

DoS: Negação de Serviço e formas de defesa

Viva o Linux Day - RJ

<http://volcon.org/volday1/>

Elgio Schlemer

Ulbra Gravataí

<http://gravatai.ulbra.tche.br/~elgio>

06 de Março de 2010

Introdução

- Problemas de segurança
 - Engenharia Social
 - "A Arte de Enganar"
 - Programas maliciosos
 - virus, worms, capturadores de senhas, ...
 - Bugs de programação
 - invasões, comprometimento do sistema
 - **DoS: *Denied of Service***
 - "Apenas" tirar do ar

Tipos de negação de serviço

- Existem dois tipos de negação de serviço:
 - Negação de serviço Local
 - usuário malicioso precisa executar comandos na máquina
 - Negação de serviço Remota
 - usuário malicioso não executa comandos na máquina

DoS Local

- precisa executar comandos na máquina, isto é, ter login
 - ou é um usuário legítimo malicioso (ou descuidado)
 - ou roubou a senha de um usuário legítimo
 - ou explorou alguma vulnerabilidade para executar comandos na máquina
- O que um DoS local pode fazer?
 - esgotar algum recurso do sistema

Recursos da máquina: disco

- capacidade do disco rígido
 - pode escrever no disco até esgotar sua capacidade
 - serviços que precisam de espaço ficarão indisponíveis
 - Exemplo: se lotar o /var/spool/mail não se recebe mais mensagens de email
 - Soluções:
 - correto particionamento do disco rígido
 - quotas de uso

Recursos do SO: processos

- capacidade do escalonamento de processos
 - atacante pode criar processos até lotar
 - novos processos não poderão ser criados
 - serviços ficam indisponíveis (ou lentos)
 - Demonstração:
 - usuário vol executa `byebyeProc`
 - root não consegue se logar
 - Soluções:
 - quotas de processos por usuário com `Pam Limits`

Recursos da máquina: memória

- capacidade da memória
 - atacante pode alocar memória até lotar
 - serviços que precisam alocar memória não conseguirão
 - Demonstração:
 - usuário vol executa `byebyeMem`
 - root não consegue se logar
 - Soluções:
 - quotas de memória por usuário com `Pam Limits`

Demonstração Pam Limits

- Permite definir recursos que usuário pode usar
 - disco: quota + particionamento
 - memória, processos: pam limits
- Mostrando o arquivo `/etc/security/limits.conf`

```
# limites para grupo root
@root          hard    maxlogins      50
@root          hard    nproc          500
@root          hard    data           2000000

# limites para todos (exceto grupo root,
# pois as regras anteriores precedem)
*              hard    maxlogins      15
*              hard    nproc          20
*              hard    data           20000
```

DoS Local: Conclusão

- Fácil de resolver, é só configurar
 - Particionar corretamente o disco durante a instalação
 - para servidores!! Planejar antes
 - Configurar quotas de disco nas partições que usuários poderão escrever
 - Configurar quotas de recursos
 - Permitir login apenas para quem realmente precisar
- DoS Local: não deveria ser um problema!!

Negação de serviço remota

- Derruba serviço ou sistema sem precisar executar comandos
 - Obs: se explorou um bug para executar comandos, não é DoS remoto, é local (apenas usou um bug para poder executar comandos)
- DoS remoto:
 - bugs nos aplicativos
 - ataques ao protocolo

DoS Remoto: Bugs em programas

- O serviço possui problemas em sua programação
- Basta fazer algo para que ele caia
 - Exemplo clássico: ping da morte:
 - ICMP com dados maiores que o *buffer* do Windows.
 - Pilha TCP caia (TUDO!)
 - Um típico problema de *buffer overflow*

Bugs em programas: Solução

- correção dos problemas de programação
 - crítico: depende do fornecedor
 - Linux: como eu posso mexer no código...
 - proprietários: somente quando vier os tais Patches... :-D
- boas práticas de programação
 - requer mudança de mentalidade
 - programar pensando em segurança

DoS Remoto: Ataques a protocolos

- Exploração de detalhes da pilha TCP/IP
- Alguns, não tem correção:
 - correção seria implementar novos protocolos
- Exemplo: ping *broadcast*
 - Envia-se um ping com ip *spoofado* para todos
 - todos respondem para o ip que cai
 - Solução (fácil):
 - tratar *ip spoofing* no firewall
 - não responder a pings *broadcast*

Ataques de DNS Recursivo

- Explora má configuração em servidores de DNS
- Um servidor DNS deveria atuar de forma recursiva apenas para seus clientes
- Recursiva: o DNS aceita qualquer consulta, mesmo não sendo para o domínio dele
 - Demonstrar consultas DNS

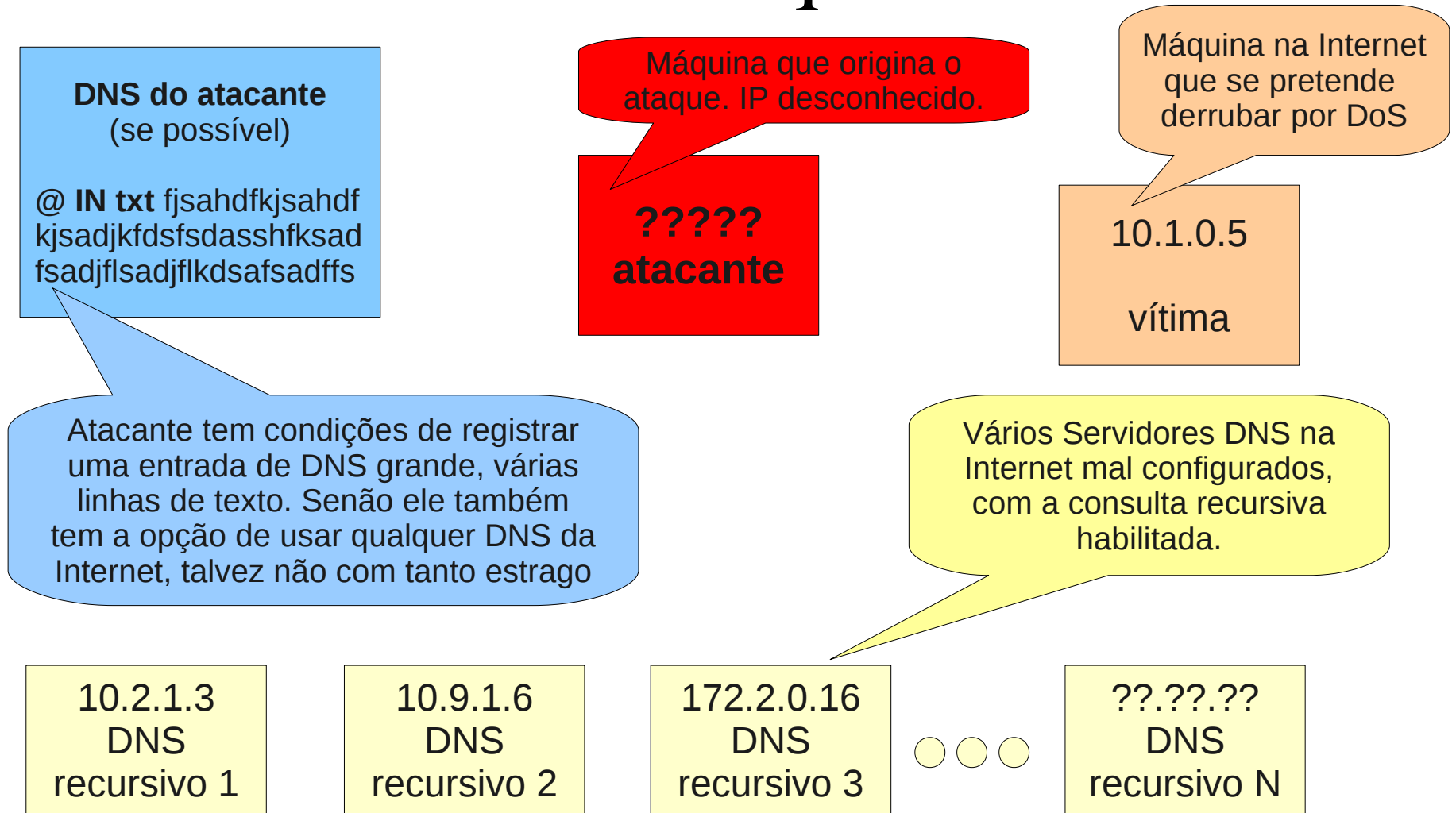
Forma de ataque DNS

- Atacante descobre lista de DNS recursivos
- Registra em seu DNS, se possível, uma entrada longa
 - para potencializar o ataque
 - se o atacante não possuir um DNS, pode usar uma entrada longa de outro DNS que venha a descobrir
 - demonstrar

Forma de ataque DNS

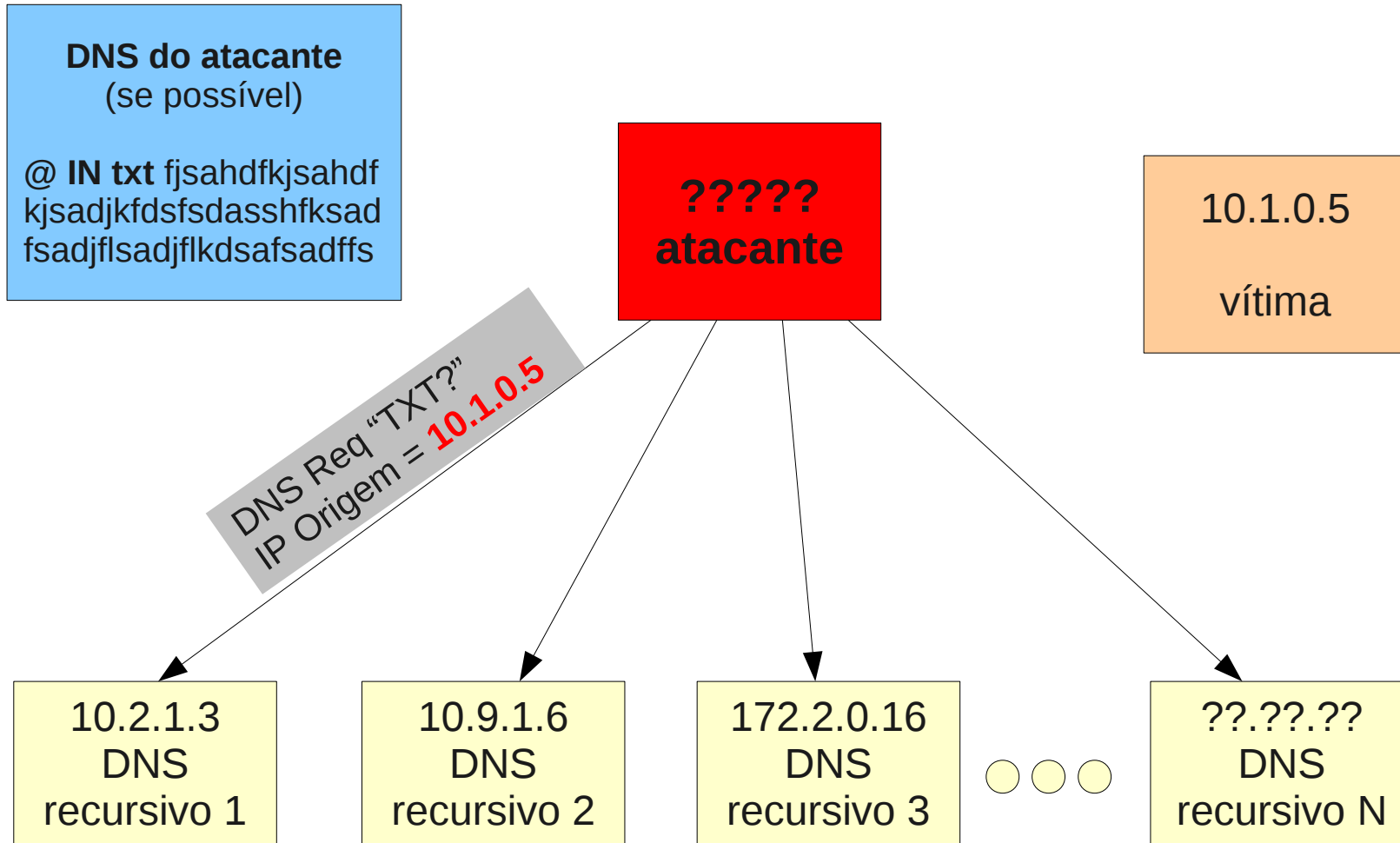
- Envia para os DNS recursivos inúmeras falsas consultas:
 - solicitando uma consulta grande
 - se passando (*ip spoofing*) pelo IP da vítima
- Consequência:
 - todos os servidores DNS irão realizar a consulta
 - colocar em sua cache
 - responder ao IP da vítima
 - vítima cai por excesso de tráfego

Forma de ataque DNS



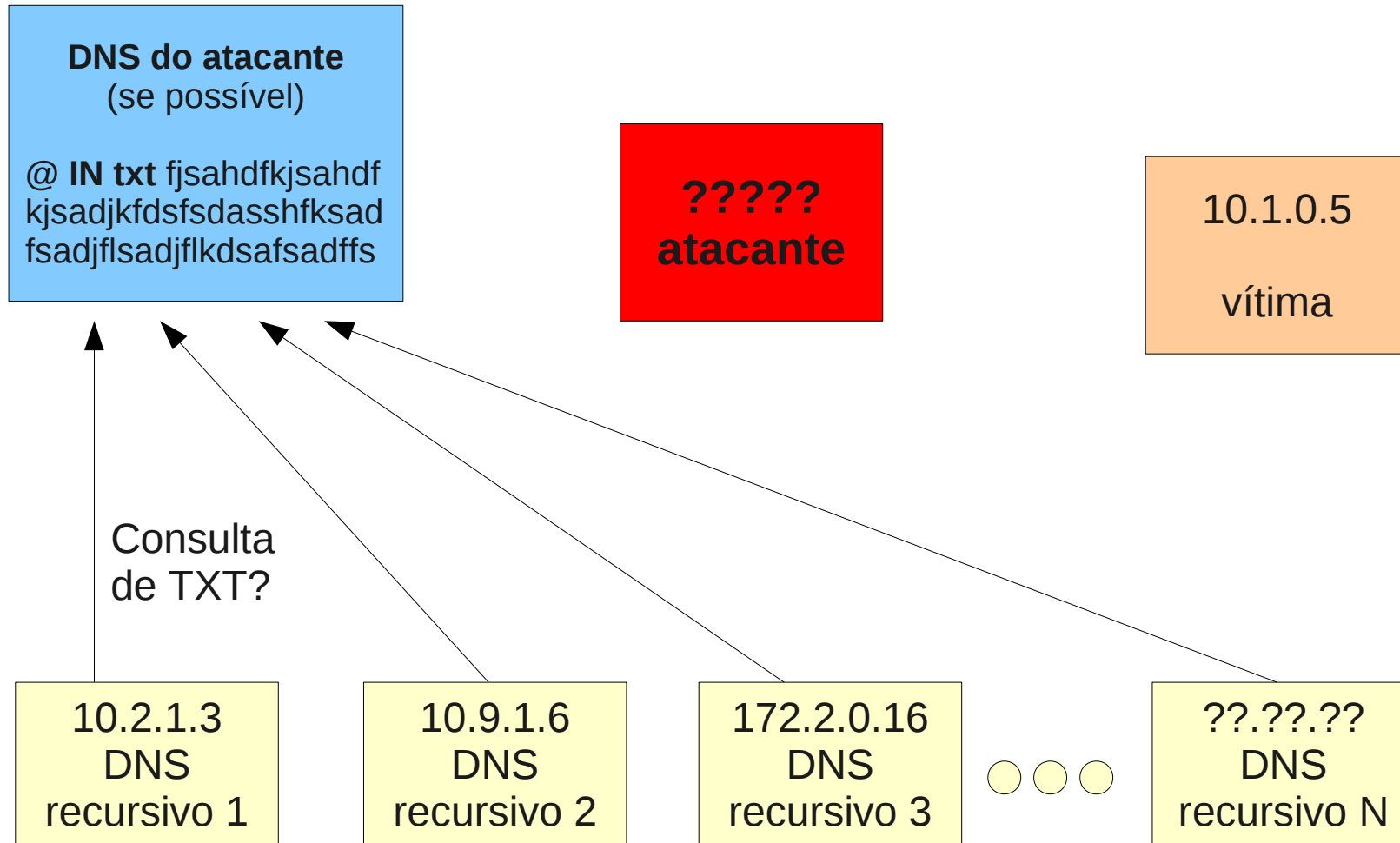
Neste exemplo serão usados apenas IPs privados, mas na prática todos são ips públicos com servidores reais espalhados pela Internet.

Forma de ataque DNS



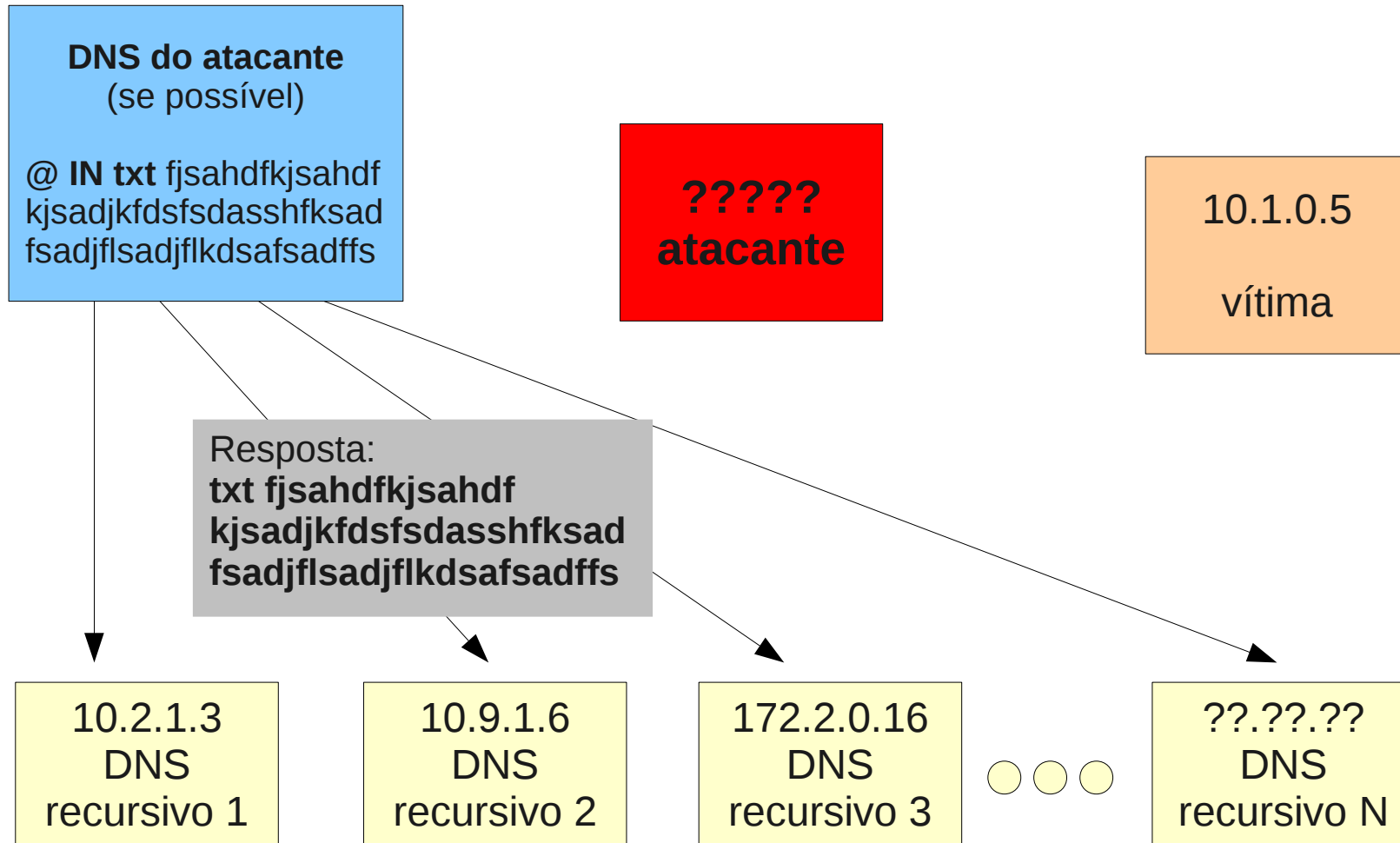
Atacante envia consultas falsas de DNS, perguntando pela entrada txt grande e fingindo ser a vítima.

Forma de ataque DNS



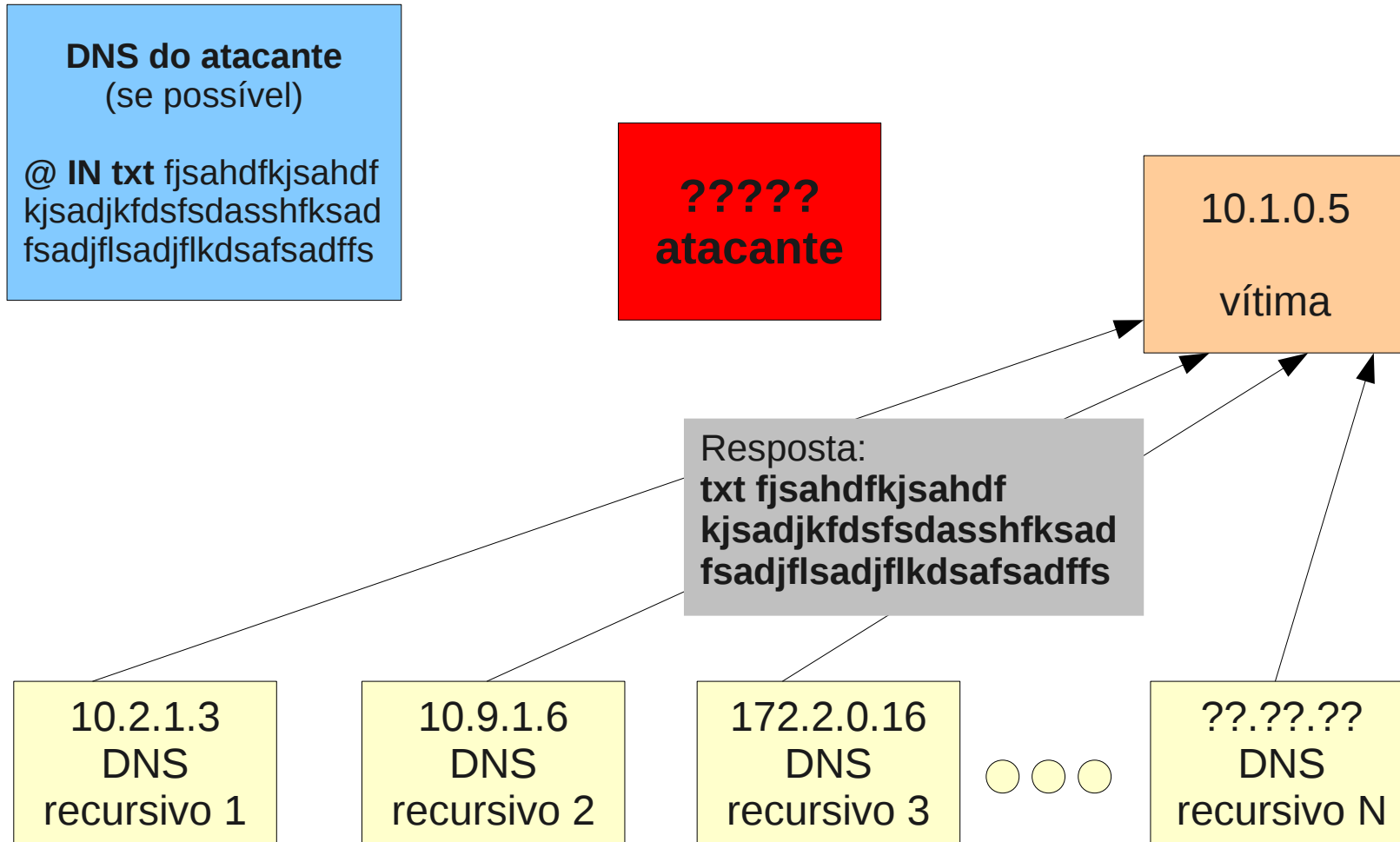
Se DNS não tiverem esta informação em cache, irão buscar, isto é, atuarão de forma recursiva (e esta é a má configuração: não deveriam fazer isto)

Forma de ataque DNS



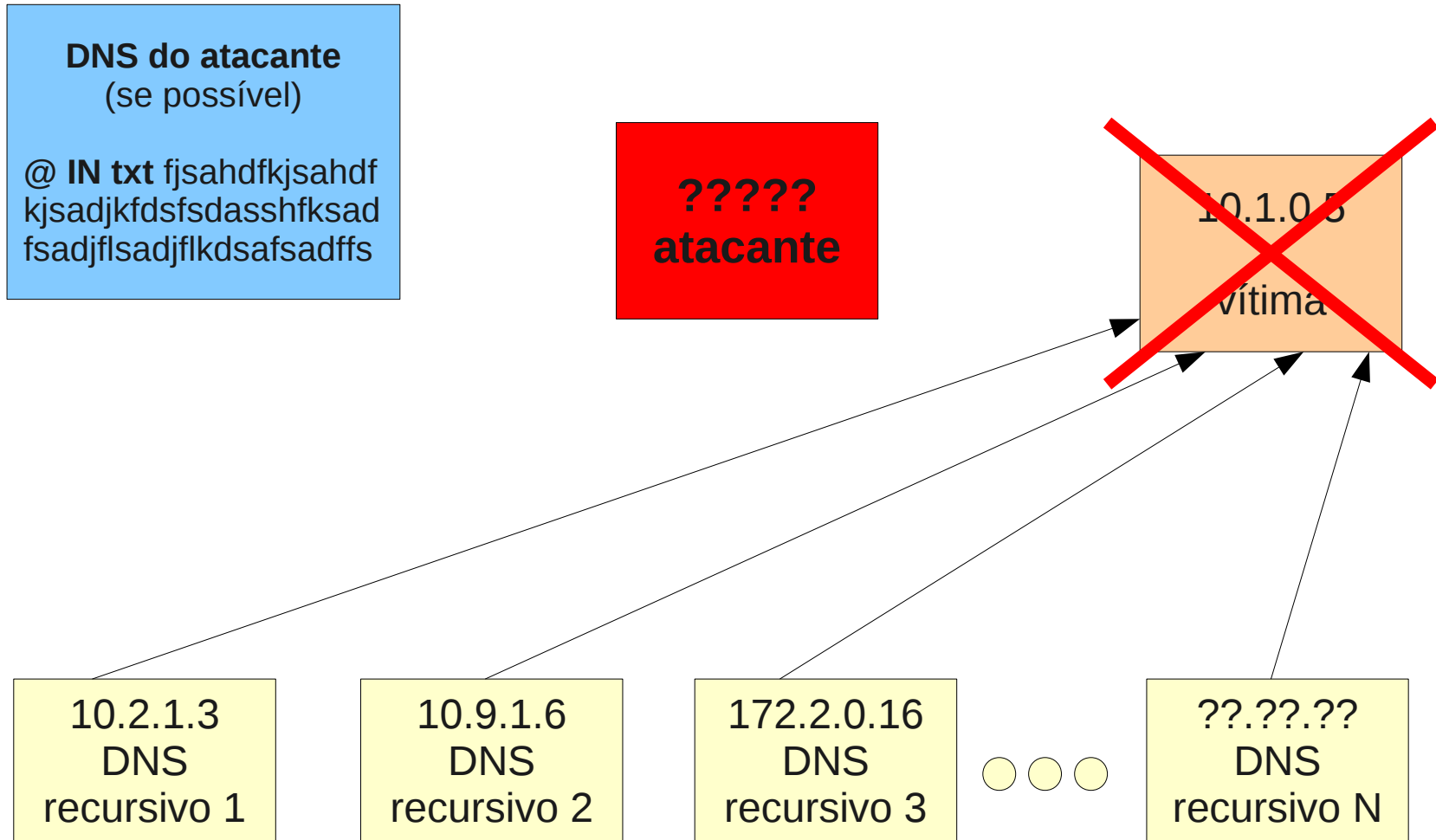
O servidor master do domínio consultado responde as solicitações feita pelas DNS. Que se diga que a resposta é muito maior que a pergunta, pois tem muitos dados. Todos guardam em sua cache.

Forma de ataque DNS



E todos os DNS respondem para a pobre da vítima, pois a consulta feita pelo atacante mentiu o número IP dizendo ser do atacante

Forma de ataque DNS



Vítima cai por não conseguir lidar com tantos pacotes UDP. O atacante continua enviando solicitações o mais rápido que pode.

Solução ataque DNS

- Tratar IP spoofing
 - evitará que máquinas de sua rede sejam atacantes
- Configurar corretamente DNS
 - evitará que o teu servidor de DNS seja usado para derrubar uma vítima
- A vítima pouco pode fazer para se defender

Ataque de *Syn Flood*

- Baseado na forma como o TCP implementa confiabilidade
- Não há correção!
 - defesa complicada, mas possível
 - defesa **impossível** com regras de firewall
 - muito embora alguns tutoriais na Internet digam o contrário
 - exemplo do guia Foca Avançado que prega solução por firewall

Syn Flood

- Servidor precisa de recursos para cada conexão
 - para os *buffers* de envio e recebimento
 - para controle de confirmações, entre outros
- Ao receber o *Syn* (início de conexão), o servidor:
 - aloca estes recursos
 - define o seu número sequencial
 - responde ao cliente com um *Syn + Ack* (*handshake*)

Syn Flood

- Se cliente não completar *handshake* após certo tempo:
 - desaloca estes recursos
 - envia um RST (reset) para o cliente
- Importante:
 - até que o tempo se esgote, recursos ficam **alocados**
 - E se cliente só fizer SYNs e nunca completar o *handshake*?

Syn Flood

- Cliente pode gerar pacotes Syn muito mais rapidamente do que o servidor pode tratá-los
- *Flood*, inundação de Syns:
 - servidor aloca tantos recursos para falsos clientes que fica sem recursos para atender um cliente legítimo
 - Negação de serviço no servidor!
 - hping3 faz *flood*

Defesa *Syn Flood*

- Muito complicado!
- Bloquear ips dos atacantes?
 - **FALHO**: atacante precisa fazer *ip spoofing* para que ataque tenha sucesso
 - Qual ip bloquear?
- Limitar quantidade de Syns no firewall?
 - **FALHO**: firewall vai recusar Syns de clientes legítimos
 - O próprio firewall acaba causando o DoS
 - e o atacante pode explorar isto ainda mais fácil

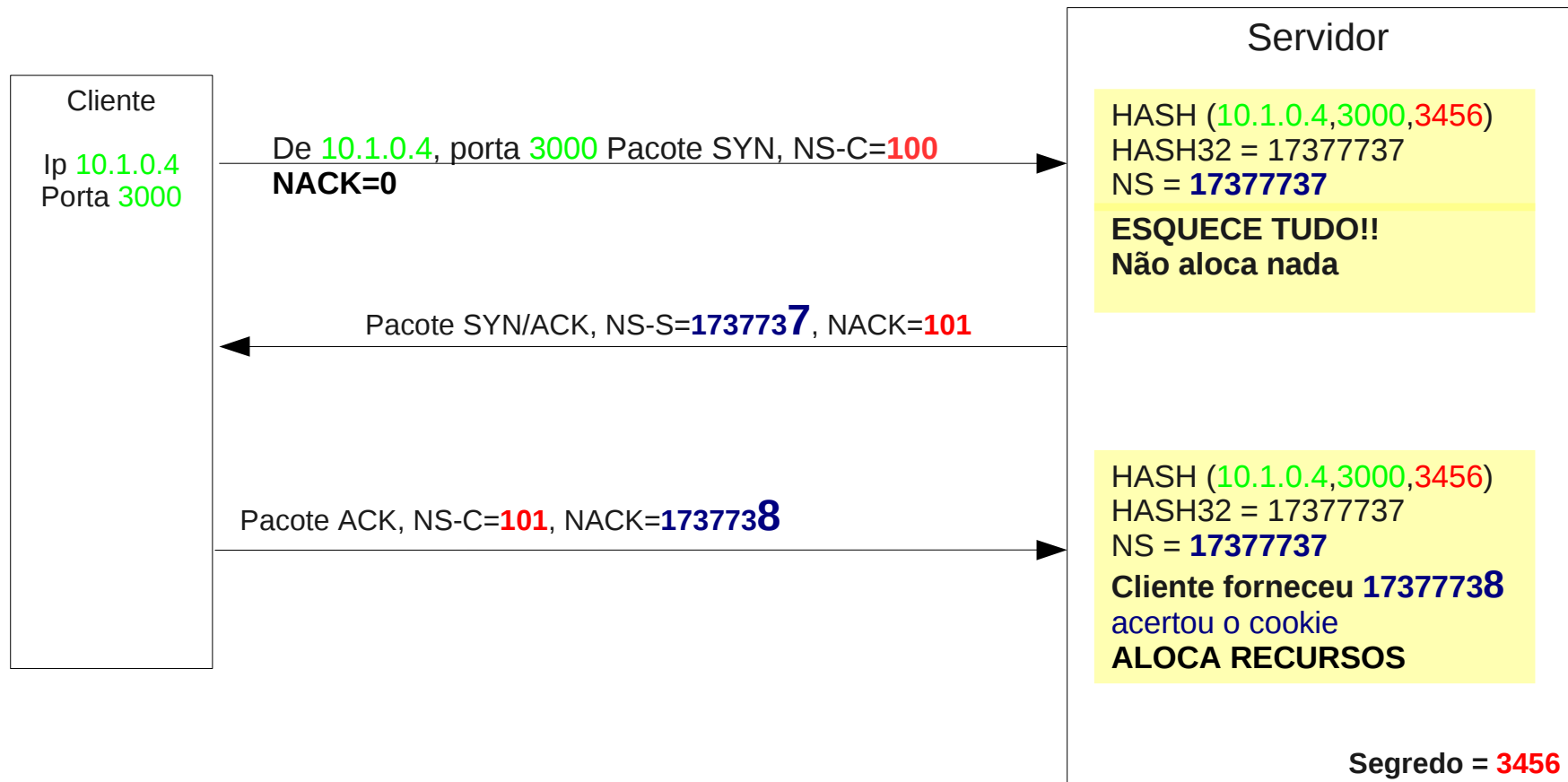
Defesa *Syn Cookie*

- Alocação de recursos feita no final do *handshake*
 - Somente se cliente for legítimo
 - Cliente legítimo: alguém que completou o *handshake*
- como saber se o cliente completou o *handshake*?
 - Armazenar Ip e porta no primeiro SYN
 - **FALHO**: isto não é alocar recursos?
 - Reconhecer cliente mesmo sem armazenar nada.
 - COMO????

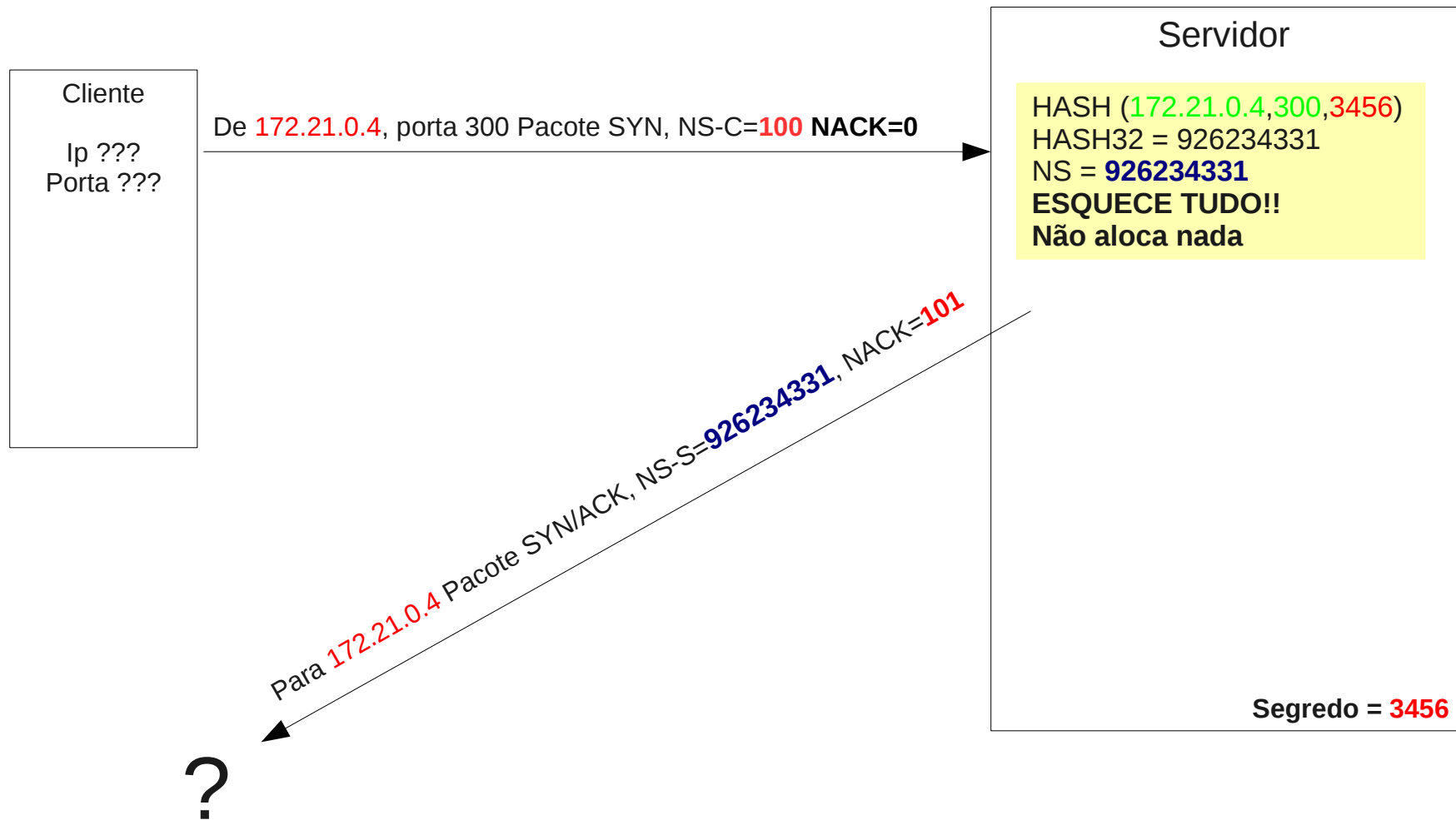
Funcionamento *Syn Cookie*

- Servidor gera seu número sequencial a partir de um HASH de 32 bits:
 - envolvendo a porta e o IP do cliente
 - informações de *time stamp*
 - informação sigilosa (senha) que só o servidor tem
- Cliente legítimo deverá no último pacote do *handshake*:
 - Numero de ACK = número seqüencia do servidor + 1

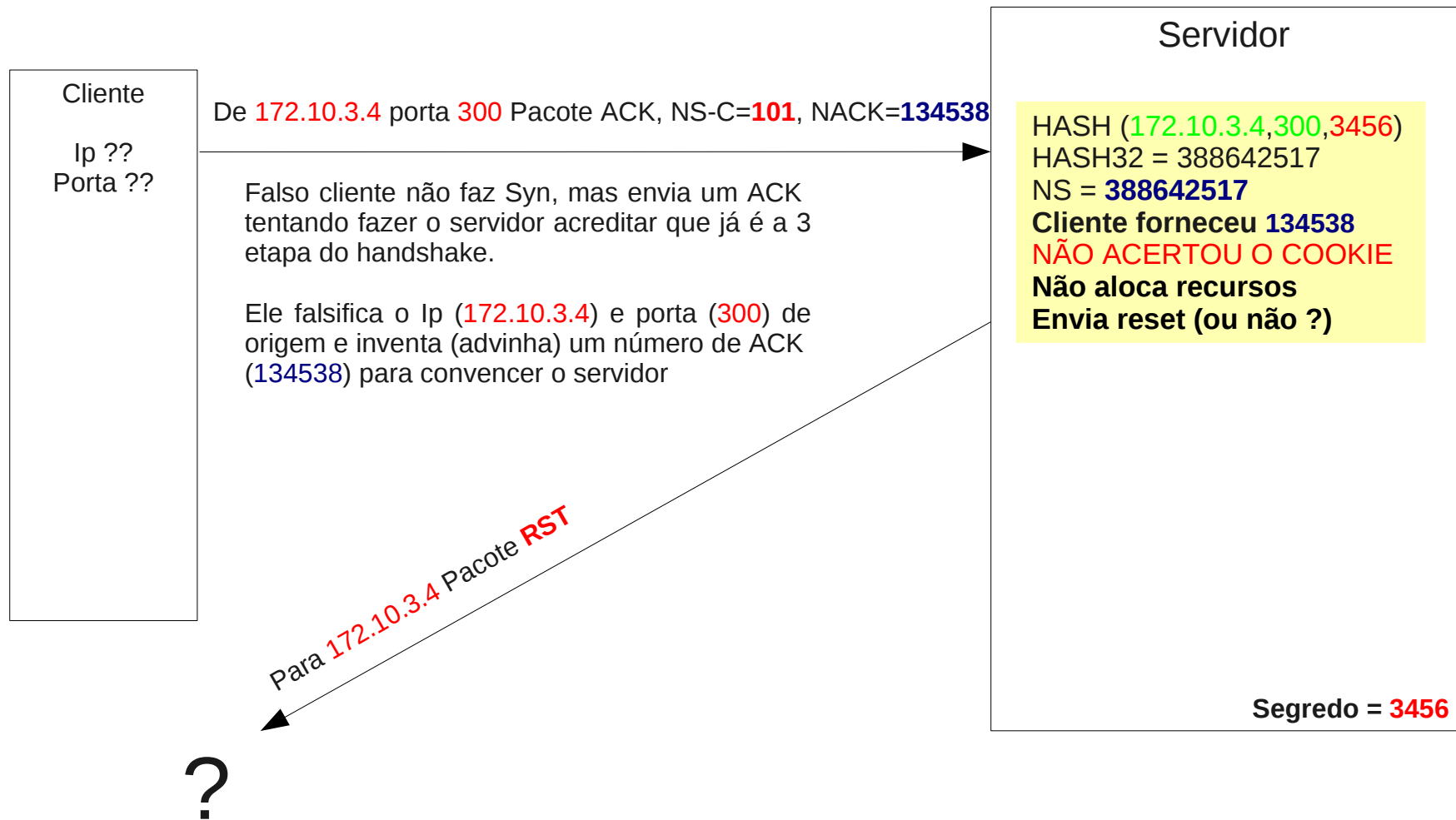
Syn Cookie: Cliente legítimo



Syn Cookie: Atacante tentando Syn Flood



Syn Cookie: Atacante tentando passar



Sobre *Syn Cookie*

- Atacante não consegue nada fazendo SYNs
 - A menos que ele não minta sobre seu IP, para poder receber o *cookie*
 - Mas ai ele se expõe e precisa participar do *handshake*
- Técnica de Syn Cookie não será portada para IPv6
 - problemas do syn cookie
 - avanço na capacidade de processador e banda
- Linux tem *Syn Cookie* a muito tempo:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Conclusão

- Negação de serviço local não é problema
 - depende do administrador configurar corretamente
- Negação de serviço remoto:
 - Protocolo: difícil, requer defesas
- Cuide-se para:
 - não ser a vítima
 - não ser o atacante

Referências

- HOEPERS, Cristine; et al. **Recomendações para evitar o Abuso de DNS Recursivos Abertos**. CERT BR, Fevereiro de 2009. Disponível em www.cert.br/docs/whitepapers/dns-recursivo-aberto/ (acesso em 05/Março/2010)
- SCHLEMER, Elgio. **Iptables protege contra SYN FLOOD?** Viva o Linux, Agosto de 2007. Disponível em www.vivaolinux.com.br/artigo/Iptables-protege-contr-SYN-FLOOD
- EDDY, Wesley. **TCP SYN Flooding Attacks and Common Mitigations**. Request for Comments (RFC4987), Agosto de 2007. Disponível em www.rfc-editor.org/rfc/rfc4987.txt
- ANÔNIMO; et al. **SYN Flood**. Enciclopédia Livre. Disponível em pt.wikipedia.org/wiki/SYN_Flood (versão de 26 de Dezembro de 2009 acessado em 05/Março/2010)

DoS: Negação de Serviço e formas de defesa

Elgio Schlemer
elgios@gmail.com

<http://gravatai.ulbra.tche.br/~elgio>

<http://www.vivaolinux.com.br/~elgio>