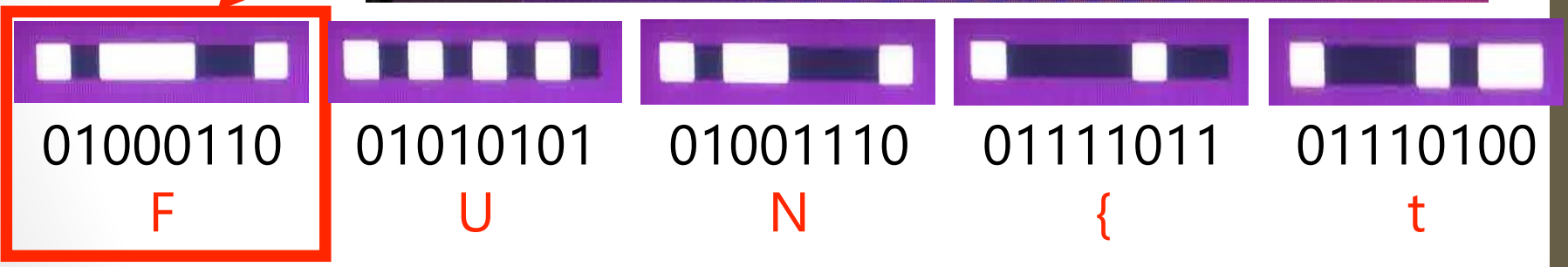
- OSが描く画面に重ね合わせ



気がつきましたか？

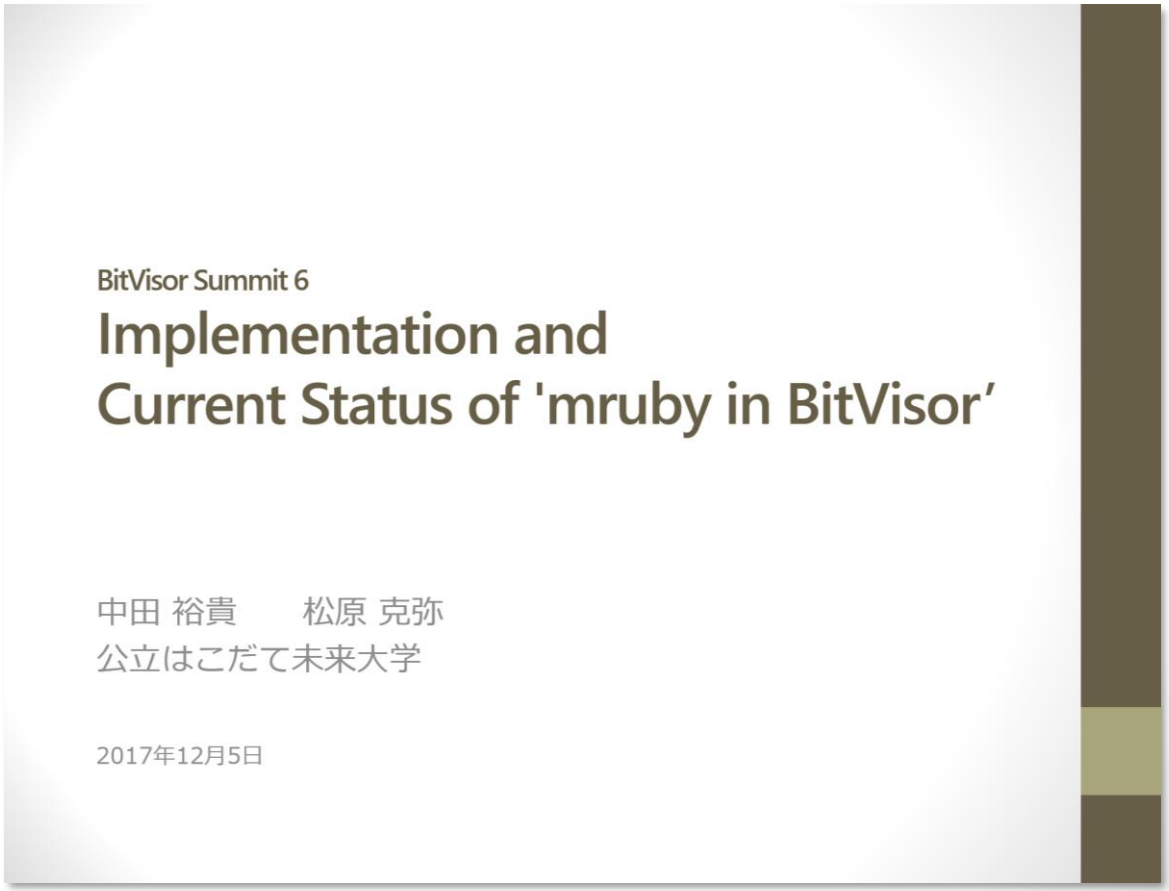
- 表示しているのはASCIIコードを2進数表記して「1->黒，0->緑」に対応させた文字列





検討：高水準プログラミング言語によるデバイス入出力の加工

- BitVisorへのmrubyの組み込み
  - BitVisor Summit 6で紹介



mrubyによる作問記述

```
# ディスク読み出し改変の設定
def network(buf, cylinder, head, sector)
  flag = “fun{test2}”
  if buf.read? && cylinder.zero? && head.zero? &&
sector.eql(1)
    flag
  end
  buf
end
```

mrubyによる作問記述 (contd.)

```
# ネットワークパケット改変の設定
def network_bits
  flag = “fun{test2}”
  # 次の3ビットを返すジェネレータ（略）
end
def network(buf, is_recv)
  if !is_recv && buf.tcp?
    buf.reserved = network_bits.next
  end
  buf
end
```

```
# 画面出力改変の設定
def display(buf)
  flag = “fun{test}”
  # LSBを用いたフレームバッファへのフラグ挿入
  lsb(buf, flag)
end
```

まとめと今後の課題

まとめ

軽量仮想マシンモニタである「BitVisor」を用いて，
CTF作問支援ソフトウェア「CTFVisor」を開発

- ツールの統一による作問の難しさの軽減
- 動的にデバイス入出力を書き換え，実践的な問題に

今後の課題

- mrubyによる作問記述
- USBなどの他デバイス入出力への対応
- CTF競技開催による評価
  - 作問者：作問しやすさ、問題内容の対応範囲
  - CTF参加者：難易度、問題のおもしろさ、学習効果