*Article*

# Contramedidas em navegador para ataques de acesso remoto do tipo dns em sistemas unix e windows

**Pedro Bernardi Alves [1], Gabriel Rodrigues Falcade[2]**

[1] Engenharia de Computação; (Pedro Bernardi) pedroalves@alunos.utfpr.edu.br
[2] Tecnologia em Análise e Desenvolvimento de Sistemase; (Gabriel Falcade) falcade@alunos.utfpr.edu.br
* Correspondente: pedroalves@alunos.utfpr.edu.br;

**Resumo:** Na comunidade de segurança houve um entendimento crescente, de que falhas que permitam ataque de DNS rebinding podem representar um grupo muito maior de vulnerabilidades do que as pessoas reconheciam anteriormente. Apesar de relevante, essas falhas têm um histórico de serem descartados pelos desenvolvedores e, muitas vezes, são deixados como um problema não resolvido. O uso de uma extensão para navegador poderia auxiliar como contramedidas a esses ataques. Este trabalho tem por objetivo contribuir para que ataques de DNS rebinding sejam bloqueados facilitando assim a segurança do usuário. Pretende-se investigar como a extensão atua. Em termos de validação, através da metodologia lean, serão conduzidos, testes em cenários específicos. Foi elaborado uma extensão chamada RebindBlock, que realizou com sucesso o bloqueio de acesso remoto.

**Palavras-Chave:** DNS Rebinding; Navegador; Ataque Remoto;

## 1. Introdução

Em 30 de novembro de 2017 foi reportada uma vulnerabilidade na aplicação Transmission [4]. Segundo o pesquisador Tavis Ormandy, o qual faz parte do "Project Zero" da Google, a falha encontrava-se sobre o sistema de gerenciamento remoto do Transmission, no qual os usuários poderiam gerenciar o programa remotamente pelo navegador.

Esse cliente bittorrent utiliza uma arquitetura cliente/servidor, a interface do utilizador é o cliente que comunica com o daemon utilizando pedidos do tipo JSON RPC. Como em todos os esquemas de HTTP RPC como este, qualquer site pode enviar pedidos ao daemon que escuta no localhost com XMLHttpRequest(). Em seu perfeito funcionamento, eles serão ignorados porque os clientes têm de provar que podem ler e definir um cabeçalho específico, X-Transmission-Session-Id. Entretanto, isso não funcionou devido a um ataque chamado "DNS rebinding". Qualquer site poderia simplesmente criar um dns com o qual está autorizado a se comunicar e, em seguida, resolver para localhost.[5]

Esse é apenas um dos casos de ataques do tipo. O pesquisador, Tavis Ormandy, descobriu vulnerabilidades de religação de DNS no mecanismo de atualização para videogames da Blizzard[6], e pesquisadores desenvolveram pesquisas na qual relataram bugs em várias carteiras Ethereum, potencialmente expondo a criptomoeda de diversas pessoas[7]. Evidências recentes sugerem vulnerabilidades em dispositivos de IOT como descrito no trabalho [8].

Durante um ataque de "DNS Rebinding", um invasor ignora o mecanismo de segurança de o firewall no roteador e se comunica interativamente com dispositivos em seu local rede usando o navegador da vítima. Isto é conseguido através da manipulação do mapeamento de nome de host e endereço IP, o que torna o navegador do invasor um proxy na rede privada da vítima.[9]

É conhecido pelo pull request no github que o patch de correção só foi integrado ao Transmission em 15 de janeiro de 2018[10]. E sabemos que sistemas baseados em linux como Debian, que possuem protocolos de atualizações restritos, boa parte dos usuários

**Citation:** Bernardi, N.; Falcade, N. Contramedidas em navegador para ataques de acesso remoto do tipo dns em sistemas unix e windows. *Appl. Sci.* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

consequentemente ficam vulneráveis ao longo desse tempo. Por isso, os resultados deste trabalho têm uma contribuição importante para o campo de segurança da informação

O objetivo deste trabalho é desenvolver uma extensão como contramedida, chamada RebindBlock, por meio de técnicas apresentadas e discutidas no artigo "Stopping DNS Rebinding Attacks in the Browser"[1].

O restante deste trabalho está organizado da seguinte forma: na subseção 1.1 Trabalhos relacionados é apresentado artigos que relacionam com a proposta a ser trabalhada; em 2 Materiais e métodos é apresentado as ferramentas utilizadas para o desenvolvimento da solução; e por fim as 4 Considerações finais.

### *1.1. Trabalhos relacionados*

O tipo de ataque de DNS Rebind é conhecido há mais de 15 anos, por isso uma quantidade considerável de literatura foi publicada sobre contramedidas. Pesquisas como a realizada por Hazhirpasand[1] mostraram as tentativas de soluções ao longo da década. A Tabela 1 fornece uma visão geral das soluções preventivas existentes e seus desvantagens.

**Table 1.** Medidas preventivas para ataques de DNS rebind e suas desvantagens.

| **Abordagem** | **Desvantagem** |
|---|---|
| Fixação de DNS | Tempo de fixação limitado, pode ser encurtado por DNS flooding |
| Fixação de DNS - Verificação de IP | Impõe muitas limitações |
| Plug-in NoScript | Bloqueia todo o conteúdo javascript do site |
| eSOP | Requer um cabeçalho de servidor, um navegador personalizado, não possui verificação de IP/Porta |
| INP | Requer implementação pelos navegadores e não possui verificação de IP/Porta |

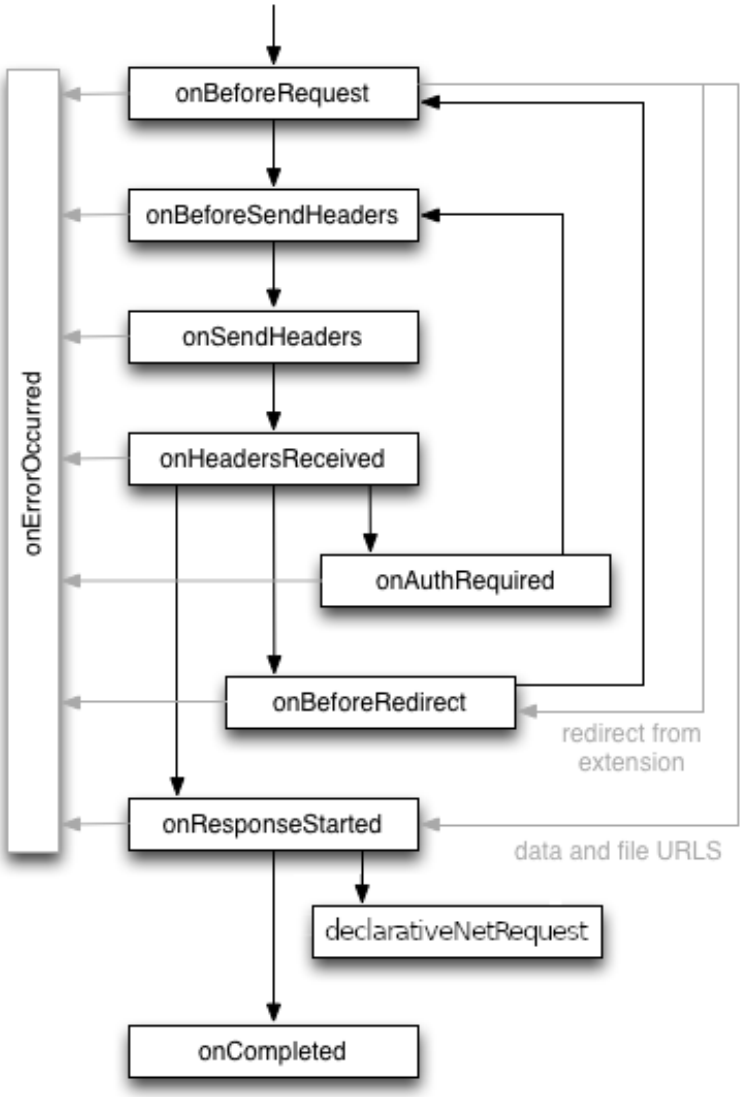
Segundo Hazhirpasand[1] nenhuma das técnicas apresentadas pode interromper totalmente esse ataque devido a vários fatores, por exemplo, a criptografia da camada de rede torna a inspeção de pacotes inviável. Examinando os fatores problemáticos anteriores, a melhor medida de proteção deve ser implementada no nível do navegador.

Jackson[2] menciona que para se defender contra ataques de DNS Rebind, destaca a importância da utilização de implementações no nível do navegador. As extensões podem reparar o DNS Rebind no nível de vulnerabilidades em soquete, adicionando verificações adicionais antes de aceitar a solicitação.

## 2. Materiais e métodos

A abordagem utilizara a API de webRequest e declarativeNetRequest do chrome. A API define um conjunto de eventos que seguem o ciclo de vida de uma solicitação da web. Esses eventos vão ser utilizados para observar e analisar o tráfego. Com eles é possível interceptar, bloquear ou modificar uma solicitação. A extensão utilizara os seguintes eventos:

**Figure 1.** Ciclo de vida do evento para solicitações.[11]




- *onBeforeRequest(opcionalmente síncrono)*: Dispara quando uma solicitação está prestes a ocorrer. Este evento é enviado antes que qualquer conexão TCP seja feita e pode ser usado para cancelar ou redirecionar solicitações.
- *declarativeNetRequest*[12]: É usada para bloquear ou modificar solicitações de rede especificando regras declarativas. Isso permite que as extensões modifiquem solicitações de rede sem interceptá-las e visualizar seu conteúdo, proporcionando mais privacidade.
- *onResponseStarted*: Dispara quando o primeiro byte do corpo da resposta é recebido. Não permite modificar ou cancelar o pedido.

Para parar os ataques de reenvio DNS, temos de obter o endereço IP de um pedido juntamente com o seu nome de domínio. Isto é conseguido por meio de uma API, no chrome, o endereço IP de um pedido pode ser obtido no evento onResponseStarted.

A função de checar o IP assegura que os pedidos são iniciados a partir de domínios confiáveis e não apontam para nenhum IP privado. Caso contrário, serão bloqueados no declarativeNetRequest.

A checagem verifica se o IP não pertencem a nenhum ip da lista de endereços que são associados a rede privada. A definição de ip privado é vista em rfc1918[3]. Assim com uma breve comparação a partir dessas definições, é possível garantir que o IP não está dentro do endereçamento rfc1918.

Considerando que uma página já esteja a par da lista de endereçamento rfc1918,utilizando uma estratégia de ip secundário que contem o endereço privado. A função onHeadersReceived continuamente verificara todas solicitações, assim que a solicitação do endereço privado chegar, ela consequentemente será bloqueada.

Quando se trata de invasão, o melhor e mais seguro método é praticar dentro de um ambiente virtual. Sendo assim, vamos usar o docker para construir e configurar um ambiente contendo apenas as ferramentas de que precisamos e, em seguida, iniciar um contêiner e trabalhar a partir dele. Nosso ambiente sempre será exatamente o mesmo que é iniciado a partir de uma imagem.

Docker é uma tecnologia que fornece virtualização em nível de sistema operacional, também conhecida como contêineres. O docker é constituído de imagens contendo um sistema operacional e ferramentas, sendo esses o sistema Debian e instalação do navegador chrome a ser instalado a extensão para testes. Além disso, uma porta deve ser compartilhada com o sistema para o framework de ataques do tipo DNS rebinding.

A ferramenta Manifest V3 para extensões google chrome foi utilizada no desenvolvimento. No entanto, existem algumas desvantagens associadas com o seu uso como a deprecação do bloqueio pelo webRequest, dessa forma a solução encontrada foi a utilização declarativeNetRequest. Apesar dessa controversa, a extensão ganha uma grande vantagem com a utilização, pois a ferramenta Manifest V2, a versão anterior a usada, deixará de funcionar em Junho/2023[15], deprecando qualquer extensão que a utiliza.

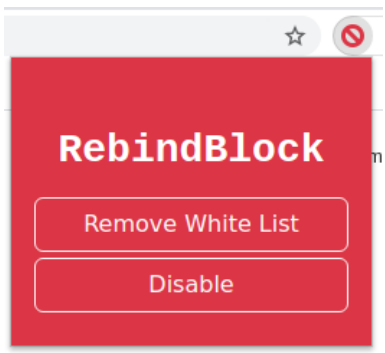
Singularity[14] é uma ferramenta para realizar ataques de DNS rebinding. Ele inclui os componentes necessários para ligar o endereço IP do nome DNS do servidor de ataque ao endereço IP da máquina de destino e servir informações úteis de ataque para explorar software vulnerável na máquina de destino. Ele visa fornecer uma estrutura para facilitar a exploração de software vulnerável. No caso de testes, a recomendação é utilização do Jenkis[13], que é um servidor de automação.

## 3. Resultados

Para validar a extensão RebindBlock, é necessário criar o cenário de teste. O primeiro docker a ser iniciado é o ambiente de teste, que contem o navegador chrome em sua versão 90.0.4430. O segundo docker contém nosso software a ser invadido, o servidor de automação Jenkis versão 2.332.3 LTS.

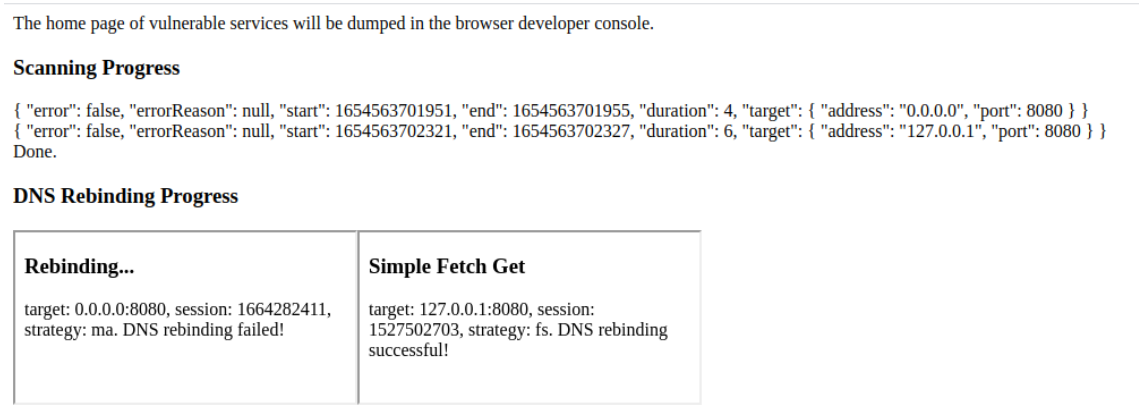
Essa fase tem como objetivo a obtenção dos dados e acesso remoto local. Com o intuito de validar as atividades de verificação e validação, a ferramenta Singularity é inicializada. A Figura 2 ilustra a adição da página a whitelist Nota-se claramente que é um passo para que o teste seja realizado.

**Figure 2.** Adicionado a página a whitelist.



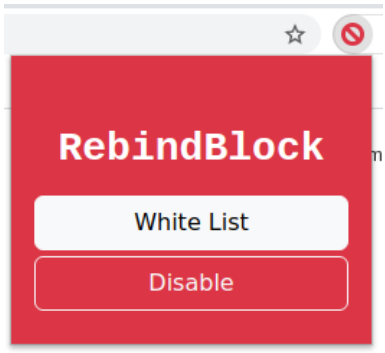
Com a página na whitelist e feito o ataque, é exemplificado na Figura 3, onde destaca-se que o ataque é concluído com sucesso. Nessa fase ele busca acesso ao IP publico 127.0.0.1, na qual está rodando o Jenkis. Feito o ataque, a conexão com o IP 35.185.206.165 ao IP publico é concluída, obtendo todos dados e acesso a rede local que está executando o Jenkis.

**Figure 3.** Ataque de DNS Rebind concluído.



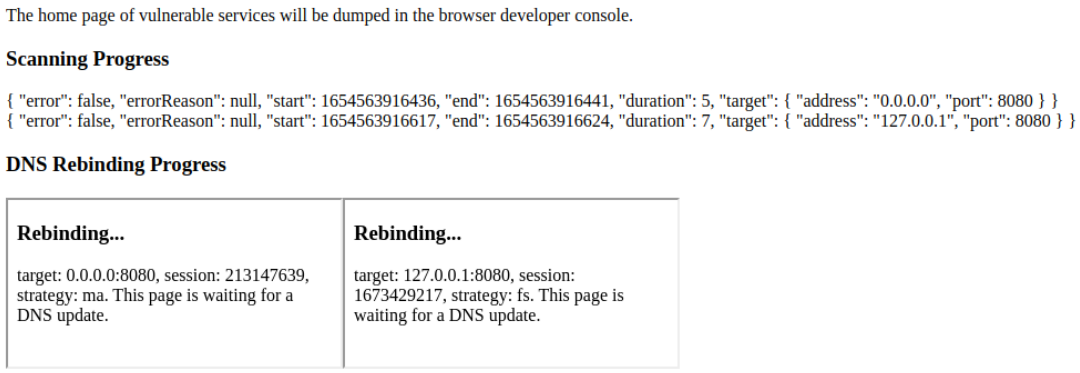

Vale destacar as iterações existentes entre as fases, o que significa que agora devemos retirar a página da whitelist, como mostrado na Figura 4. Essa fase tem como objetivo validar o funcionamento da extensão.

**Figure 4.** Removendo a página da whitelist.




A página maliciosa do framework estabelece sua primeira conexão, nesse cenário a conexão inicial não buscará acesso ao IP local, por isso será aceita no onBeforeRequest. A API onBeforeRequest recebe o Host e URL, e verifica se o endereço pertence a faixa de endereços privados do rfc1918.

**Figure 5.** Ataque de DNS Rebind sendo realizado.




Nessa fase, a página maliciosa estabelece solicitações contínuas, o RebindBlock verifica todas solicitações. Cada solicitação possui a URL e o IP, que são recebidas pelo onHeadersReceived. A solicitações iniciais contem o IP 35.185.206.165, nesse caso não há problemas em tais conexões. Porém, quando é recebido um novo endereço, como o 127.0.0.1. É realizado pela extensão, a comparação entre o IP recebido com a faixa de IPs públicos. Visto que o IP pertence a essa faixa, é encontrado uma tentativa de invasão a rede local, e portanto a conexão é bloqueada.

Desse modo, assim que recebido a conexão maliciosa, uma nova regra é criada no declarativeNetRequest, não permitindo nenhuma nova conexão de nenhum host relacionado a essa página, levando a uma página de bloqueio como ilustrado na Figura 6.

**Figure 6.** Página de bloqueio com detalhes sobre a tentativa de ataque


## 4. Conclusão

Espera-se que este trabalho contribua para segurança em ataques de acesso remoto do tipo DNS, que tem sido uma fonte recorrente de vulnerabilidades em navegadores desde a década passada. Este trabalho mostrou que, em geral os métodos utilizados em navegadores possuem desvantagens em sua proteção. Os resultados deste trabalho mostraram que a extensão RebindBlock, é uma contramedida para ataques de DNS rebind, dado a verificação de cada solicitação realizada dentro do navegador.

A limitação mais importante reside em futuras atualizações em navegadores, novos trabalhos estão sendo realizados internamente pelos proprietários. Como foi salientado na introdução deste trabalho, o presente trabalho limita-se para usuários que já não possuem uma atualização constante de sistema, e consequentemente o navegador.

Para perspectivas futuras, a seguinte extensão RebindBlock foi publicada como código aberto, desse modo contribuições em seu desenvolvimento podem ser realizadas a qualquer momento.

## References


1. Hazhirpasand., M., Ale Ebrahim., A. & Nierstrasz., O. Stopping DNS Rebinding Attacks in the Browser. *Proceedings Of The 7th International Conference On Information Systems Security And Privacy - ICISSP,*. pp. 596-603 (2021)
2. Jackson, C., Barth, A., Bortz, A., Shao, W. & Boneh, D. Protecting browsers from DNS rebinding attacks.. *TWEB*. **3** pp. 2 (2009,1)
3. Moskowitz, R., Karrenberg, D., Rekhter, Y., Lear, E. & Groot, G. Address Allocation for Private Internets. (RFC Editor,1996,2), https://www.rfc-editor.org/info/rfc1918
4. Ormandy, Tavis. *1447 - Project-zero - Project Zero - Monorail*. (2017,11), https://bugs.chromium.org/p/project-zero/ issues/detail?id=1447
5. McClure, S., Scambray, J. & Kurtz, G. Hacking Exposed 7 : Network Security Secrets and Solutions, Seventh Edition: Network Security Secrets and Solutions, Seventh Edition. (Mcgraw-hill,2012)
6. Ormandy, T. 1471 - project-zero - Project Zero - Monorail. , https://bugs.chromium.org/p/project-zero/issues/detail?id=1471
7. Dimi, Internet Wide Ethereum JSON-RPC Scans. , https://blog.3or.de/internet-wide-ethereum-json-rpc-scans.html
8. Tatang, D., Suurland, T. & Holz, T. Study of DNS Rebinding Attacks on Smart Home Devices. (2020,2)
9. Contributor, T. What is DNS rebinding attack? - definition from whatis.com. *SearchSecurity*. (2008,4), https://www.techtarget.com/ searchsecurity/definition/DNS-rebinding-attack
10. Transmission CVE-2018-5702: Mitigate DNS rebinding attacks against daemon by Taviso · pull request 468 · transmission/transmission. *GitHub.*, https://github.com/transmission/transmission/pull/468
11. Google Chrome webRequest. , https://developer.chrome.com/docs/extensions/reference/webRequest/
12. "Chrome.declarativeNetRequest - Chrome Developers." Chrome Developers, 2022, developer.chrome.com /docs/extensions /reference/declarativeNetRequest/. Accessed 7 June 2022.
13. jenkinsci. "Jenkinsci/Docker: Docker Official Jenkins Repo." GitHub, 6 June 2022, github.com/jenkinsci/docker. Accessed 7 June 2022.
14. nccgroup. "Nccgroup/Singularity: A DNS Rebinding Attack Framework." GitHub, 7 Feb. 2020, github.com/nccgroup/singularity. Accessed 7 June 2022.
15. "Manifest V2 support timeline - Chrome Developers," Chrome Developers, 2022. https://developer.chrome .com/docs/extensions/ mv3/mv2-sunset/