Parte IV

DIFERENCIAÇÃO

Diferenciação

- É a comparação de um programa, biblioteca ou outro arquivo, antes e depois de uma ação.

- Usada com frequência durante a pesquisa de segurança.

Diferenciação

- Pode ser feita em níveis de disco, arquivo e banco de dados.
- Em nível de disco, pode-se descobrir quais arquivos foram modificados.
- Em nível de arquivo, pode-se descobrir quais bytes foram alterados.

Diferenciação

- Em nível de banco de dados, pode-se descobrir quais registros são diferentes.

- Com diferenciação, pode-se descobrir como manipular os dados fora de uma aplicação para a qual, um arquivo, ou uma parte de arquivo ou registros de um banco de dados são intencionados.

Diferenciação

- O utilitário *diff* : UNIX da AT&T.
- O termo *diffing* pode ser definido como o uso do utilitário *diff*.
- A partir de uma comparação podemos reunir informações para finalidades tais como determinar o que mudou de uma revisão de software para a seguinte, se um binário é ou não diferente de outro, ou como um arquivo de dados foi alterado de uma operação para outra.

Diferenciação

- Exemplo Figura 5.1 e Figura 5.2, pag. 112-113, livro Ryan Russel.

- Exemplo Figura 5.3, pag. 114 :

  Saída de uma sessão *diff*.

Por que diferenciar ?

- Determinar a parte do arquivo do item de interesse: por exemplo, um formato de senha localizada num arquivo binário, pode-se saber que parte do arquivo representa a senha.

- Fazer mudanças diretamente no arquivo sem passar pela aplicação.

Por que diferenciar ?

- Decodificar informações ao invés de alterá-las.
- Determinar quando uma mudança ocorre e possivelmente deduzir a ação que a causou.

Por que diferenciar ?

- O processo de descoberta da pesquisa de vulnerabilidades de segurança.
- Pode expor uma vulnerabilidade quando o fornecedor de software tiver liberado um anúncio vago referente a um reparo de segurança.

Por que diferenciar ?

- Para se detectar problemas que foram resolvidos silenciosamente de uma revisão de um pacote de software para outra.

Examinando o Código-Fonte

- A descoberta de problemas no software fonte aberto são divulgados através de arquivos de *patch* produzidos e distribuídos por fornecedores do UNIX: Linux e FreeBSD.

- Ver o *patch* na Figura 5.4, pag. 116, Ryan Russel.

Examinando o Código-Fonte

- Ver Figura 5.5, pag. 117, Ryan Russel:

  Saída *diff* entre duas versões do programa pwupd.c do FreeBSD.

- Entre a versão mais antiga e a mais atual dos arquivos pwupd.c, podemos ver as mesmas mudanças que estão anunciadas no arquivo de patch da Figura 5.4

Explorando Ferramentas de Diferenciação

- Comparando arquivos no Windows com a ferramenta fc

  C:\windows\COMMAND>fc /?

  /b ou /B, compara arquivos binários;

  /l ou /L, compara arquivos texto ASCII.

Usando o comando *diff* no UNIX

- Capacidade limitada de comparação binária, mas útil para comparar arquivos-texto.

- Ver exemplos do comando *diff*, pag.123, Ryan Russel.

Usando o comando Diff (UNIX)

- Para enviar para alguém uma pequena mudança em um arquivo-texto, especialmente para o código-fonte, poderá enviar um arquivo *diff* (saída diff).

- Quando alguém postar uma vulnerabilidade para uma lista de discussão, referente a uma parte do software fonte aberto, o remetente pode usar a saída diff para consertar o fonte.

Usando o comando Diff

- Uma versão do programa diff para Windows está disponível no projeto Cygwin.

- O projeto Cygwin é um projeto com a finalidade de levar diversas ferramentas GNU e outras baseadas no UNIX, para a plataforma Windows.

Usando o comando Diff

- Todo software GNU está incluído na GNU Public License (GLP).
- Uma versão diff para Windows está disponível em: http://sourceware.cygnus.com/cygwin
- Existe o utilitário Windiff nos *Resource Kits* do Windows NT e 98.

Trabalhando com Editores Hex

- Mudança de arquivos binários.
- Um editor hex é uma ferramenta que permite que o usuário acesse diretamente um arquivo binário sem ter que usar o programa de aplicação ao qual o tipo de arquivo pertence.

Editores Hex

- Hackman
- [N] Curses Hexedit
- Hex Workshop

Hackman (Fig. 5.9, pag. 125)

- Gratuito
- Windows
- Inclui uma lista longa de recursos (pesquisa, colagem, calculadora hex, disassembler, funcionalidade em linha de comandos).
- Fácil de usar
- Funcionalidade de um editor hex básico
- Ótima interface GUI.
- http://www.tecnologismiki.com/hackman

[N] Curses Hexedit

- Obtido em:

- http://cewf.cc.utexas.edu/~apoc/programs/c/hexedit

Hex Workshop

- Windows e se edita muito em Hex ...
- Obtido em: http://www.bpsoft.com
- Editor hex comercial
- BreakPoint Software
- Funções aritméticas, conversor de base, uma calculadora, uma calculadora de soma de verificação, ...

Ferramentas de monitoração do Sistema de Arquivos

- Trabalham sobre um grupo de arquivos, como uma partição, letra de unidade ou diretório.
- Possibilita a seleção de um arquivo para que possa ser trabalhado.

Ferramentas de monitoração do Sistema de Arquivos

- Depois que um programa realiza alguma ação, poder-se-á saber o que foi alterado.
- Quase sempre a ação terá alterado um arquivo no disco, mas qual?

Comparando atributos do arquivo

- Tirar proveito dos atributos de arquivo embutidos no sistema de arquivos.
- Determinar quais arquivos foram simplesmente modificados.
- Pag. 128-129, Ryan Russel

Usando o atributo *archive*

- Para Windows/DOS
- Sistema de Arquivos FAT (File Allocation Table) inclui um atributo de arquivo chamado *bit archive*.
- Finalidade: determinar quando um arquivo foi modificado desde o último backup, precisando receber novo backup.

Examinando somas de verificação e hashes

- Existe um problema quando se conta com os atributos de arquivo para determinar se os arquivos foram alterados.
- Os atributos são fáceis de se falsificar.
- É simples definir uma arquivo para qualquer tamanho, data, hora que se desejar. ... ...

Examinando somas de verificação e hashes

- Vírus e Cavalos de Tróia fazem algo como isso para se esconderem.
- Alternativa: utilizar algoritmos de soma de verificação ou hash criptográfico sobre os arquivos e armazenar os resultados.
- Os algoritmos de soma de verificação também são fáceis se fraudar, se o programa de invasão souber qual algoritmo está sendo usado sobre os arquivos.

Hash Criptográfico

- Recomendação: usar um algoritmo de hash criptográfico forte (um algoritmo que recebe qualquer comprimento de entrada e mescla a entrada para produzir uma saída pseudo-aleatória de largura fixa).

- A propriedade essencial de um algoritmo de hash, é que a chance de dois arquivos chegarem ao mesmo valor são infinitamente pequenas.

- Portanto, não é possível que um atacante produza um arquivo diferente que chegue ao mesmo valor. Os valores de hash normalmente possuem de 128 a 160 bits de extensão, sendo muito menores do que um arquivo típico.

Hash Criptográfico

- Pode-se usar hash para determinar quando os arquivos foram alterados, mesmo que estejam tentando ocultar o fato.
- Percorremos os arquivos em que estamos interessados e apanhamos um valor hash para cada um.
- Faz-se a mudança.
- Depois, calcula-se os valores de hash novamente, procurando-se as diferenças. Os atributos de arquivos podem ser os mesmos, mas se o valor de hash for diferente, o arquivo será diferente.