Entendendo o que é Negação de Serviço (Denial of Service – DoS)

A negação de serviço é um ataque que visa impedir a comunicação normal com um recurso, desabilitando o próprio recurso ou desabilitando um dispositivo de infraestrutura que forneça conectividade a ele. O recurso desativado pode ser na forma de dados do cliente, recursos do site ou um serviço específico, por exemplo. A forma mais comum de DoS é inundar uma vítima com tanto tráfego que todos os recursos disponíveis do sistema estarão sobrecarregados e incapazes de lidar com pedidos adicionais. O invasor inunda a rede de vítimas com quantidades extremamente grandes de dados inúteis ou solicitações de dados, esmagando a rede e tornando-a inútil ou indisponível para usuários legítimos.

Então, quais são os sinais de um potencial ataque DoS?

- Falta de disponibilidade de um recurso
- Perda de acesso a um site
- Lentidão
- Aumento de e-mails de spam

As vítimas típicas dos ataques DoS variam de recursos de propriedade do governo a fornecedores on-line e outros, e a intenção do ataque é geralmente o fator decisivo em termos de como o alvo será atacado. Considere alguns exemplos simples para lhe dar uma idéia do impacto de um ataque de DoS bem-sucedido. Um ataque bem sucedido DoS contra a página da Web de uma empresa ou a disponibilidade de recursos de back-end poderia facilmente resultar em uma perda de milhões de dólares em receita, dependendo do tamanho da empresa. Além disso, considere o impacto negativo para a marca e reputação da empresa. Como você pode ver, o impacto de um único ataque DoS

com intenção dirigida específica pode se tornar extremamente prejudicial para a vítima em muitos níveis diferentes.

Outro tema que permeia os ataques DoS, bem como outras formas de ataque, são os hackers que tomam medidas contra um alvo baseado no princípio ou um senso de missão pessoal, que é conhecido como hacktivismo. Hacktivistas são uma ameaça particularmente preocupante, porque seu foco não é necessariamente no ganho pessoal ou reconhecimento; Seu sucesso é medido pelo quanto suas ações maliciosas beneficiam sua causa. Este processo de pensamento se encaixa bem com ataques DoS em que a mensagem que está sendo enviada pode ser deixada para a interpretação ou, mais comumente, ser reivindicada por um grupo ou indivíduo.

Vítimas do DoS

Os ataques DoS resultam em muitas conseqüências. Vejamos alguns exemplos comuns do que é visto no mundo real:

- Web Server comprometido – Um ataque de DoS bem sucedido e o comprometimento subsequente de um servidor web constitui a maior exposição pública contra um alvo específico. O que você vê com mais freqüência é uma perda de tempo de atividade para uma página da web da empresa ou recurso da web.
- Recursos de back-end – Os recursos de back-end incluem itens de infra-estrutura que suportam um recurso voltado para o público, como um aplicativo da Web. Os ataques DoS que eliminam um recurso back-end, como um banco de dados de clientes ou um farm de servidores, tornam essencialmente indisponíveis todos os recursos de front-end.
- Os ataques DoS específicos de rede ou de computador também são iniciados a partir de uma rede de área local, com a intenção de comprometer a própria rede ou comprometer um nó específico, como um servidor ou

sistema cliente.

Tipos de Ataques

Ataques DoS vêm de muitas maneiras, cada um dos quais é fundamental para a sua compreensão da natureza da classe de ataque DoS.

Inundações de solicitação de serviço

Nesta forma de ataque DoS, um serviço como um servidor web ou aplicativo da Web é inundado com solicitações até que todos os recursos sejam usados. Isso seria o equivalente a ligar para o telefone diversas vezes para que eles não pudessem atender quaisquer outras chamadas devido à sua ocupação. Quando um único sistema está atacando outro, é difícil oprimir a vítima, mas pode ser feito em alvos menores ou ambientes despreparados.

As inundações de solicitação de serviço são normalmente realizadas configurando conexões TCP repetidas para um sistema. As conexões TCP repetidas consomem recursos no sistema da vítima até o ponto de exaustão.

Ataque de SYN Flood

Este tipo de ataque explora o handshake de três vias com a intenção de amarrar um sistema. Para que esse ataque ocorra, o invasor forja pacotes SYN com um endereço de origem falso. Quando o sistema de vítima responde com um SYN-ACK, ele vai para este endereço falso, e uma vez que o endereço não existe, faz com que o sistema de vítima espere por uma resposta que nunca virá. Este período de espera prende uma conexão ao sistema porque o sistema não receberá um ACK.

Ataque ICMP Flood

Uma solicitação ICMP requer que o servidor processe a solicitação e responda, consumindo recursos da CPU. Os ataques

no ICMP incluem ataques de smurf, ICMP flood e ping flood, todos os quais se aproveitam dessa situação inundando o servidor com solicitações ICMP sem esperar pela resposta.

Ping da Morte (Ping of Death)

Um verdadeiro clássico realmente, que se originou em meados dos anos 1990, o ping da morte foi um pacote de ping que era maior do que o permitido de 64 K. Embora não seja uma ameaça significativa hoje devido ao bloqueio de ping. Em seu auge o ping da morte era um exploit de DoS formidável e extremamente fácil de usar.

Teardrop

Um ataque de Teardrop ocorre quando um invasor envia pacotes fragmentados personalizados com valores de deslocamento que se sobrepõem durante a tentativa de reconstrução. Isso faz com que a máquina de destino se torne instável ao tentar reconstruir os pacotes fragmentados.

Smurf

Um ataque smurf faz spoof do endereço IP da máquina de destino e envia inúmeros pacotes ICMP echo requests para os endereços de difusão de sites intermediários. Os sites intermediários amplificam o tráfego ICMP de volta para o IP de origem, saturando assim o segmento de rede da máquina de destino.

Fraggle

Um ataque fraggle é uma variação de um ataque smurf que usa UDP echo requestes em vez de ICMP. Ele ainda usa um intermediário para amplificação. Comumente um ataque de fraggle enviam as UDP echo requestes à porta do chargen (gerador de caracteres) dos sistemas intermediários através de um pedido de transmissão. Assim como em um ataque de smurf, o atacante falsifica o endereço IP da vítima como fonte. Cada cliente que recebe o echo para a porta chargen irá, por sua vez, gerar um caractere a ser enviado para a vítima. Uma vez

recebido, a máquina vítima ecoará de volta à porta de carga do intermediário, reiniciando assim o ciclo.

Land

Um ataque Land envia tráfego para a máquina de destino com a fonte falsificada como a própria máquina de destino. A vítima tenta reconhecer o pedido repetidamente sem fim.

Ataques DoS permanentes

A maioria dos ataques DoS são temporários e só precisa ser interrompida, e qualquer bagunça que eles criaram voltará tudo ao normal. No entanto, alguns tipos de ataques DoS destroem um sistema e podem fazem ficar permanentemente off-line.

O Phlashing é uma forma de DoS permanente que envolve empurrar atualizações falsas ou incorretas para o firmware de um sistema alvo. Quando isso é feito, o hardware torna-se inutilizável em muitos casos e deve ser substituído. Quando um sistema é atacado de tal maneira, é dito ser bricked. Em outras palavras, é inútil como um computador e agora é um tijolo.

Ataques no nível de aplicativo

Os ataques em nível de aplicativo são aqueles que resultam em perda ou degradação de um serviço até o ponto em que ele é inutilizável. Esses ataques podem até resultar na corrupção ou perda de dados em um sistema. Normalmente, esses tipos de ataques assumem a forma de um dos seguintes:

- **Flood** – Este ataque oprime o alvo com o tráfego para dificultar ou impossibilitar responder aos pedidos legítimos.
- **Disrupt –** Este ataque geralmente envolve atacar um sistema com a intenção de bloquear ou bloquear um usuário ou usuários, por exemplo, tentando fazer login em um sistema várias vezes para bloquear a conta para

que o usuário legítimo não possa usá-lo.

- **Jam –** Neste ataque, normalmente o atacante está criando consultas SQL para bloquear ou corromper um banco de dados.

Buffer Overflow (Estouro do buffer)

Buffer overflow é uma técnica DoS que tira proveito de uma falha na codificação de um programa, introduzindo mais dados do que o buffer do programa, ou espaço de memória, tem de espaço. Uma vez que o buffer de um programa está no estado de estouro, todas as entradas adicionais gravadas no buffer podem ter consequências negativas, como falhas, problemas de segurança ou outros problemas. Tal como acontece com muitos ataques DoS, a intenção é colocar o programa ou sistema em um estado imprevisível ou inesperado. Isso se relaciona com o estouro de buffer, uma vez que um programa está em um estado inesperado, o potencial para uma condição DoS é extremamente alto.

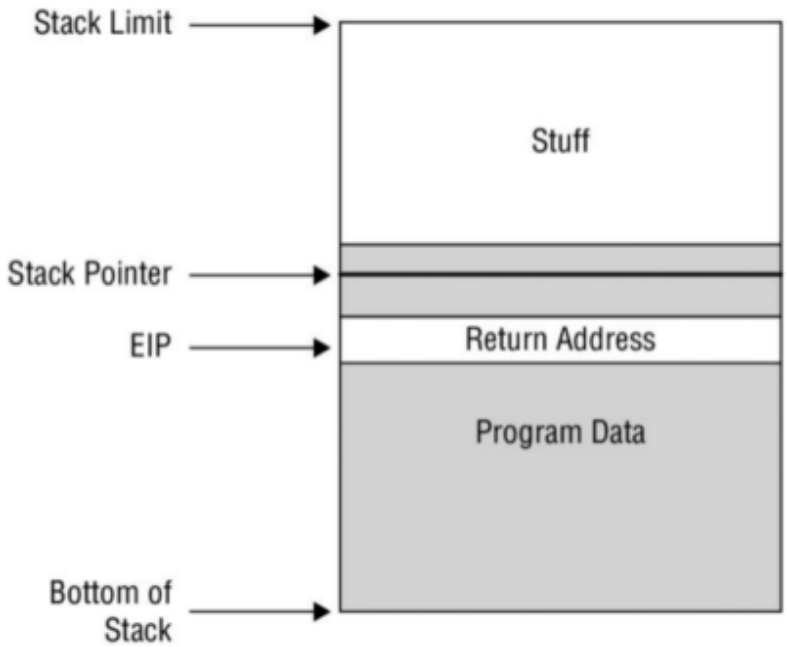
O Heap e o Stack

A Heap e o Stack são duas áreas de memória que um programa usa para armazenamento:

Heap – É um local de armazenamento dinâmico que não tem restrições sequenciais ou um esquema organizacional. Considera-se o conjunto maior de armazenamento livre para programas para serem utilizados conforme necessário. Uma vez que o espaço de memória dinâmico não é mais necessário e o programa recuperou os dados necessários, o espaço ocupado no heap é liberado para uso futuro.

Stack (Pilha) – A pilha refere-se ao pool menor de armazenamento livre: memória alocada a um programa para processamento de curto prazo. Esta é a área de ação principal, onde as variáveis de programa são temporariamente armazenadas, adicionadas e removidas conforme necessário para executar uma

função específica. O nome Stack (pilha) vem do fato de que acessar seus recursos é semelhante em função à forma como você acessa informações de uma pilha de dominós, por exemplo. Você pode ver o valor do dominó superior, você pode remover um dominó da parte superior, e você pode empilhar outro dominó na parte superior. Se você puxar o dominó inferior ou médio da pilha, a pilha inteira vem caindo para baixo. Assim, você está limitado a manipular a pilha de cima para baixo. É assim que uma pilha de programas também funciona. Outro nome para este tipo de acesso é last-in, first-out (LIFO). O último item a ser empilhado é o primeiro item a ser removido. Na linguagem de programação, o termo *push* é usado para descrever a adição de um novo item à pilha e *pop* descreve a remoção de um item. Assim, se um programa quer adicionar ou remover algo da pilha, ele usa as ações push e pop de acordo, e ele faz isso de forma linear de cima para baixo.



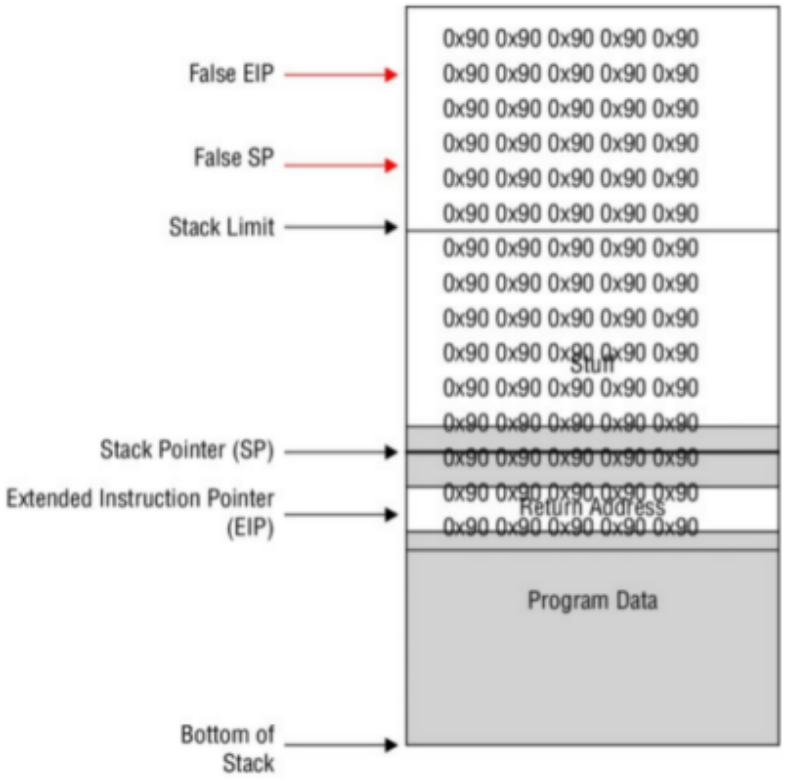
Pilha de um programa básico

A ideia para tirar disto é entender como a pilha pode ser “transbordada” e assim criar uma condição DoS dentro do programa ou sistema. Conhecer os conceitos básicos de como a pilha é usada dará a ideia de como pode ser comprometido.

Vamos ver alguns conceitos-chave que serão importantes:

Smashing the Stack – Refere-se ao uso de buffer overflow para comprometer a integridade da pilha e obter acesso ao nível do programa para execução de códigos maliciosos.

Consulte a figura da pilha de um programa básico; Quebrar a pilha modifica a operação normal da pilha enviando o excesso de dados à pilha, ultrapassando seus limites normais (se deixado desmarcado). O excesso de dados substitui as variáveis ??legítimas na pilha e redefine o valor do ponteiro de instrução estendido (EIP) para apontar para o código malicioso injetado.



Smashing

A figura merece apenas um pouco mais de explicação, porque pode parecer um pouco confuso neste momento. Vamos ver uma peça de cada vez. Subjacente ao bloco 0x90 está a pilha de programas básicos da primeira figura. Lembre-se de que a primeira figura representa a operação normal, onde as variáveis ??do programa e os dados armazenados permanecem

dentro dos limites de memória normais, que estão entre o ponteiro da pilha (parte superior da pilha) e a parte inferior da pilha. A sobreposição 0x90 na figura do Smashing representa a porção de excesso que foi aplicada ou empurrada na pilha normal. O excesso de dados, que ultrapassou em muito o limite de pilha, colocou a pilha em uma condição de estouro. Uma vez que isso seja alcançado, o ponto de referência do programa para a próxima execução de instrução legítima foi deslocado para o código transbordado do atacante. Neste ponto, o programa executa o código malicioso do invasor com privilégios idênticos aos do programa legítimo original.

NOP Sled – Refere-se ao shellcode (código de máquina) usado em um ataque de buffer overflow que usa vários comandos "No Operation" em um pedaço sequenciado. NOP por si só significa "Sem Operação"; Assim, o que vem seguindo o NOP é uma grande sequência de chamadas sem função de operação. O valor 0x90, que você viu na figura de Smashing, é o valor hexadecimal de uma instrução NOP que se aplica a processadores Intel; Portanto, uma instrução NOP com um valor de 0x90 instruirá um processador Intel para executar um ciclo de um relógio em um processo vazio. Em linguagem simples, 0x90 forçará uma CPU Intel a disparar um único ciclo. Agora, uma série de valores 0x90, como você viu na figura de Smashing, e você terá um grande "padding" na pilha que pode definir o estágio para a execução de código malicioso.

Um resumo rápido é que um programa usa a pilha e o heap para armazenamento. O heap é dinâmico, enquanto que a pilha é linear em operação (superior, inferior, LIFO). O buffer overflow sobrecarrega o heap, excedendo os limites de memória. Isso, por sua vez, cria uma condição imprevisível na qual o sistema operacional agora vê o programa como operando fora de seu espaço de memória alocado. Provavelmente acontecerá um dos seguintes:

- O sistema operacional encerra o programa ofensivo devido ao programa estar operando fora do espaço de memória

alocado.

- O endereço do código malicioso do hacker, que agora reside na pilha transbordada, termina no EIP, fazendo com que esse código seja executado.

Sugestões de livros: