# ブラウザが属するネットワークの情報を採取する Browser Scanner の提案

細井 理央† 高須 航† 山田 智隆† 武居 直樹†
石川 貴之† 高橋 和司‡ 安田 昂樹‡ 齋藤 孝道‡

†明治大学大学院
214-8571 神奈川県川崎市多摩区東三田 1-1-1

‡明治大学
214-8571 神奈川県川崎市多摩区東三田 1-1-1

**あらまし** HTML5を中心とした新たな技術の登場により，ブラウザ経由で動作中のホスト名やサーバアプリケーション名といったネットワークに関する情報を採取することが可能となった．本論文では，それらのネットワークの構成情報に加え，Webサイト閲覧者の属するネットワーク内のホストに関する情報やプロキシの有無の採取を可能とする，Browser Scannerを提案する．

# A Browser Scanner：Collecting Intranet Information

Rio Hosoi† Ko TAKASU† Naoki TAKEI† Tomotaka YAMADA†
Takayuki ISHIKAWA† Kazushi TAKAHASHI‡ Koki YASUDA‡ Takamichi SAITO‡

†Graduate School of Meiji University
1-1-1, Higashi-Mita, Tamaku, Kawasakishi, Kanagawa 214-8571, JAPAN
‡Meiji University
1-1-1, Higashi-Mita, Tamaku, Kawasakishi, Kanagawa 214-8571, JAPAN

**Abstract** HTML5 and new browser technologies can make us collect intranet information such as hostname or a name of server application on active host. In this paper, we propose a penetration test tool that a browser can collect information of active hosts or the existence of proxy over channel to the Web server.

## 1 はじめに

ブラウザ技術の進歩に伴い，Web サイトにアクセスするだけで Web サイト閲覧者の端末に関する情報を採取することが可能となった．UserAgent やハードディスク空き容量などに加え[1][2]，XMLHttpRequest2（XHR2）や Web Real-Time Communications（WebRTC）などの技術により，Web サイト閲覧者の使用するブラウザが属するネットワークの情報（以降，ネットワーク構成情報と呼ぶ）を採取することが可能になった．

採取可能なネットワーク構成情報には，動作中のホスト名やサーバアプリケーション名があるが，我々の研究により新たにホストのOS に関する情報[3]やルータの有無，プロキシの有無の情報も採取可能となった．

これら技術のネガティブな面として，企業組織のネットワークの規模や構成を推測できるので，クライアントへの攻撃やネットワークへの侵入などが行われる際の足がかり，もしくは，攻撃法の一つになる可能性がある．そのため，防衛する側の観点からも，ブラウザから Web サイトにアクセスするだけで採取されてしまうネットワーク構成情報を把握しておくことは，重要である．

そこで本論文では，現状の Web の技術を用いて，対象のブラウザが稼働する端末が属するネットワーク構成情報の採取を可能とする Browser Scanner を提案する. 本提案では，既存の採取法に加え，新たな採取法を示す．本提案である Browser Scanner は, ペネトレーションテストツールとしての利用を想定する．

# 2 関連技術

本章では，本提案において利用する技術について説明する．また，類似する既存のツールである BeEF について紹介する．

## 2.1 利用する技術

XHR2 [4]とは, クライアント側でスクリプトが実行されることにより，クライアント・サーバ間でリクエストやレスポンスなど，データ転送の機能を提供する API である．XMLHttpRequest（XHR）は，同一のオリジン[5]へのみリクエストが可能であったが，XHR2 は異なるオリジンへのリクエストを可能にさせる仕組みである Cross Origin Resource Sharing（CORS）[6]に対応している[7].

WebRTC[8]とは, ブラウザでリアルタイムコミュニケーションを可能にさせるためのフレームワークである．WebRTC は，ブラウザ間で Peer-to-Peer（P2P）通信を行うために用いられ，一般的に STUN サーバ[9]と TURN サーバ[10]が利用される. STUN サーバとは，P2P 通信を行う際に必要となる Network Address Translator（NAT）越えを行う過程で利用されるサーバである. NAT 越えとは，NAT 機器を用いているネットワーク同士が通信を行う際に，それぞれのプライベートネットワークにある端末間の通信を可能にさせる技術である. また TURN サーバとは, NAT の種類により NAT 越えが不可能である場合に利用されるサーバである．これらのサーバを利用し，プライベート IP アドレスやグローバル IP アドレス，ポート番号などの情報を持つ Session Description Protocol（SDP）[11]を通信相手と共有することで，P2P 通信の実現が可能となる．

## 2.2 BeEF ツール

本節では，本提案と類似する BeEF（The Browser Exploitation Framework）[12]について説明する．

BeEF とは，ブラウザにおけるペネトレーションテストツールである. BeEF では, Web サイト閲覧者が BeEF のスクリプトが埋め込まれたサイトにアクセスした際に，BeEF をインストールしてある Web サーバからブラウザを通してクライアントシステムに攻撃を行わせることで，ブラウザが稼働する端末が属するネットワーク内の情報など，以下に示す情報を採取することができる．

- ホスト名
- サーバアプリケーション名
- 存在するホストのIPアドレス

# 3 Browser Scanner

## 3.1 概要

提案する Browser Scanner は, ブラウザが稼働する端末が属するネットワークの構成情報を採取するために，ブラウザからネットワーク内をスキャンする仕組みである．我々は Browser Scanner を利用して，ネットワーク構成情報を採取する Web サイト http://www.saitolab.org/bscan/ （以降，Browser Scanner サイトと呼ぶ）を構築し運用している．本論文では，スキャンの対象とするブラウザが稼働する端末が属するネットワークのことを対象ネットワークと呼ぶ．また，Browser Scanner サイトと Browser Scanner サイトにより提供される JavaScript（以降，採取コードと呼ぶ）を総じて，Browser Scanner とする．

Browser Scanner サイトでは, Web サイト閲覧者のアクセスによりブラウザから送信されるグローバル IP アドレスと，採取コードをブラウザ上で実行させることで得られる情報を用いて，ネットワーク構成情報を採取している．この採取コードは，ネットワーク内のホストや STUN サーバにリクエストを送信する処理，および SDP から必要な情報を抽出する処理を行う．Browser Scanner サイトでは，以下のネットワーク構成情報を採取することができる．

- ホスト名

  対象ネットワーク内のクライアントで使用されているホストの名前を指す．Fully Qualified Domain Name（FQDN）がwww.example.comの場合，ホスト名はwwwである．ホスト名の例として，adminやnsなどがあげられる．

- サーバアプリケーション名

  対象ネットワーク内にあるサーバで，動作しているアプリケーションの名称を指す．サーバアプリケーション名の例として，ApacheやQNAP NASなどがあげられる．

- 対象ネットワーク内に存在するホストのIPアドレスとOSの情報

  対象ネットワーク内に存在する，IPアドレスが割り当てられているホストのIPアドレスとそのOSがWindowsか否かを指す．

- ルータの有無

  対象ネットワーク内に，Softbank社のYahoo!BB 光 ルータ（以降，Softbank ルータと呼ぶ）が存在するかどうかを指す．

- プロキシの有無

  スキャン対象のブラウザとBrowser Scannerサイトとの間にプロキシが存在するかどうかを指す．

## 3.2 Browser Scanner サイトの動作

本節では，Browser Scanner サイトの動作を図 1 により説明し，実際に Browser Scanner サイトで採取した際の結果画面を図２に示す．

図1のブラウザと Browser Scanner サイト間の実線の矢印は画面遷移を伴う通信を表し，破線の矢印は画面遷移を伴わない XHR2 による通信を表す．

(1) Web サイト閲覧者が Browser Scanner サイトに訪れると，採取コードを含むページのリクエストが Browser Scanner サイトへ送信される．この採取コードには，ネットワーク構成情報を採取し，Browser Scanner サイトへ送信する処理が記述されている．
(2) Browser Scanner サイトは，(1) のリクエストに含まれる以下の情報をデータベースに保存する．

- グローバル IP アドレス
- サーバの時刻

(3) Browser Scanner サイトは，(1) でブラウザからリクエストされた採取コードを含むページをブラウザに返信する．

(4) ブラウザは，グローバル IP アドレスとプライベート IP アドレスの情報を持つ SDP を取得するために，STUN サーバにリクエストを送信する．ここで利用する STUN サーバは，Mozilla が運営し，公開している STUN サーバである．

(5) ブラウザは，(3) で受け取った採取コードにより，STUN サーバから受け取った SDP からグローバル IP アドレスとプライベート IP アドレスを抽出する．なお SDP の中に，プライベート IP アドレスが複数存在する場合，すべてのプライベート IP アドレスを抽出する．

(6) ブラウザは，(5) で採取したプライベート IP アドレスを基に対象ネットワーク内の各 IP アドレスにリクエストを送信し，3.1 節に示すネットワーク構成情報（ただし，プロキシの有無を除く）を採取する．

(7) ブラウザは，XHR2 を利用して (5) と (6) で採取した結果を Browser Scanner サイトへ送信する．Browser Scanner サイトは，3.3 節で示す採取方法でプロキシの有無を判定し，受け取った結果に加えてデータベースに保存する．

(8) Browser Scanner サイトは，(7) でデータベースに保存したネットワーク構成情報をデータベースから読み出し，ブラウザに送信する．

(9) ブラウザは，Browser Scanner サイトから受信したネットワーク構成情報を採取結果として表示する．

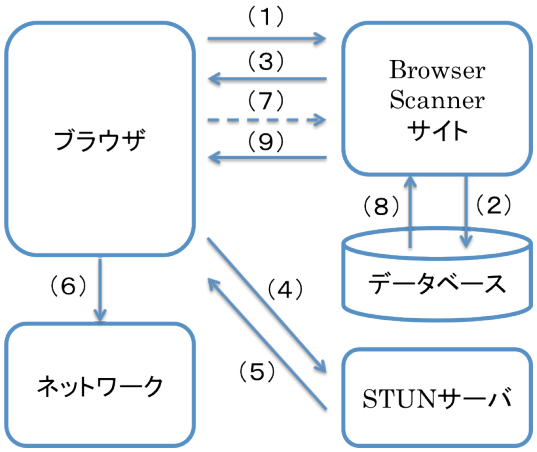

図 1 Browser Scanner サイトの動作

## 3.3 ネットワーク構成情報の採取

本節では，3.1 節で述べたネットワーク構成情報の採取方法を示す．ただし，ネットワークの構成によっては，採取できない場合がある．

- ホスト名

  一般的に広く利用されているホスト名のリストをあらかじめ用意する．ブラウザにより実行される採取コードは，このリスト内にあるホスト名の名前解決を行うためにXHR2を用いてDNSに問い合わせを行う．問い合わせに対するDNSからの応答が，Browser Scanner サイトで指定した時間以内に返ってくる場合は該当するホスト名が存在し，指定した時間を超えた場合は該当するホスト名が存在していないことがわかる．

- サーバアプリケーション名

  端末にアプリケーションをインストールすると，特定のディレクトリに当該アプリケーションのファイル名を持つ画像ファイルが配置される場合がある．ブラウザにより実行される採取コードは，この画像ファイルに問い合わせを行い，問い合わせに対するエラー応答の有無

により，サーバ上で動作中のアプリケーションがわかる．ただし，インストールされているアプリケーションが画像ファイルを配置しない場合や，該当の画像ファイルが既に削除されていた場合，正しく採取することができない．

- 対象ネットワーク内に存在するホストのIPアドレスとOSの情報

  文献[3]では，3.2節の（5）で受け取ったSDPに含まれるプライベートIPアドレスを用いて，ホストの検出が可能であることを示した．Webサイト閲覧者のプライベートIPアドレスが192.168.10.5の場合，192.168.10.1から192.168.10.254までのIPアドレス宛に，XHR2を用いて445番ポートへリクエストを送信する．エラー応答が返される場合はホストが存在し，返されない場合はホストが存在しないので，ホストの検出が可能である．

  また，OSがWindows，Linux，Max OS X, Android, iOSであるホストにおいて，Windowsであるか，Windows以外であるかを識別する手法も示した．検出した各ホストのIPアドレス宛に，446番ポートへリクエストを送信する．エラー応答が返される場合は，ホストのOSはWindows以外であり，返されない場合はWindowsであることがわかる．

- ルータの有無

  Softbankルータには，特定のIPアドレス（172.16.255.254）が設定されている[13]．そこで，このIPアドレスに問い合わせを行うことで，ネットワーク内にSoftbankルータが存在するかどうかがわかる．ただし，ネットワーク内で利用されているプライベートIPアドレスがクラスBの172.16.0.0/12の場合などは，正しく採取できない可能性がある．

- プロキシの有無

  3.2節の（2）で採取したグローバルIPアドレスと3.2節の（4）で受け取ったSDPに含まれるWebサイト閲覧者のグローバルIPアドレスを用いることで，プロキシの有無がわかる．採取した2つのグローバルIPアドレスが同じであればプロキシは存在せず，異なればプロキシが存在することがわかる．

Browser Scanner サイトの出力画面を図 2 に示す．

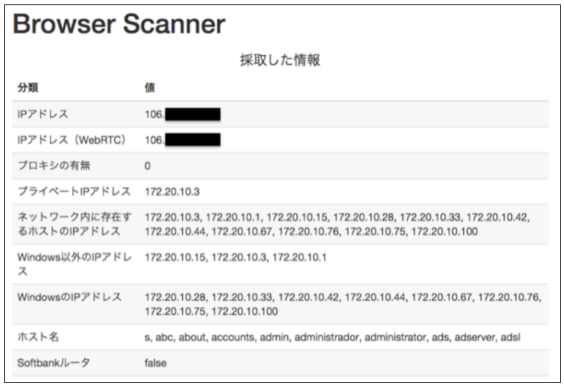


図 2 採取結果画面の例（一部省略）

図 2 の「IP アドレス」の項目には，3.2 節の（2）で述べたグローバル IP アドレスを，「IP アドレス（WebRTC)」の項目には，3.2 節の（5）で述べた WebRTC により得られるグローバル IP アドレスを表示している．図 2 の例では，これら 2 つの IP アドレスが一致しており，ブラウザと Browser Scanner サイトとの間にはプロキシが存在しないと判断できるので，「プロキシの有無」の項目には，0 と表示している．なお，「プロキシの有無」では，プロキシがある場合は 1 を，ない場合は 0 を表示する．

「プライベート IP アドレス」の項目には，3.2 節の（5）で述べた WebRTC により得られるプライベート IP アドレスを表示している．そして，このプライベート IP アドレスを利用して得られる，ネットワーク内に存在するホストの IP アドレス，Windows 以外の IP アドレス, Windows の IP アドレスをそれ

ぞれ表示している．

「ホスト名」の項目には，DNS へ問い合わせることで採取したホスト名を表示している．

「Softbank ルータ」の項目には，Softbank ルータの有無を表示しており，ネットワーク内に存在する場合は true を，存在しない場合は false を表示する．なお，本実験においてサーバアプリケーション名の採取結果は，スペースの都合上載せていない．

# 4 考察

本章では，本提案手法の採取コードの利用とネットワーク構成情報の採取への対策手法について説明する．

## 4.1 採取コードの利用

Browser Scanner サイトの採取コードは JavaScript で記述されたページの一部として提供されるので，ダウンロードとインストールを行う必要がない．そのため，Web サーバは，Web サイト閲覧者に気づかれずにネットワーク構成情報を採取することができる．また，第三者として提供するスクリプトファイルとしても利用できる．

## 4.2 スキャンへの対策

本節では，主要な 3 つのブラウザ（Firefox，Google Chrome，Internet Explorer）における，スキャンへの対策について説明する．

### 4.2.1 JavaScript の無効化

Browser Scanner サイトの採取コードは，JavaScript で実装されているので，JavaScript を無効化することでスキャンができなくなる．

主要な 3 つのブラウザでは，各ブラウザの設定により，JavaScript を無効化することができる．

また，Firefox と Google Chrome では，JavaScript の実行を妨げる拡張機能として，それぞれ NoScript[14]と ScriptBlock[15]がある．NoScript は，ホワイトリストを用いており，信頼のおける Web サイトでのみ，JavaScript，Java，Flash やプラグインの実行を許可する．ScriptBlock は，NoScript と同様に，ホワイトリストにある Web サイトにのみ，JavaScript，iframe やプラグインの実行を許可する拡張機能である．これにより，主要な 3 つのブラウザにおいて，JavaScript の無効化により対策を行うことは可能である．

### 4.2.2 WebRTC の無効化

WebRTC を無効化することで，ネットワーク内のクライアントのスキャンを実施できなくすることが可能となる．

Firefox（バージョン 39 時点）では，ブラウザの設定で，WebRTC の機能を無効にすることができる．Google Chrome（バージョン 44 時点）では，ブラウザの設定で WebRTC の機能を無効にすることは出来ないが，拡張機能として提供されている WebRTC Block[16]を用いることで WebRTC の機能を無効化することができる．また Internet Explorer では，デフォルトで WebRTC に対応していない[17]ため，現状対策を行う必要はない．これにより，主要な 3 つのブラウザにおいて，WebRTC の無効化により対策を行うことは可能である．

# 5 まとめ

本論文では，Browser Scanner を提案した．これは，Web サイト閲覧者が Web サイトにアクセスするだけで，Web サーバがブラウザの属するネットワーク構成情報の採取を実現する．

今後の課題として，ネットワーク構成情報の採取における処理時間の短縮，より正確なネットワーク構成情報の採取があげられる．

ネットワーク構成情報は，利用環境によっ

ては変化しにくいと考えられるので, Web サイト閲覧者の端末を識別するための特徴として利用できる可能性がある.

## 参考文献


[1] Ko Takasu, Takamichi Saito, Tomotaka Yamada, Takayuki Ishikawa, 2015, A Survey of Hardware Features in Modern Browsers: 2015 Edition, in Proc. of the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS) 2015

[2] 磯 侑斗, 桐生 直輝, 塚本 耕司, 高須 航, 山田 智隆, 武居 直樹, 齋藤 孝道, 2014, Web Browser Fingerprint を採取する Web サイトの構築と採集データの分析, コンピュータセキュリティシンポジウム 2014 論文集 CD-ROM p.378-p.385

[3] 細井 理央, 磯 侑斗, 桐生 直輝, 塚本 耕司, 高須 航, 山田 智隆, 武居 直樹, 石川 貴之, 齋藤 孝道, XHR2 を用いた Web サイト閲覧者の属するネットワーク構成情報の採取法の提案, 第 77 回情報処理学会全国大会

[4] XMLHttpRequest Living Standard https://xhr.spec.whatwg.org

[5] The Web Origin Concept https://tools.ietf.org/html/rfc6454

[6] Cross-Origin Resource Sharing http://www.w3.org/TR/cors/

[7] 齋藤孝道, マスタリング TCP/IP 情報セキュリティ編, オーム社, 2013

[8] Web Real-Time Communication Use Cases and Requirements https://tools.ietf.org/html/rfc7478

[9] Session Traversal Utilities for NAT https://tools.ietf.org/html/rfc5389

[10] Traversal Using Relays around NAT https://tools.ietf.org/html/rfc5766

[11] SDP: Session Description Protocol https://tools.ietf.org/html/rfc4566

[12] BeEF http://beefproject.com/

[13] Yahoo!BB 光 ルーター機能の設定方法 http://ybb.softbank.jp/support/connect/hikari/router/

[14] NoScript https://noscript.net/

[15] ScriptBlock https://chrome.google.com/webstore/detail/scriptblock/hcdjknjpbnhdoabbngpmfekaecnpajba/

[16] WebRTC Block https://chrome.google.com/webstore/detail/webrtc-block/nphkkbaidamjmhfanlpblblcadhfbkdm

[17] WebRTC http://www.webrtc.org/