Analyzing the Effects of Disk-Pointer Corruption

João Dias e João Isento

Estrutura da Apresentação

- Introdução;
- Sistemas de Ficheiros;
- Motivações;
- Type-Aware Point Corruption;
- Resultados;
- Conclusões

Introdução

- A disponibilidade a longo prazo de dados armazenados num sistema informático depende de quão esses sistema salvaguarda os caminhos de acesso a esses dados, isto é, apontadores.

- Infelizmente, estes apontadores estão susceptíveis a ficarem corruptos por várias razões:

  - Controladores de disco;
  - Falhas de transporte;
  - Bugs de Sistema;

Introdução (Cont.)

- Hoje os Sistemas de Ficheiros usam várias técnicas de protecção contra a corrupção destes apontadores.
  - ReiserFS;
  - JFS;
  - Windows NTFS;
  - FAT;
  - HFS;
  - UFS;
  - Veritas File System;
  - ZFS;

Sistemas de Ficheiros

- Um sistema de ficheiros é um conjunto de estruturas lógicas e de rotinas, que permitem ao sistema operativo controlar o acesso ao disco rígido.

- Realiza algumas verificações tais como:
  - ***type checking***, isto é, garante que o bloco que está a ser lido contém o tipo de dados esperado;
  - ***sanity checking*** que verifica que valores particulares nas estruturas de dados seguem certas restrições.
  - Para a recuperação de falhas, alguns sistemas, como o NTFS, replicam as estruturas de dados chave, possibilitando a recuperação dessas estruturas.

Sistemas de Ficheiros - NTFS

- O **NTFS** (*New Technology File System*) é o sistema de arquivos standard para o Windows NT e seus derivados (2000, XP, Vista, 7,Server -- 2003 e 2008);

- Desenvolvido para superar as limitações do sistema FAT, utiliza algumas estruturas em 64 bits (p.ex., para endereçamento de blocos – *clusters*);

- A NTFS surgiu como forma de garantir um nível de segurança maior permitindo proteger arquivos que estão a ser executados pelo sistema, e também para promover permissões de acesso conforme a ACL do Windows.

Motivação

- A corrupção dos discos tem origem na pilha de armazenamento;
- Podem ser provocados:
  - Sistemas de ficheiros;
  - Firmware dos discos;
  - Componentes eléctricos, mecânicos do disco;
  - Drivers dos dispositivos;
  - Controlador de bus;
  - Bugs de Software
  - Layer de Transporte;

Motivação (Cont.)

- Drivers dos dispositivos:
  - Podem esconder dados corrompidos que são escritos directamente para um local errado, ou podem informar que os dados foram escritos quando na verdade não foram;
- Controlador de Bus:
  - Podem indicar incorrectamente que os pedidos feitos ao disco estão completos ou até mesmo trocar os bits de estado dos dados;
- Bugs de Software
  - Podem fazer com que o sistema de ficheiros escreva os dados de forma incorrecta para o disco;

Motivação (Cont.)

- Num estudo envolvendo 1.53 milhões de discos, foi concluído que 0,66% do drivers SATA e 0,06% dos drivers FC desenvolvem corrupções de disco ao fim de 17 meses de uso.

Motivação (Cont.)

- Algumas corrupções de disco provocam mais danos que outras;
  - Se um bloco de dados de um ficheiro está corrompido, apenas a aplicação que lê esse ficheiro é atingida;
  - Se um bloco de disco que pertence ao sistema de ficheiros está corrompido, então todo o sistema de ficheiros pode ser afectado;
    - Se um sector de boot está corrompido, então o sistema de ficheiros pode não ser montado;
    - Por outro lado, se um apontador corrompido se refere a dados que pertencem a uma estrutura de dados diferente daquela que queremos, faz com que esses mesmo dados sejam reescritos e possam ficar corrompidos;

Type-Aware Pointer Corruption (TAPC)

- O estudo da corrupção dos apontadores dos discos é difícil pois é impossível corromper todos os pontos do disco para todos os valores possíveis num espaço de tempo considerado razoável;

- Para resolver esta dificuldade foi criado uma nova técnica de injecção de falhas denominada Type-Aware Pointer Corruption (TAPC)

TAPC (Cont.)

- Reduz o espaço de exploração, assumindo que o comportamento do sistema depende de apenas de dois tipos:
  - O tipo de apontador que está corrompido;
  - O tipo de bloco para que o apontador faz referência depois de corrompido;
- Com esta técnica é possível cobrir a maior parte dos casos de interesse no espaço de exploração;

TAPC - Terminologias

- Container:
  - É o bloco do disco em que o apontador do disco se encontra. Para corromper o apontador é necessário modificar os conteúdos do container;
- Target Original:
  - É o bloco do disco para o qual o apontador do disco deve apontar. Isto é, é o bloco do disco apontado quando não existe corrupção;
- Target Corrupt:
  - É o bloco do disco que está a ser apontado por um apontador corrompido;

TAPC – Corruption Framework

- Foi desenvolvida para executar sem a necessidade de um código fonte de um sistema de ficheiros;
- Consiste numa corrupter layer que injecta corrupção nos apontadores e ainda num equipamento de teste que controla a experiência;
- Esta layer foi implementada usando uma uma driver de filtro Windows para o NTFS e um pseudo-dispositivo para o Ext3;

Corruption Framework (Funcionamento)

- A Layer corrompe os apontadores do disco e observa o tráfego do disco;
- Por sua vez o corrupter tem conhecimento do sistema de ficheiros;
- Por fim, o equipamento de teste executa as operações do file system e controla o corrupter;

Corruption Framework (Exemplificação)

- O equipamento de teste cria um sistema de ficheiros com poucos ficheiros e directorias;
- De seguida, dá instruções ao corrupter para corromper um determinado apontador com um valor específico e executa as operações sobre o ficheiro, para verificar o comportamento do apontador em estudo
  - NTFS -> CreateFile, mount
  - Ext3 -> creat, mount

Corruption Framework (Exemplificação)

- Depois são executadas as operações sobre o ficheiro por um utilizador com permissões limitadas;
  - Leitura;
- É neste ponto que o corrupter intercepta o acesso ao disco realizado pelo sistema de ficheiros e procura pedidos feitos ao container;
- Todos os acesso ao disco, chamadas ao sistema que retornam valores e o log de eventos do sistema são analisados para identificar o comportamento do sistema de ficheiros;

Resultados

- Terminologia:
  - **Detecção:** o sistema de ficheiros identifica a corrupção de um determinado apontador;
  - **Recuperação**: o sistema de ficheiros é capaz de regenerar os dados perdidos usando informação redundante e continua a sua execução sem erros;
  - **Relatório**: o sistema de ficheiros informa a aplicação ou o utilizador que foi descoberto um erro;
  - **"Tentar Novamente" (Retry)**: o sistema repete o conjunto de acessos em disco necessários para a operação de montagem (mount);
  - **Reparação**: o sistema modifica as estruturas de dados corrompidas para continuar a execução.

Resultados (Cont.)

- A **detecção** é essencial para o resto das acções ocorrerem.
- A **recuperação** é a acção ideal que qualquer sistema pode fazer.
- Se esta não for possível, a **reparação** é um método alternativo para a continuação da execução.
- Se uma operação falhar é esperado que seja **reportado** o erro e, se possível, a sua causa.

Detecção

- No NTFS são usados os métodos de type e sanity checking para a detecção de apontadores corrompidos.

- **1ª Observação**: NTFS detecta erros primeiramente através de type checking:

  - Foi observado que o NTFS detecta erros depois de ler o $Target_{corrupt}$ para muitos apontadores. Uma verificação de correspondente estrutura de dados mostra que contêm "palavras chaves" (por exemplo "INDX" para *index buffers*) que indetificam um bloco com um certo tipo de dados.

Detecção (Cont.)

- O type checking é muito útil para a detecção de apontadores corrompidos. Contudo, sistemas que utilizam este método não devem *sobrecarregar* os tipos de dados.
  - Não é detectado nenhum erro quando um ponteiro de index buffer(RootIndxBuf, SDH, SII, ou DirIndxBuf) aponta para o index errado. Neste caso, o tipo "INDX" é *sobrecarregado*; é usado para representar diferentes estruturas usados para diferentes finalidades. Portanto, a não detecção deste erro leva a futuros erros causados pelo NTFS.

Detecção (Cont.)

- O type checking não funciona para todos os apontadores. Portanto o sanity checking deve ser realizado.
  - O type check não é útil para apontadores como FileData, umavez que não é possível identificar o tipo de dados concreto. Nestes casos, o sanity checking assume grande importância.
- É de salientar que nem todo o comportamento do NTFS pode ser explicado através destes dois métodos de verificação.

Reações

- O NTFS reage de diferentes maneiras quando detecta um erro: pode recuperar da corrupção, informar o erro à aplicação, tenta novamente, ou tenta reparar os dados.

- **2ª Obsevação**: Tipicamente, é usada a replicação para a recuperação de apontadores corrompidos.

  - No entanto, o NTFS não reescreve o apontador corrompido com o valor correcto, logo, é uma recuperação temporária. É necessário a mesma recuperação para cada montagem do sistema. Apenas quando ocorre um valor "out-of-bounds" do apontador é feita a recuperação permanente do apontador.

Reações

- **3ª Observação:** Usa a informação de erro quando não é capaz de recuperar.
  - Para um subconjunto de casos, O NTFS tenta novamente a operação de montagem, talvez esperando que a corrupção seja temporária e que a montagem seja bem sucedida a segunda vez.

- **ARMADILHA 1:** Detecção que o alvo do apontador está corrompido em vez da detecção da corrupção do apontador.
  - O sistema NTFS confia que o apontador é o que está correcto, enquanto que não confia nos "dados alvo". Portanto, a tentativa de reparar um aparente alvo corrompido causa maior dano que a própria corrupção do apontador. Em geral, observou-se que isto acontece em muitas instancias onde o NTFS não detecta qualquer erro ou detecta mas não recupera dele.

- **ARMADILHA 2:** Gestão de réplicas ineficaz:
  - a) Não utilização das réplicas quando disponíveis;
  - b) Destruição das réplicas secundárias sem a verificação das principais;
  - c) Não utilização de caminhos de acesso independentes para as réplicas.

Resultados Visíveis aos Utilizadores

- O Sistema trabalha correctamente quando recupera duma corrupção
  - Isto em 61 cenários (17%). Noutros 10, apesar de não recuperar, não causou problemas.
- O resultado mais frequente é a impossibilidade da montagem do sistema de ficheiros
  - Acontece em 133 cenários (37%).
- Outros resultados incluem:
  - a) Perda de dados: 102 cenários(28%);
  - b) Falha da maior parte das operações de ficheiros: 127 cenários(35%);
  - c) Dados do utilizador corrompidos : 8 cenários (2%).
- Apontadores corrompidos e não detectados pode comprometer a segurança da informação.
  - Em 22 casos, estes apontadores causaram o crash de todo o sistema. Mas dependendo dos valores exactos presentes num bloco, pode provocar tanto o crash do sistema como a perda de dados.

Resultados – Ext3

- Contrariamente ao NTFS, este sistema baseia-se mais em sanity checks do que em type checks;
- Assume tipicamente que os dados alvos são aqueles que estão corrompidos, tal como no NTFS;
- Apesar do Ext3 criar réplicas, estas nunca são utilizadas mesmo quando acontece alguma detecção de erro;
- A reacção típica deste sistema é reportar o erro ao utilizador e remontar o sistema em modo de leitura. O Ext3 não recuperou de uma única situação de corrupção;

Conclusões

- Os sistemas de ficheiros têm como base apontadores para aceder aos dados do disco;
- À medida que vão sendo desenvolvidas novas técnicas para proteger contra a corrupção dos apontadores, será necessário perceber o comportamento e o desempenho destas na realidade;
- Num futuro próximo os sistemas de ficheiros serão mais cuidadosos na implementação de técnicas para evitar a corrupção dos apontadores;