Atividade S3

Rafael Sartori M. Santos, 186154

27 de março de 2020

Avaliação

O problema proposto sugere avaliar a importância da segurança na interação entre aplicativos em *userspace* e o *kernel* do sistema operacional.

Enunciado do problema

O que é necessário ser feito para proteger o *kernel* do sistema durante chamadas do sistema (*syscalls*)? Por que cada alteração foi necessária?

Resposta

Como nas trocas de modos de operações, vista com maior profundidade na arquitetura x86, *syscalls* também precisam de cuidado para não permitir ações maliciosas por parte de aplicativos, de rigor para negar acessos inválidos e de resiliência para que qualquer erro não cause a quebra do sistema.

Os principais cuidados são:

- **Tratamento dos argumentos:** como as *syscalls* são chamadas a partir do *userspace*, é necessário buscar na memória do aplicativo (em modo usuário) os argumentos das chamadas. Por isso, é necessário copiar para a memória do *kernel* para impedir que haja modificações em etapas posteriores à verificação dos argumentos, impedindo ataques ao *kernel* em formato de alterações de valores de argumento.
- **Verificação dos argumentos:** com os argumentos copiados para a memória protegida do *kernel*, verifica-se quanto a validade do pedido, por exemplo: o caminho do arquivo é uma *string* válida? o aplicativo está tentando abrir algum arquivo que exista? o aplicativo possui permissão para escrita no arquivo? o aplicativo pode executar mais um *fork*? o aplicativo pode requisitar mais memória? São questões importantes para impedir o abuso por parte de aplicações: mesmo se não parar o programa, o sistema permanecerá responsivo para que o usuário o faça? A verificação precisa assegurar isso.
- **Tratamento da resposta:** a resposta da *system call* não pode permanecer na memória protegida do *kernel*, é necessário copiá-la novamente à memória do aplicativo para torná-la acessível. É necessário verificar se as fronteiras para escrever a resposta são válidas, notando mudanças enquanto a *syscall* era resolvida, senão o *kernel* poderia quebrar escrevendo em área que não é mais do aplicativo devido ação maliciosa alterando posições de memória.