# DNSリバインディングにどう立ち向かうべきか

EGセキュアソリューションズ株式会社
取締役CTO 徳丸浩

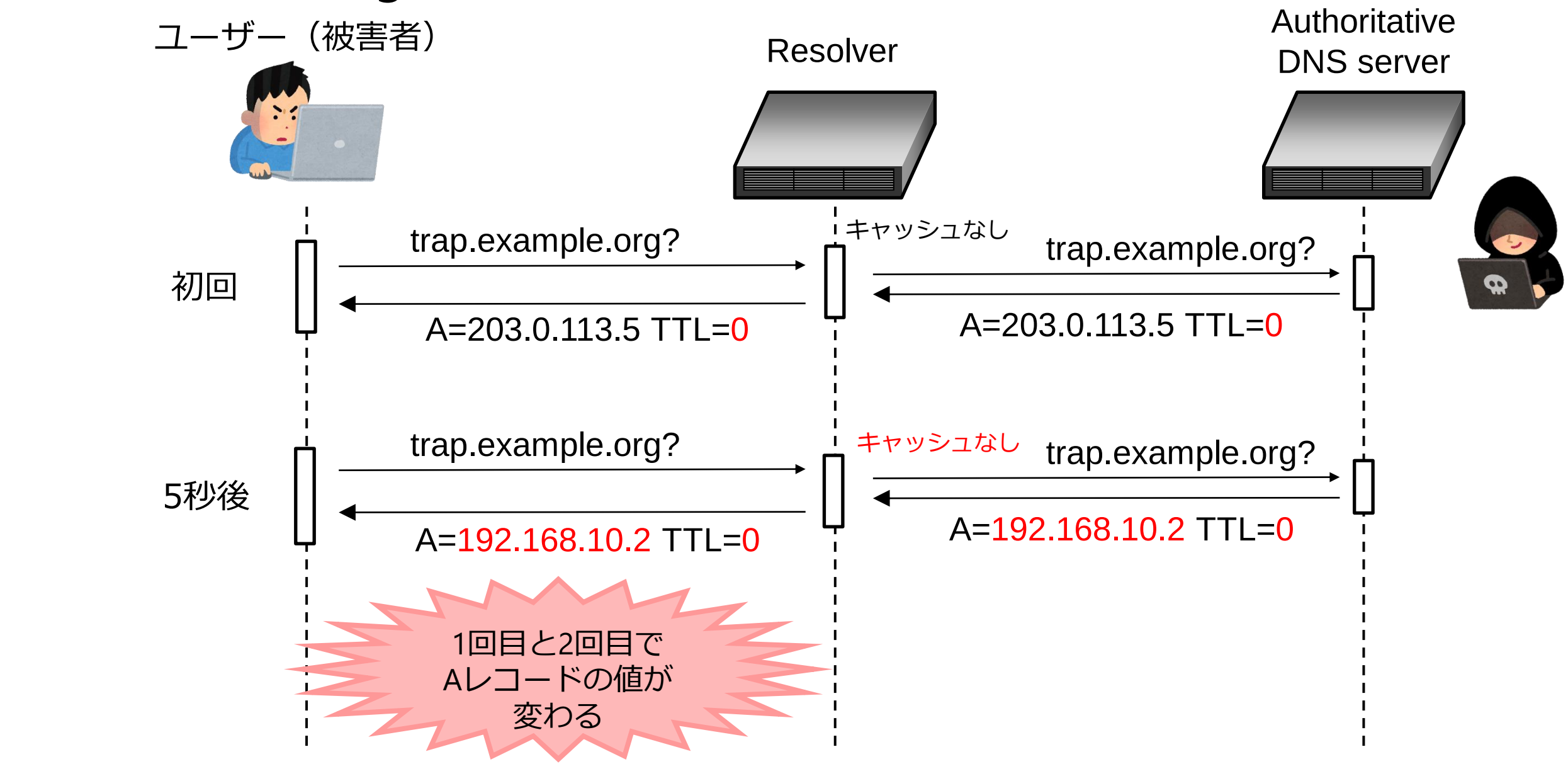
# 本日お伝えしたいこと

- 再び注目を集めるDNS Rebinding
- DNS Rebindingはどのような攻撃か
- DNS Rebindingの対策
- 結局DNS Rebindingはどうすべきか

# DNS Rebindingとは

- 「時間差」を用いた攻撃の一種
- 複数回のDNSクエリに対して異なるIPアドレスを返すことにより、ネットワーク的に到達できないサーバーに対して、ブラウザ経由で攻撃する
- DNSのキャッシュ時間（TTL=Time to Live）を非常に短く（0秒～5秒程度）設定して攻撃する

# DNS Rebindingの様子



# 再び注目を集めるDNS Rebinding

- DNS Rebindingは2006年に指摘されている「歴史のある」攻撃手法
- 1996年にはDNS Spoofingという呼称でJavaアプレットに問題提起され、Sunが対策を行う
- 古典的な攻撃
  - ルータやファイアウォールへの攻撃
  - イントラネット内の端末やサーバーへの攻撃
  - ガラケーの「かんたんログイン」に対する指摘（2009年11月）
- 最近再注目されている様子
  - IoT機器や家庭用IT機器に対する攻撃手法
  - 開発者のパソコン上で動く開発ツールへの攻撃(Ruby on Rails等)
  - SSRF(Server-side Request Forgery)との合わせ技

# ケータイtwitter(twtr.jp)においてDNS Rebinding脆弱性

- twtr.jpはフィーチャーフォン（ガラケー）向けtwitterフロントエンド
- twtr.jpの「かんたんログイン」機能にDNS Rebinding脆弱性があることをIPAに届け出
  - 2010/01/14 深夜  脆弱性の確認完了
  - 2010/01/15 11:37 デジタルガレージ社への通報
  - 2010/01/15 12:52 IPAへの届出 取扱い番号 IPA#04364080 として受信される
  - 2010/01/15 19:18 IPAより届出受理および取り扱い開始の連絡
  - 2010/01/15 21:51 デジタルガレージ社の担当者より修正済みの返信。手元でも修正を確認。
  - 2010/01/19 10:53 IPAより修正完了の連絡。その後取り扱い終了となる。

参考: https://www.tokumaru.org/d/20100222.html#p01

# Rails 6にDNSリバインディング攻撃防止機能が追加された（翻訳）

Rails 6にはDNSリバインディング攻撃から保護する機能が#33145で追加されました。この機能はdevelopment環境ではデフォルトで有効になっており、他の環境でもオプションで有効にできます。

## DNSリバインディング攻撃とは

DNSリバインディング攻撃は、攻撃者が悪意のあるクライアントサイドスクリプトを仕込んだWebページを用いて標的ネットワークに侵入し、そのネットワーク内のブラウザをプロキシとしてネットワーク内の他のデバイスを攻撃することを指す。

## Railsアプリで生じる影響

ローカル実行されているRailsアプリケーションに対して、攻撃者がDNSリバインディングを用いてリモートコード実行（RCE）する可能性があります。攻撃者が的確にアプローチすると、ローカルENV（環境変数）情報や、ローカルRailsアプリの全情報にアクセスできてしまうことがあります。



https://techracho.bpsinc.jp/hachi8833/2020_02_05/83154 より引用

# Google Nest WiFiのマニュアルから…

## DNS リバインディングに対する保護機能

Google Nest スピーカー、ホームメディア サーバー、IoT（モノのインターネット）デバイスのような接続されたデバイスをホストするホーム ネットワークは、DNS リバインディングと呼ばれる攻撃を受けるおそれがあります。Google Wifi では、この種の攻撃を防止するため、DNS リバインディングに対する保護機能で、公開ドメインにプライベート IP アドレス範囲を使用されないようブロックすることができます。この機能はデフォルトで有効になっています。

ただし、サービスによっては DNS リバインディングが許可されていないと機能しないものがあります。ローカル ネットワークで DNS リバインディングを許可する場合は、ご自身の責任でカスタム DNS サーバーを設定することで、DNS リバインディングに対する保護機能を無効にすることができます。

https://support.google.com/googlenest/answer/9144137?hl=ja

# EC2上でDNS RebindingによるSSRF攻撃可能性を検証した

AWS EC2環境でのDNS Rebindingについて検証したので紹介します。
まずは、「前回までのおさらい」です。先日以下の記事でSSRF攻撃およびSSRF脆弱性について紹介しました。

- SSRF(Server Side Request Forgery)徹底入門

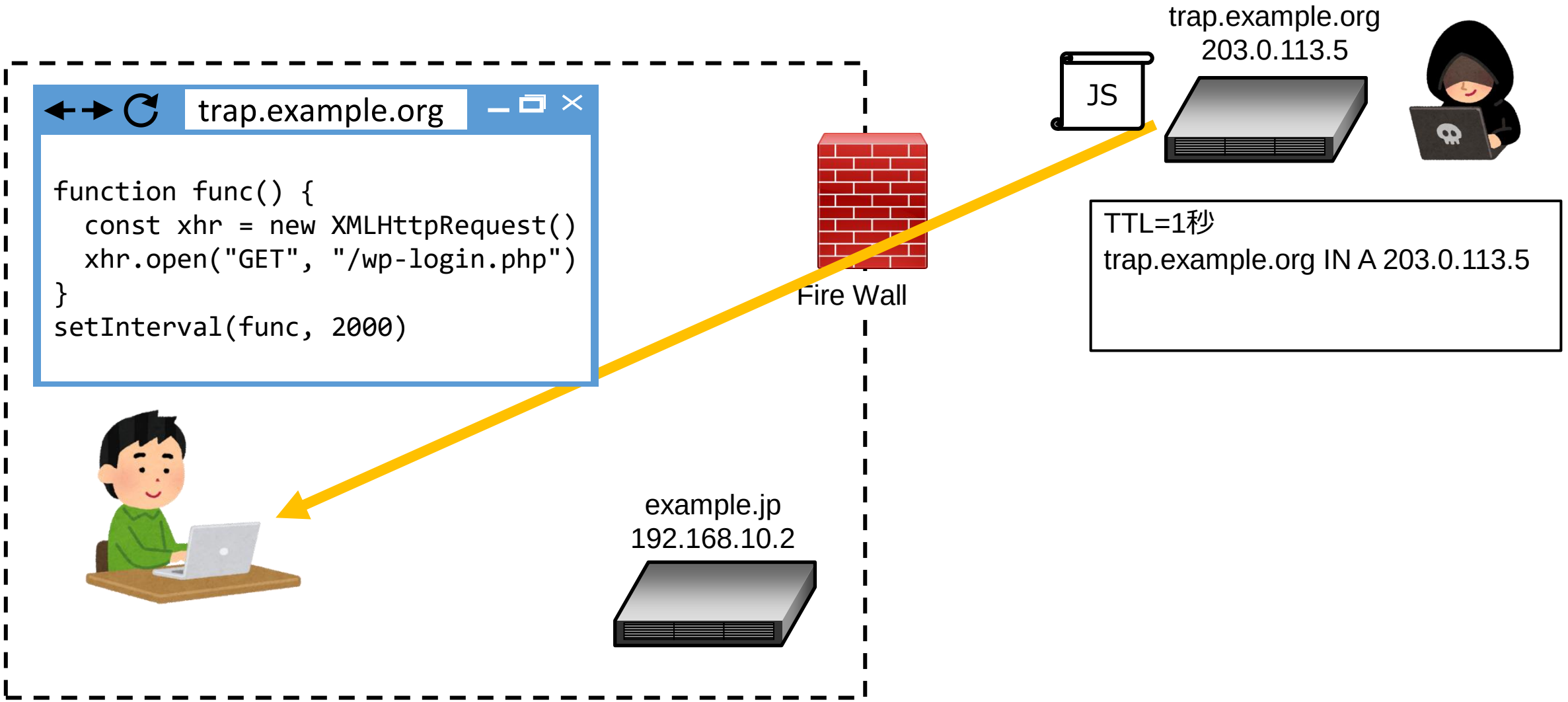
この記事の中で、以下のように紹介しました。

> ホスト名からIPアドレスを求める際にも以下の問題が発生します。
>
> - DNSサーバーが複数のIPアドレスを返す場合の処理の漏れ
> - IPアドレスの表記の多様性（参考記事）
> - IPアドレスチェックとHTTPリクエストのタイミングの差を悪用した攻撃（TOCTOU脆弱性）
> - リクエスト先のWebサーバーが、攻撃対象サーバーにリダイレクトする
>
> 上記のTOCTOU(Time of check to time of use)問題は、DNSの名前解決の文脈ではDNS Rebindingとも呼ばれます。

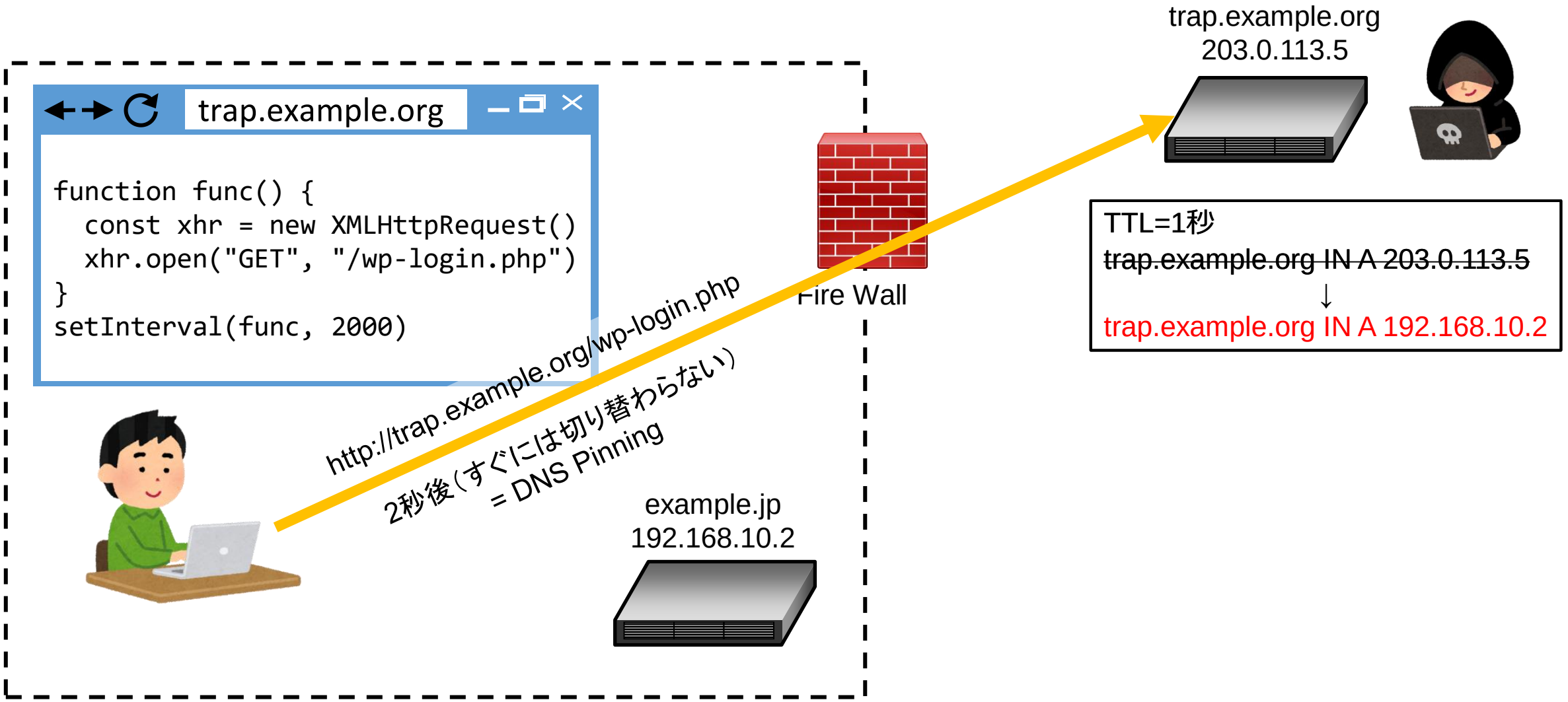
DNS Rebinding攻撃についてはインフラ側で対策が取られている場合もあります。そこで、EC2上のウェブサイトにて、DNS RebindingによるSSRF攻撃が可能か検証してみました。

# デモ：イントラネット内のWordPressサイトに侵入

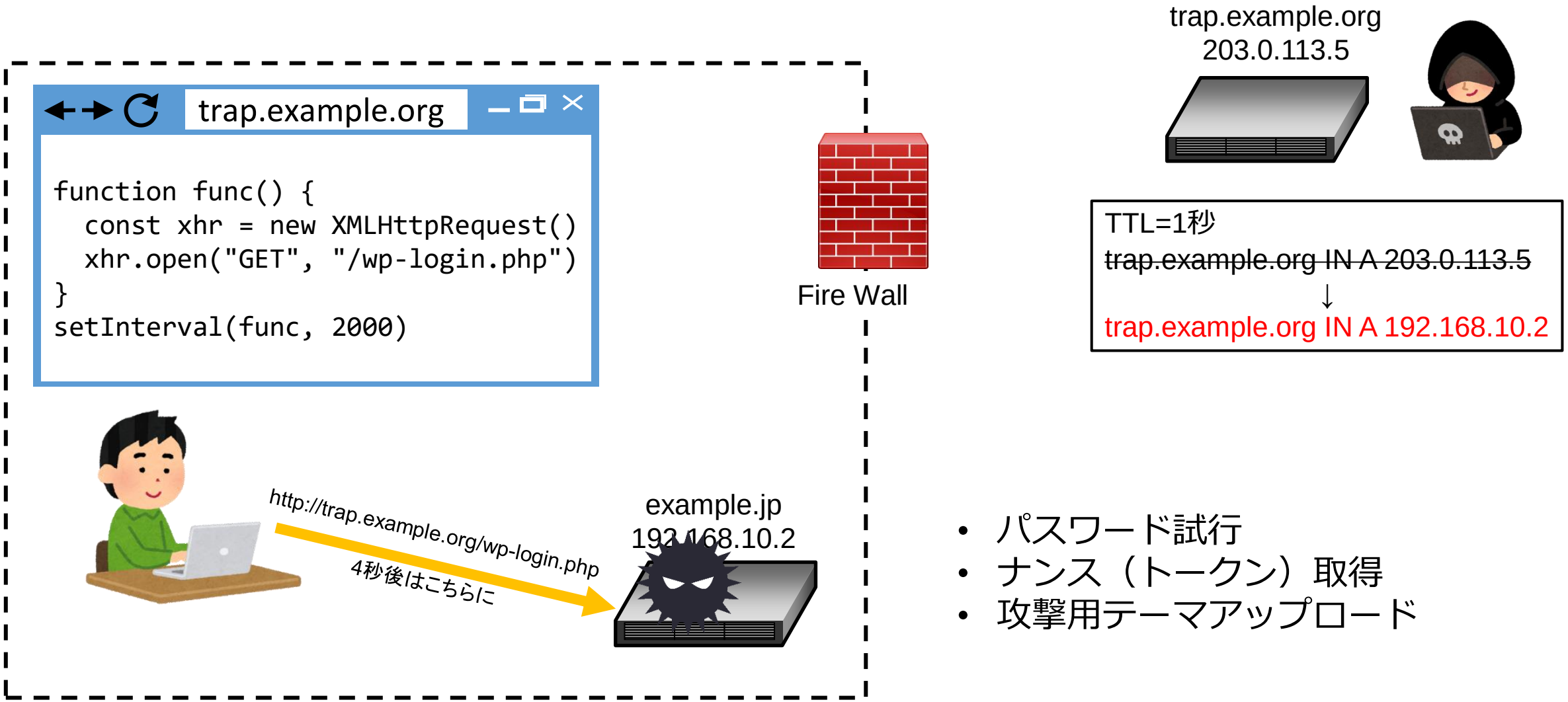
# DNS Rebindingによるイントラネット内サイトへの攻撃



# DNS Rebindingによるイントラネット内サイトへの攻撃

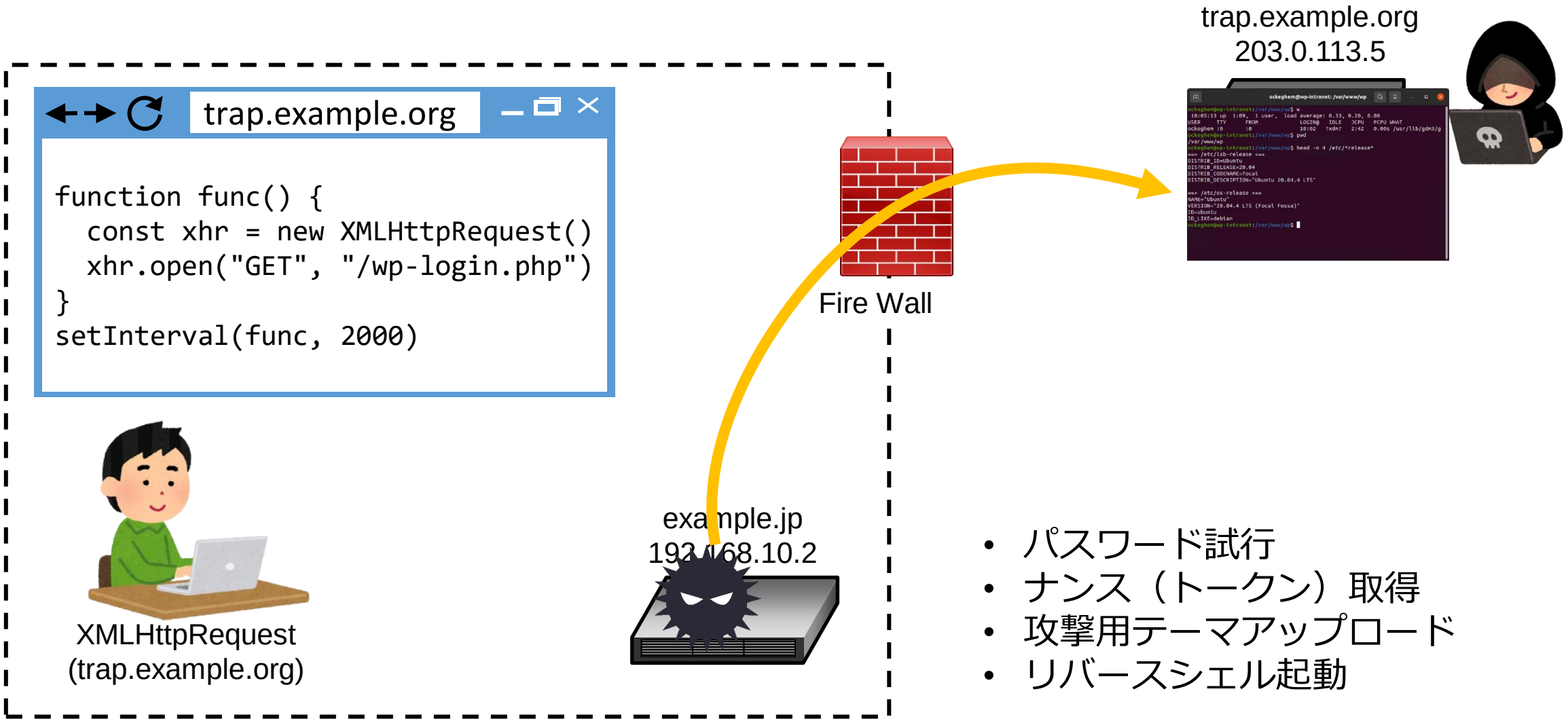


# DNS Rebindingによるイントラネット内サイトへの攻撃




- パスワード試行
- ナンス（トークン）取得
- 攻撃用テーマアップロード

# DNS Rebindingによるイントラネット内サイトへの攻撃

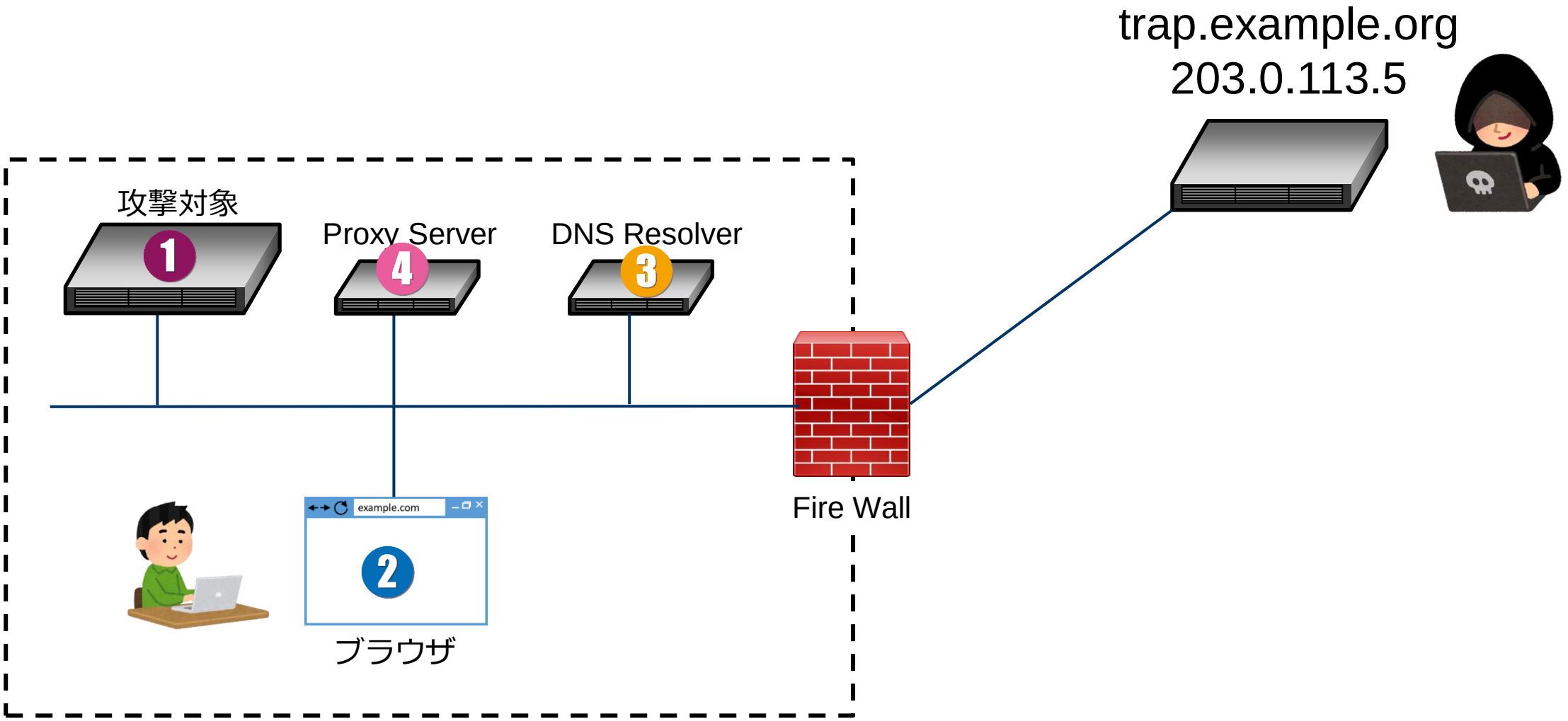



- パスワード試行
- ナンス（トークン）取得
- 攻撃用テーマアップロード
- リバースシェル起動

# Demo

# 対策編

# DNS Rebindingの対策できる箇所



# 対策1: 攻撃対象での対策

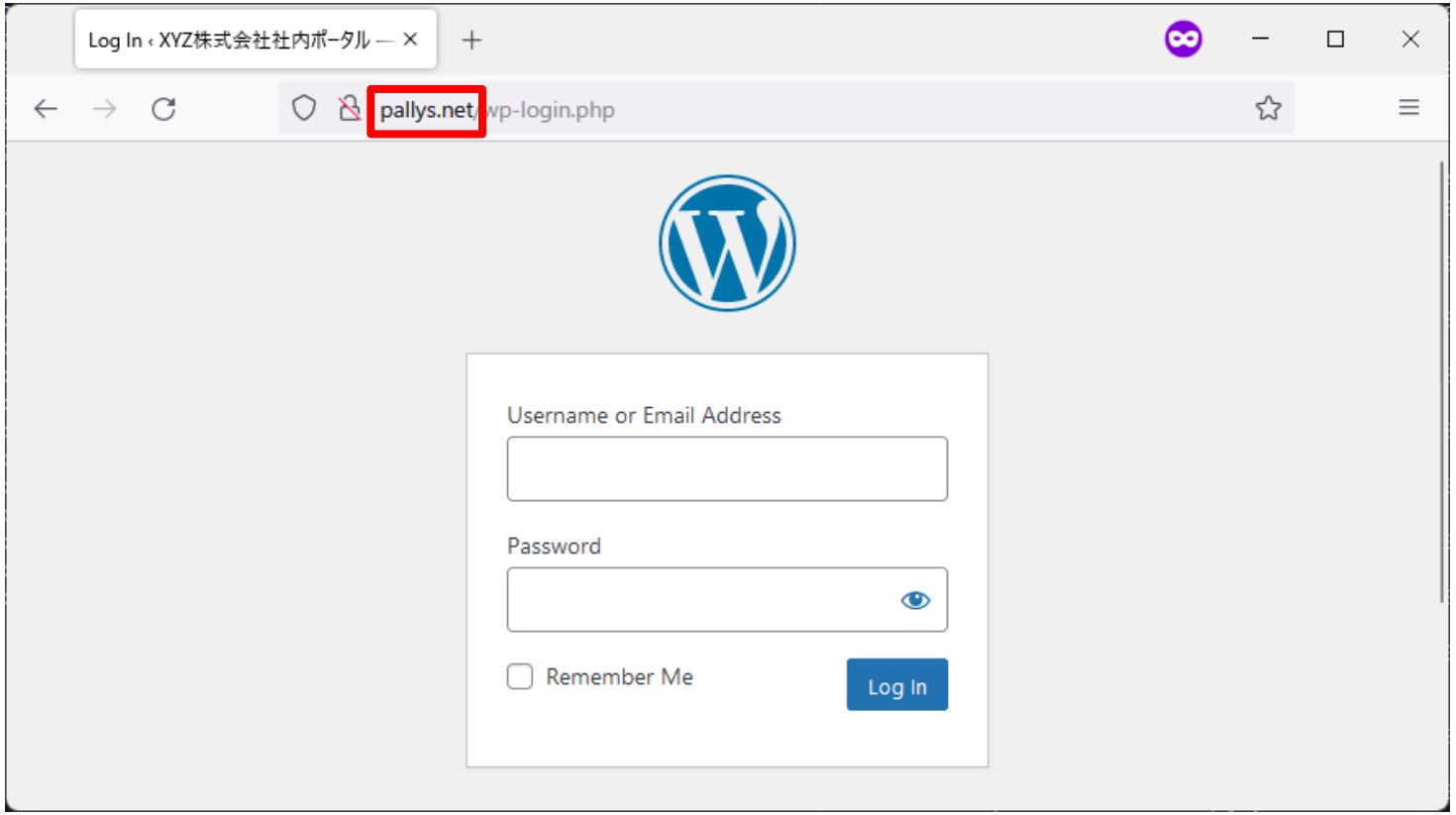
# DNS Rebinding対策

- DNS Rebinding攻撃はXSSやCSRFとは異なり、HTTPリクエストのHostヘッダが罠サイトのホスト名になる
  - Cookie等は飛ばずセッションが乗っ取られるわけではない（重要）
- DNS Rebinding対策としては以下が有効
  - Hostヘッダのチェックを行う または
  - 認証機能を設け安全なパスワードを設定する
- 古い文献では、Hostヘッダを改変できる可能性を懸念するものがあるが、現在ではHostヘッダはForbidden header nameとして改変が仕様として禁止されている

# ダミーのバーチャルホストによる対策（対策前）

```
# Before
server {
        listen 80;
        server_name 0.0.0.0;
        …
```

全てのホストを受付



# ダミーのバーチャルホストによる対策（対策後）

```
# After (ダミーのデフォルトサーバーを追加)
server {
        listen 80 default_server;
        server_name '_';
        …
}
server {
        listen 80;
        ## server_name 0.0.0.0;
        server_name wp.example.jp;
        …
```

ダミーのデフォルトサーバー

ホスト名を明示

# 対策2: ブラウザでの対策

# DNS Pinning

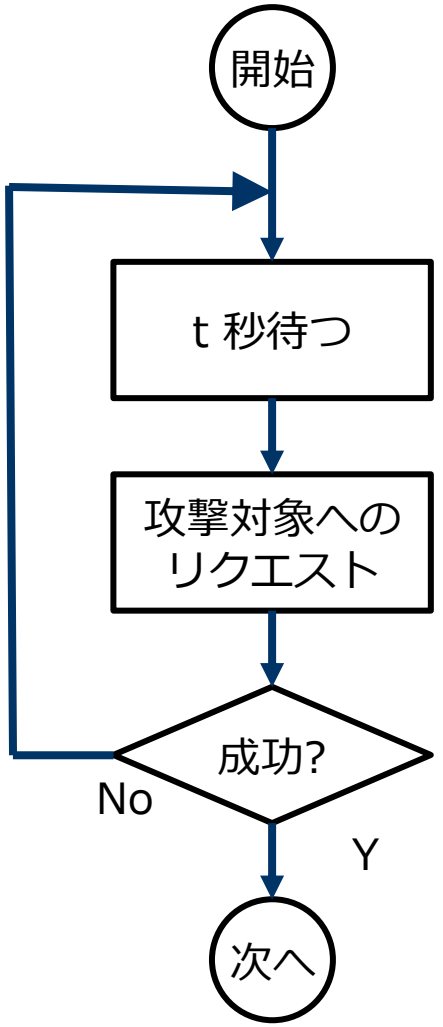
- ブラウザでのDNS Rebinding対策としては、DNS Pinningが一般的
- DNS Pinningはリゾルバの結果を一定時間ブラウザが保持すること
- 主要ブラウザのDNS Pinningの保持時間（実測値）

| Google Chrome | Firefox | Safari | IE |
|---|---|---|---|
| 1分程度 | 1秒～70秒 | 15秒～30秒 | 無期限? |

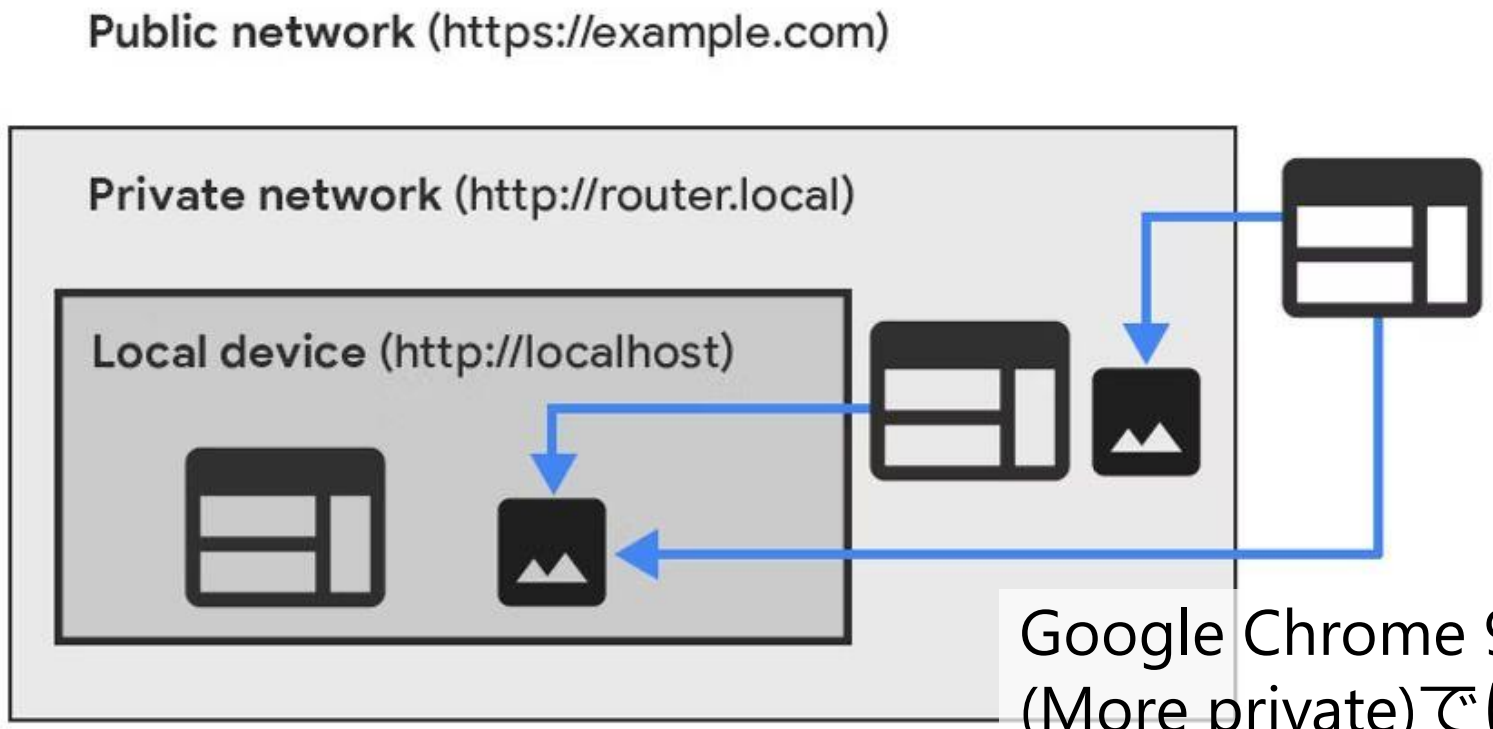
- 以前はDNS Pinningの期間はもっと長かったが、最近はCDN利用のためDNSのTTLが短くなっており、DNS Pinningの期間も短くなっている

# Anti DNS Pinning

- ブラウザのDNS Pinningを回避するテクニックとして、古来、「一度アクセスをわざと失敗される」方法が知られてる（金床本など）
  - 一度当該ポートを閉じるなど
  - 実験の範囲では、404を返すことでも効果あり
- 右のフローチャートでよいことになるが…
- Nginxだと t = 10 等にすると無限ループになる
- DNS Pinningが固いのかと思いきや
- Nginx のKeepAliveのデフォルト値が70秒なので、無限にKeepAliveし続けるためだった
- KeepAliveを無効にすれば前ページの結果となる



# Google Chrome での最近の対策 "More private" という概念




Google Chrome 94以降では、青矢印の遷移(More private)では、HTTPSが要求される
→ 証明書エラーになりDNS Rebindingできない

https://developer.chrome.com/blog/private-network-access-preflight より引用

Google Chrome のPrivate-network Accessの保護機能は強力だが、
PROXY経由でのアクセスには効果がない

# 対策3: PROXYサーバーでの対策

# PROXYサーバーでのDNS Rebinding対策方針

- Squidの場合以下の二種類の対策がとれる
  - IPアドレスによるアクセス制御
  - DNS Pinning
- IPアドレスによるアクセス制御例

```
acl localnet dst 192.168.0.0/24
http_access deny localnet
```

- 特定ドメインのみローカルネットワークを許可する例

```
acl example dstdomain .example.jp
acl localnet dst 192.168.0.0/24
http_access allow example
http_access deny localnet
```

# SquidのDNS Pinning

- デフォルトで以下の設定になっている

| ディレクティブ | 意味 | デフォルト値 |
|---|---|---|
| positive_dns_ttl | DNSキャッシュの上限 | 6時間 |
| negative_dns_ttl | DNSキャッシュの下限 | 1分 |

- DNS側のTTLに関わらず、TTLは1分～6時間の値に丸められる
- negative_dns_ttl を大きくすると、DNS Pinning相当のことができるが、DNSクエリの失敗も長時間保持される
- 結論としては、これらはいじらずに、IPアドレスによるアクセス制御を用いるがよい

# 対策4: リゾルバでの対策

# 著名リゾルバのDNS Rebinding対応

| | 0秒のTTL | TTL下限値の変更 | TTL上限 | 拒否リストの設定 |
|---|---|---|---|---|
| bind9 | 許容 | min-cache-ttl | 90 | deny-answer-addresses |
| unbound | 許容 | cache-min-ttl | 無制限? | private-address |
| dnsmasq | 許容 | min-cache-ttl | 3600 | stop-dns-rebind |
| PowerDNS recursor | TTL>=1 | minimum-ttl-override | 無制限? | Luaスクリプトで可能 |
| KNOT DNS resolver | TTL>=5 | cache.min_ttl() | 無制限? | modules.load('rebinding < iterate') |
| 1.1.1.1 | 許容 | - | ? | - |
| 8.8.8.8 | 許容 | - | ? | - |

主要リゾルバは
DNS Rebinding対
策機能がある

# BIND 9 Administrator Reference Manual 8.2.16.17. より

Note that this is not really an attack on the DNS per se. In fact, there is nothing wrong with having an “external” name mapped to an “internal” IP address or domain name from the DNS point of view; it might actually be provided for a legitimate purpose, such as for debugging. As long as the mapping is provided by the correct owner, it either is not possible or does not make sense to detect whether the intent of the mapping is legitimate within the DNS. The “rebinding” attack must primarily be protected at the application that uses the DNS. For a large site, however, it may be difficult to protect all possible applications at once. This filtering feature is provided only to help such an operational environment; turning it on is generally discouraged unless there is no other choice and the attack is a real threat to applications.

これは、実際にはDNSそのものに対する攻撃ではないことに注意してください。実際、DNSの観点からは、「外部」の名前が「内部」のIPアドレスやドメイン名に マップされることは何の問題もありません。マッピングが正しい所有者によって提供されている限り、DNS内でマッピングの意図が正当であるかどうかを検出することは不可能であるか、意味がないかのどちらかです。リバインディング攻撃は、主にDNSを使用するアプリケーションで保護されなければなりません。しかし、大規模なサイトでは、すべてのアプリケーションを一度に保護することは困難な場合があります。このフィルタリング機能は、そのような運用環境を支援するためにのみ提供されています。他に選択肢がなく、攻撃がアプリケーションにとって本当に脅威とならない限り、この機能をオンにすることは一般に推奨されません。

# 結局DNS Rebindingはどうすべきか

- ブラウザ、PROXY、リゾルバでDNS Rebindingの対策機能はあるがいずれも完全な対策にはならない
- 閉域といえども、インターネット公開サイト同様に対策をしておけば影響ない
  - 閉域内のサーバーでも認証ちゃんとやる
- ウェブサーバーの設定によりホストヘッダを確認するとよい
  - DNS Rebindingの簡単で安全な対策
  - IPアドレス直打ちの総当たり的な攻撃にも効果あり

# ご清聴ありがとうございました
# Any Questions?