明治大学総合数理学部

2025 年度

卒　業　研　究

2FA をバイパスする中間者攻撃ツール Evilginx によるフィッシング攻撃の脅威分析

学位請求者　先端メディアサイエンス学科

太田 和希

目次

第 1 章

はじめに

サイバーセキュリティの脅威の一つにフィッシング攻撃がある．2022 年に発生したデータ侵害の 36 %にフィッシングが関与している [1]．そこで，その対策として 2FA(二要素認証) が広く講じられている．しかし，Evilginx[2]，Muraena[3]，Modlishka[4] などのフィッシングツールキットの中には，中間者攻撃を行い，2FA をバイパスして不正アクセスをできるものがある．

そこで，本研究では，フィッシングツールキットの一つである Evilginx の動作確認を行い，2FA を破るかどうかを検証する．さらに，主要な Web サービスの二要素認証方式を調査して，本攻撃に対する影響を仕様から評価する．

第 2 章

準備

2.1 2FA

2FA とは，安全性を高めるために用いられる認証方式であり，認証の際に ID とパスワードだけでなく追加の認証要素を要求する認証方式である．認証要素には知識要素 (本人のみが知っていること)，所有要素 (本人のみが所有するもの)，身体要素 (本人の身体的特徴) がある．

2.2 AiTM 攻撃

Adversary-in-the-Middle(AiTM) 攻撃は中間者攻撃の一種であり．攻撃者はユーザと正規サイトの間でリバースプロキシとして働き，ログイン情報を取得する．

図 2.1 に AiTM 攻撃の概要を示す．1. 攻撃者 B からユーザ A に対して，フィッシング URL を含むメール等を送り，攻撃者が作り出したフィッシングサイト B に誘導する．2. 攻撃者は ID とパスワードを取得し，それらを正規サイト C に転送する．3. 正規サイト C はユーザ A に二要素認証の要求をして，メールや SNS などで 2FA コードを送る．4. 攻撃者 B はユーザ A から 2FA コードを取得し，正規サイト C に転送する．

これらにより，攻撃者は認証済みのセッションを取得し，サイトに不正にログインする．

2.3 Evilginx

Evilginx とは，フィッシングツールキットの一つである．リバースプロキシとして働き，2FA をバイパスしてユーザと Web サイト間のセッションを不正に取得することができる．

Evilginx は Go 言語で書かれており，OS は Windows と Linux の両方で使用することができる．CLI でコマンドを打ち込み操作を行う．図 2.2 に取得したセッションの例を示す．

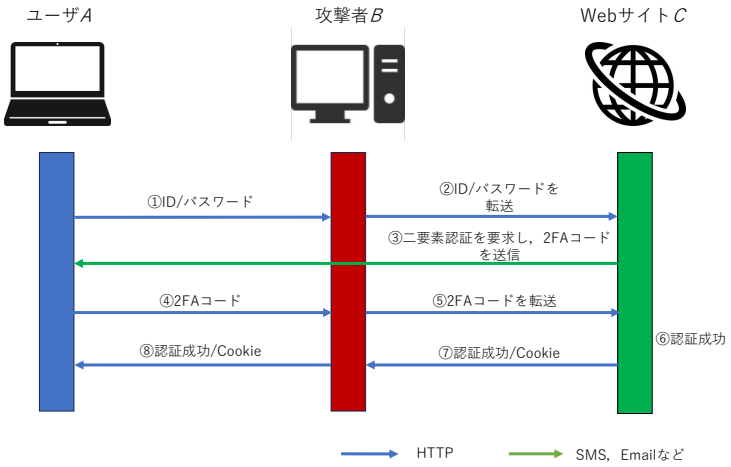

図 2.1　AiTM 攻撃の概要

: sessions 127

 id   : 127
 phishlet   : example
 username   : ■■■■■■■■■■@■■■■■■
 password   : ■■■■■■■■
 tokens   :
 landing url : https://academy.kikn.fms.meiji.ac.jp/JDxUeqmf
 user-agent   : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/131.0.0.0
Safari/537.36
 remote ip   : 127.0.0.1
 create time : 2024-12-23 14:27
 update time : 2024-12-23 14:29

図 2.2　セッション

第 3 章

Evilginx の評価

3.1　基本動作

Evilginx は，GitHub からダウンロードし実行する．図 3.1 に Evilginx の起動画面を示す．次に，config, phishlets，lures を設定する．config コマンドで，中間者として働くサーバのドメインと IPv4 を入力する．phishlets コマンドで，phishlet と呼ばれる特定のサイトをターゲットにするための設定ファイル (yaml) を読み込む．lures コマンドで，フィッシングサイトの URL を取得する．そして，取得した URL をユーザが開くことで攻撃が実行される．

また，Evilginx は TLS を欺くために Let's Encrypt から偽の公開鍵証明書を自動で取得する．

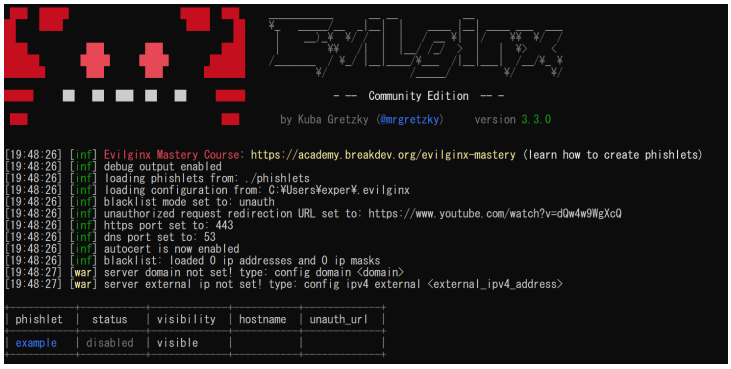

図 3.1　起動画面

3.2　処理速度計測

正規サイトと Evilginx を用いたフィッシングサイトでページの読み込み時間，ログインにかかる時間の計測を行った．ローカル環境で行い，ターゲットとなる Web サイトには，Evilginx の学習サイトである Enilginx Mastery[5] を用いた．

3.2.1 実験目的

フィッシングサイトのほうが正規サイトよりも処理速度が遅いと仮定し，正規サイトとフィッシングサイトの処理時間の違いからフィッシングサイトのシステムの遅延を計測する．また，その遅延によって，一般ユーザが異変に気付くことができるか推測する．

3.2.2 実験方法

正規サイトとフィッシングサイトでページの読み込み時間を 50 回ずつ，ログイン時間を 30 回ずつ計測し，平均の時間を比較する．ページの読み込み時間は，ページ上のすべての要素が表示されるまでの時間として計測を行った．時間の計測には Selenium[6] を使用した．

3.2.3 結果

表 3.1 に実験結果を示す．読み込み時間では約 0.8 秒，ログイン時間では約 3.0 秒の差が発生した．証明書の読み込み時間，中間者を介した通信によって処理速度に差が生じたと考えられる．

一方で，計測した時間は本来使用するネットワークの状況で前後するものであるため，これらの差が生じても一般ユーザは攻撃の存在に気づくことができないと推測される．

表 3.1　実験結果

		平均値	最小値	最大値	標準偏差
読み込み時間	正規	4.28	1.19	8.89	0.84
	フィッシング	5.04	3.64	8.02	0.87
ログイン時間	正規	4.10	3.38	4.97	0.41
	フィッシング	7.14	4.85	12.40	1.70

第 4 章

Web サイトの脆弱性評価

4.1 仮定

二要素認証方式を 8 つの方式に分類し，AiTM 攻撃に対して脆弱であるかを仮定する．表 4.1 に各二要素認証方式とそれらの AiTM 攻撃への耐性を示す．

表 4.1 各二要素認証方式の AiTM 耐性

二要素認証方式	AiTM 耐性	認証コード秘匿
SMS	×	×
phone call	×	×
Email	×	×
TOTP	×	×
App	×	○
push notification	×	○
USB key	○	○
passkey	○	○

SMS，phone call，Email

SMS，phone call，Email を用いた二要素認証は脆弱であると考える．これらの認証方式は，4 桁や 6 桁のコードを取得し，打ち込む．従って，図 2.1 に示した AiTM 攻撃の概要のように攻撃を受けると推測される．

また，これらの方法は AiTM 攻撃以外の攻撃にも脆弱性がある．特に，SMS では SIM スワッピング攻撃 [7] や PRMitM 攻撃 [8] などを受けるリスクがある．

TOTP

認証アプリを用いた二要素認証は脆弱であると考える．この認証方式では，アプリが TOTP と呼ばれる 6 桁のコードを生成し，ユーザがそれを打ち込む．この方式は，SMS，phone call，Email などの認証方式のようにサーバ側が認証コードをインターネットにより送信することがないため，それらの認証方式よりかは安全だと言える．

しかし，ユーザが認証コードを送信する際は，特別に暗号化を施しておらず中間者によって傍受されてしまう．従って，AiTM 攻撃に対して脆弱であると推測される．

専用アプリ，プッシュ通知

専用アプリやプッシュ通知を用いた二要素認証は脆弱であると考える．これらの認証方式は Web サイトとユーザの持つアプリやデバイスの間で直接，認証要求と認証を行う．そのため，認証コードを中間者に取得されることはない．

しかし，ユーザが認証をすることで Web サイトは正常に認証が成功したと判断し，セッションを中間者に送信してしまう．従って，AiTM 攻撃に対して脆弱であると推測される．

セキュリティキー，パスキー

セキュリティキーによる認証方式は Evilginx に対して安全であると考える．セキュリティキーは，USB や NFC などのハードウェアを用いて認証する方式である (例：YubiKey[9])．パスキーは，PC やスマートフォンで顔や指紋などを用いて認証する方式である (例：Windows Hello[10])．これらの認証方式は，デバイスが認証サーバの正しい URL を保存しているため，中間者の偽の URL を区別することが可能である．そのため，これらの認証方式は，Evilginx に対して安全であると推測される．

4.2 Web サイトの二要素認証方式の調査

4.2.1 調査目的

国内，国外のサイトを含む 102 のサイトを調査対象とした．二要素認証を導入しているか，どのような二要素認証方式をとっているのかを明らかにする．

4.2.2 調査方法

サイトの公式ドキュメントが公開されていれば参照して，対応している二要素認証の種類を調べた．ドキュメントがないサイトは自らアカウントを作成して調査をした．二要素認証の対応状況やカテゴリの分類に 2FA Directory[11] というサイトを利用した．

4.2.3 選定基準

国内のサイトを 48 サイト，国外のサイトを 54 サイト調査した．対象サイトは様々なカテゴリから網羅的に選択した．また，国外のサイトは，日本でも知名度が高いサイトを優先して選択した．

4.2.4 結果

二要素認証の導入率

表 4.2 にそれぞれの二要素認証導入率を示す．ソーシャルログインとは，他のサービスの ID を用いてログインすることである．そのサービスが二要素認証を導入している場合，同じように二要素認証を利用することができる．

全体では，102 サイトのうち 68 サイト (67 %) が，ソーシャルログインを含めると 102 サイトのうち 85 サ

イト (83 %) が二要素認証を導入していた．また，ソーシャルログインを含めるかにかかわらず，国内サイトより国外サイトのほうが二要素認証の導入率が高い．

表 4.2 から，ソーシャルログインを含めると 8 割を超えるサイトが二要素認証を導入している．これは，フィッシング攻撃が依然として増え続けている [14] からであると推測される．既存のフィッシング攻撃への対策として，2FA を用いた認証方式は有効である．そのため，ほとんどのサイトが二要素認証を導入していると考えられる．

一方，国内と国外サイトの普及率は，ソーシャルログインを含まない場合で約 23 %，含む場合で約 11 %国外サイトが上回っている．

この差は単に国内と国外の差ではなく，本調査でのサイトの選定基準に起因するものであると考える．国外サイトの選定基準として，日本でも知名度が高いサイトを優先して選定した．そのため，国外サイトは国内サイトよりユーザ数が多いサービスや大企業が運営するサービスが多いと考えられる．つまり，本調査からは国内と国外のセキュリティ意識に違いがあるとは言えず，むしろ運営する企業の規模によって差が生じるのではないかと推測される．

表 4.2　二要素認証導入率

ソーシャルログイン	国内 [%](件)	国外 [%](件)	全体 [%](件)
含まない	54.17 (26)	77.78 (37)	66.67 (68)
含む	77.08 (42)	88.89 (48)	83.33 (85)
計	(48)	(54)	(102)

二要素認証方式

調査したサイトで用いられていた二要素認証方式を 8 つの種類に分類し集計を行った．図 4.1 に二要素認証方式別の件数を示す．二要素認証を導入しているサイトには複数の認証方式を利用しているサイトが多いため，それぞれの認証方式の合計数は 102 を超える．複数の認証方式をサポートしているサイトはあるが，複数を必須に，すなわち，単一の認証を禁じているサイトは存在しなかった．

最多は TOTP(Time-based One-Time Password) を用いた認証アプリで 51 サイト，二番目は SMS で 38 サイトとなった．

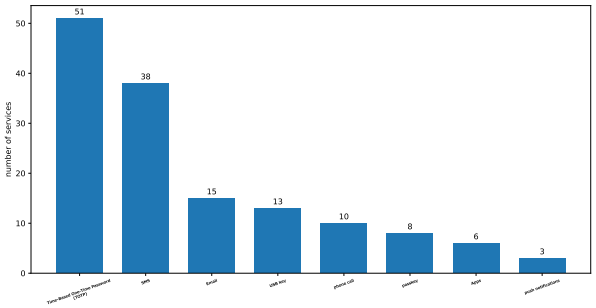

図 4.1　二要素認証方式

カテゴリごとの二要素認証導入率

対象 102 サイトを 2FA Directory[11] に従い 38 カテゴリに分類した．その中から 1 カテゴリに 4 サイト以上あるカテゴリを抽出し，15 カテゴリで比較を行った．

図 4.2 にカテゴリごとの二要素認証導入率，表 4.3 に調査結果の一部を示す．Communication カテゴリと Hosting/VPS(例：さくらインターネット [12]) カテゴリはともに二要素認証の導入率が 100 %であった．また，Hotels & Accomodation(例：NH Hotels[13]) カテゴリは二要素認証の導入率が 0 %であった．

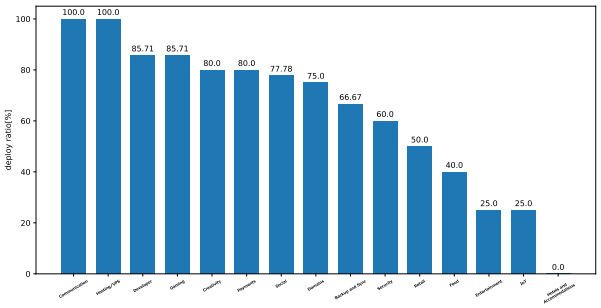

図 4.2　カテゴリごとの二要素認証導入率

4.3　考察

調査した結果，4.1 節で脆弱でないと仮定した方式であるセキュリティキー，パスキーを導入していないサイトは 102 サイトのうち 87 サイト (85 %) であった．つまり，調査を行った 85 %のサイトは AiTM 攻撃に対して脆弱であると推測される．

表 4.3　調査結果 (1/2)

カテゴリ	サイト名	国	多要素認証	ソーシャルログイン	SMS	phone call	Email	認証アプリ	専用アプリ	セキュリティキー	パスキー	プッシュ通知
Backup and Sync	Dropbox	アメリカ	○		○			○		○		○
	icloud(AppleID)	アメリカ	○		○	○				○		○
	Files.com	アメリカ	○		○		○	○		○		
	Evernote	アメリカ	○	○				○				
	firestroage	日本		○								
	infinicloud	日本										
Communication	Discord	アメリカ	○		○			○			○	
	slack	アメリカ	○		○			○				
	zoom	アメリカ	○		○			○				
	miro	アメリカ	○					○				
	LINE	日本	○						○			
	Chatwork	日本	○		○			○				
Creativity	Canva	オーストラリア	○					○				
	Adobe	アメリカ	○	○	○		○		○			
	pixiv	日本	○	○				○				
	小説家になろう	日本										
	studio	日本	○	○				○				
Developer	unity	アメリカ	○		○			○				
	readme	アメリカ	○					○				
	openAI	アメリカ	○	○				○				
	Github	アメリカ	○		○			○	○	○	○	
	Monaca	日本		○								
	Qiita	日本	○					○				
	zenn	日本	○	○								
Domains	Crazy Domains	オーストラリア	○	○	○	○						
	101domains	アメリカ	○					○				
	namecheap	アメリカ	○					○		○		
	ライブドアブログ	日本										
Entertainment	Netflix	アメリカ										
	DAZN	イギリス										
	Twitch	アメリカ	○		○			○				
	Spotify	スウェーデン		○								
	DMM	日本	○	○				○				
	コミックシーモア	日本		○								
	SHOWROOM	日本		○								
	ブックライブ	日本		○								
Food	wolt	フィンランド		○								
	UberEats	アメリカ	○	○	○			○				
	Starbucks	アメリカ			○							
	食べログ	日本	○	○								
	cookpad	日本		○								
Gaming	VRChat	アメリカ	○					○				
	Nvidia	アメリカ	○	○						○	○	
	Electronic Arts	アメリカ	○	○	○	○	○	○				
	Nintendo	日本	○	○				○				
	KONAMI	日本	○				○					
	CAPCOM	日本	○					○				
	gamewith	日本		○								
Hosting/VPS	KnownHost	アメリカ	○					○				
	PlanetHoster	カナダ	○					○				
	さくらインターネット	日本	○		○	○	○	○				
	ConoHa	日本	○					○				
Hotels and Accommodations	NH Hotels	スペイン										
	トリバゴ	ドイツ		○								
	Hotels.com	アメリカ		○								
	一休	日本		○	○							
	エアトリ	日本										
	HIS	日本										
IoT	Xiaomi	中国		○								
	IFTTT	アメリカ	○	○				○				
	Nature	日本		○								
	TOLIGO	日本										
Payments	paypal	アメリカ	○					○				
	Stripe	アメリカ	○		○			○		○	○	
	paypay　アプリ	日本	○		○							
	Vpass	日本										
	MIXI M	日本	○	○	○		○				○	
Retail	Amazon	アメリカ	○		○	○		○				
	ZARA	スペイン										
	楽天市場	日本										
	zozotown	日本		○								
	minne	日本	○	○	○	○						
	Yahoo! JAPAN	日本	○		○		○					

表 4.4　調査結果 (2/2)

カテゴリ	サイト名	国	多要素認証	ソーシャルログイン	SMS	phone call	Email	認証アプリ	専用アプリ	セキュリティキー	パスキー	プッシュ通知
Security	Norton	アメリカ	○		○	○		○		○		
	DNSFilter	アメリカ	○					○				
	Malwarebytes	アメリカ		○								
	Trend Micro	日本	○		○		○	○				
	ソースネクスト	日本										
Social	X	アメリカ	○	○	○ (有料)			○		○	○	
	instagram	アメリカ	○	○				○				
	facebook	アメリカ	○		○			○		○		
	TikTok	中国	○	○	○		○	○				
	LinkedIn	アメリカ	○	○	○			○				
	Reddit	アメリカ	○	○				○				
	yay	日本	○	○				○				
	Wantedly	日本		○								
	mixi	日本										
Cloud Computing	DigitalOcean	アメリカ	○		○			○				
	Heroku	アメリカ	○					○	○	○		
Education	udemy	アメリカ	○				○					
Task Management	Nifty	アメリカ	○					○				
Universities	Oh-o! Meiji	日本	○		○	○		○				
VPN Providers	Cyberghost	ルーマニア										
other	Google	アメリカ	○		○	○		○		○	○	○
	Microsoft	アメリカ	○		○	○	○	○		○	○	
	リクルート	日本	○		○		○	○				
	マイナビ	日本	○		○		○					
	one career	日本										
	GMO	日本	○	○			○	○				
	Mercari	日本	○		○							
	au(au ID)	日本	○		○		○		○			
	docomo(d アカウント)	日本	○		○				○			

第 5 章

おわりに

本稿では，正規サイトと Evilginx を用いたフィッシングサイトの読み込み時間とログイン時間の計測を行った．その結果，読み込み時間では約 0.8 秒，ログイン時間では約 3.0 秒の差が発生した．そのことから，一般ユーザは攻撃の存在に気づくことができないと推測された．

また，主要な Web サイトの AiTM 攻撃に対する脆弱性を検討した．調査の結果，85 %の Web サイトが脆弱であることがわかった．

謝辞

本研究を行うにあたり，多くの方より御指導いただきました．特に，多大なる御指導を受け賜りました，明治大学総合数理学部先端メディアサイエンス学科，菊池浩明教授に深く感謝申し上げます．また，共同で付録 A の研究を行った三浦晃暉さん，研究にご協力，ご助言いただいた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] Dataprot,“Phishing Statistics”(https://dataprot.net/statistics/phishing-statistics/，2024 年 11 月参照)

[2] Kuba Gretzky,“Evilginx”(https://github.com/kgretzky/evilginx2，2024 年 11 月参照)

[3] Muraena,“Muraena”(https://github.com/muraenateam/muraena，2024 年 11 月参照)

[4] Modlishka,“Modlishka”(https://github.com/drk1wi/Modlishka.，2024 年 11 月参照)

[5] BREAKDEV Academy,“Evilginx Mastery”(https://academy.breakdev.org/evilginx-mastery，2024 年 11 月参照)

[6] Selenim,“Selenium”(https://www.selenium.dev/，2024 年 12 月参照)

[7] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert,“ SMS-based one-time passwords: Attacks and defense”, Proceedings of the 10th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment，DIMVA，pp.150-159，2013.

[8] Nethanel Gelernter, Senia Kalma, Bar Magnezi, Hen Porcilan,“The Password Reset MitM Attack”, IEEE Security and Privacy，pp.251-267，2017.

[9] Yubico,“YubiKey”(https://www.yubico.com/yubikey/，2024 年 12 月参照)

[10] Microsoft,“Windows Hello”(https://www.microsoft.com/en-us/windows/tips/windows-hello/, 2024 年 12 月参照)

[11] 2FA Directory,“2FA Directory”(https://2fa.directory/，2024 年 11 月参照)

[12] さくらインターネット,“さくらインターネット”(https://www.sakura.ad.jp/，2024 年 12 月参照)

[13] NH Hotel,“NH Hotels & Resorts”(https://www.nh-hotels.com/，2024 年 12 月参照)

[14] フィッシング対策協議会,“月次報告書”(https://www.antiphishing.jp/report/monthly/，2024 年 11 月参照)

付録 A

ARP スプーフィングを用いたインターネットの利用時間を管理するホームセキュリティの開発

A.1　はじめに

近年，YouTube や TikTok といった様々なサービスが登場し，SNS やゲーム依存をはじめとするインターネット依存症の子どもの人数が増加傾向にある [1]．スマートフォンの普及率は年々増加 [2] しており，幼年期のうちからインターネットを利用する人数が増えていくと予想される．そのため，保護者が子どものインターネット利用を制限することが重要である．しかし，既存の利用制限アプリやサービス [3] では，制限したいデバイスに直接インストールする必要があり，パスワードの漏洩や抜け道により，制限が容易に解除される事例が指摘されている [4]．企業内 LAN で広く用いられている専用のファイアウォール製品は，高度な運用知識が必要であり，ホームネットワークには適していない．

そこで，本研究では，ホームネットワークの環境に配慮した，別のデバイスから対象のデバイスに制限をかけるシステムを開発する．同一ネットワーク内の管理用 PC にて本システムを運用し，専用ファイアウォールなしでネットワーク内のデバイスの管理，制限を行うことを目的とする．本研究では，IP アドレスから MAC アドレスを導出するためのプロトコルである ARP(Address Resolution Protocol)[5] に注目する．それぞれのデバイスがもつ，IP アドレスと MAC アドレスを対応づける ARP テーブルを意図的に書き換える ARP スプーフィングを応用し，デバイスの利用制限を実現する．

2022 年，井窪 [11] は，ARP スプーフィングを用いた，ネットワーク内のファイアウォールシステムである SHS(Simple Home Security) を開発した．SHS は，家庭内ネットワークのセキュリティ対策を目的とし，Wi-Fi に接続しているデバイスの検出，それらの IP アドレス，MAC アドレス，ベンダー情報の検出，サイトのアクセス制限を実現している．それに対し本研究では，子どものデバイスのセキュリティ対策と利用時間の管理を目的としている．検出されたデバイスについて，Wi-Fi 接続時間と閲覧可能コンテンツの管理の機能を新しく実現する．

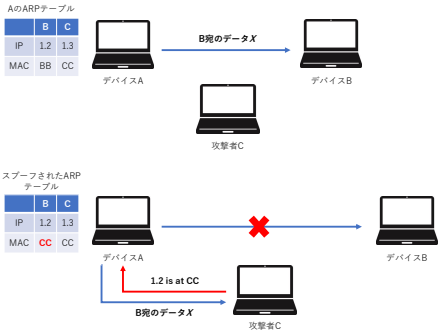

図 A.1　ARP スプーフィングのしくみ

A.2　提案システム構成

A.2.1　ARP

ARP[5] は，LAN 内で IP アドレスから MAC アドレスを求めるプロトコルである．デバイスは，解決したい IP アドレスを ARP リクエストをして送信し，目的のデバイスは ARP リプライを送信する．対応付けられた IP アドレスと MAC アドレスを自身の ARP テーブルに格納している．デバイスは他のデバイスと通信する際，自身の ARP テーブルを参照して，通信相手の IP アドレスから MAC アドレスを取得し，Erthernet フレームを送信する．

A.2.2　ARP スプーフィング

ARP スプーフィングは，不正な ARP リプライを対象に送信し，ARP テーブルを書き換え，通信経路を強制的に変更する攻撃である．図 A.1 にネットワーク内での本来の通信と ARP スプーフィングでの通信の例を示す．通常，デバイス A は自身の ARP テーブルを参照してデバイス B の IP アドレス 1.2 の MAC アドレス BB を取得し，データ X を送信している．しかし，攻撃者 C がデバイス A に不正な ARP リプライを送信して ARP テーブルを書き換えることで，デバイス B の MAC アドレス BB を攻撃者 C の MAC アドレス CC に改変させる．そのため，B 宛のデータ X は攻撃者 C に届いてしまう．

A.2.3　システム構成

図 A.2 に本研究のシステム構成を示す．本来であれば，ネットワーク内のデバイス A，B，C がインターネットを利用する際，ルータを介して通信する．本研究では，ルータと A，B，C に ARP スプーフィングを仕掛け，全てのアウトバウンドの通信を X を経由するようにする．このシステムはネットワーク内にある X にて実行する．

A.2.4　実装する機能

本研究で開発するシステムで表 A.1 の機能を実装する．

表 A.1 の 1，2 はネットワーク内の全てのデバイスに対して行う．それ以外の機能は，利用者が管理対象の

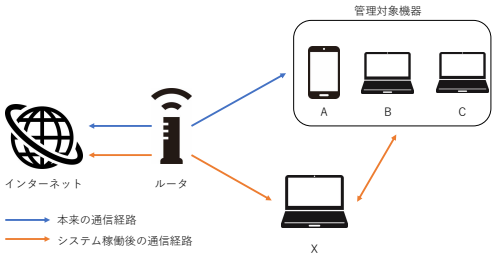

図 A.2　システム構成図

表 A.1　研究分担

研究者	分担内容	章
太田	1. ネットワークに接続されているデバイスを検知する機能	3.2.1
	2. デバイスが Wi-Fi に接続，切断した際に通知する機能	3.2.2
	3. Wi-Fi に接続している時間を管理し，利用制限する機能	
三浦	4. 管理するコンテンツの IP アドレスを検索し，リストに登録する機能	3.2.3
	5. コンテンツの閲覧時間を管理する機能	3.2.4

機器やコンテンツを個別に設定する．

A.3　システムの開発

A.3.1　開発環境

本研究では，python を用いてシステムの開発を行った．また，ARP スプーフィングには，ライブラリである scapy[7] を用いた．

A.3.2　主要な機能の仕組み

1. デバイス検知
   デバイス検知は，インストールしている PC の ARP テーブルを取得することによって行う．これにより，ネットワーク内にあるデバイスの IP アドレス，MAC アドレスを取得する．
2. Wi-Fi 接続時間による利用制限
   利用者は管理対象機器を設定する際，個別に制限時間を分単位で指定する．制限時間を超えると，そのデバイスとルータに ARP スプーフィングを行う．表 A.2 のように ARP リプライを送信し ARP テーブルを書き換えることで通信が遮断される．
3. 閲覧管理するコンテンツのリスト登録
   閲覧サイトのホワイトリスト登録は，対象の Web サイトのドメインによって行う．対象 Web サイトの IP アドレスを以下のフォーマットで json ファイルに記録する．

```
{
    "example.com": [
```

```
            "192.0.2.1",
            "192.0.2.2",
            "192.0.2.3"
        ]
    }
```

4. コンテンツ閲覧時間の管理
   3 の json ファイルに記録されている Web サイトを閲覧している時間を計測し、設定した閲覧制限時間を比較する.

表 A.2　ARP リプライ

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000	1.1	1.2	ARP	60	1.2 is at AA
2	0.001	1.1	1.3	ARP	60	1.3 is at BB
3	0.002	1.1	1.4	ARP	60	1.4 is at CC

A.3.3　システム開発

本システムは，以下のライブラリが必要となる.

- scapy[7]
- tabulate[8]
- manuf[9]

デバイス検知と Wi-Fi 接続時間による利用制限，閲覧管理するコンテンツのリスト登録，コンテンツ閲覧時間の管理のソースを表 A.3 に示す.

main.py は，デバイスの検知を自動で行い，表 A.4 のような表を出力する．一行ごとに検知したデバイスを表示し，デバイスごとに六つの情報を出力する．IP アドレスの欄は，デバイスの IP アドレスを表示する．MAC アドレスの欄は，デバイスの MAC アドレスを表示する．デバイス名の欄は，利用者が個別に設定するデバイス名が表示される．接続状況の欄は，デバイスが Wi-Fi に接続しているかを表示する．制限状態の欄は，Wi-Fi の接続時間による利用制限をするための管理対象機器に設定しているかを表示する．接続時間の欄は，一日のデバイスの Wi-Fi 接続時間を分単位で表示する.

接続時間管理

接続・切断の通知は自動的に行う．デバイスが接続（切断）すると，「MAC アドレス，デバイス名が接続（切断）しました」と通知する.

Wi-Fi 接続時間による利用制限は，コマンドを実行した後，制限したいデバイスの IP アドレスと制限した

表 A.3　各機能の起動コマンド

機能	ソース
デバイス検知と Wi-Fi 接続時間による利用制限	main.py
閲覧管理するコンテンツのリスト登録	save_domain_ip.py
コンテンツの閲覧時間の管理	calc_browsingtime.py

表 A.4　main.py の起動画面の例

IP アドレス	MAC アドレス	デバイス名	接続状況	制限状態	接続時間
192.168.1.1	00:0b:a2:a3:ef:f4	未登録	接続中	対象外	0
192.168.1.2	58:27:8c:3a:cd:f8	未登録	接続中	対象外	0
192.168.1.3	aa:2d:d5:f8:fd:97	未登録	接続中	対象外	0
192.168.1.10	38:78:62:7a:40:e1	未登録	接続中	対象外	0

い時間をコンソールから入力する．入力した時間を超えると，自動的に ARP スプーフィングを開始し，デバイスはインターネットを利用できなくなる．表 A.5 に利用制限の実行画面の例を示す．一行目で利用制限のコマンドを入力し，二行目で制限したいデバイスの IP アドレスを入力し，三行目で制限したい時間を分単位で入力する．

表 A.5　利用制限の例

コマンドを入力してください (終了するには 'exit' を入力): add
IP アドレスを入力してください: 192.168.1.10
制限時間を入力してください: 30

コンテンツアクセス制御

閲覧管理するコンテンツのリスト登録は，閲覧管理する Web サイトのドメインを入力する．

コンテンツ閲覧時間の管理は，閲覧管理したい Web サイトのドメインと制限したい時間を入力する．入力した時間を超えると，「IP アドレス X における example.com の閲覧時間が Y 分を超えました」と通知する．

A.3.4　評価

ネットワーク内にあるデバイスが正確に検知されているか調べる．ネットワーク内に以下の 6 デバイスを用意した状態でシステムを実行し，デバイスが検知されるかを検証する．

- Windows PC
- mac PC
- iPhone
- iPad
- Android
- Nintendo Switch

また，Web サイトの閲覧時間が正確に計測されているか調べる．システムを実行した状態で，以下の 5 つの Web サイトにアクセスして 10 分間閲覧する．システムで計測した時間と手動で閲覧した時間を比較して正確に計測されているかを検証する．

- Yahoo!JAPAN (https://www.yahoo.co.jp/)
- X (https://twitter.com/)
- CookPad (https://cookpad.com/)
- Steam (https://store.steampowered.com/)
- Discord (https://discord.com/)

表 A.6　デバイス検知結果

デバイス	結果
WindowsPC	検知
macPC	検知
iPhone	検知
iPad	検知
Android	検知
Nintendo Switch	検知

表 A.7　閲覧時間計測結果

Web サイト	**閲覧時間 (分)**	**計測時間 (分)**	**誤差 (分)**
Yahoo!JAPAN	10	10	0
X	10	10	0
CookPad	10	10	0
Steam	10	10	0
Discord	10	10	0

デバイスの動作結果を表 A.6 に示す．6 デバイスすべてが正常に検知された．また，検知以外の動作についても正常に行われていた．

サイトのアクセス結果を表 A.7 に示す．5 つのサイトすべてにおいて，目測による実際の閲覧時間とシステムによる自動計測時間の差は発生しなかった．

A.4　おわりに

本研究では，ARP スプーフィングを応用し，保護者が子どものインターネット利用を管理するホームセキュリティを開発した．しかし，本ホームセキュリティは Web サイトの閲覧を停止する機能が実装されていないため，今後は URL フィルタリングの実装を課題とする．

参考文献

[1] KDDI 株式会社，“コロナ禍で変化するスマートフォンの利用方法と、スマホ依存などへの影響を調査”，(https://news.kddi.com/kddi/corporate/newsrelease/2021/10/12/5468.html，2023 年 12 月参照)

[2] 総務省，“令和 2 年版 情報通信白書”，(https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/html/nd252110.html，2023 年 12 月参照)

[3] NTT ドコモ，“あんしんフィルター for docomo”，(https://www.docomo.ne.jp/service/anshin_filter，2023 年 1 月参照)

[4] CNET Japan，“親のかけたスマホ制限を自力で解除する子どもたち”，(https://japan.cnet.com/article/35160831/，2023 年 12 月参照)

[5] ITmedia，“ARP”，(https://atmarkit.itmedia.co.jp/aig/06network/arp.html，2023 年 1 月参照)

[6] 井窪，“ホームネットワークにおける全ホストを管理する Simple Home Security の開発”，2022 年度菊池研究室卒業論文，2023.

[7] Scapy，“Scapy”，(https://scapy.net/，2023 年 1 月参照)

[8] PyPI，“tabulate”，(https://pypi.org/project/tabulate/，2023 年 1 月参照)

[9] PyPI，“manuf”，(https://pypi.org/project/manuf/，2023 年 1 月参照)

[10] 北原，菊池，“ARP スプーフィング攻撃のリスク評価”，情報処理学会第 84 回全国大会，2022.

[11] 瀬川，中村，“仮想環境における L 2 ネットワークの脆弱性分析と ARP スプーフィング緩和手法の提案”，コンピュータセキュリティシンポジウム 2022，pp.540-545，2022.