明治大学総合数理学部

2021 年度

卒　業　研　究

Residential IP Proxy サービスを悪用した不正行為の調査

学位請求者　先端メディアサイエンス学科

住友 孝彰

目次

第 1 章

はじめに

近年，住宅用の IP アドレスで動作する PC やスマートフォンをプロキシとして利用する Residential IP Proxy(以下 RESIP とする) サービスの市場規模が拡大している．本来は検閲によりネットワークの利用が制限されているユーザが，アクセス制限を回避して自由なインターネットサービスを利用するために提供されているが，2017 年に Mi らによって悪用されていることが報告されている [1]．匿名通信路 Tor と同様に，アクセスされたサーバ側からブラウザ側は秘匿されているので，攻撃に悪用されている可能性がある．Mi らの 2017 年を契機として，2019 年には半澤らが国内における状況を調査している [2].

そこで，本研究では RESIP サービスの不正利用の最新状況を明らかにすることを目的とする．代表的なサービスである ProxyRack[4] と Bright Data[5] の proxy IP アドレスを収集し，分析基盤 NONSTOP[3] を介して利用できる情報通信研究機構が保持する非対話型ハニーポッドで得られた情報と脅威情報分析プラットフォーム VirusTotal などを用いて，2 つのサービスの不正行為の頻度，種類，および，不正な proxy の分布を報告する．

第 2 章

準備

2.1 Residential IP Proxy

RESIP サービスは住宅用 IP をプロキシとして提供する proxy サービスである．サービスの利用者はプロバイダが提供している IP アドレスをプロキシとして利用し，任意のサイトにアクセスを行う．本調査対象である Bright Data[5] は，IP アドレスを提供することでメリットを得られる旨を主張している．

2.2 データセット

本研究で使用したデータセットについて説明する．

NICTER Darknet データセットは情報通信研究機構が保有するダークネットで観測された不正通信の情報である．同機構が提供する分析基盤 NONSTOP[3] から，パケットの到着時間，送信元，送信先の IP アドレス，ポート番号などを得る．

GeoLite2City データベースは MaxMind 社が提供している無償のデータベースである．IP アドレスから国や緯度、経度などの情報を得る．

VirusTotal は Google 社が運営する脅威情報分析プラットフォームである．WEB サイト上でマルウェアやファイルのスキャン，URL や IP アドレスの悪性判定を提供している．API を用いて IP アドレスの悪性判定を行うこともできる．

Shodan はインターネットに接続されたデバイス情報の検索エンジンである．インターネットをクロールし，解放されているポート番号，位置情報，応答から得られるバナー情報を収集し，データベース化して提供している．web ページからの利用，もしくは API が提供されている．

それぞれの概要と本研究での利用目的について表 2.1 にまとめる．

表 2.1: 使用したデータセット・データベースの概要と使用目的

データセット データベース	内容	使用目的
NICTER Darknet[6]	情報通信研究機構が保有する ダークネットで観測されたパケットの情報	Proxy の IP アドレスから国内のダークネットに ポートスキャンが行われているか調査する
VirusTotal[7]	Google 社が提供する 脅威情報分析プラットフォーム	Proxy の IP アドレスの悪性判定
shodan[8]	インターネットに接続された デバイスの情報を検索する検索エンジン	Proxy の IP アドレスで 解放されているポートの調査
GeoLite2City[9]	MaxMind 社提供している Geo location データベース	Proxy の IP アドレスの地理情報の調査

2.3 先行研究

2017 年に Mi らは RESIP サービスで提供される IP アドレスを収集し，RESIP サービスのサービスの基盤，規模を明らかにした [1].

2021 年に半澤らは，Mi らの研究 [1] で収集された Residential IP Proxy as a Service データセットに含まれる RESIP ホストの IP アドレスから日本のネットワークに不正な通信が継続的に到達していると結論づけた [2]．また，ホストの詳細な所在，ISP，RESIP の不正利用の頻度，RESIP ホストになっているデバイスベンダ等を明らかにした．

第 3 章

RESIP サービスを利用した内部からの RESIP ホストの調査

3.1 実験目的

本研究の実験目的は以下の 3 つである．

1. 2 つのサービスのホストの差を明らかにする．
2. Proxy 経由の通信が国内のダークネットに到達しているか調査する．
3. 悪性利用のポートと用途，国名，時系列の変化を明らかにする．

3.2 実験方法

図 3.1 に IP アドレスの収集方法の概要，図 3.2 に IP アドレスの分析方法の概要を示す．クライアント PC から RESIP サービスのプロキシを経由して研究室のサーバ (windy.mind.meiji.ac.jp) にアクセスすることで RESIP ホストの IP アドレスを収集する．アクセスの際に User-Agent 属性に特徴的な文字列を挿入することにより通常の WEB アクセスと実験によるアクセスを判別する．1，NICTER Darknet データセット，2，VirusTotal，3，Shodan，4，GeoLite2City を用いて収集したホストの IP アドレスについて調査する．

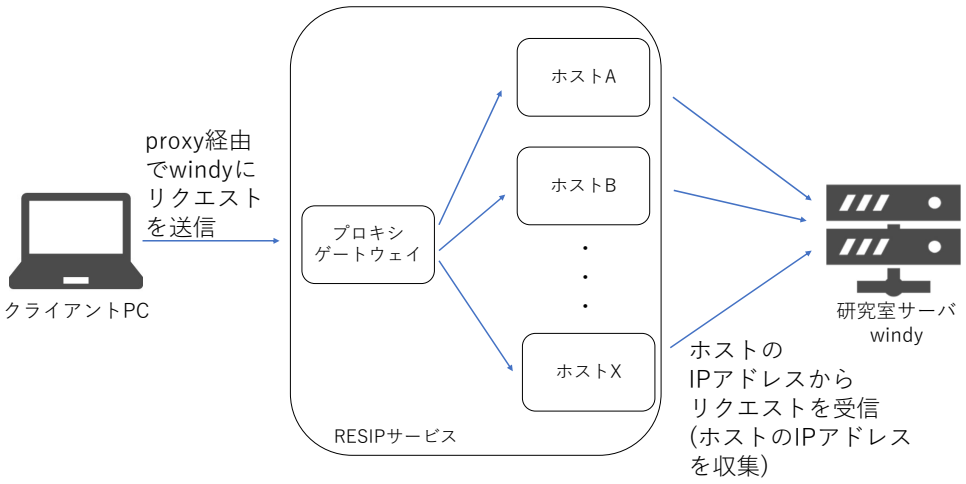


図 3.1: IP アドレス収集方法の概要図

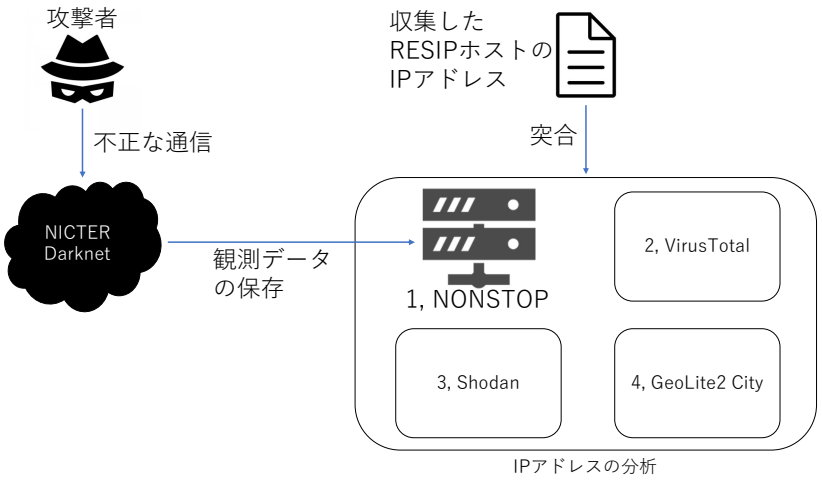


図 3.2: 収集した IP アドレスの分析方法の概要図

3.3 実験結果

ダークネットは未使用のアドレスブックなので，ここへの通信路要求は全てポートスキャン等の不正目的であると考えられるので，国内のダークネットに不正な通信を行った IP アドレスを不正 IP アドレス，国内のダークネットで観測されたパケットを不正パケットと呼ぶ．また VirusTotal で 1 つでも悪性と判定されたものを悪性と呼ぶ．

表 3.1 に収集期間，収集した IP アドレスの総数，不正 IP アドレスの個数，不正 IP アドレスの中の悪性 IP アドレスの個数，不正パケットの総数，悪性 IP アドレスからの不正パケット数を示す．観測期間で取得されたユニークな IP アドレス数を約 7 万個で共通にした時，ダークネットにスキャンした不正 IP アドレスの個数，不正 IP アドレスの内悪性と判定された個数は，ProxyRack の方が多い．ProxyRack で 44 個，Bright Data で 2 個の IP アドレスは，期間中毎日観測されるアクティブなアドレスであった．

表 3.1: 収集したデータの概要

	サービス	ProxyRack		Bright Data	
	期間	2021/7/22-8/7, 8/16-23 計:25 日		2021/9/30-10/13, 10/15-23, 11/19-21 計:26 日	
IP アドレス	IP アドレスの総数	69,369		70,253	
	不正 IP アドレス数	3,092	4.5%	1,545	2.2%
	同アドレス中 悪性 IP アドレス数	1,087	35.2%	307	19.9%
パケット	不正パケット数	526,674		32,724	
	悪性 IP アドレスから 到達したパケット数	252,454	47.9%	15,379	47.0%

図 3.3 に NICTER Darknet で観測された RESIP ホストの数の変化を示す．図 3.4 に 1 ホスト当たりの 1 日の平均パケット数を示す．*_mal は不正な通信を行った IP アドレスの内，悪性と判定された数である．

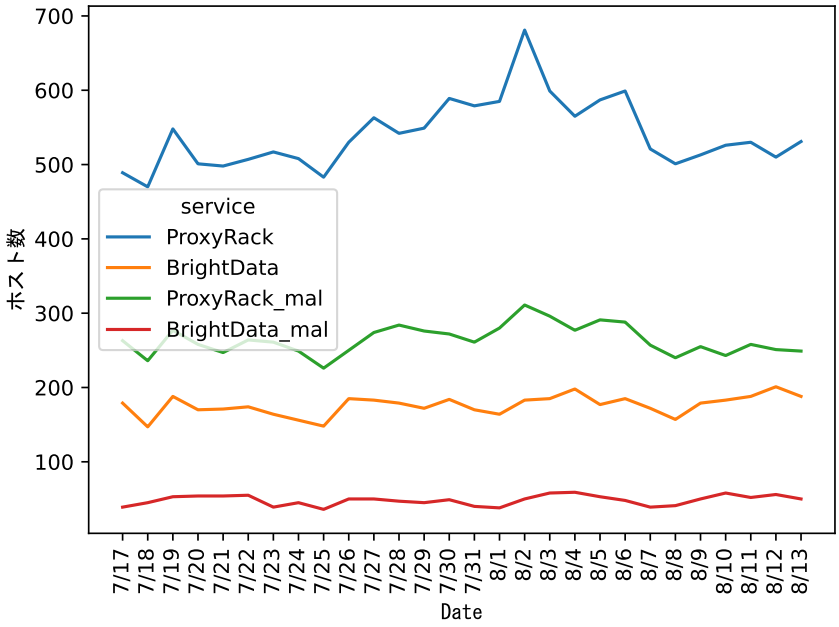


図 3.3: NICTER Darknet で観測されたホスト数

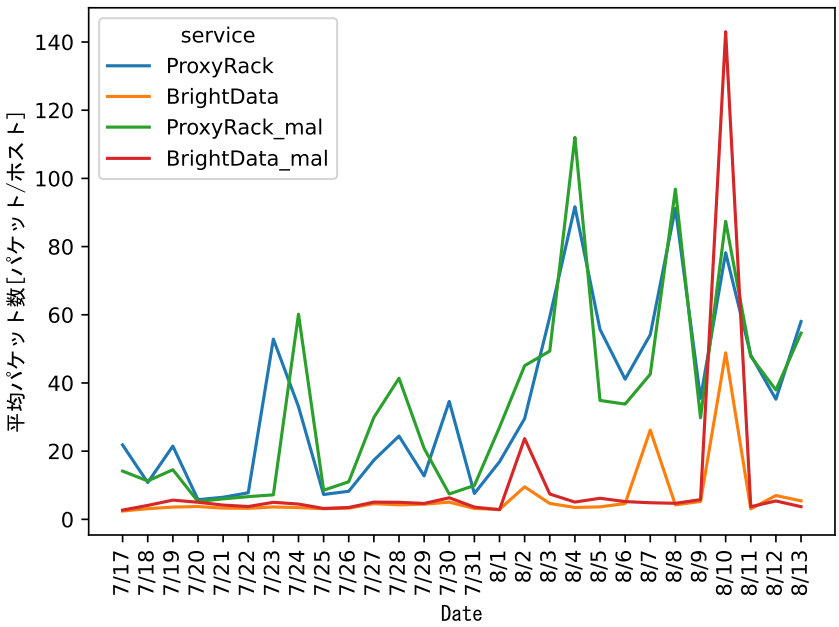
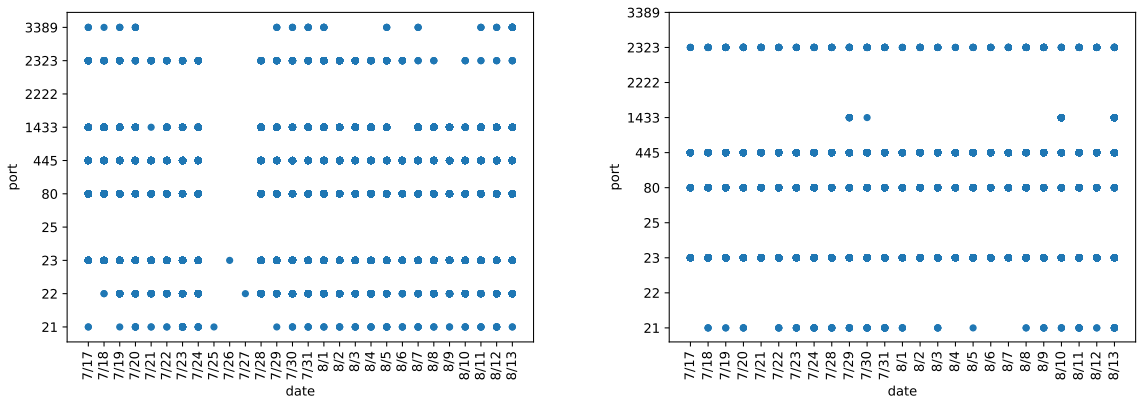


図 3.4: NICTER Darknet で観測された 1 ホスト当たりの通信数

表 3.2 に宛先のポート別パケット観測数を示す. 図 3.5a(ProxyRack), 図 3.5b(Bright Data) に観測された日付と宛先ポートの散布図を示す.

表 3.2: サービス別の宛先ポート別パケット観測件数

	ProxyRack		Bright Data		半澤	
宛先ポート番号 (サービス)	観測件数	[%]	観測件数	[%]	観測件数	[%]
21(FTP)	112	0.0	125	0.4	193,917	11.5
22(SSH)	38592	**7.3**	0	0.0	49,767	2.9
23(Telnet)	32300	6.1	4051	12.4	613,606	**36.4**
25(SMTP)	0	0.0	0	0.0	21,732	1.3
80(HTTP)	15150	2.9	2044	6.2	97,780	5.8
445(SMB)	19682	3.7	11284	**34.5**	399,250	23.7
1433(MSSQL)	4671	0.9	524	1.6	144,928	8.6
2222(SSH)	0	0.0	0	0.0	16,838	0.1
2323(Telnet)	754	0.1	363	1.1	43,310	2.5
3389(RDP)	64	0.0	0	0.0	9,782	0.5



(a) ProxyRack

(b) Bright Data

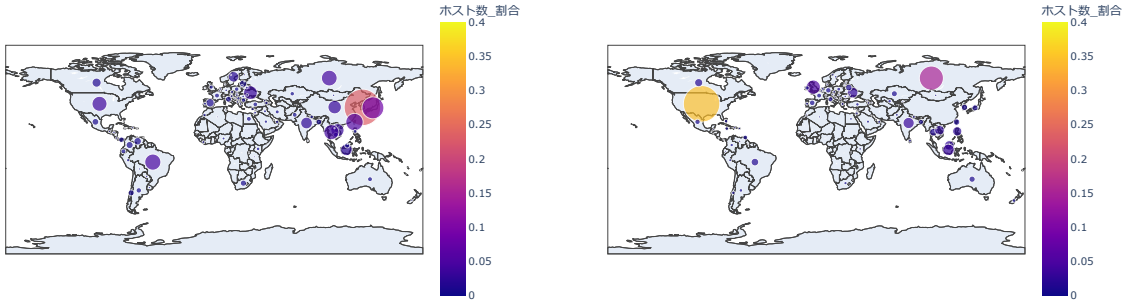
図 3.5: 宛先ポート

表 3.3 に Shodan を用いて調査した不正 IP アドレスで外部に開放しているポートを示す.

表 3.3: サービス別の不正 IP アドレスのホストのソース

	ProxyRack				Bright Data			
ポート番号	不正 IP 数	[%]	悪性な 不正 IP 数	[%]	不正 IP 数	[%]	悪性な 不正 IP 数	[%]
2000	169	**5.47**	33	19.53	48	**3.11**	4	8.33
80(HTTP)	119	3.85	15	12.61	28	1.81	2	7.14
1723(PPTP)	100	3.23	21	21.00	32	2.07	2	6.25
8291	75	2.43	11	14.67	18	1.17	2	11.11
53(DNS)	66	2.13	13	19.70	25	1.62	2	8.00
21(FTP)	53	1.71	10	18.87	6	0.39	0	0.00
22(SSH)	49	1.58	5	10.20	10	0.65	1	10.00
443(SMB)	48	1.55	8	16.67	28	1.81	2	7.14
7547(CWMP)	41	1.33	4	9.76	4	0.26	0	0.00
8080(HTTP)	29	0.94	7	**24.14**	5	0.32	1	**20.00**
5555	20	0.65	3	15.00	1	0.06	0	0.00
23(Telnet)	20	0.65	3	15.00	7	0.45	0	0.00

収集した IP アドレスすべてについて GeoLite2City データベースを用いて国を判別した．図 3.6a(ProxyRack)，図 3.6b(Bright Data) に世界地図のマッピングを示す．図 3.7a(ProxyRack)，図 3.7b(Bright Data) に不正 IP アドレスの割合の世界地図のマッピングを示す．表 3.4a(ProxyRack)，表 3.4b(Bright Data) にホスト数の上位 10 位とその国の不正ホスト率を示す．表 3.5 に 2 つのサービスでホスト数の上位 15 位以内で共通している国のホスト数と不正ホスト率を示す．



(a) ProxyRack　　**(b)** Bright Data

図 3.6: ホストの分布

表 3.4: ホスト数上位 10 か国と不正ホスト率

国名	ホスト数	不正ホスト率 [%]
大韓民国	16114	1.9
日本	5836	1.0
香港	5521	4.2
台湾	3655	3.2
ブラジル	3151	3.9
ロシア連邦	2988	5.6
タイ	2757	7.7
アメリカ合衆国	2715	3.6
ウクライナ	2293	7.0
中華人民共和国	2181	**7.5**

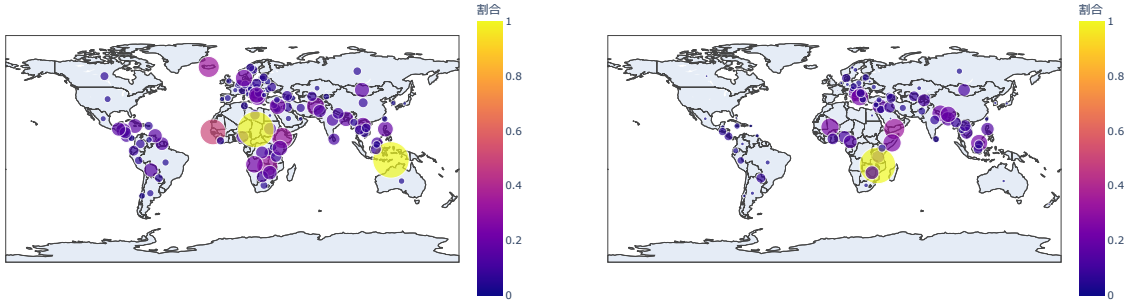
(a) ProxyRack

国名	ホスト数	不正ホスト率 [%]
アメリカ合衆国	24258	0.05
ロシア連邦	10578	3.0
イギリス	3715	0.1
インドネシア	2415	7.0
インド	2284	7.2
ウクライナ	2104	2.1
マレーシア	1691	6.0
ベトナム	1677	4.2
フィリピン	1555	**13.5**
カナダ	1214	0.6

(b) Bright Data

表 3.5: ホスト数上位 15 か国と不正ホスト率の比較

	ProxyRack		Bright Data	
国名	ホスト数	不正ホスト率 [%]	ホスト率	不正ホスト率 [%]
ブラジル	3151	3.9	1060	1.9
カナダ	1002	6.1	1214	0.6
インドネシア	1656	8.8	2415	7.0
インド	1647	11.8	2284	7.2
大韓民国	16114	1.9	695	0.7
ロシア連邦	2988	5.6	10578	3.0
タイ	2757	7.7	1058	3.9
台湾	3655	3.2	749	1.5
ウクライナ	2293	7.0	2104	2.1
アメリカ合衆国	2715	3.6	24258	0.05
ベトナム	2109	6.9	1677	4.2



(a) ProxyRack　　**(b)** Bright Data

図 3.7: 不正ホストの割合

3.4　考察

本研究で対象とした期間において，両サービスのホストから継続的に国内のダークネットにポートスキャンが行われていることが分かった．不正な通信を行っているホストの割合，悪性と判定された割合，パケットの総数において ProxyRack の方が多い．ホスト数，1 ホスト当たりの通信数ともに全ての日数においても ProxyRack の方が多い．

パケットの到達ポートに着目してみると，ProxyRack は 748 個，Bright Data は 365 個と ProxyRack の方が幅広いポートに通信が到達し，スキャン目的の通信が多いといえる．表 3.2 で挙げた 10 個のポートにおいても ProxyRack は 8 個，Bright Data は 6 個と差が見られた．例えば，22(SSH) は ProxyRack では 10 個の中で第 1 位であったが Bright Data では 0 であった．半澤らの研究では 10 個のポート全てに通信が確認されており，変化が見られた．一方，Bright Data は到達パケット数こそ少ないものの，ポート別の割合に着目すると Telnet と SMB への通信の割合が高い．スキャンではない意図的な行動の拠点になっていると考える．

外部に開放されているポートに着目してみると，マルウェア Mirai やその亜種が標的とするポートと報告されている 7547[10]，マルウェア Mirai やその亜種に探索対象のポートとして追加されたと考えられている，5555[11]，MikroTik 製のルータの Winbox がデフォルトで利用するポート番号でリモートから任意のコードを実行できる (RCE) 脆弱性が報告されている 8291[12]，同様に MikroTik 製のルータを標的とする通信の標的ポートである 2000[13] が解放されていた．特に，2000 は両サービス共に割合が第 1 位であり，悪性と判定されたホストの割合も第 4 位と高い．このことより，両サービスの proxy ホストは Mirai などのマルウェアに感染したデバイス，もしくは脆弱性を利用されたデバイスの割合が高いと考える．

ホストの分布はサービスごとに差があることが分かった．ProxyRack はアジア圏に集中しているのに対し，Bright Data は北米に分布している．

第 4 章

おわりに

2 つの代表的な RESIP サービスを調査した．両サービスのホストから継続的に国内のダークネットに通信を行っていることが分かった．不正 IP アドレスの割合や宛先ポート，不正 IP アドレスの解放ポートには RESIP サービスごとの差があり，不正 IP アドレスの割合は ProxyRack の方が高く，通信の宛先ポートも ProxyRack の方が幅広いことが分かった．不正 IP アドレスで解放されているポートには Mirai やその亜種の標的とされるポート，脆弱性が報告されているポートが含まれていることもわかった．

本研究では IP アドレスを収集した期間に差があるため，今後，同一の期間で条件を揃えて再実験する予定である

謝辞

本研究を行うにあたり，多くの方よりご指導いただきました．特に，指導教官である明治大学総合数理学部先端メディアサイエンス学科の菊池浩明教授には多大なるご指導を受け賜わりました．深く感謝申し上げます．また，研究にご協力，ご助言いただいた松本寛輝さん，半澤映拓さん，平山夏輝さん，梶間大地さん，井窪竜矢さん，福田ひかりさん，研究室の皆様に深く感謝の意を表すとともに，謝辞とさせていただきます．

参考文献

[1] Xianghang Mi et al. “Resident Evil: Understanding Residential IP Proxy as a Dark Service”, IEEE Symposium on Security and Privacy (SP), volume: 1, pp. 170-186, 2019.

[2] 半澤，“Residential IP Proxy サービスに悪用される住宅用ホストの調査”，2020 年度明治大学菊池研究室修士論文，2021.

[3] 竹久達也，神薗雅紀，笠間貴弘，中里純二，衛藤将史，井上大介，中尾康二，サイバーセキュリティ情報遠隔分析基盤 NONSTOP の利活用について，コンピュータセキュリティシンポジウム 2014 論文集，volume: 2，pp. 207-214, 2014.

[4] ProxyRack (https://www.ProxyRack.com/) (2021/11/27 参照)

[5] Bright Data (https://BrightData.com/) (2021/11/27 参照)

[6] NICTER WEB (https://www.nicter.jp/) (2021/11/27 参照)

[7] Virsutotal (https://www.virustotal.com/) (2021/11/27 参照)

[8] Shodan (https://www.shodan.io/) (2021/11/27 参照)

[9] MAX MIND, GeoLite2 Free Geolocation Data (https://dev.maxmind.com/geoip/geolite2-free-geolocation-data?lang=en) (2021/11/27 参照)

[10] TrendMicro,「Mirai」亜種か？海外製ルータを狙うアクセスが急増 (https://www.trendmicro.com/jp/iot-security/news/3086) (2021.12.19 参照)

[11] NICTER Blog (https://blog.nicter.jp/2018/10/android-5555/) (2021.12.19 参照)

[12] tenable, MikroTik RouterOS Vulnerabilities: There’s More to CVE-2018-14847 (https://www.tenable.com/blog/mikrotik-routeros-vulnerabilities-there-s-more-to-cve-2018-14847) (2021.12.19 参照)

[13] NICTER 観測レポート 2018 (https://www.nict.go.jp/cyber/report/NICTER_report_2018.pdf) (2021.12.19 参照)

付録 A

ARP スプーフィング攻撃の調査

A.1　背景

ネットワークの普及に伴いネットワークを用いて情報をやり取りすることが増えている．ネットワークは便利なものであるが，悪意ある人がネットワークを利用する人をターゲットに攻撃を仕掛けることもある．本稿では通信を盗聴する中間者攻撃の一種である ARP スプーフィング攻撃をいくつかの OS に仕掛けて攻撃が成功するか．また攻撃に利用する不正なパケットを送信する間隔を変えることでネットワークの速度に変化が起こるかを調べることで ARP スプーフィング攻撃の対策に有効なものがないか調査した．

A.2　ARP スプーフィング攻撃

A.2.1　概要

ARP スプーフィング攻撃とは同一ネットワーク内の通信を盗聴する中間者攻撃の一種であり，ARP プロトコルの特性を用いた攻撃である．ARP とは IP アドレスから MAC アドレスを得るために利用されるプロトコルである．ネットワーク内の端末は ARP テーブルを所持しており，それを参照することで通信を行っている．端末がネットワークに接続された際，また一定間隔で ARP パケットを送りあうことで互いの情報を交換し，それぞれの端末の ARP テーブルを更新している．しかし，この ARP テーブルの書き換えは容易であり，ARP パケットを送信することで書き換えることができる．もし，このパケットの中身が不正に書き換えられたものであった場合，端末は本来意図しない端末に対して通信を行うことになる．これが ARP スプーフィング攻撃である．攻撃前と攻撃後で通信の経路がどのように変わるかを示す．正常時の通信の様子が図 A.1，攻撃時の通信の様子が A.2 である．

A.2.2　先行研究

1 つ目は TARP[1] である．これは ARP を暗号化することで通信に安全性を持たせるものである．しかし暗号化することによりアドレス解決に時間を要するという問題点がある．2 つ目は ARPwatch[2] である．これは ARP テーブルを監視することで ARP テーブルの書き換えを検知するものであるが，書き換えられてしまっているためすでに攻撃は完了しているという問題点がある．3 つ目は動的 ARP 検査 [3] である．これは Cisco 社のスイッチに内蔵されている機能であり，有効な ARP リクエスト，ARP リプライのみ中継する機能である．ARP のパフォーマンスを低下させることはないが Cisco 社のスイッチに内蔵されている機能であるため

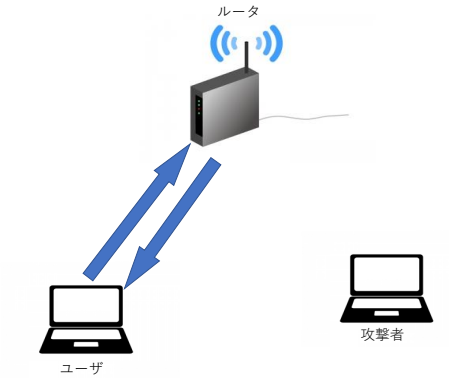


図 A.1: 正常な通信

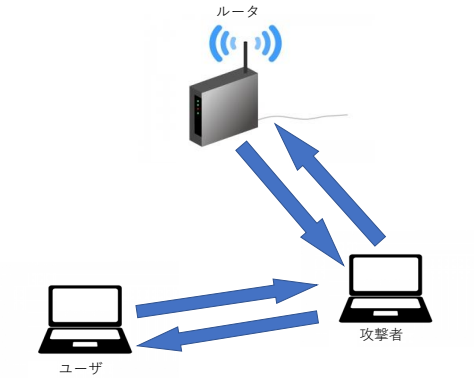


図 A.2: 攻撃時の通信

導入にコストがかかる．4 つ目は ARP リクエスト内の送信元 MAC アドレスと Ethernet ヘッダ内の MAC アドレスを比較することで不正なパケットかどうか判別する手法 [4] である．

A.2.3　ARP スプーフィングの活用

本稿では ARP スプーフィングを攻撃手法の 1 つとして扱ったが，ARP スプーフィングは攻撃としてだけでなく，セキュリティ対策機器にも用いられている．用いられている機器としてトレンドマイクロ社のウイルスバスター for Home Network が挙げられる．この製品についての調査は [5] で行われている．

A.3　ARP スプーフィング攻撃下での RTT の計測

A.3.1　実験目的

ネットワークの速度の変化を観測することで ARP スプーフィング攻撃を検知するために有用だと考えられる情報がないか調査する．

A.3.2　実験方法

本実験は家庭の Wi-Fi 環境で行った．攻撃機は VirtualBox 上で動作している ubuntu，攻撃対象は windows10，macOS，IOS の 3 つの OS である．攻撃が成功するか．攻撃前，攻撃後のネットワーク速度の変化．また攻撃端末が攻撃対象に送る不正な ARP パケットの間隔を変えることでネットワーク速度のが変化するかを観測する．攻撃が成功するかは攻撃機で起動している wireshark で，攻撃対象機から 8.8.8.8 に送信される ping パケットが観測できるかを判断の基準としている．ネットワーク速度の観測にはラウンドトリップタイム (以下 RTT) を用いた．攻撃対象機から 8.8.8.8 に ping を 20 回送り，得られる RTT の時間の平均を取ることで変化しているかを確かめた．

本実験に用いたプログラムは python のライブラリ scapy を用いて作成したプログラムであり，不正な ARP パケットを一定間隔で繰り返し攻撃対象機に送信し攻撃対象機の ARP テーブルを不正に書き換えるものであ

る．また ubuntu の設定で IP フォワーディングを有効にすることで攻撃機である ubuntu を経由した通信を可能にして ARP スプーフィング攻撃を行った．

A.3.3 実験結果

実験の結果以下のデータが得られた．

表 A.1: 攻撃が成功するか

Windows10	macOS	ios
×	×	×

表 A.2: RTT の変化 [ms]

送信間隔 [s]	Windows10	macOS	ios
平常時	11.6	14.7	25.4
2	130.2	60.7	106.2
3	110.6	51.2	104.6
4	77	-	70.2
5	81.2	-	57.2

表 A.3: RTT の平均と標準偏差

送信間隔 [s]	平均 [s]	標準偏差
平常時	17.2	5.9
2	99	28.8
3	88.8	26.7
4	73.6	3.4
5	69.2	12

A.3.4 考察

本実験の結果より攻撃の前の 3 機種の RTT の平均に比べて攻撃後の RTT の平均は 4.8 倍大きいことが分かった．不正パケットを送信する間隔の変化での通信速度の変化も見られた．本実験で用いたプログラムでは不正なパケットを送信することのみを行っているため，タイミングによっては正常なパケットが送信され ARP テーブルが正常なものに書き変わっていることが考えられる通信速度も観測できた．また同じ間隔で送信している間での通信速度の観測でもタイミングによって大きく速度が遅くなる場面が観測できた．

A.4　まとめ

本実験の結果より攻撃の前の 3 機種の RTT の平均に比べて攻撃後の RTT の平均は 4.8 倍大きいことが分かった．全体的に不正パケットを送信する間隔が開くことで通信速度が向上するといえるが，間隔があけば必ず通信速度も向上するわけではないことも分かった．ただし，本実験の結果は python で作成したプログラムを用いた際に得られた結果であり，実際に攻撃が行われる際に用いられるプログラムでも同じ挙動が観測されることを保証するものではない．

A.5　ソースコード

本実験に用いた python のソースコードを記載する．本プログラムを作成するにあたり，[6] を参考にした．

```
from scapy.all import *
import time

target_ip = 対象の機器のアドレスIP
gateway_ip = 対象機器の所属するゲートウェイのアドレスIP

def getmac(ip):
    arp = Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip)
    res, v = srp(arp)
    for s, r in res:
        return r[Ether].src

def ArpSpoofing(target_ip, target_mac, gateway_ip, gateway_mac):
    arp_target = Ether(dst=target_mac)/ARP(op=2, psrc=gateway_ip, hwdst=
        target_mac, pdst=target_ip)
    arp_gateway = Ether(dst=gateway_mac)/ARP(op=2, psrc=target_ip, hwdst=
        gateway_mac, pdst=gateway_ip)
    for i in range(100):
        sendp(arp_target)
        sendp(arp_gateway)
        time.sleep(2)
    print("finish")


def main():
    target_mac = getmac(target_ip)
```

```
    gateway_mac = getmac(gateway_ip)
    ArpSpoofing(target_ip, target_mac, gateway_ip, gateway_mac)


if __name__ == '__main__':
    main()
```

参考文献

[1] Wesam Lootah, William Enck, and Patrick McDaniel, ”TARP: Ticket-based Address Resolution Protocol” (https://www.enck.org/pubs/acsac05.pdf) (参照 2021/01/11)

[2] ”arpwatch(8) - Linux man page” (https://linux.die.net/man/8/arpwatch) (参照 2021/01/11)

[3] ” ダイナミック ARP インスペクション（DAI）” (https://www.cisco.com/c/ja_jp/td/docs/sw/campuslanswt-coredistribution/cat6500swt/cg/002/15-1-sy-c4-swcg/dynamic-arp-inspection.html#98273) (参照 2021/01/11)

[4] 松藤央，落合秀也，江崎浩，” 無線端末による ARP を用いたセグメント内の通信妨害攻撃とその対策”，マルチメディア，分散，協調とモバイル (*DICOMO*2018) シンポジウム，2018，pp.210-213

[5] 平山夏輝，” ウイルスバスター for Home Network の調査研究”，明治大学菊池研究室 2020 年度卒業論文，2020

[6] Justin Seitz(2015) 『サイバーセキュリティプログラミング-Python で学ぶハッカーの思考』（青木一史ほか訳）株式会社オライリージャパン