通信可視化と動的解析の連携による攻撃解析支援

○義則隆之[†]　伴拓也[†]　宮嵜仁志[†]　松井拓也[†]　佐藤両[†]　岡崎亮介[†]
篠田昭人[†]　廣友雅徳[‡]　毛利公美[††]　神薗雅紀[‡‡]　白石善明[†]

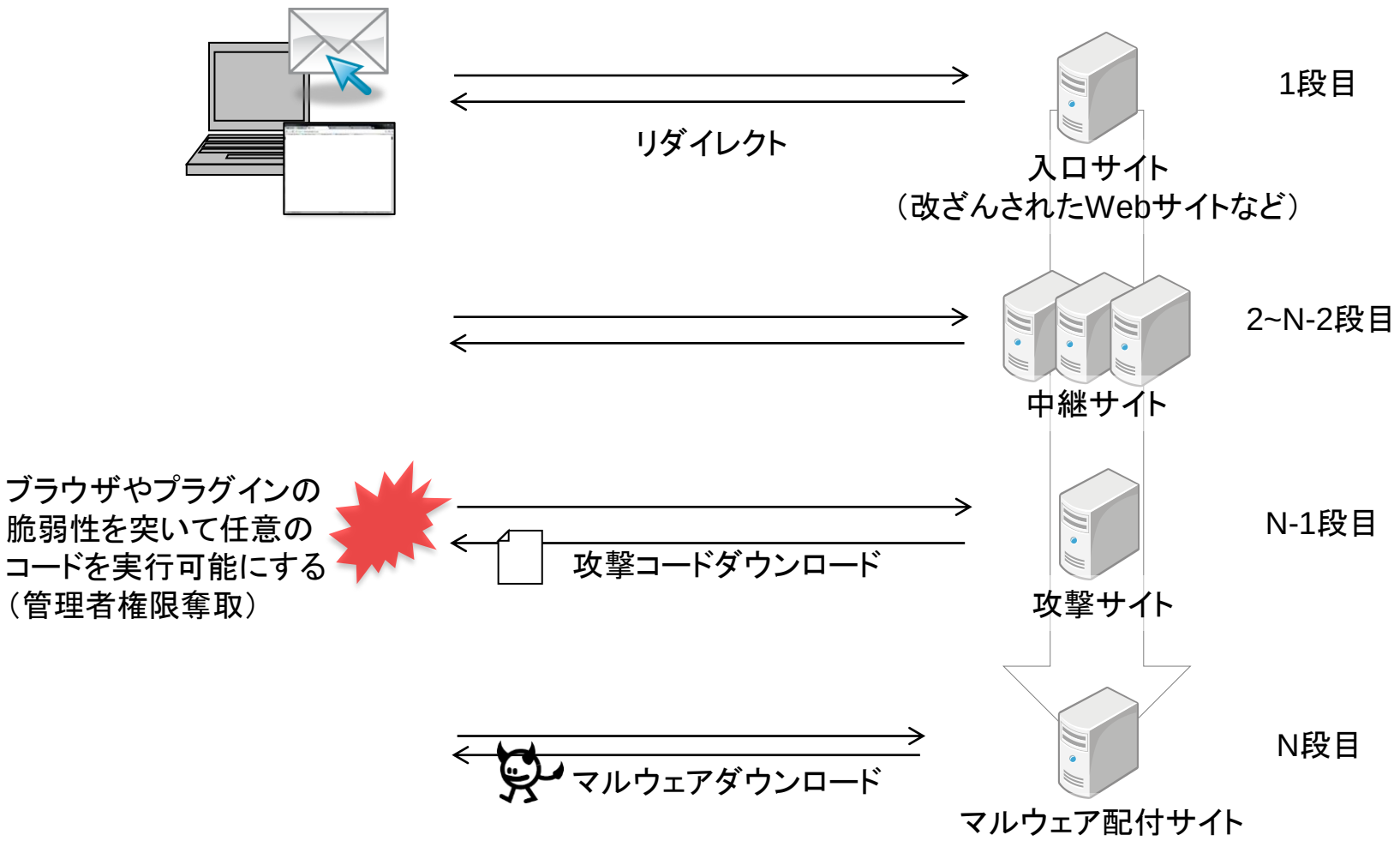
[†]名古屋工業大学　[‡]佐賀大学　[††]岐阜大学　[‡‡]株式会社セキュアブレイン

2012/10/31

発表概要

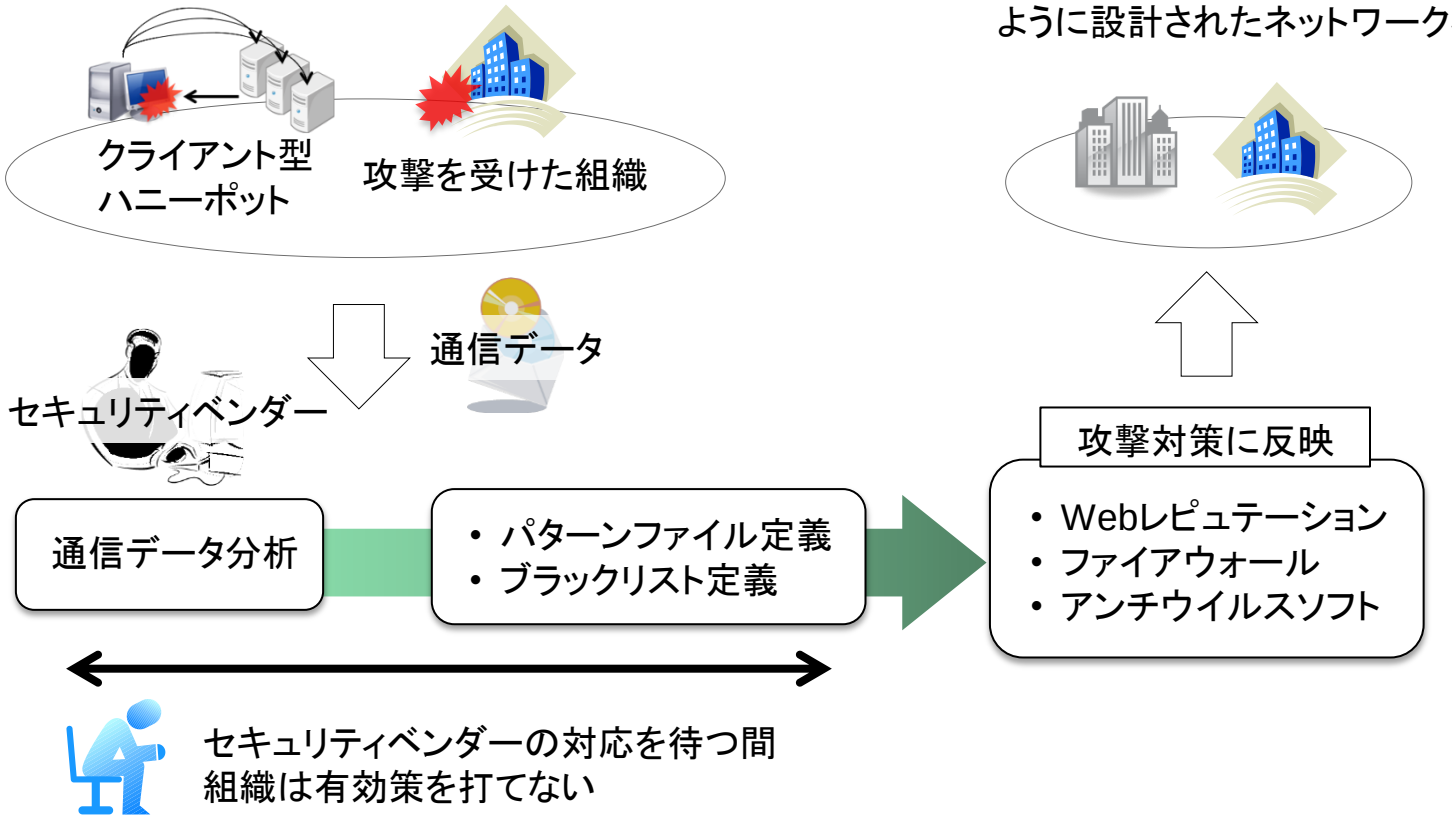
- 目的
  - DBD（Drive-by-Download）攻撃の被害の拡大を抑える ← 各組織で迅速に対策を講じられるよう支援
    - 攻撃フローの全容を把握できれば適切な対策を講じられる
    - 通信データから脅威を見つけ出すのは時間のかかる作業である
- 課題
  - 人が通信データを解析し攻撃フローの全容を把握することを支援する
- 本研究のコンセプト
  - 支援方法1： 攻撃フローを可視化し，攻撃の一連の流れを把握する
  - 支援方法2： 脆弱性を突くコードを配布するサイトを特定する
- 提案システム
  - 通信データに含まれる通信フローを出現した時間順にビューワーで世界地図上に描画する
  - 誤って描画されていると思われるフローをクリックすると，配布されたファイルを自動で動的解析しフローを自動で修正する
  - 攻撃サイト，マルウェア配布サイトのフローをクリックすると，配布されたファイルを自動で動的解析し不正なコードの有無を確認する
- 評価
  - 9個中9個の攻撃を特定することができた
  - 一連の流れは27秒を超えずに行えることを確認した

DBD（Drive-by-Download）攻撃



DBD攻撃の対策を講じるまでの流れ

※ハニーポット：意図的に脆弱性を有し攻撃を受けやすいように設計されたネットワーク機器のこと



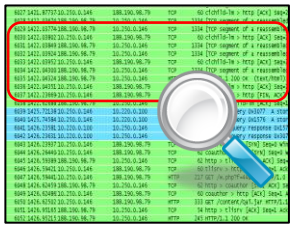
攻撃の防御・被害の抑制を目指す

⟶ 各組織が通信データを解析し，各組織で迅速に（＆連携して）攻撃対策を講じられるのが望ましい

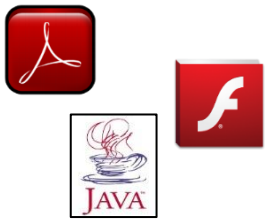
通信データ分析の課題

通信データ分析手順

疑わしい通信フローに
当たりをつける



通信フローに含まれる
ファイルを復元する



ファイルを解析する

静的解析

動的解析

攻撃全容把握

問題

- 通信データから疑わしい通信フローを探し出すのは相応の時間を要する

- リダイレクトコードや攻撃コードは難読化されていることが多く，静的な解析だけでは正確に通信の流れを追うのは困難である

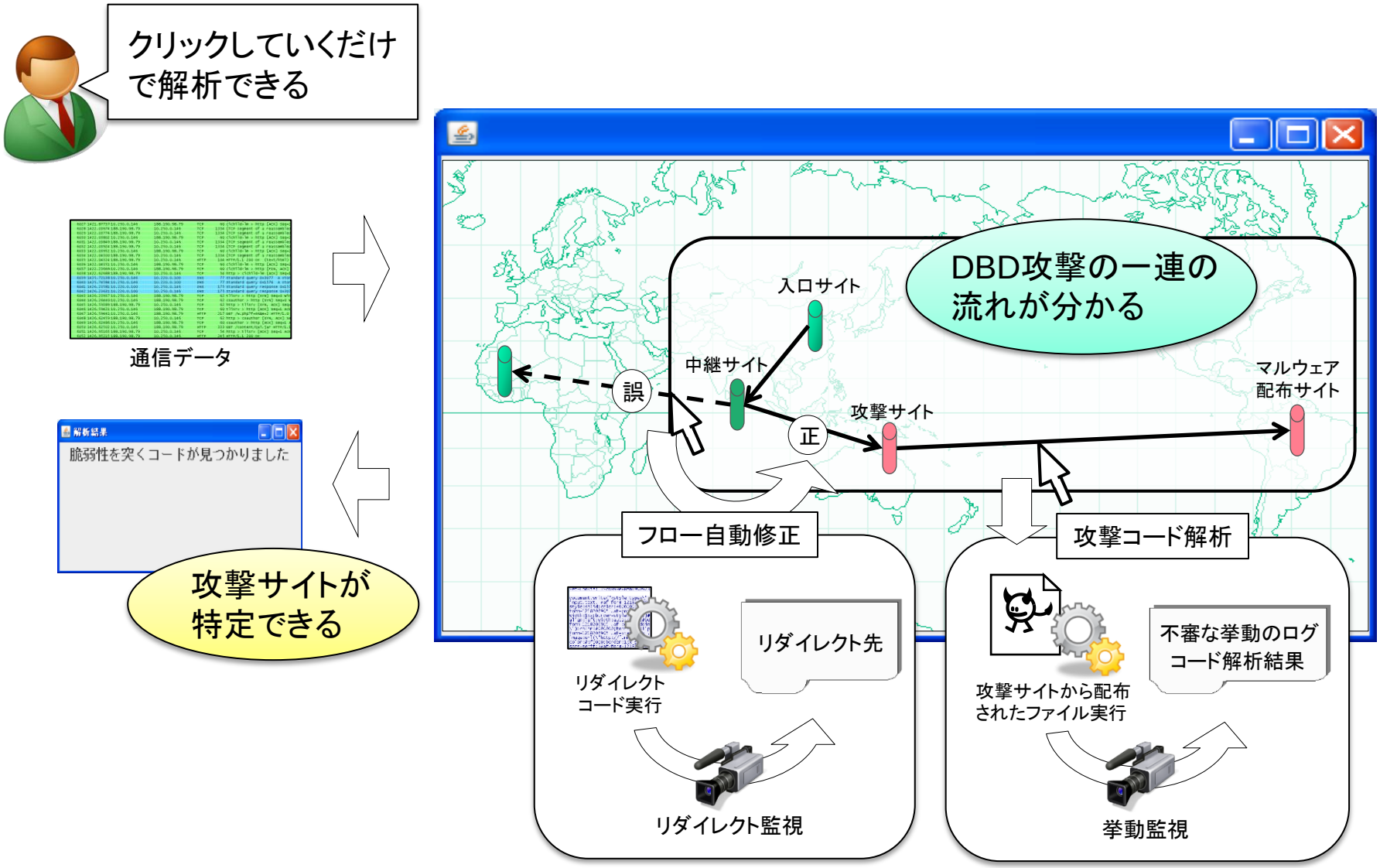
```
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+
,/n{n+,/+#n+,/#;¥
#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l q#'+d'K#!/+k#;¥
q#'r}eKK{#}w'r}eKK{nl]'/#;#q#n'}{}#}w'){}{nl]'/+#n';d}rw'
i;# ){nl]!/n{n#'; ¥
r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#
```

難読化されたコード

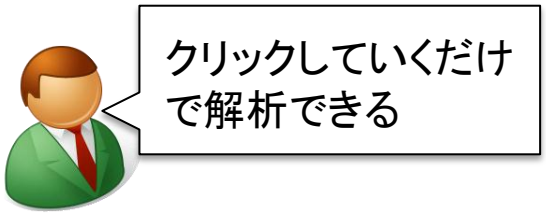
課題

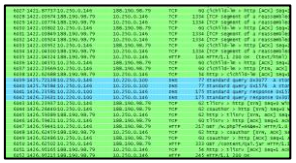
通信データを人が解析するときに，攻撃フローの全容を把握することを支援する

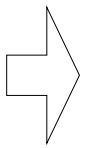
提案システムのコンセプト

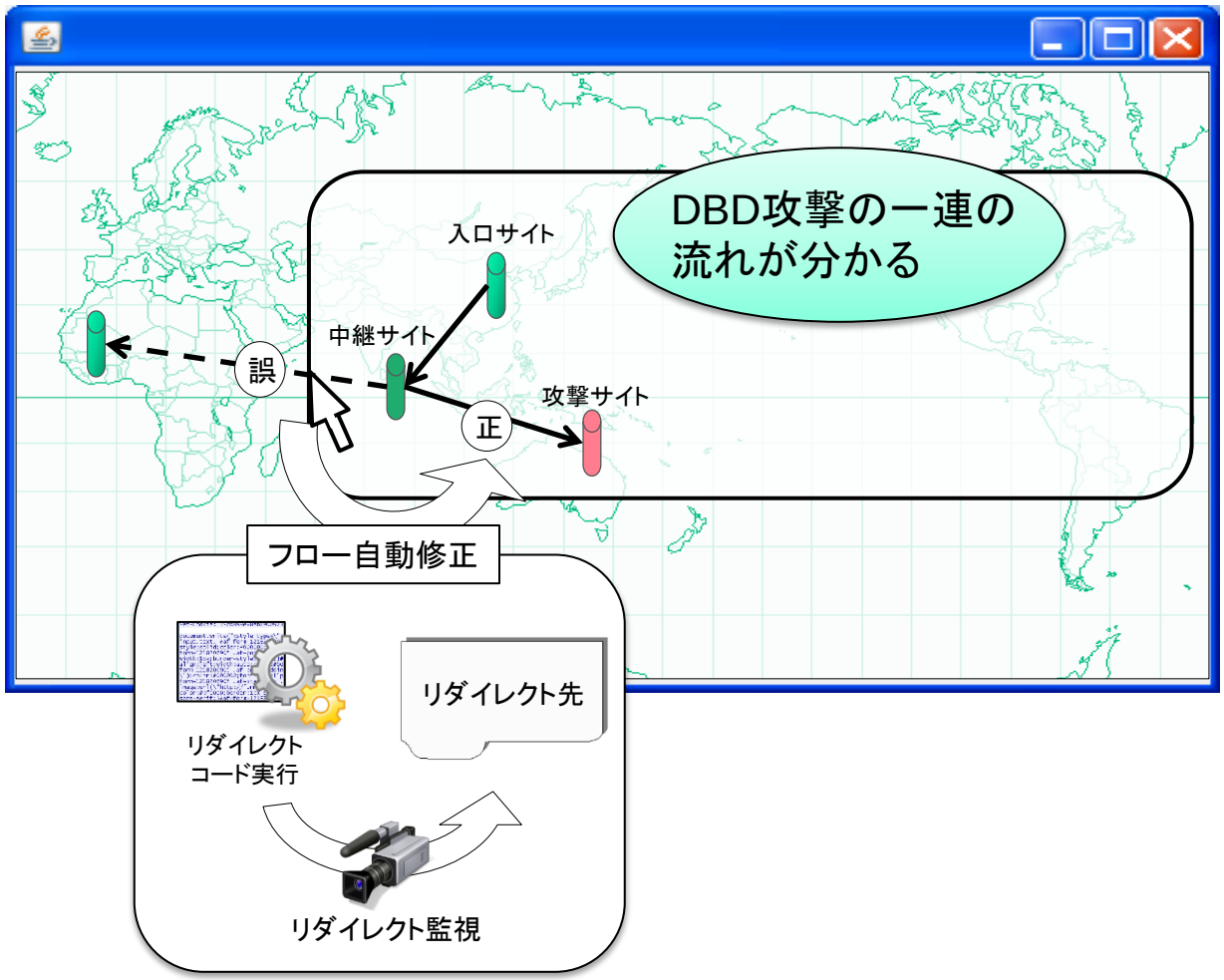


提案システムのコンセプト-リダイレクトフロー可視化と修正-




通信データ





通信データの可視化：正しい例

GETリクエスト同士の間隔が
小さければリダイレクトと判断

GETリクエストが
出てくる順に
フローを描画

No.	sourceIP	DestinationIP	Info
1	10.220.0.101	A	GET
2	10.220.0.101	B	GET
3	10.220.0.101	C	GET
4	10.220.0.101	D	GET
5	10.220.0.101	E	GET
6	10.220.0.101	F	GET

0.5s
0.5s
30s
1.0s
4.0s

GETリクエスト同士の間隔が大きいと
一連の通信の切れ目と判断する

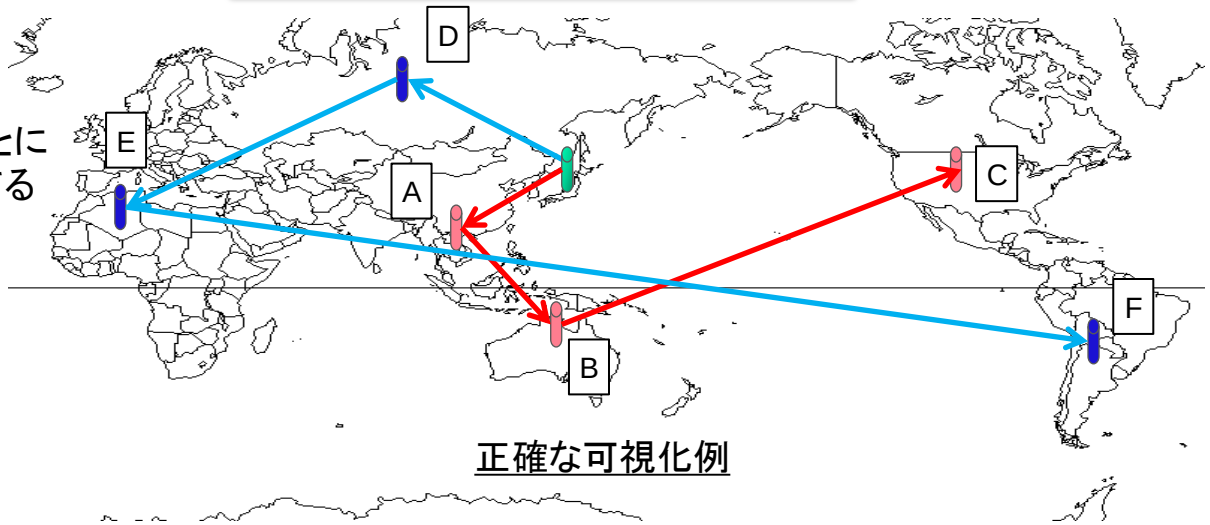
フロー1

No.	sourceIP	DestinationIP	Info
1	10.220.0.101	A	GET
2	10.220.0.101	B	GET
3	10.220.0.101	C	GET

フロー2

No.	sourceIP	DestinationIP	Info
4	10.220.0.101	D	GET
5	10.220.0.101	E	GET
6	10.220.0.101	F	GET

IPアドレスごとに
ポールを立てる



正確な可視化例

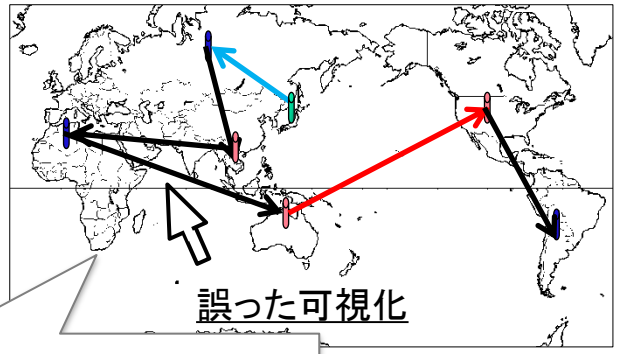
通信データの可視化：誤った例　と　修正方法

No.	SourceIP	DestinationIP	Info
1	10.220.0.101	A	GET
2	10.220.0.101	B	GET
3	10.220.0.101	C	GET
4	10.220.0.101	D	GET
5	10.220.0.101	E	GET
6	10.220.0.101	F	GET

0.5s
1.0s
0.5s
1.0s
3.0s

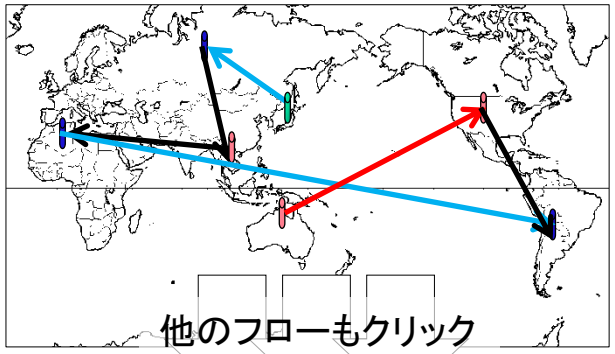
区別がつかないフローが混在
→　一つの通信フローだと認識

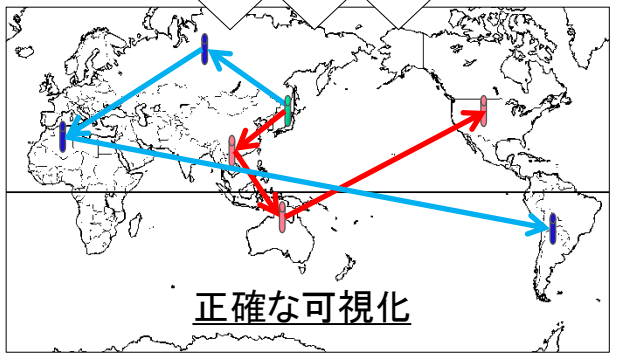
←　：誤って描画されたフロー



誤って描画されていると思うフローをクリック

自動修正



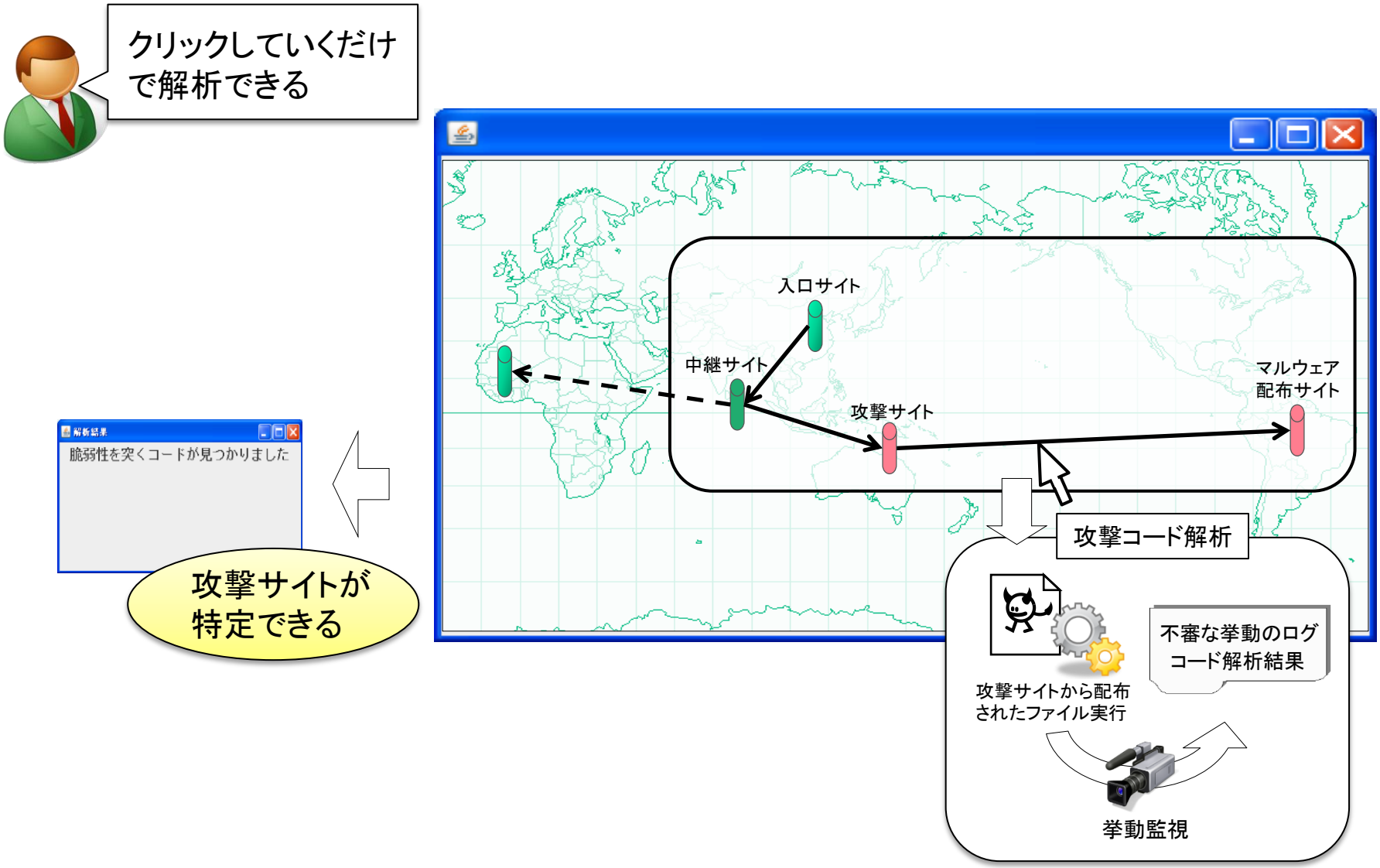


フローに含まれるリダイレクトコード

正しいリダイレクト先

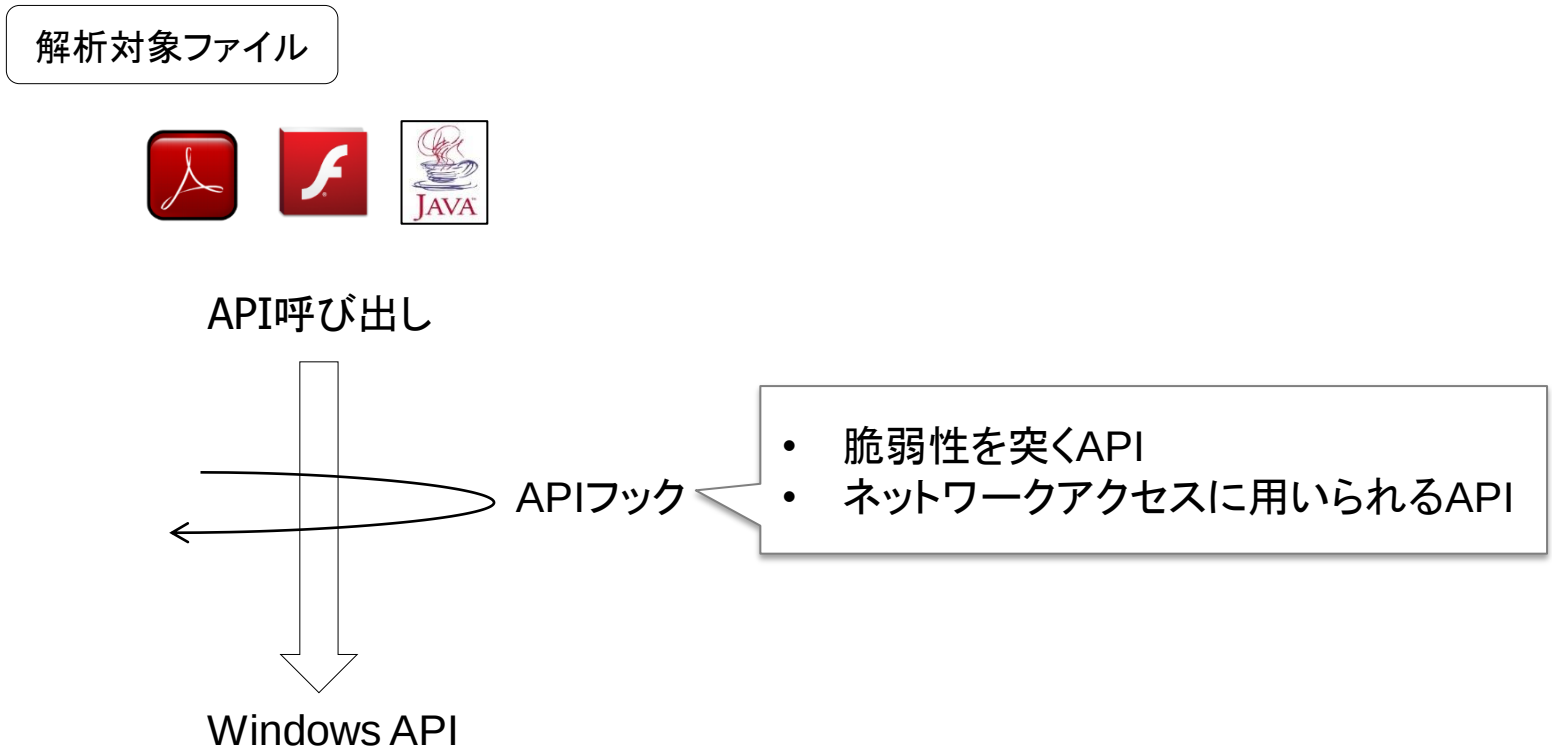
動的解析環境

提案システムのコンセプト -攻撃サイトの特定-

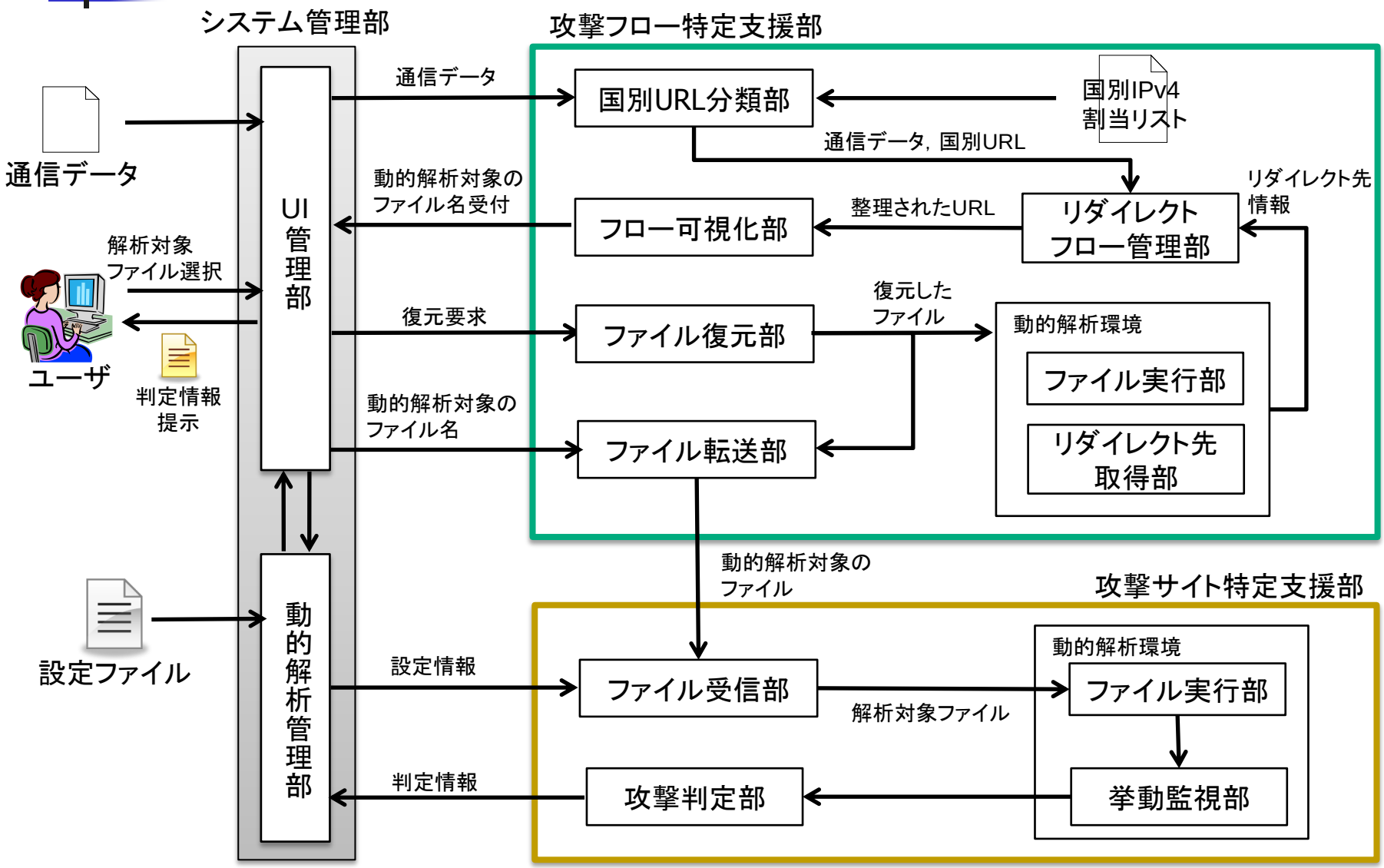


攻撃サイト，マルウェア配布サイトを特定

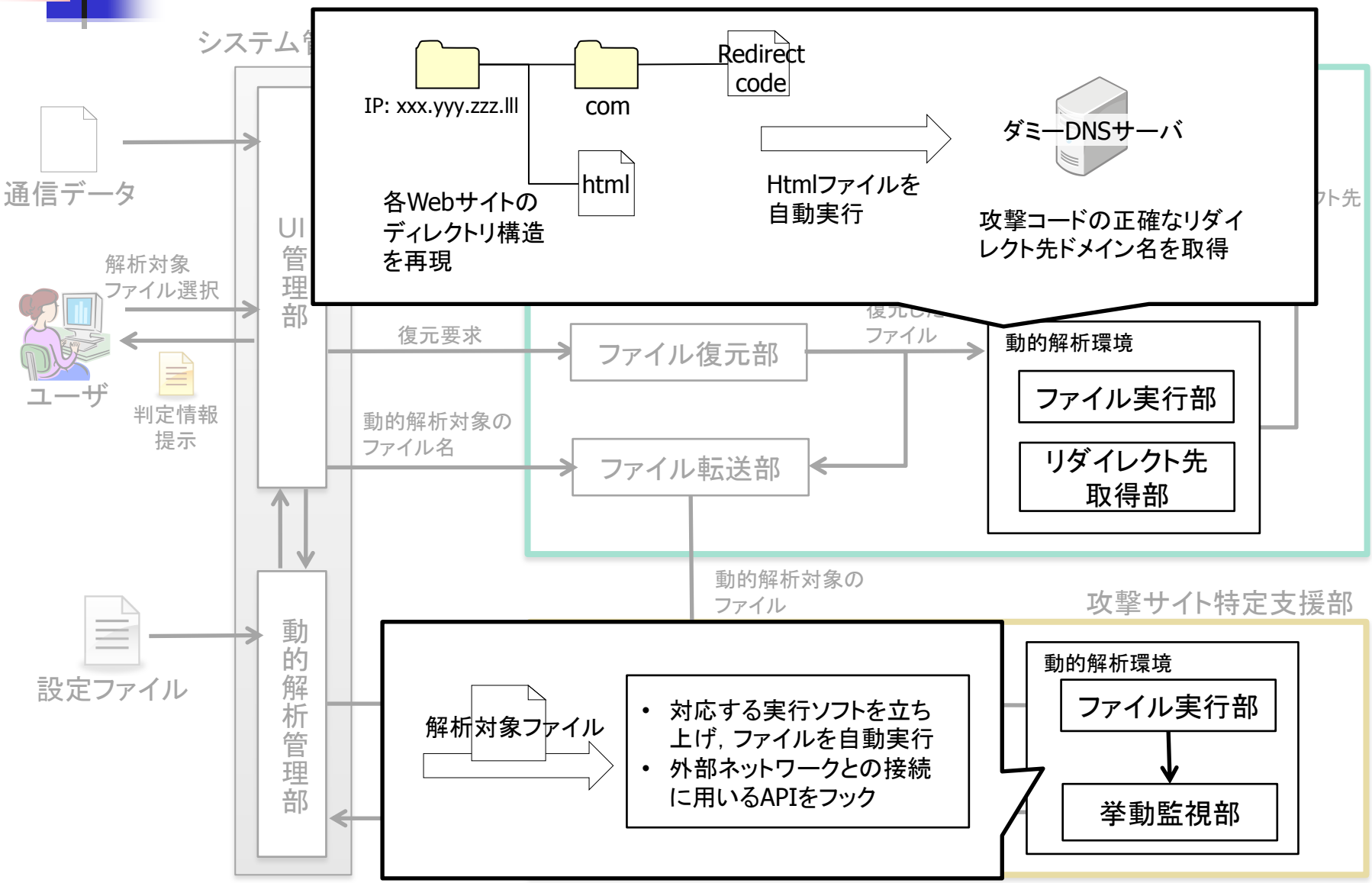
- 攻撃サイトとマルウェア配布サイトを繋ぐフローに含まれるファイルを解析する
  →攻撃の有無を検知する



提案システムの構成


システム管理部
攻撃フロー特定支援部
通信データ
UI管理部
通信データ
国別URL分類部
国別IPv4割当リスト
通信データ，国別URL
動的解析対象のファイル名受付
フロー可視化部
整理されたURL
リダイレクトフロー管理部
リダイレクト先情報
解析対象ファイル選択
ユーザ
判定情報提示
復元要求
ファイル復元部
復元したファイル
動的解析環境
ファイル実行部
リダイレクト先取得部
動的解析対象のファイル名
ファイル転送部
動的解析対象のファイル
攻撃サイト特定支援部
設定ファイル
動的解析管理部
設定情報
ファイル受信部
解析対象ファイル
動的解析環境
ファイル実行部
判定情報
攻撃判定部
挙動監視部

提案システムの構成


システム管
通信データ
UI管理部
解析対象ファイル選択
ユーザ
判定情報提示
設定ファイル
動的解析管理部
IP: xxx.yyy.zzz.lll
com
Redirect code
html
各Webサイトのディレクトリ構造を再現
Htmlファイルを自動実行
ダミーDNSサーバ
攻撃コードの正確なリダイレクト先ドメイン名を取得
復元要求
ファイル復元部
ファイル
動的解析環境
ファイル実行部
リダイレクト先取得部
動的解析対象のファイル名
ファイル転送部
動的解析対象のファイル
攻撃サイト特定支援部
解析対象ファイル
対応する実行ソフトを立ち上げ，ファイルを自動実行
外部ネットワークとの接続に用いるAPIをフック
動的解析環境
ファイル実行部
挙動監視部

評価用システムの実装

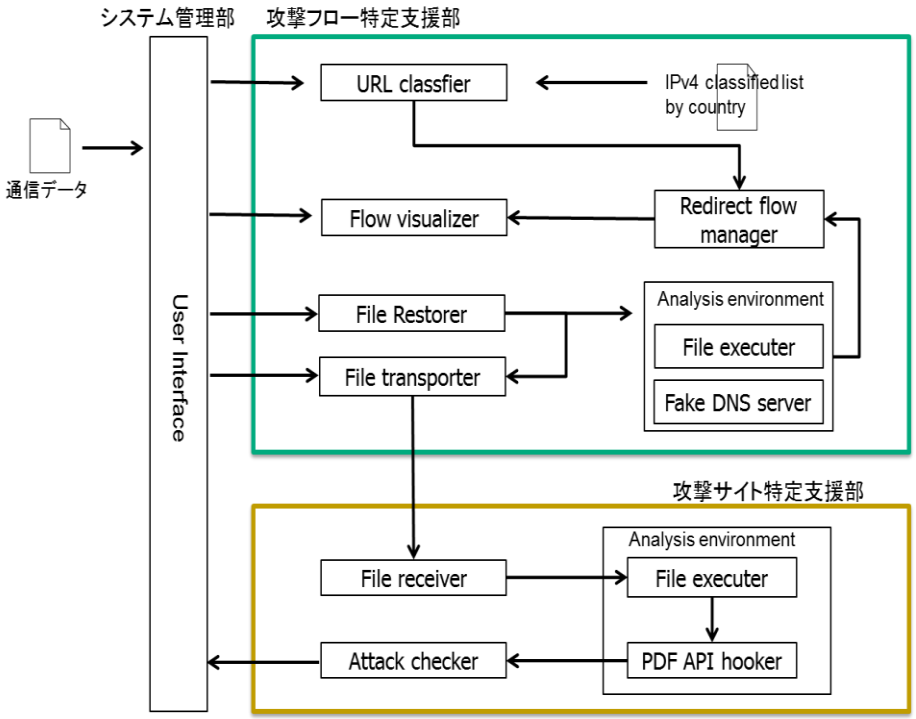
- 提案システムの主要な機能を有し，一連の流れが進むことを確認するためのシステムを実装した
- 動的解析環境にはD3M 2012を収集したハニーポットと同じブラウザ・プラグインを導入した
  - APIフックの試作としてAdobe ReaderのネットワークI/OをフックするPDF API Hookerを実装した
    - 外部と通信を行うPDFファイルを攻撃コードとみなす

動的解析環境

仮想環境	Vmware Workstation 8.0
OS	Windows XP SP2
ブラウザ	Internet Explorer 6.0
プラグイン	Adobe Reader，Flash Player，WinZip，QuickTime，JRE （全てセキュリティパッチ未適用）

実装環境

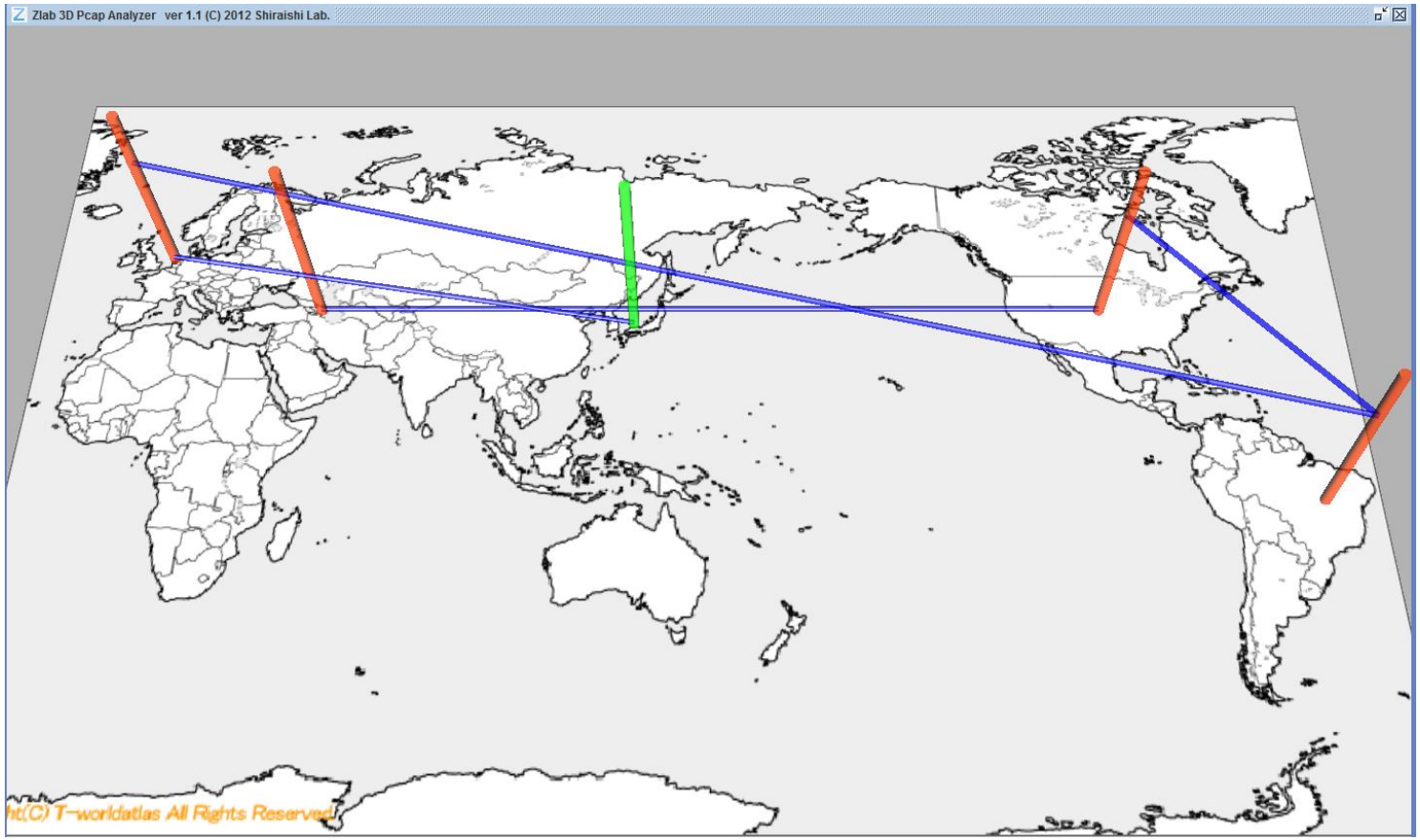
URL classfier，File transporter，File receiver，Attack checker，File executer	JDK 1.7.0_05
Flow Visualizer	Java 3D 1.5.1
File Restorer	jNetPcap 1.3.b4 （WinPcapのJavaのラッパー）
Fake DNS server	Python 2.7
PDF API Hooker	C++



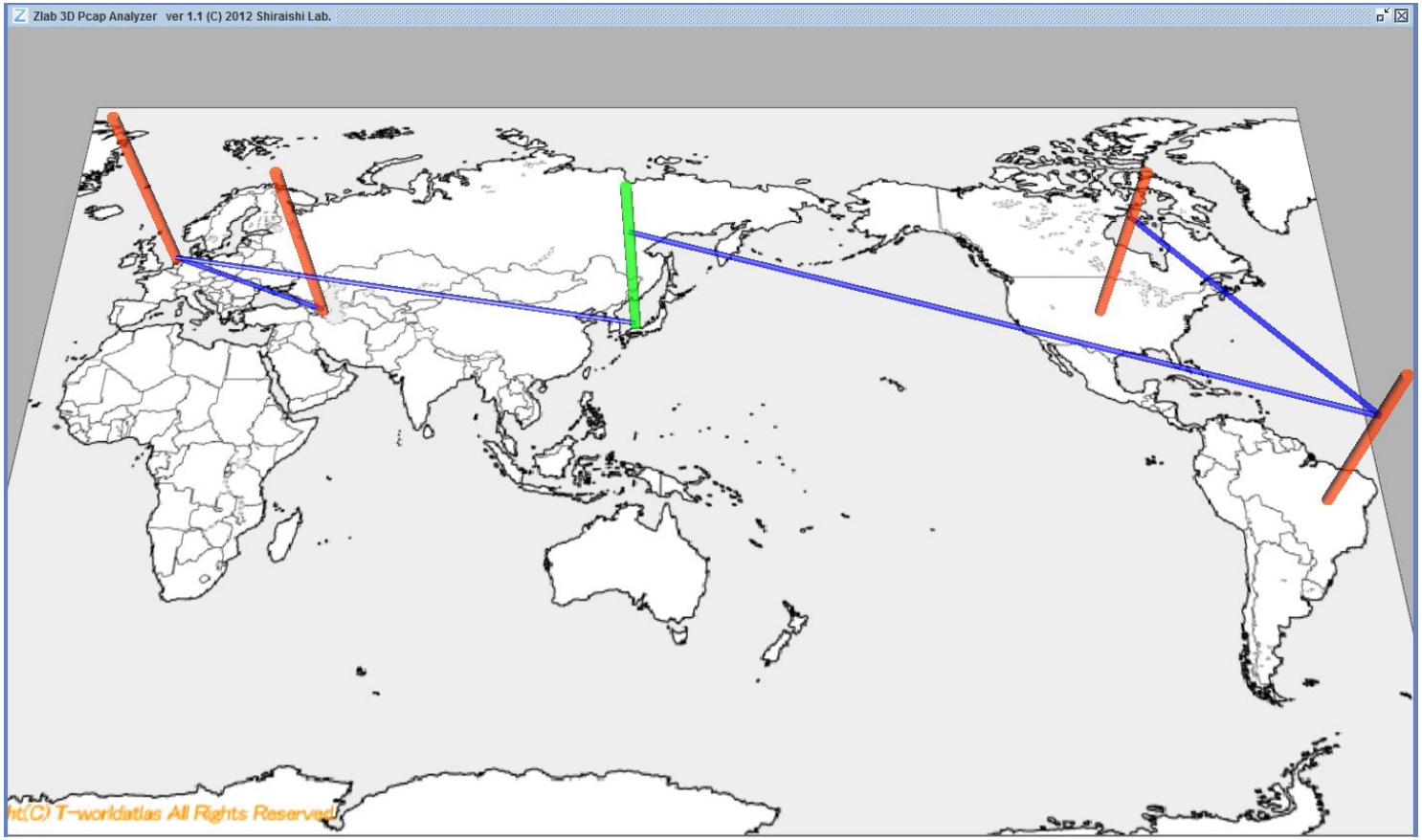
評価実験と結果

- 評価実験1
  - 目的: 提案システムの一連の流れを行えることを確認
  - 手段: 攻撃フローと正規通信フロー(正規サイトへのWebアクセス)を混合した通信データを用いて評価 ← 組織の通信データを想定
  - 事前準備:
    - 脆弱性を突く攻撃コードを含むPDFタイプのマルウェア3検体をD3M 2012(3/25付の通信データ)から抽出
      - 攻撃に利用したCVEナンバーを特定できたものに限定した
    - Wiresharkで正規通信フローを一つ取得
    - 一つの正規通信フローに一つの検体を含む通信フローを混合した通信を1セットとする ← 同じ2つのフローで混合させるタイミングを変えて3セット作成する
    - 同様に，別の検体を含む通信フローを利用して3セットずつ作成し計9セット作成
  - 実験手順:
    - 9個の評価用データに対して，通信データ可視化から解析結果提示までの一連の解析の操作を行い，その操作の正確性を確認
  - 評価結果 （通信データ9個中）
    - 9個 ・・・ 解析結果の提示までできた
- 評価実験2
  - 提案システムの一連の流れに要する時間を計測した
  - →解析結果の提示まで27秒を超えずに行えることを確認した

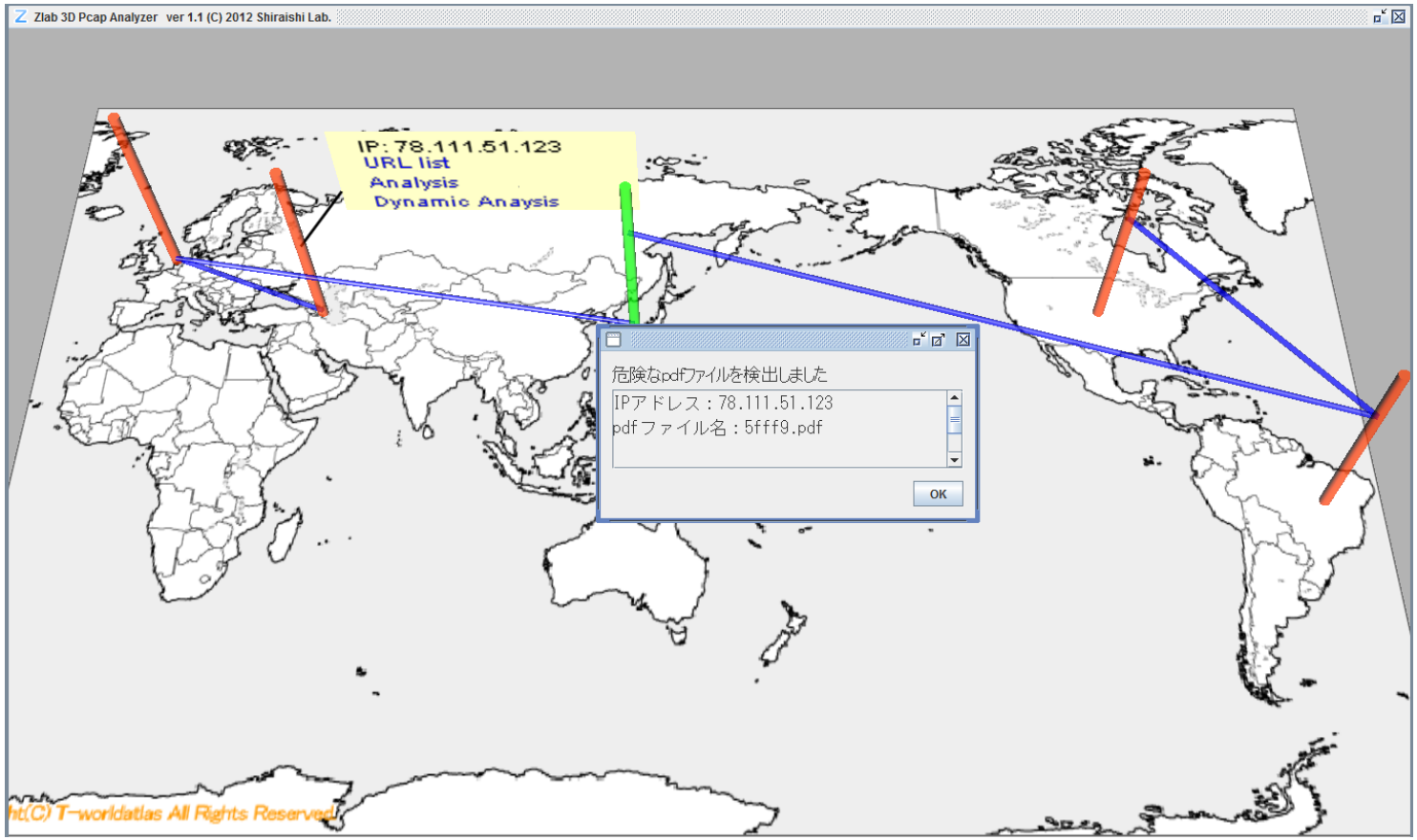
評価システムの動作確認

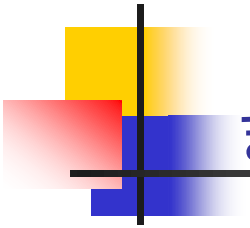


評価システムの動作確認



評価システムの動作確認





まとめ

- 目的
  - DBD（Drive-by-Download）攻撃に対して，各組織で迅速に対策を講じられるよう支援することで被害の拡大を抑える
- 課題
  - 人が通信データを解析し攻撃フローの全容を把握することを支援するシステムを実現する
- 本研究のコンセプト
  - 攻撃フローの可視化
  - 脆弱性を突くコードを配布するサイトを特定
- 提案システム
  - 通信データに含まれる通信フローを出現した時間順にビューワーで世界地図上に描画する
  - 誤って描画されていると思われるフローをクリックすると，配布されたファイルを自動で動的解析しフローを自動で修正する
  - 攻撃サイト，マルウェア配布サイトのフローをクリックすると，配布されたファイルを自動で動的解析し不正なコードの有無を確認する
- 評価実験
  - 検体を含む通信データ9個中9個の攻撃サイトを特定することができた
  - 一連の流れは27秒を超えずに行えることを確認した
- 今後の課題
  - 検体の数を増やし，より大きなデータサイズの通信データを用いた評価実験を行う