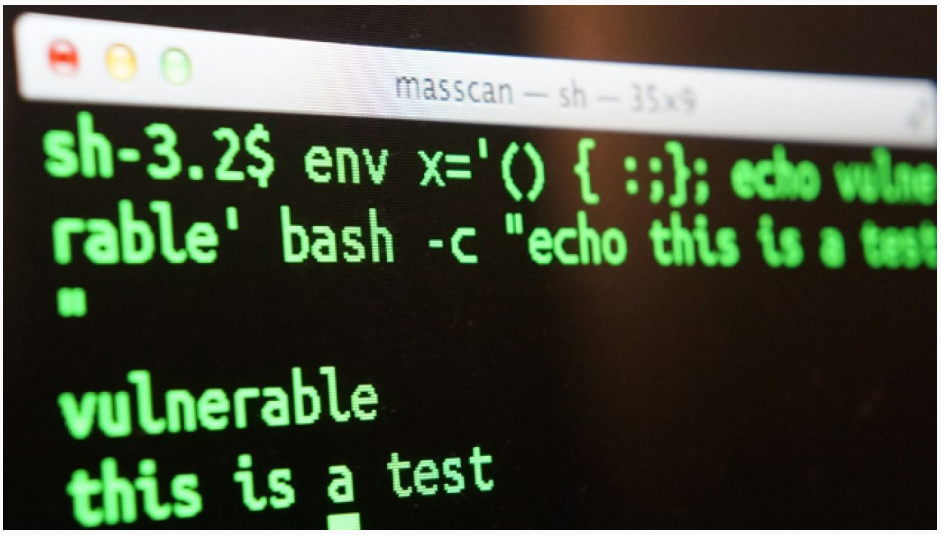
Entenda o Shellshock, a falha no Bash tão grave quanto o Heartbleed

29/9/14, 10H05 • RODRIGO GHEDIN



Stephane Chazelas, um entusiasta de software livre, descobriu uma falha de 22 anos (!) no Bash, interpretador de comandos bastante popular em sistemas *nix, como Linux e o OS X, da Apple.

A falha permite que alguém tome o execute comandos remotamente em máquinas afetadas. Dada a amplitude com que o Bash é usado, de servidores a sistemas embarcados (câmeras fotográficas, roteadores, terminais comerciais), o potencial de danos é tão grande, ou até maior, que o do Heartbleed, outra falha encontrada no OpenSSL no começo do ano.

O Instituto Nacional de Padrões e Tecnologia dos EUA atribuiu ao Shellshock nota 10, a máxima na escala, em termos de gravidade, impacto e exploração, e para piorar o mesmo órgão disse que a falha é de baixa complexidade, o que significa que pode ser facilmente usada.

A origem do Shellshock e a questão do open source

O Bash é um interpretador, o mais popular em sistemas *nix. Ele foi criado em 1989 por Brian J. Fox e, cinco anos depois, passou a ser mantido por Chet Ramey, um arquiteto de software que trabalha no Bash como um hobby, sem receber nada, há 22 anos.

Em entrevista ao New York Times, Ramey confessou que pode ter introduzido o Shellshock, sem querer, em 1992 (!) ao implementar um novo recurso no Bash.

Apesar de ter o código aberto, ou seja, disponível para qualquer um analisar, estudar e encontrar erros, levou mais de duas décadas para que o Shellshock fosse descoberto. O Heartbleed também passou um tempão despercebido.

Esses dois incidentes trouxeram críticas ao modelo, sempre defendido como mais seguro justamente por contar com auditores independentes e interessados, ao contrário do software fechado, cujo código-fonte é guardado a sete chaves pelas empresas que os produzem.

Também ao NYT, Ramsey minimizou a polêmica: “Não acho que esse seja um problema do código aberto. O software está dominando o mundo. A má notícia é que software é difícil e complexo.”

Com toda essa exposição, o Bash pode deixar de ser um hobby a Ramsey e outros colaboradores e passar a receber mais apoio financeiro. A Fundação Linux já sinalizou a possibilidade de injetar dinheiro no projeto, a exemplo do que fez com o OpenSSL após os eventos do Heartbleed.

Como a falha atua?

O Bash permite criar variáveis de ambientes, um tipo de código global que pode ser invocado posteriormente pelo próprio usuário ou automaticamente, dentro de um script. Digamos que você queira chamar a palavra “vermelho” em vários locais; em vez de escrevê-la sempre, é possível definir uma variável $COLOR que se converterá em “vermelho” sempre que é executada.

O Shellshock atua aí. Ele permite transformar uma string de texto (“vermelho”, no exemplo) em um comando e, pior, executa-o mesmo quando a variável *não* está presente no comando executado. Tomo emprestadas algumas telas da Vox, que explica muito bem a situação:

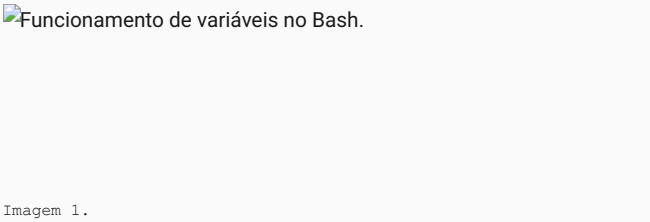


Imagem 1.

Na *imagem 1*, são definidas as variáveis `$COLOR` e `$NAME`. Note que ao executar um comando o texto recorre a elas e que, na saída (as linhas sem `$` no começo) o Bash interpreta a variável e a transforma no texto salvo anteriormente. Aqui, tudo ok.

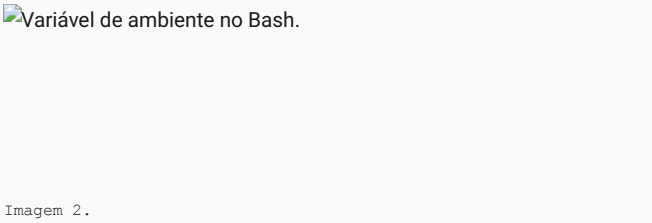


Imagem 2.

Temos, na *Imagem 2*, uma variável de ambiente (indicada pelo `env` no início da linha). Novamente, tudo ok aqui: a variável é chamada pelo comando `echo` e interpretada corretamente pelo Bash.

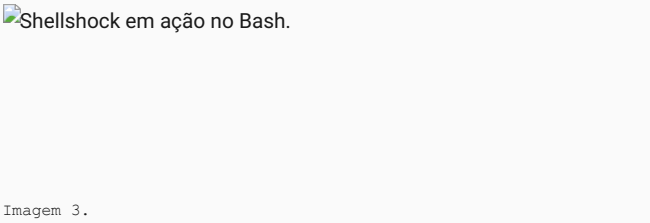


Imagem 3.

Problemas na *Imagem 3*. O comando `'() { :;}; echo vulnerable'` explora a falha. Note que o comando `echo` sequer menciona a variável `$COLOR`, mas o "vulnerable" aparece na saída. Esse é um exemplo simples, uma mera string de texto que aparece onde não deveria. Em situação real, poderia ser um comando destruidor, algo que transforme a máquina em uma zumbi, emissora de spam, mineradora de bitcoins ou sabe-se lá mais o quê.

Ou seja, alguém que queira explorar sistemas vulneráveis poderia escrever um script que acrescente comandos inadvertidos que seriam executados, automaticamente, quando qualquer outro comando fosse disparado.

O perigo maior é que o simples acesso a um servidor preparado para explorar o Shellshock com uma máquina vulnerável pode causar estragos. Quando uma máquina cliente acessa um servidor web, interpretadores como o Bash são usados no processamento da página a fim de determinar características e entregar partes dinâmicas dela — pense, por exemplo, em sites que têm versões desktop e móvel; eles servem a versão correta após interpretar sinais do pedido pelo Bash ou algum equivalente.

O que posso fazer? Como isso me afeta?

Como no caso do Heartbleed, pouca coisa. A boa notícia é que entre sistemas domésticos, apenas o OS X, da Apple, é um dos que *podem* ser afetados. (A Apple diz que a "grande maioria" dos usuários está livre; veja aqui se o seu Mac está vulnerável e dicas paliativas para amenizar o problema.) Windows, Windows Phone e Android não usam interpretadores ou, se usam, são interpretadores diferentes do Bash. O trabalho pesado ficará a cargo, pois, de profissionais, gente de quem se espera conhecimento e competência para sanar esse problema o quanto antes.

A parte mais séria é em servidores web e equipamentos embarcados. No primeiro, porque depende da implementação da correção por parte dos sysadmins. As principais distribuições Linux já oferecem patches que solucionam o bug, mas ninguém garante se todos farão o trabalho, nem quando.

Em equipamentos embarcados, como roteadores, a situação é ainda mais delicada. Há pouco interesse das fabricantes em lançarem atualizações para o firmware deles, e o processo de atualização é mais complicado e arriscado que atualizar o Android do seu smartphone.

Indiretamente isso pode afetar pessoas comuns. Imagine um roteador hackeado que direcione o site do banco a uma cópia idêntica e maliciosa, exibindo-a no domínio oficial. Ou, então, um servidor que processe pagamentos e que, após comprometido pelo Shellshock, passe a direcionar as informações de cartão ou mesmo meta dados a um terceiro. Ou ainda servidores web com endereços DNS modificados, ou seja, entregando páginas diferentes da requisitada mas com o o endereço "certo" exibido no navegador.

Software é escrito por humanos e mesmo com ferramentas auxiliares para minimizar erros, eles ainda são suscetíveis. Ainda ouviremos falar bastante do Shellshock e, por ora, isso é tudo o que você precisa saber dele.

Foto do topo: Robert Graham/Twitter. (Edição: Engadget).

« YAHOO ANUNCIA ENCERRAMENTO DO YAHOO

COM PHOTOSHOP POR STREAMING, CHROMEBOOKS PODEM FICAR MAIS ATRAENTES »

Newsletter

O **Manual** no seu e-mail. Três edições por semana — terça, sexta e sábado. Grátis. Cancele quando quiser.

E-mail Cadastrar

Acompanhe

RSS Bluesky Flipboard

Google News LinkedIn Mastodon

PeerTube Telegram Tumblr

YouTube Todos os canais sociais

Deixe um comentário

⚠️

É possível formatar o texto do comentário com HTML ou Markdown. Seu e-mail não será exposto. Antes de comentar, leia isto.

Comentário *

Nome *

E-mail *

Salvar dados para futuros comentários

Inscrever-se para receber comentários por e-mail? Você também pode inscrever-se sem comentar.

Apenas respostas a meus comentários

Publicar comentário

2 comentários

Luis Henrique

29/9/2014 ÀS 10:22

É tanta informação que nem sei o que comentar… Acho que a sensação d e"segurança" na web é coisa do passado, e alguns hábitos deverão ser modificados em virtude dessa falha. Quando começarem os ataques em massa (se começarem), vamos ver em prática os efeitos do Shellshock.
O que mais me preocupa mesmo é esse redirecionamento de páginas. Se hoje o usuário mais leigo cai em armadilhas óbvias, imagina com esse toque de sofisticação.

RESPONDER

Vagner "Ligeiro" Abreu
29/9/2014 ÀS 10:50

Em resumo desta informação, a parte necessária a ser entendida é o fato de que um "desleixo" (por assim dizer) na criação de um comando resulta em uma possibilidade de que qualquer pessoa possa fazer um uso de um sistema.

Quanto a questão de segurança na web, noto que desde sempre é falha: desde os primórdios as pessoas se acostumaram a usar um sistema sem tanto controle ou proteção. Os primeiros comunicadores, os primeiros sistemas de e-mail… tudo de alguma forma é interceptável. Até eu, podemos dizer.

Basicamente, é que nem andar na rua: NUNCA você estará 100% protegido de alguma ocorrência. Por mais que uma pessoa tente ficar anônima, sempre haverá alguma forma de reconhece-la.

Acho que o problema não está nem na questão da criação de software, mas no fato de quem cria e trabalha com softwares. "O ser humano é um animal triste". Difícil confiar em qualquer pessoa, e mais difícil ainda controlar o ímpeto de alguns, que é de infelizmente "fazer o mau", prejudicar o próximo. E a informática acabou deixando o espaço livre para este tipo de pessoa. O mau tem valor monetário hoje. O mau enriquece…

RESPONDER