Uma Arquitetura para Mitigar Ataques DDoS em Serviços Web sob Nuvem

Fernando Gielow, Fernando Bernardelli,
Cinara Menegazzo, Nadine Pari, Aldri Santos

Acadêmico: Matheus Bauer

Introdução

- Os ataques de Denial of Service (DoS) ainda representam sérias ameaças a muitos servidores na Internet e se configuram como um dos principais desafios de segurança atualmente propagado para a IF, que interconectará muito mais dispositivos e indivíduos.
- Um ataque DoS não visa invadir um computador para obter informações confidenciais, nem tão pouco alterar informações armazenadas nele. Seu objetivo é a indisponibilização de um serviço fornecido, utilizando-se do encaminhamento de grandes quantidades de tráfego ao hospedeiro do serviço.
- Essa questão torna-se ainda mais severa quando diversos geradores de tráfego intensificam o encaminhamento de tráfego de maneira distribuída, caracterizando um ataque de Distributed Denial of Service (DDoS).

- Embora tal carga seja um problema apenas momentâneo, em se tratando de aplicações destinadas ao comércio eletrônico, por exemplo, uma parada do serviço representa grandes perdas financeiras.

- O ataque DDoS é um dos tipos de ameaça que se tornou famoso nos últimos meses justamente por ter sido o tipo de ataque mais executado pelo Anonymous para derrubar diversos sites pelo mundo.

- Um Distributed Denial-of-Service ATTACK é uma maneira relativamente simples de derrubar algum *service*.

- Para efetuar o processo, os hackers precisam criar uma rede zumbi (BotNet), que inclui uma infinidade de computadores infectados de maneira que eles possam ser controlados por um host “mestre”. Quando o hacker escolhe o alvo, ele envia o IP para o mestre, que se encarrega de distribuí-lo por toda a rede zumbi. Essa rede pode incluir milhares de computadores que são responsáveis por sobrecarregar o alvo até que ele se torne indisponível.

- Embora a maioria das soluções comumente oferecidas para mitigar DDoS em cloud se baseie na maior alocação de recursos, essas abordagens tornam-se inadequadas pois a premissa da possibilidade de maior alocação de recursos nem sempre é viável por ser custosa demais. Este comportamento caracteriza o economic DDoS (eDDoS).

- Este trabalho propõe uma arquitetura reativa e tolerante a falhas para a mitigação de ataques de DDoS executados contra aplicações hospedadas em uma cloud. Tal arquitetura é baseada na instanciação de uma réplica da aplicação e no redirecionamento apenas de requisições legítimas a esta réplica. A arquitetura monitora o tráfego de uma aplicação e ao detectar uma possível anomalia, isto é, a ocorrência de um ataque de DDoS, ela estabelece uma nova instância desta aplicação, garantindo que nenhum tráfego malicioso a alcance.

Trabalhos Relacionados

- Em 2010, os ataques são tratados através da criação de uma nova instância da aplicação. Uma vez que um ataque DDoS é detectado, o mecanismo proposto busca identificar os atacantes através de PINGs: caso um cliente suspeito de ser atacante não responda ao PING, ele é considerado como um atacante, não sendo redirecionado para a nova instância da aplicação. Entretanto, essa solução assume que sempre e apenas clientes genuínos responderão a PINGs, o que as vezes não condiz com a realidade.

- A eficácia desses esquemas de mitigação depende diretamente da capacidade de identificação ou filtragem dos clientes legítimos.

Arquitetura para Mitigação de DDoS em Cloud

- Esta Seção descreve uma arquitetura para mitigar ataques de DDoS em clouds de forma autônoma e independente. A arquitetura proposta pode ser utilizada por qualquer aplicação web hospedada em uma cloud que, ao sofrer indícios de um ataque DDoS, filtra o tráfego legítimo e encaminha apenas este para uma nova instância da mesma aplicação.

- Esta arquitetura, ilustrada na Figura 1, é composta por um módulo geral chamado de Gerenciador de Tráfego (GT), que não se comunica diretamente com a aplicação. Esse módulo possui os submódulos INA, GB, AT e RT.

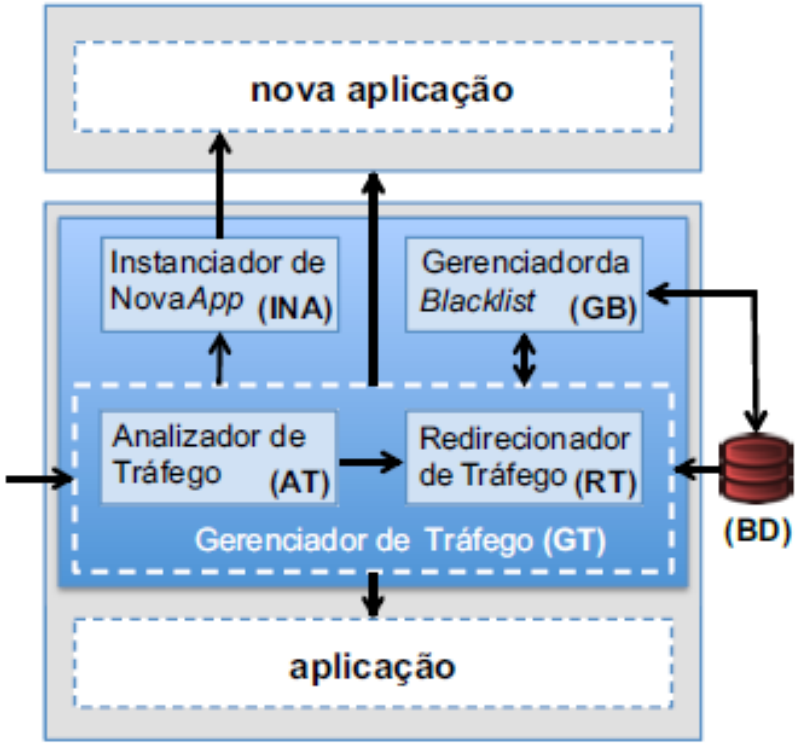

Figura 1. Arquitetura de mitigação de DDoS

- O submódulo AT observa o comportamento do tráfego de entrada para a aplicação de forma proativa. Ele foca na estimativa da quantidade de tráfego e de processamento no servidor, e realiza medição para detectar a existência de um possível ataque DDoS. Caso um ataque seja detectado, o submódulo INA é ativado. O INA criará uma nova instância da aplicação em outro servidor na cloud, consequentemente com um endereço IP diferente. O submódulo RT trata todo o tráfego de entrada, respondendo com um redirecionamento para a nova instância da aplicação.

- Ao tentar redirecionar os clientes para a nova instância, o endereço do cliente, seja ele legítimo ou não, será adicionado em uma blacklist. Os clientes presentes nesta lista têm suas requisições descartadas, a fim de reduzir o custo de processamento de respostas no servidor. Entretanto, como o cliente legítimo é informado do redirecionamento antes de seu endereço entrar na blacklist, ele terá acesso à esta nova instância replicada e poderá enviar uma nova requisição. Registros com tempo de validade são empregadas nesta blacklist, dado que as respostas podem ser perdidas. O tempo de validade na lista aumenta exponencialmente, para diminuir ainda mais a sobrecarga. Cabe ao GB, o papel de adicionar e gerenciar a saída de endereços de clientes à blacklist, assim como o tempo de validade da entrada que aumenta exponencialmente.

Implementação

- Para a implementação da arquitetura, a solução em nuvem foi utilizada. Ela oferece infraestrutura como serviço de hospedagem, possibilitando o desenvolvimento em RoR (Ruby on Rails).
- **RoR** é um framework livre que promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a banco de dados, já que é possível criar aplicações com base em estruturas pré-definidas.
- A arquitetura do framework é completamente baseada no paradigma Model View Controler (MVC).
- Então a estrutura do código escrito em RoR é composta de componentes de Modelo, de Visão e de Controle

- Os componentes de **modelo** correspondem a regras de negócios, funções e aos dados - como eles são armazenados, obtidos.
- A parte de visão corresponde à parte gráfica da aplicação.
- os controladores realizam a manipulação de dados como um todo, e correspondem à parte lógica e funcional do código.
- o submódulo analisador de tráfego (AT) da arquitetura corresponde a um controlador. Uma requisição à aplicação será interceptada por esse componente de controle, que realizará a medição de estatísticas, e imediatamente acionará o controlador que corresponde ao funcionamento da aplicação em si.

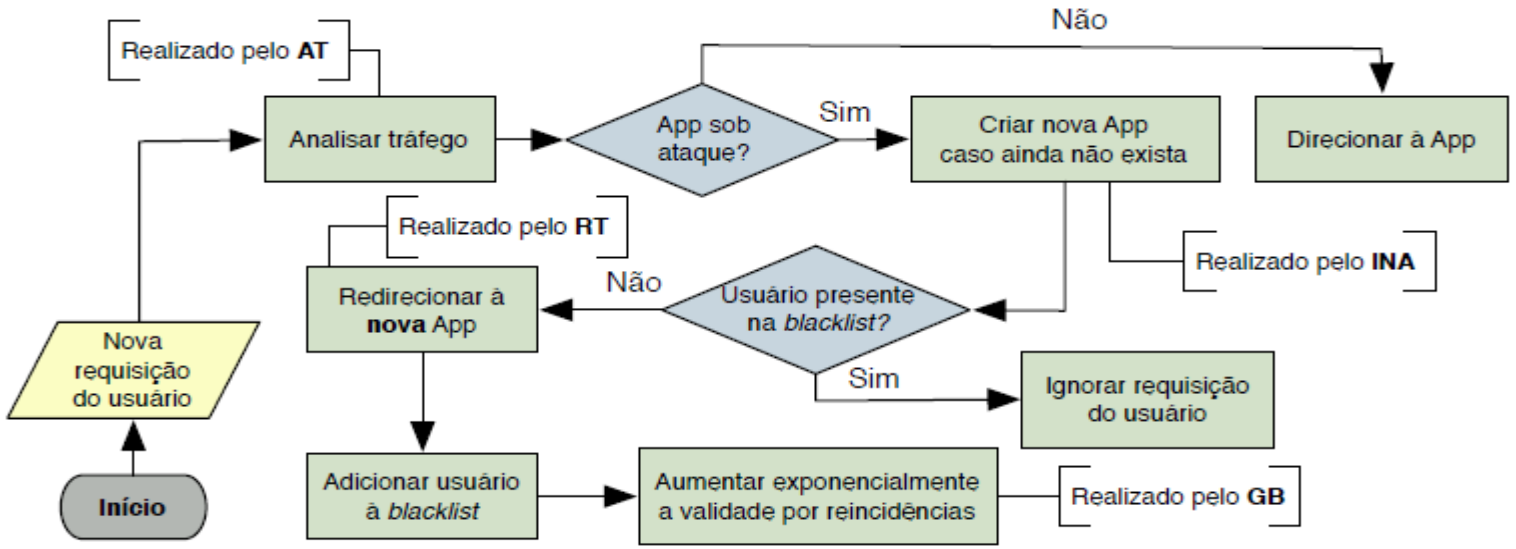

Figura 3. Operações da implementação para a mitigação

- Quando o AT detectar a existência de um possível ataque, uma nova instância cloud é criada pelo submódulo INA e a aplicação é replicada para esta instância, paralisando a aplicação original, que passa a responder apenas como redirecionador.
- A implementação do submódulo redirecionador de tráfego (RT) é realizada em cima de um arquivo de rotas chamado routes.rb. Ele é utilizado para a adição de clientes na blacklist e filtragem dos clientes bloqueados pelo gerenciador da blacklist (GB). No redirecionamento do tráfego para uma nova instância, uma entrada será adicionada, bloqueando o cliente em questão por determinado tempo.

Avaliação

- A avaliação da arquitetura proposta consiste na análise da capacidade do servidor em atender novas requisições, sendo que o atendimento pode ser apenas o redirecionamento. Se o ataque de DDoS for devidamente mitigado, as requisições dos atacantes serão ignoradas, após a sua inclusão na blacklist.

- Logo, o servidor na cloud deverá ser capaz de redirecionar apenas clientes legítimos para a nova instância e garantir que eles terão acesso direto nas próximas requisições.

- Para a experimentação, como nós atacantes, foram utilizadas oito máquinas do laboratório de pesquisa para processar os ataques.

- Cada uma destas máquinas operou com 25, 50, 75 ou 100 instâncias de um script atacante, que utiliza o comando curl para bombardear o servidor com requisições HTTP do tipo GET. Tais experimentos foram realizados diversas vezes, obtendo resultados de comportamento similar.

Resultados

- Primeiro, foi avaliado o impacto da mitigação de DDoS em tempo de resposta e, em seguida, em relação à taxa de resposta e overhead. Como observado na Figura 4, com um intervalo de confiança de 95%, o uso da arquitetura proposta, gráfico à esquerda, reduziu o tempo de resposta às requisições legítimas em comparação ao gráfico à direita, que mostra o tempo gasto para atender o mesmo número de requisições originadas sem o nosso mecanismo.

- Tal comportamento ocorre porque com o uso da arquitetura, a blacklist impedirá uma aplicação de responder ao mesmo cliente múltiplas vezes, garantindo ainda assim que o cliente legítimo seja capaz de atingir a nova instância.

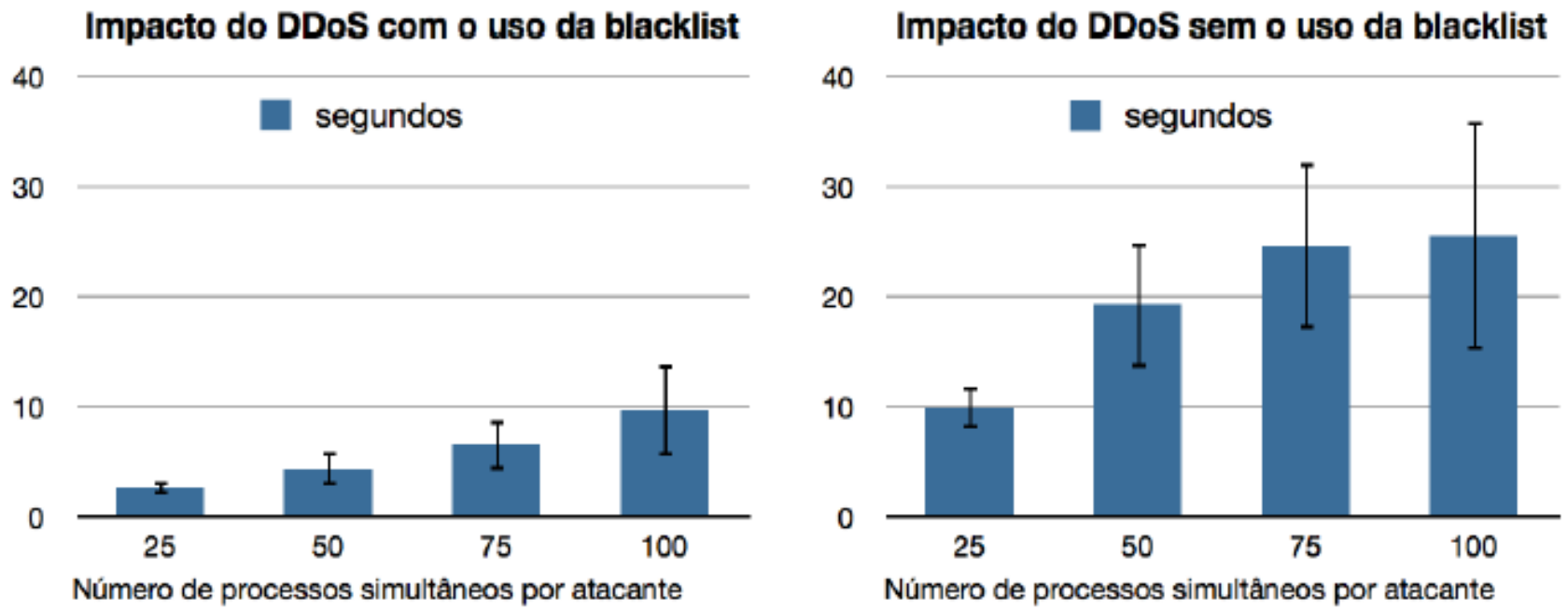

Figura 4. Tempo de resposta para clientes legítimos

- Outra métrica avaliada é a taxa de páginas solicitadas recebidas com sucesso, mostrada na Figura 5. Há uma queda no número de respostas apenas quando não foram utilizadas a filtragem pela blacklist e o consequente redirecionamento. Nestes casos, a aplicação envia uma resposta ao atacante, que descarta esta resposta de imediato, dando continuidade ao ataque. Nota-se que a taxa de entrega sem o uso da blacklist é afetada pela ocorrência de timeouts de requisições HTTP não respondidas, pois a aplicação permanece ocupada com as requisições atacantes.

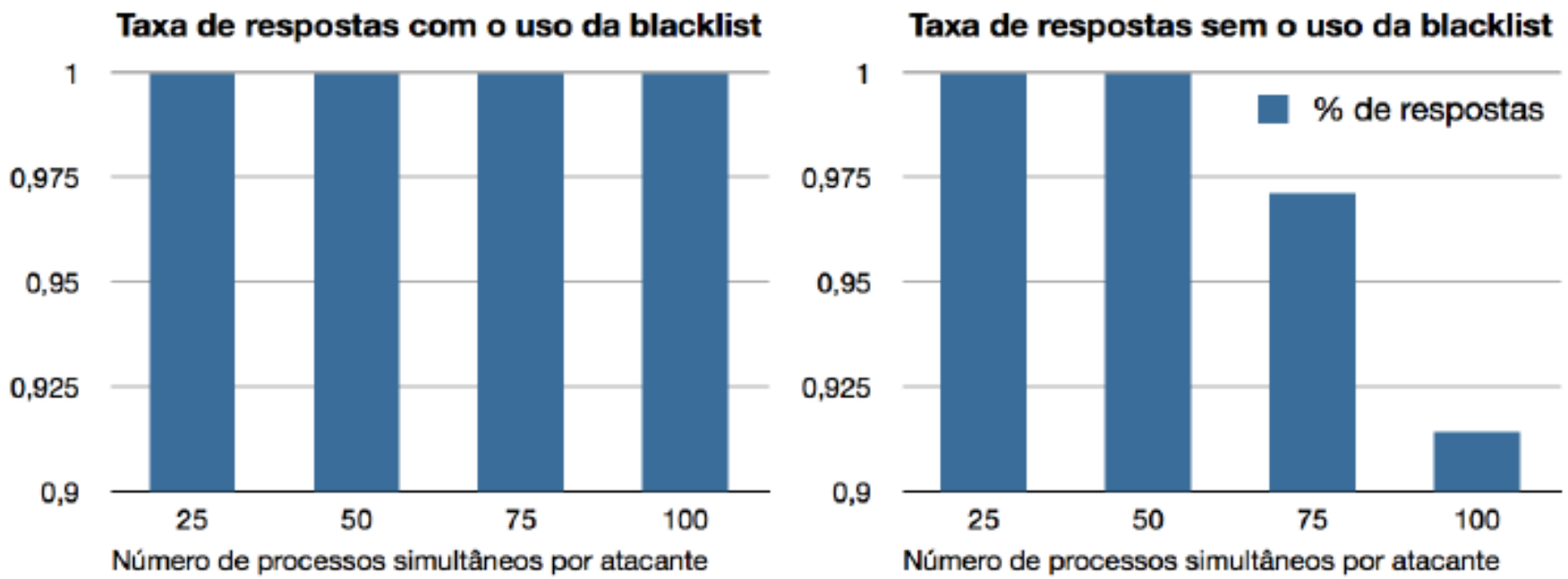

Figura 5. Taxa de respostas do servidor a clientes legítimos

Conclusão

- Os resultados alcançados nas experimentações demonstram a validade da solução proposta, pois conseguem direcionar o tráfego legítimo de modo satisfatório, impossibilitando os atacantes de acessarem a nova instância criada. Mecanismos mais robustos para a checagem da blacklist em níveis mais baixos e otimizados serão desenvolvidos como trabalhos futuros, complementando a solução atual. Além disso, experimentos em maiores escalas tanto no ataque DDoS como também nos recursos alocados às instâncias da aplicação serão realizados.