

Deception ネットワークを構成するフレームワークの提案

竹中 幹^{1,a)} 高野 祐輝^{1,b)} 宮地 充子^{1,c)}

概要: 標的型攻撃やゼロデイ攻撃などの防御が難しい攻撃に対して、攻撃相手を騙してネットワーク環境を守る deception ネットワークに関する研究が進んでいる。様々な deception の手法が研究されているが、要素技術の研究開発にとどまっておらず統合的に deception ネットワークを構築できる環境は整っていない。本研究では deception ネットワークを容易に構築できるフレームワークを提案する。本フレームワークを用いるとプロトコルの応答の偽装や仮想サーバの構築、仮想ネットワーク環境の管理などの機能を作成、統合して行えるようになる。

キーワード: ネットワークセキュリティ, deception ネットワーク

A Framework for Building Deception Networks

MOTOKI TAKENAKA^{1,a)} YUUKI TAKANO^{1,b)} ATSUKO MIYAJI^{1,c)}

Abstract: For targeted or zero-day attacks, which cannot be prevented easily, deception network technologies are studied. However, existing researches deal with only underlying technologies for deception network. In this paper, we propose a framework for building deception networks easily. By using this framework, deceptive replies of network protocols, constructing virtual servers, and management of virtual networks are performed integrately.

Keywords: network security, deception network

1. はじめに

1.1 研究背景

今日では様々なネットワーク攻撃が世界中で行われている。攻撃側と防御側で日々多様な技術が取り入れられ進化し続けているが、現実的には攻撃者の侵入を完全に防ぐことは難しい。しかし、従来までの防御手法では、攻撃者に1度攻撃対象のネットワークへ侵入を許すと、その後の被害拡大を防ぐことは難しい。また防御手法として従来より使用されている手段は発見された攻撃の特徴をもとに防御を行うため、防御が後手にまわってしまうのが現状である。

1.2 最近のサイバーセキュリティ技術動向

従来より利用されてきたネットワークの防御技術として、ファイアーウォール、IDS/IPS、(WAF) が挙げられる。ファイアーウォールは送信元、送信先の IP アドレスや接続先ポート番号から通信の可否を決める。IDS/IPS はパケット内部の情報から、ファイアーウォールでは理解できない既存の攻撃の特徴を持つ攻撃を検知 (IDS)、破棄 (IPS) する。WAF は WebApplication レベルでの攻撃 (SQL インジェクションなど) の攻撃を防ぐ。前述の防御手段はどれも境界防御であり、それぞれの境界を突破された場合、攻撃者は内部で自由に行動ができる。ネットワークの正当な利用者に自らマルウェアをインストールさせて攻撃を実行する標的型攻撃や、防御手段が配布されるまでに脆弱性について攻撃するゼロデイ攻撃などが境界を突破する攻撃である。そのような攻撃に対抗するために相手を騙してネットワーク環境を守る deception ネットワーク ([1], [7], [8], [9])

¹ 大阪大学
Osaka University

a) takenaka@cy2sec.comm.eng.osaka-u.ac.jp

b) ytakano@cy2sec.comm.eng.osaka-u.ac.jp

c) miyaji@comm.eng.osaka-u.ac.jp

の研究が進んでいる。deception ネットワークとは侵入されることを前提としたネットワークで、侵入された後に攻撃者に偽物のネットワーク情報を掴ませることで攻撃の緩和、検知を行うという従来の防御手段とは全く異なる新しい防御手段である。

1.3 本研究の目的

現在までに公表された様々な論文によって、日々新しい deception システムの技術が生まれている。しかし、生み出される技術はネットワークや deception に関する専門的な知識が非常に多く、その技術を必要とする企業などの一般ユーザが利用できない状態である。本研究では、専門的な知識を有していない一般ネットワークユーザでも deception ネットワークを統合的に利用できるフレームワークを作成することを目的とする。設置の対象はネットワーク内とする。ネットワーク内部に設置したノードにおいて、ネットワーク偽装を行うことができる関数を Ruby プログラムで実行すると、接続された各ノードに対してそれぞれ異なるネットワークの構造を提供する。

1.4 本論文の構成

本論文の構成は次のとおりである。2章では、deception システムの概要、および本論文の内容について考える際に参考にした既存研究について記載している。3章では本研究の設計原理について書きまとめる。4章では実際に提案するシステムの動作や仕組みについてまとめる。5章では、実際の本フレームワークの実装手段、使用方法について記載する。6章では、本フレームワークの評価について、定性的、定量的に行ったものを書きまとめる。7章では、本研究のまとめと今後の課題について述べる。

2. 既存研究

Open vSwitch [3] は OpenFlow スイッチの一つであり、流れてきたパケットを用意されたフローに従って処理する。またこのスイッチの特徴として、カーネルデータパスモジュールにキャッシュを持っているため、パケットに対して実行されたフローは一時データパスのキャッシュに保存される。OpenvSwitch はフローキャッシュを2層で持ち、第1層と第2層を状況に応じて使い分けることでパフォーマンスの向上に努めている。

ASyDS [2] は SDN の技術を用いて考えられた deception 技術である。ネットワークに接続されたホストごとに IP アドレスやネットワークの構成が全く異なる仮想ネットワーク環境を与え、ネットワーク内部の偵察の複雑化や多数ホストからの共同攻撃の防御、また攻撃検知の正確性と可能性の向上を目的として考えられたシステムである。OpenFlow スイッチ (以下スイッチ) に複数台のホストやサーバを接続し、それらの通信によって生成されるパ

ケットの流れをすべて OpenFlow コントローラ (以下コントローラ)、スイッチに用意したフローテーブル、パケットの複雑な加工や応答パケットの作成が行うことができる deception サーバを用いて管理する。またホストごとにそれぞれの仮想ネットワーク環境が与えられるため、ホスト接続時にそのホスト専用の仮想ネットワーク環境を作成、管理するための Deception View Generator (DVG)、Deception View Database (DVDB) が用意されている。実ネットワーク環境や他のホスト目線のネットワーク環境の情報は得ることができないようになっている。

Zhan Shu らは、高度標的型攻撃に対して deceptionFTP サーバを用意し、実物の FTP サーバに侵入してきた攻撃者を deceptionFTP サーバへ誘導して騙す技術を提案した [6]。攻撃能力の高い攻撃者は、自分の置かれているネットワーク環境が違和感があればすぐにおとりサーバであることを疑い、攻撃を中断するので容易に騙されることはない。この論文では攻撃者が侵入してからの行動で得ているネットワーク情報と、攻撃を検知してから誘導する deceptionFTP サーバで一貫性を保つ方法を紹介している。実物の FTP サーバで攻撃者に取得された情報をコマンドログとしてデータベースに保存し、情報の一貫性を保つためそのまま deceptionFTP サーバへ動的に反映して、そこへ攻撃者を誘導することで攻撃者に気づかれないように相手の目的の確認や行動の観察を行う。

3. 設計原理

以下に本研究の設計原理について説明する。現在に至るまで様々な deception システムが考えられ公表されているが、deception の要素技術開発にとどまった内容であり、実用化は行われていない現状となっている。そこで我々が行う設計のモチベーションとして掲げるものは、deception ネットワークを統合的に構築できるフレームワークを作成するという点である。

3.1 複雑な動作の統合

前述のとおり、現在公表されている deception システムは総じて要素技術の開発にとどまっているため、実用可能な環境が整っておらず、一般的に deception システムを利用することは非常に難しい。本研究の最終目標の一つは deception システムを構築する際に必要な動作を統合して、いくつかの単純な関数とその内部で利用する引数を入力するだけで十分利用可能な deception システムを構築できるフレームワークを作成することである。また、引数を用いることで deception システムのオンオフやテストモードの実現、また deception システムによって作られる deception ネットワークの規模の変更ができるようにする。

3.2 攻撃者を騙す仮想ネットワーク

最近利用されている SDN は、ネットワークを仮想化しソフトウェアでパケットを制御している。本フレームワークの内部構造として、SDN の仕組みを拡張しパケットのソフトウェア制御による書き換えのみでなくパケットの偽装応答や仮想サーバの構築、deception ネットワークの管理を行えるようにしたい。最終的には、攻撃に長けた者から見ても本物と区別が付かない deception ネットワークを構築することをもう一つの目標とする。本研究で設計されるフレームワークは容易に利用できることも目標の一つとするため、システム実行の環境が作成しやすいネットワーク内部に置かれたサーバで実行し、ネットワークに侵入した攻撃者の攻撃の横展開を防ぐ、または検知することを前提とする。deception ネットワークはまず攻撃者の IP アドレススキャンを騙す必要がある。攻撃者が個々のノードからネットワークの IP アドレススキャンを行ったときに存在しない IP アドレスへの要求に対して適切な応答を返すことができればその攻撃者には本来存在しない IP アドレスを持ったノードが見えることになり、すなわちそれが deception ネットワークの土台となる。次に攻撃者は存在の確認できた IP アドレスに対してポートスキャンを行う。deception ネットワークはポートスキャンも騙す必要がある。攻撃者によって存在が確認された、実際は存在しない IP アドレスに対してポートスキャンが行われたときに、通常のノードの応答と区別つかない適切な応答を返すことで攻撃の横展開の妨害をより大きく行うことができる。他にも正常な利用方法ではアクセスを行うことがない IP アドレス、ポートに対してアクセスがあった場合は攻撃者の行動であると認識できるため、ハニーポットの動的配置を行い、そこへアクセスを転送するようにして攻撃者の観察や妨害、侵入目的の把握を deception ネットワークとして行う必要がある。

本論文内では deception ネットワーク構築において最も基礎的な、存在しない IP アドレスへのアドレススキャンを騙すための ARP[4] の偽装応答と ICMP[5] の偽装応答の部分拡張性を持たせつつ簡潔に行えるフレームワークの実装までを行う。その他の設計については本研究の後の課題とする。

4. 設計

4.1 ARP 応答の偽装

一般的には同一ネットワーク内で使用されている IP アドレスを確認する際には、ping コマンドを様々な IP アドレス宛に使用して応答があるかどうかで判断する。ping コマンドの動作は、(1)ARP テーブルに IP アドレスの情報がない場合と (2)ARP テーブルに IP アドレスの情報がある場合で少し異なる。ARP 要求が送信されるのは前者のみなのでここでは前者についてのみ言及する。まず IP ア

ドレスに対応する MAC アドレスを取得しなければいけないので、ping コマンド使用者は ARP 要求をブロードキャストで送信する。その際に本システムは送信元の IP アドレスに対して偽の応答を返す IP アドレスを送信元 IP アドレスごとにランダムに定め、定められた IP アドレス宛に ARP 要求が来たときのみ ARP 応答を偽装して作成、送信する。このようにすることで ARP 送信元の ARP テーブルに偽の MAC アドレス情報を付加することができ、送信元に偽のネットワークを見せることができる。また、送信元 IP アドレスとそれに対応するランダムに作成された偽 IP アドレスデータはデータベースに deception ネットワークとして保持するため、以後同じ送信元から通信がある場合はデータベースをもとに応答をする。

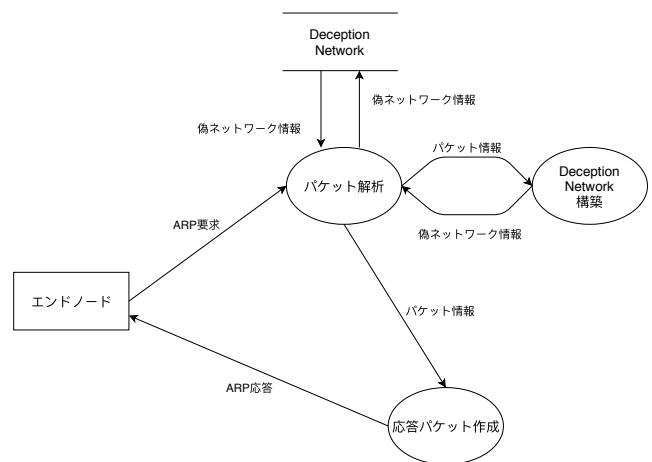


図 1 ARP 応答の偽装
 Fig. 1 Deception of ARP reply

1 は ARP 応答を偽装するための手順を表している。以下ではそれぞれのデータ処理部について説明する。パケット解析部ではエンドノードからブロードキャストで送られてきた ARP 応答パケットを解析する。まず受信したパケットの送信元の IP を Deception Network データベース (以下データベース) へ転送し、データベースに登録されているか確認する。登録されていなければ Deception Network 構築部へ送信元 IP アドレスを送信する。Deception Network 構築部では送られてきた送信元 IP アドレスに対して、パケットの応答を行う IP アドレスをランダムに決定し偽ネットワーク情報としてパケット解析部に送信する。パケット解析部はデータベースに偽のネットワーク情報を送信し保存する。次にパケット解析部はエンドノードからの受信パケットの宛先 IP アドレスを確認し、データベースの偽ネットワーク情報にその IP アドレスが登録されていれば、送信元 IP アドレス、送信元 MAC アドレス、送信元 IP アドレスを応答パケット作成部へ渡す。最後に応答パケット作成部は得た情報から ARP 応答パケットを作成し、エンドユーザへ応答する。図 2 は ARP 応答の偽装のシーケンス図である。

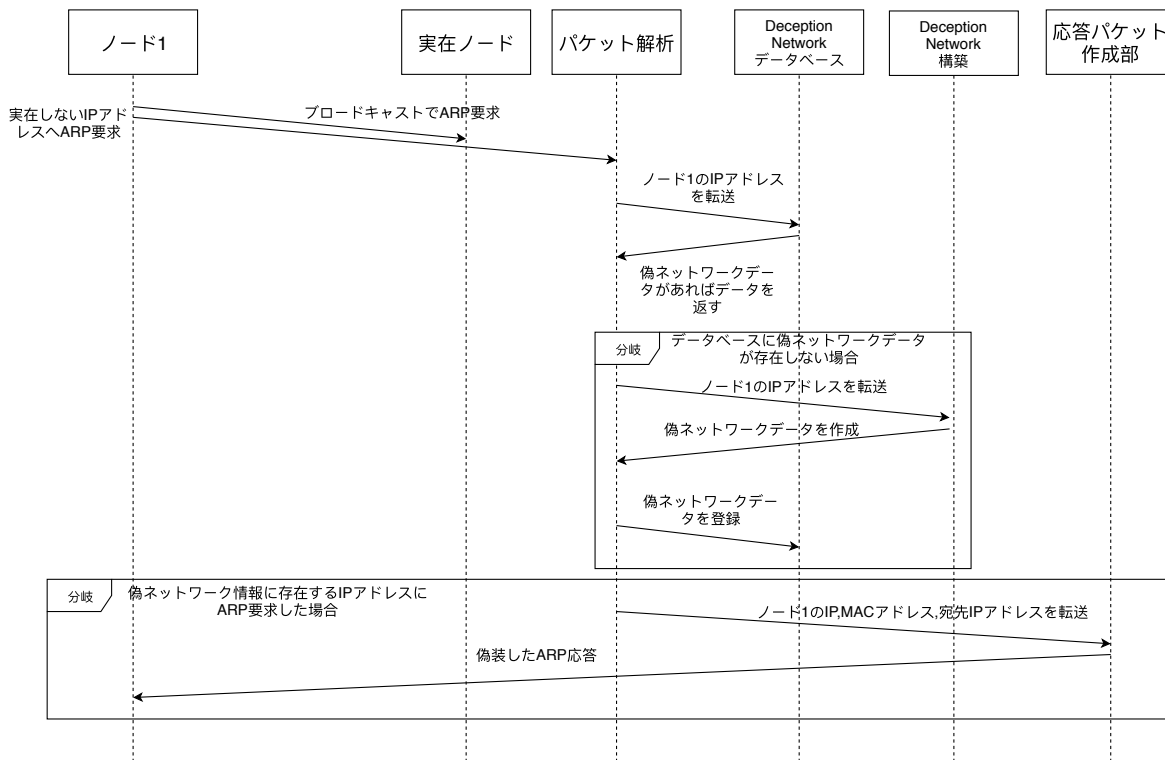


図 2 ARP 偽装応答のシーケンス図

Fig. 2 Sequence diagram of deception of ARP reply

4.2 ICMP 応答の偽装

ICMP を用いた ping コマンドを使用すると、ARP 応答があった IP アドレス、もしくは ARP テーブルに登録されている IP アドレス宛に ICMP 要求を送信する。本システムでは ARP 要求に応答する際にデータベースに偽ネットワーク情報を保持しているため、送信元が偽ネットワークデータに対応する偽 IP アドレスに ICMP 要求を送信した場合 ICMP 応答を偽装して作成、送信する。このようにすることで、攻撃者には偽 IP アドレスを持つホストがネットワーク内に本当に存在するかのように見せることができる(図 3)。

図 4 は ICMP 応答を偽装するための手順を表している。以下ではそれぞれのデータ処理部について説明する。ARP 応答偽装と基本的には同じで、まずパケット解析部では送信元 IP アドレスを取得する。ここで ICMP 要求が届いているとき、必ずデータベースには送信元 IP アドレスの偽ネットワーク情報が存在するため新規に偽ネットワーク情報を作る必要がない。受信パケットの宛先 IP アドレスを確認し、データベースの偽ネットワーク情報にその IP アドレスが登録されていれば、送信元 IP アドレス、送信元 MAC アドレス、宛先 IP アドレス、宛先 MAC アドレス、ICMP パケットのデータ部を応答パケット作成部に渡す。最後に応答パケット作成部は得た情報から ICMP 応答パケットを作成し、エンドユーザへ応答する。

5. 実装

5.1 内部構成

まず使用する言語についてだが、パケット解析部、偽装パケットの作成部では生のネットワークパケットを扱う必要があるため、raw socket がプログラム内で利用できる C++ を用いて実装した。前述の設計ではパケット解析部、偽装パケット作成部、Deception ネットワークデータベースは役割を分けて設計したが、実装ではプロセスとデータベースを同一のプログラム内で実装した。なお、データベースは C++ 上で map を用いて管理している。プログラムの内部構成は、まず raw socket をプロミスキャスモードで使用しすべてのパケットを生の状態で得ていく。得たそれぞれのパケットに対してパケットのタイプに応じて処理を行う。以下に処理方法を示す。

ARP パケット ARP パケットが届いた場合は、まず map に送信元の IP アドレス情報が保管されているか確認する。保管されていなければ送信元 IP アドレスを後述の deception ネットワーク構築を行う Ruby プログラムへ送り、ネットワーク情報を得て map へ保存する。次に map のネットワーク情報と宛先 IP アドレスを比較し、ネットワーク情報の中に宛先 IP アドレスが存在すれば応答パケットを作成する関数へ送信元、宛先 IP アドレスと送信元 MAC アドレスを渡す。最後にパケット作成の関数で応

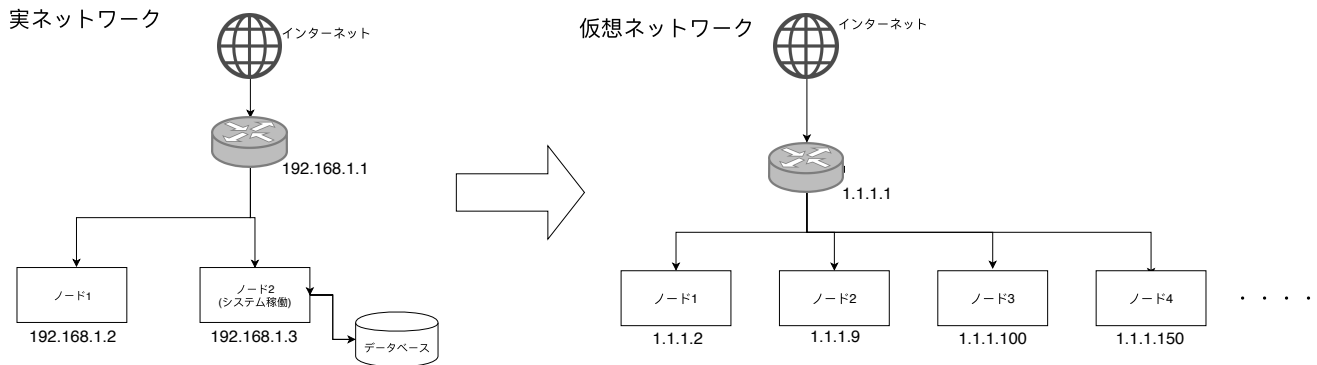


図 3 偽装した仮想ネットワーク
 Fig. 3 Deceptive virtual network

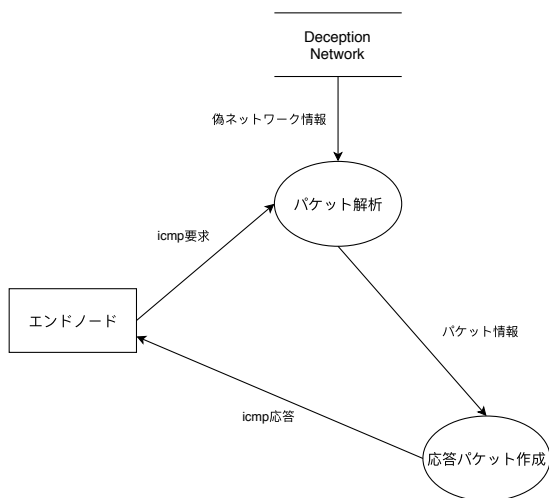


図 4 ICMP 応答の偽装
 Fig. 4 Deception of ICMP reply

ソースコード 1 実装した関数

```
1 make_deception_network(interface,
2                         *virtual_node_num)
```

ソースコード 2 使用例: 偽装ノード数を 50 に指定

```
1 require './icmp_decep'
2
3 make_deception_network("eth1", 50)
```

答パケットを作り応答を行う。

ICMP パケット ICMP パケットが届いた場合は、事前に ARP 応答を行っていることがわかっているので map には送信元 IP アドレスが登録されていることを前提として処理を行う。まず map のネットワーク情報と ICMP パケットの宛先 IP アドレスを比較し、ネットワーク情報の中に宛先 IP アドレスが存在すれば応答パケットを作成する関数へ送信元、宛先の IP、MAC アドレスを渡す。最後に

パケットの作成の関数で応答パケットを作り応答を行う。
 次に deception ネットワーク作成部について、このプロセスは Ruby を用いて実装した。本フレームワークの内部構造として、このプロセスを軸として関数の形で用意し、この Ruby の関数を実行することで C++ で書いた前述のプログラムを常駐で実行するようにする。これによって Ruby の簡潔な関数のみを用いることで複雑なネットワーク偽装を実現できる。以下に Ruby で利用できる、本研究で実装した関数の動作について説明する。

ソースコード 1 は実行すると本研究で実装した ARP、ICMP の応答偽装を行う。引数 interface はパケットの監視を行うネットワークに接続されたインターフェース名を代入する。可変長引数 (*virtual_node_num) はリストの引数で、偽装したい IP アドレスの数、すなわち各ノードに見せたい偽装ノードの数を代入する。この関数において、もし引数を取らなければ、もしくは引数を "random" とすれば偽装ノードの数をランダムで生成する。この関数の動作として、まず Open3.popen3 を用いて前述の C++ プログラムを実行しその標準入力、標準出力、標準エラー出力と接続されたパイプを持つ。C++ のプログラムから deception ネットワークの作成用の IP アドレスが接続したパイプの標準出力から届けば virtual_node_num[0] の値分のランダムな IP アドレスをリストに収めリストを deception ネットワーク情報とし、それを文字列化して接続したパイプに標準入力として出て C++ のプログラムへ渡す。また、動作を視覚化するために Ruby 側では IP アドレスの文字列、すなわち deception ネットワーク情報を作成した時にコンソールへ出力し、また C++ 側では受け取ったパケットのタイプを標準エラー出力で Ruby のプログラムへ渡し、Ruby プログラム上でコンソールへ出力するようにしている。なお、C++ からの標準エラー出力と標準出力でどちらから受け取っても適切に処理するために IO.select を使用して Ruby 側で受け取る情報を区別している。

5.2 関数の利用

前述の関数の実際の利用について紹介する。本実験ではどのユーザでも容易に deception ネットワークの構築を行えることが目的であるため、利用は非常に簡潔である。ソースコード 2 に使用例を示す。このように引数の値を変更するだけで deception ネットワークの規模を変更することができる。

6. 評価

6.1 定性的評価

本研究の強みとして最も大きい部分はネットワーク一般ユーザが統合的に本フレームワークを利用できるということである。今までで公表されている研究内容は専門的知識を持っていること前提で、一般向けで利用するにはかなり複雑な設計となっていた。また、今までに公表されている研究内容は専門的かつ複雑ということもあり、設計段階のものが多く、実際の実装後の使用例が取り上げられていることがあまりなかった。一方本研究では実装まで行い実際の使用例を挙げ、結果を出せているという点も強みである。

6.2 機能評価

以下、5章で記載した使用例について実行結果を示す。まず、システムを実行したネットワーク環境について説明する。

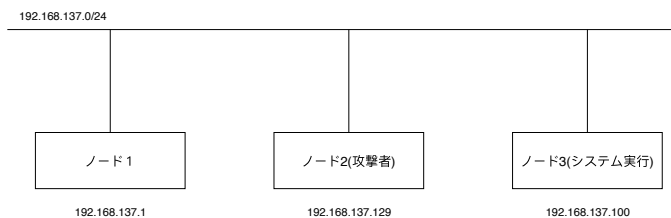


図 5 実行ネットワーク環境
Fig. 5 Used network environment

図 5 内、ノード 3 において本システムを実行した。また、攻撃者側であるノード 2 においては `fping` コマンドを用いたネットワーク内部偵察を仮定して、次のコマンドを実行した。

ソースコード 3 攻撃者ノードにおける実行コマンド

```
1 fping -g 192.168.137.0/24 2>/dev/null | grep -v unreachable
```

また、ソースコード 3 記載のコマンド実行結果を本システム非利用、5章に記載した使用例の利用時に分けてソースコード 4, 5 へ記載する。

deception ネットワークを展開していない環境では、ソー

ソースコード 4 非 deception ネットワークにおける ping 応答

```
1 192.168.137.1 is alive
2 192.168.137.100 is alive
3 192.168.137.129 is alive
```

スコード 4 のように実存する IP アドレスのみが応答しているが、偽装ノード数を 50 に指定した deception ネットワーク環境ではソースコード 5 のように実存する IP アドレスに加えて、存在しない IP アドレスが 50 個存在しているように見えることがわかる。このように関数利用時に指定した数によってユーザ自らが deception ネットワークの規模が調節できるということが本研究の強みの一つである。

7. まとめ

本論文では Ruby を用いて利用できる統合的な deception ネットワークシステムのフレームワークの設計と実装について記載した。5章のとおり、ARP 応答の偽装と ICMP 応答の偽装を統合し、deception ネットワークを専門的な知識や複雑な手順なしで利用できるフレームワークの作成が行えた。このフレームワークを用いれば内部へ侵入してきた攻撃者が攻撃の横展開をする際の IP アドレススキャンを攪乱することが可能となった。

今後の課題として偽装パケットの応答速度が通常の応答とどれ程の差異があるかについての評価、改善が挙げられる。また実装面ではデータベースの作成とデータベースの照会、削除を行えるようにすること、3章でも記載したようにポートスキャンに対する開放ポートの偽装、攻撃者を誘導するハニーポットの動的配置のためのフレームワークを実装すること、が挙げられる。本研究は要素研究段階である deception システムを実用化へ助長するものであるため、意義のある研究であると考えられる。

謝辞

本研究の一部は JSPS 科研費基盤 C (JP15K00183), Microsoft Research Asia の共同研究費, 科学技術振興機構 (JST) の CREST(JPMJCR1404) と国際科学技術協力基盤整備事業 (日本-台湾研究交流), 及び文部科学省の情報技術人材育成のための実践教育ネットワーク形成事業 分野・地域を越えた実践的情報教育協働ネットワークさらに文部科学省の平成 30 年度「Society 5.0 実現化研究拠点支援事業」の助成を受けています。

参考文献

- [1] Stefan Achleitner, Thomas F. La Porta, Patrick D. McDaniel, Shridatt Sugrim, Srikanth V. Krishnamurthy, and Ritu Chadha. Deceiving network reconnaissance using sdn-based virtual topologies. *IEEE Trans. Network and Service Management*, 14(4):1098-1112, 2017.

ソースコード 5 使用例: 偽装ノード数 50 における ping 応答

```
1 192.168.137.1 is alive
2 192.168.137.5 is alive
3 192.168.137.12 is alive
4 192.168.137.19 is alive
5 192.168.137.23 is alive
6 192.168.137.24 is alive
7 192.168.137.27 is alive
8 192.168.137.35 is alive
9 192.168.137.40 is alive
10 192.168.137.44 is alive
11 192.168.137.46 is alive
12 192.168.137.48 is alive
13 192.168.137.55 is alive
14 192.168.137.57 is alive
15 192.168.137.69 is alive
16 192.168.137.71 is alive
17 192.168.137.72 is alive
18 192.168.137.81 is alive
19 192.168.137.86 is alive
20 192.168.137.91 is alive
21 192.168.137.97 is alive
22 192.168.137.100 is alive
23 192.168.137.110 is alive
24 192.168.137.113 is alive
25 192.168.137.120 is alive
26 192.168.137.129 is alive
27 192.168.137.131 is alive
28 192.168.137.134 is alive
29 192.168.137.135 is alive
30 192.168.137.142 is alive
31 192.168.137.146 is alive
32 192.168.137.150 is alive
33 192.168.137.160 is alive
34 192.168.137.166 is alive
35 192.168.137.167 is alive
36 192.168.137.169 is alive
37 192.168.137.188 is alive
38 192.168.137.193 is alive
39 192.168.137.194 is alive
40 192.168.137.195 is alive
41 192.168.137.197 is alive
42 192.168.137.200 is alive
43 192.168.137.205 is alive
44 192.168.137.226 is alive
45 192.168.137.227 is alive
46 192.168.137.235 is alive
47 192.168.137.236 is alive
48 192.168.137.239 is alive
49 192.168.137.240 is alive
50 192.168.137.241 is alive
51 192.168.137.243 is alive
52 192.168.137.248 is alive
53 192.168.137.254 is alive
```

- [2] Cho-Yu Jason Chiang, Yitzchak M. Gottlieb, Shridatt James Sugrim, Ritu Chadha, Constantin Serban, Alexander Poylisher, Lisa M. Marvel, and Jonathan Santos. Acyds: An adaptive cyber deception system. In Jerry Brand, Matthew C. Valenti, Akinwale Akinpelu, Bharat T. Doshi, and Bonnie L. Gorsic, editors, *2016 IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, November 1-3, 2016*, pages 800–805. IEEE, 2016.
- [3] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan J. Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, Keith Amidon, and Martín Casado. The design and implementation of open vswitch. *login.*, 40(2), 2015.
- [4] D. Plummer. An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, IETF, November 1982.
- [5] J. Postel. Internet Control Message Protocol. RFC 792, IETF, September 1981.
- [6] Zhan Shu and Guanhua Yan. Ensuring deception consistency for FTP services hardened against advanced persistent threats. In Massimiliano Albanese and Dijiang Huang, editors, *Proceedings of the 5th ACM Workshop on Moving Target Defense, CCS 2018, Toronto, ON, Canada, October 15, 2018*, pages 69–79. ACM, 2018.
- [7] Shridatt Sugrim, Sridhar Venkatesan, Jason A. Youzwak, Cho-Yu Jason Chiang, Ritu Chadha, Massimiliano Albanese, and Hasan Cam. Measuring the effectiveness of network deception. In *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018, Miami, FL, USA, November 9-11, 2018*, pages 142–147. IEEE, 2018.
- [8] Vincent E. Urias, William M. S. Stout, and Caleb Loverro. Computer network deception as a moving target defense. In *International Carnahan Conference on Security Technology, ICCST 2015, Taipei, Taiwan, September 21-24, 2015*, pages 1–6. IEEE, 2015.
- [9] Todd Vollmer and Milos Manic. Cyber-physical system security with deceptive virtual hosts for industrial control networks. *IEEE Trans. Industrial Informatics*, 10(2):1337–1347, 2014.