



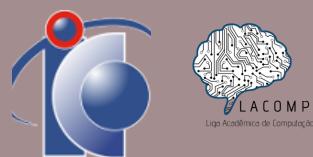
# Info Hacker

RESUMO QUINZENAL OFICIAL DA ACADEMIA HACKER



**01: Buffer Overflow e Heartbleed**

**02: Computação Forense**



# Buffer Overflow e Heartbleed

por Sarah de Lima Domingos

Para aqueles que já tiveram contato com a linguagem de programação C, seja por curiosidade ou obrigatoriedade, dentre os vários desafios os quais você pode ter enfrentado, a manipulação de *strings* em *array* de caracteres pode ter sido a mais perigosa. Na criação de uma simples frase, o uso descuidado do tamanho do *array* para a quantidade de caracteres digitados ocasionará o que chamamos de *buffer overflow*. Esse termo é usado para descrever um “estouro” na memória, ou seja, há uma entrada de dados maior que o espaço reservado.



Diante disso, o principal problema gerado por essa pequena falha é o acesso indevido à memória por um invasor. Isso acontece graças a tentativa do programa em alocar os dados recebidos, mesmo que para isso ele precise sobrescrever em porções da pilha de memória. Dessa forma, um *hacker* pode injetar um código malicioso que modifique a execução do programa, lhe dando total controle de forma silenciosa. Entretanto, essa não é a única forma de explorar essa falha, sendo o *heartbleed* um exemplo catastrófico no mundo da segurança da informação.

Em meados de 2014 foi divulgado um *bug* na biblioteca *open source* OpenSSL. Até então, ela era amplamente utilizada em softwares, como o Apache e Nginx, e sites, como o Yahoo e StackOverFlow. Com a descoberta do problema, que aconteceu de forma independente cerca de dois anos antes da sua ampla divulgação, os desenvolvedores agiram de forma rápida para lançar uma versão corrigida e diminuir os prejuízos. Apesar disso, estima-se que muitos dados foram indevidamente acessados, não havendo uma estimativa exata devido à forma silenciosa e limpa de exploração do *bug*.



Para entender melhor a razão do nome *heartbleed*, ou sangramento, é preciso falar um pouco do conceito de *heartbeat*. Como o nome sugere, *heartbeats* são “batimentos cardíacos” feitos entre máquinas cliente-servidor para que ambas saibam que a conexão está estável e ativa. Isso acontece de forma síncrona como uma conversa, pois, por exemplo, o cliente envia “oi” para o servidor, este, por sua vez, responde “oi” e assim sucessivamente. Ainda que pareça algo simples, um erro em uma única linha de código foi capaz de expor dados sensíveis de forma indetectável:

**memcpy(bp, pl, payload);**

Com esse comando, o servidor copia a “batida” enviada pelo cliente para enviá-la de volta. Todavia, a função `memcpy()` não verifica o tamanho da “batida”, sendo possível enganar o servidor e fazê-lo enviar dados que estavam contidos em sua memória, podendo ser chaves de criptografia, nomes de usuários, senhas etc. Assim, ao enviar um arquivo vazio, o *hacker* pode driblar o sistema fazendo-o pensar que existe uma mensagem no arquivo, porém, no momento de copiar essa mensagem para devolvê-la o servidor não encontra nada, sendo forçado a retirar de sua memória informações do mesmo tamanho ao qual ele acredita que foi o *heartbeat*. Dessa forma, o *heartbleed* é executado discretamente, visto que, por ser erro de código, não ativa nenhum sistema de segurança.

Por ser um problema antigo, muitas forças já foram movidas para erradicar o *buffer overflow* e suas consequências, como linguagens de programação com mecanismos internos de segurança e a ampla divulgação de sua existência para novos programadores. Apesar disso, novos *bugs* do gênero continuam sendo descobertos, sendo imprevisível o alcance de sua exploração por *hackers*. Portanto, é sempre bom seguir bons costumes de programação, ainda que, em um primeiro momento, pareça algo irrelevante, pois é preciso lembrar que não há limites para a criatividade humana.

## Referências:

- <https://andreybleme.com/2019-07-06/etendendo-explorando-buffer-overflow/>
- <https://www.cloudflare.com/pt-br/learning/security/threats/buffer-overflow/>
- <https://gitbook.ganeshicmc.com/pwning/buffer-overflow-e-a-stack>
- <https://heartbleed.com/>
- <https://gizmodo.uol.com.br/como-funciona-o-heartbleed/>

# Computação Forense

por Yanka Ribeiro

Não deve causar surpresa ao leitor que a computação está inserida nos mais variados campos do nosso cotidiano. Hoje, trarei destaque à sua aplicação na ciência forense e como isso ajuda na resolução de crimes. Ao final, um pequeno vislumbre do salário médio dos investigadores forenses brasileiros.

A ciência forense é a coleta e análise, por peritos especialistas, de vestígios/evidências encontrados em uma cena de crime. Como os vestígios podem variar de formato e fonte (computador, corpo, documento, líquido, e etc), várias áreas da ciência podem estar envolvidas. Portanto, a computação forense estará voltada para a análise de informações de equipamentos eletrônicos, seja celular, computador, tablet, e-mails, redes sociais, bem como informações providas de logs de acesso, servidores, redes físicas e etc. Evidências decorrentes de investigações forenses computacionais têm sido usadas em vários casos importantes e são amplamente aceitas no Sistema Judiciário e nos Tribunais do Brasil, Estados Unidos e Europa.

Embora seja mais frequentemente associada à investigação de uma ampla variedade de crimes de informática, a computação forense também pode ser usada em processos civis. A disciplina envolve técnicas e princípios semelhantes à recuperação de dados, mas com diretrizes e práticas adicionais projetadas para criar uma trilha de auditoria legal. Existem três tipos de peritos nesta área, resumidos a seguir.

- *Perito oficial*: Foi o que prestou concurso, federal ou estadual, e trabalha para algum órgão da lei.
- *Perito judicial*: Especialistas na sua área de atuação que não fazem parte de um órgão da lei, mas que realizaram um cadastro estadual e assim podem ser nomeados, por juízes, para atuar em perícias.
- *Perito particular*: Assim como os judiciais, não prestaram concurso. Nesta categoria, são contratados por qualquer uma das partes de um processo para a elaboração de laudos periciais, que podem ser apresentados em tribunal.

Ainda não existe formação específica para área de investigações forenses, porém é fácil encontrar cursos de pós-graduação em segurança da informação, análise de dados, forense digital ou ainda perícia forense computacional. O salário médio de um Perito Criminal no Brasil é superior a R\$9 mil, para uma jornada de 40 horas/semana e com um teto salarial de apróx. R\$29 mil\*\*. Por ser um cargo público, os benefícios adicionais à remuneração podem variar entre órgãos e estados.

## Referências:

[Artigo sobre Computação Forense](#), do CDP - Canal da Perícia

[Artigo sobre Computação Forense](#), do Computação Forense

\*\* [Artigo sobre Salário do perito criminal brasileiro](#), do Indeed BR

## **Info Hacker - Edição 015**

### **27/09/2022**



#### **Autores:**

Sarah de Lima Domingos  
Yanka Ribeiro

#### **Revisão**

Bruno Severo  
Everton Borges



## **Edições Anteriores**

<https://github.com/Academia-Hacker/Info-Hacker>