TÉCNICO LISBOA

DDoS Attack Mitigation

Segurança Informática em Redes e Sistemas

Grupo A35 (Alameda)

Miguel Marques
83532

Pedro Figueiredo
83545

Rui Ribeiro
83562

Problema

Um ataque *Denial of Service* (DoS) é caraterizado por enviar um número intolerável de pedidos para uma máquina que se encontre acessível, numa tentativa de sobrecarregar este sistema, levando a que este deixe de conseguir processar qualquer tipo de pedidos ou mesmo ficando *offline*, até que seja possível filtrar os pedidos maliciosos. No entanto, a filtração de pedidos pode ser mais complicada se o ataque em questão for distribuído (*Distributed Denial of Service* ou DDoS), ou seja, se os pedidos forem enviados por várias máquinas diferentes ao mesmo tempo.

Num ambiente vulnerável, este ataque pode levar a que determinados serviços que são prestados através da internet fiquem indisponíveis por tempo indefinido, não conseguindo manter o seu funcionamento normal. Por isso, é necessária uma solução para permitir que o serviço se mantenha ativo em cenários como este e, principalmente, para garantir a qualidade do serviço.

Exemplificando, uma rede de pagamentos que seja alvo deste tipo de ataque e que fique desligada da internet durante algumas horas, poderá não só perder clientes como sofrer prejuízo monetário.

Os alvos da *botnet* são primariamente *IoT devices*, que são normalmente mais vulneráveis e possuem recursos limitados. Como tal, é necessário ter em conta estes parâmetros para aumentar a eficiência dos ataques.

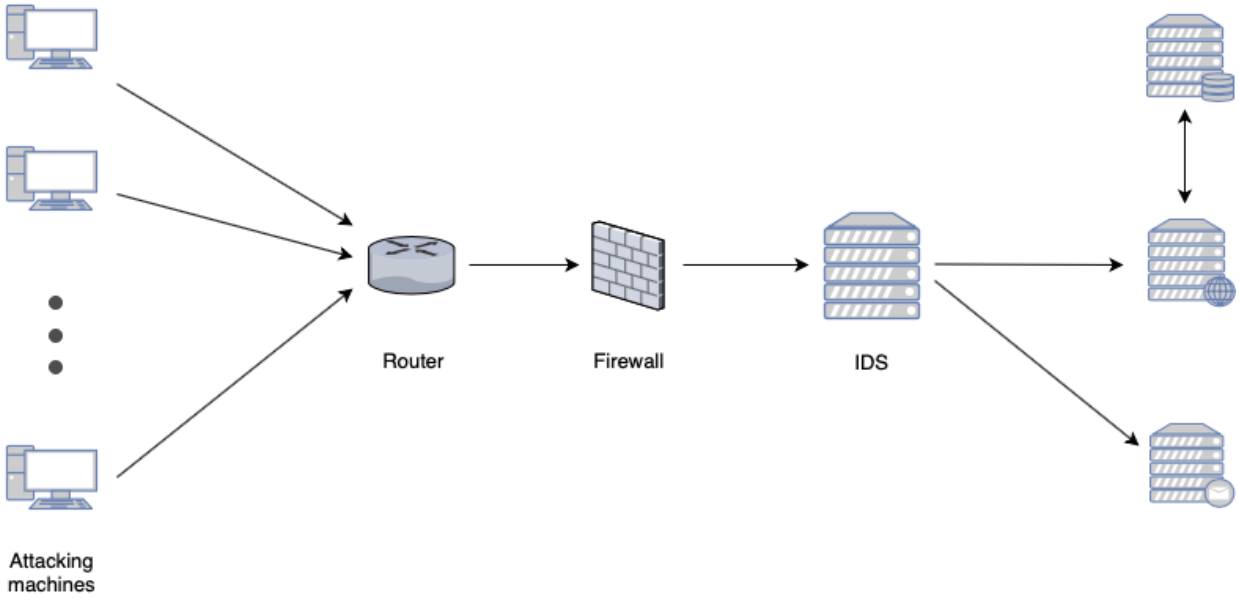
Requisitos

- O sistema de destino deve identificar determinados padrões de tráfego e distinguir pedidos legítimos de pedidos feitos por atacantes.
- O sistema deve conseguir bloquear ou mitigar pedidos fraudulentos.
- O cliente tem que assegurar que mecanismos básicos de defesa, tais como manter o sistema atualizado e utilizar *passwords* seguras.
- Monitorizar o tráfego com destino à infraestrutura em causa.
- Os *edge devices* devem filtrar tráfego à entrada da infraestrutura.
- O Sistema de Prevenção de Intrusões deve fornecer à firewall informações relativas ao tráfego suspeito.

Ferramentas

- Python libraries (well-tested): *nmap*, *telnet*, *ssh*, *socket*, *threading*
- *Snort* (tested)
- LOIC (simulador de DDoS, tested)
- *IPFire* (firewall, tested)
- *dnsmasq* (well-tested)

Proposta de Solução



Versão básica

Nesta versão apenas é considerado o lado atacante.
Vamos começar por desenvolver uma bot*net que* tem 2 funcionalidades:

1. Realizar ataques DDoS enviando pacotes UDP para portos aleatórios da infraestrutura de destino (UDP f*looding)*
2. Auto-replicar-se na rede onde se encontra:
   a. Encontra todos os *hosts* nessa rede;
   b. Tenta ganhar acesso testando passwords *default* (p.e., *admin:admin*);
   c. Copia-se para dentro das máquinas vulneráveis e executa esse script.

Versão intermédia

Observando o diagrama acima, vai ser criado um servidor com Sistema de Prevenção de Intrusões (IPS) onde vai ser configurado o *Snort* (IDS). Deste modo podemos aliviar recursos de servidores que prestam serviços, como, por exemplo, o servidor web. O S*nort t*erá como funcionalidade analisar e detetar tráfego e este servidor irá posteriormente bloquear a entrada de pacotes cuja origem está assinalada como sendo suspeita.

Versão avançada

Nesta última versão, vamos explorar os recursos dos *edge devices.* Vamos configurar o router para adotar mecanismos de defesa básicos: *rate limiting,* descartar pacotes malformados e diminuir o UDP *flood threshold.* O IPS terá regras para detectar fluxo anormal de tráfego em

UDP bem como verificar se o endereço IP da origem é legítimo. Se o IP for *spoofed*, o IPS descartará os pacotes, caso contrário, irá comunicar com a Firewall para que esta impeça que o IP em questão envie mais pacotes.

Plano de Trabalho

	Miguel Marques	Pedro Figueiredo	Rui Ribeiro
29/10 - 2/11	Aprofundamento de conhecimentos relativos às ferramentas a utilizar.	Aprofundamento de conhecimentos relativos às ferramentas a utilizar.	Aprofundamento de conhecimentos relativos às ferramentas a utilizar.
5/11 - 9/11	Preparação da infraestrutura da *botnet*.	Preparação da infraestrutura da *botnet*.	Elaboração do script *botnet*.
12/11 - 16/11	Configuração do servidor a atacar. Teste e *tweaks botnet*. Versão básica.	Teste e *tweaks botnet*. Versão básica.	Teste e *tweaks botnet*. Versão básica.
19/11 - 23/11	Configuração do IPS e do *Snort*. Versão intermédia.	Configuração do IPS e do *Snort*. Versão intermédia.	Configuração do IPS e do *Snort*. Versão intermédia.
26/11 - 30/11	Configuração avançada do *router*. Versão avançada.	Comunicação entre o IPS e o *firewall*. Versão avançada.	Configuração avançada do *router*. Versão avançada.
3/12 - 12/12	Últimas modificações aos sistemas de mitigação. Testes e *tweaks* finais (foco no lado da vítima).	Últimas modificações aos sistemas de mitigação. Testes e *tweaks* finais (foco no lado atacante).	Últimas modificações aos sistemas de mitigação. Testes e *tweaks* finais (foco no lado da vítima).