Vulnerabilidade crítica do kernel do Linux permite que invasores obtenhar

Data: 2025-08-09 11:38:53

Autor: Inteligência Against Invaders

*9 de agosto de 202*5: Uma grave vulnerabilidade de segurança no kernel do Linux, apelidada de CVE-2025-38236, foi descoberta pelo pesquisador do Google Project Zero, Jann Horn, expondo um caminho para invasores que vão desde a execução de código nativo na sandbox do renderizador do Chrome até o controle total no nível do kernel em sistemas Linux.

A falha, ligada ao recurso de MSG_OOB obscuro nos soquetes de domínio UNIX, afeta os kernels Linux versão 6.9 e superior, levantando alarmes sobre a segurança das sandboxes do navegador e os riscos das funcionalidades esotéricas do kernel.

MSG_OOB bug abre portas para a exploração do kernel

Descoberta no início de junho durante uma revisão de código de um novo recurso do kernel Linux, a vulnerabilidade decorre da funcionalidade MSG_OOB (fora de banda) introduzida no Linux 5.15 em 2021.

Usado principalmente em aplicativos Oracle de nicho, MSG_OOB foi ativado por padrão em kernels que suportam soquetes de domínio UNIX e foi inadvertidamente acessível na sandbox do renderizador Linux do Chrome devido a sinalizadores de syscall não filtrados.

O bug aciona uma condição de uso após liberação (UAF), que Horn demonstrou que pode ser explorada com uma sequência direta de operações de soquete, permitindo que os invasores manipulem a memória do kernel e potencialmente obtenham privilégios elevados.

A exploração, circunstanciado no rastreador de bugs do Google Project Zero, mostra um ataque sofisticado a um sistema Debian Trixie executando a arquitetura x86-64.

Ao aproveitar um primitivo de leitura induzido por UAF, os invasores podem copiar memória kernel arbitrária para o espaço do usuário, ignorando as restrições de proteção de cópia do usuário.

A abordagem de Horn envolve a realocação de memória liberada como páginas de pipe ou pilhas de kernel, usando técnicas como manipulação de tabela de páginas e ponteiro mprotect() toOCB), que permanece pendente, levando a um UAF quando uma chamada RECV(…, MSG_OOB) subsequente acessa ele.

O sucesso do exploit depende do recurso CONFIG_RANDOMIZE_KSTACK_OFFSET do Debian, que randomiza os deslocamentos de pilha por syscall.

Horn transformou essa mitigação em uma vantagem, usando o primitivo de leitura para detectar alinhamentos de pilha ideais, permitindo a corrupção precisa da memória.

Desde então, o kernel do Linux foi corrigido e o Chrome atualizou sua sandbox para bloquear mensagens MSG_OOB, fechando esse vetor de ataque específico.

Pontos fracos do sandbox e limitações do fuzzer expostos

O bug inicial foi detectado pela ferramenta de fuzzing syzkaller do Google em agosto de 2024, exigindo seis syscalls para ser acionado, enquanto um problema relacionado e mais complexo encontrado por Horn precisava de oito.

Isso destaca o desafio que os fuzzers enfrentam ao navegar em estruturas de dados complexas do kernel, como buffers de soquete (SKBs).

Horn sugere aprimorar os fuzzers para se concentrar em subsistemas específicos do kernel para descobrir melhor essas vulnerabilidades, pois a probabilidade de atingir aleatoriamente a sequência de syscall adequada cai exponencialmente a cada chamada adicional.

A exploração também revela a extensa superfície de ataque na sandbox do renderizador Linux do Chrome, que expõe interfaces como VMAs anônimos, soquetes UNIX, pipes e syscalls como sendmsg() e mprotect(). Muitos deles são desnecessários para a funcionalidade do renderizador, aumentando o risco de exploração.

Vulnerabilidades anteriores do Chrome, incluindo aquelas envolvendo futex(), memfd_create() e pipe2(), ressaltam como recursos obscuros do kernel podem introduzir vulnerabilidades não intencionais quando expostos em sandboxes.

As descobertas de Horn também desafiam a eficácia de mitigações probabilísticas, como randomização de pilha por syscall contra invasores com recursos de leitura arbitrária, pois eles podem ser contornados verificando repetidamente os resultados da randomização.

O uso de mprotect() para atrasar operações copy_from_user() sugere ainda que restringir recursos como userfaultfd pode não mitigar totalmente esses riscos, pois métodos alternativos podem alcançar atrasos semelhantes.

Essa vulnerabilidade ressalta a necessidade de restrições de sandbox mais rígidas e uma reavaliação dos recursos do kernel expostos a processos sem privilégios.

Horn planeja realizar uma análise mais profunda da sandbox do renderizador Linux do Chrome em um relatório futuro.

Por enquanto, os usuários do Linux são incentivados a aplicar os patches mais recentes do kernel, e os desenvolvedores são encorajados a examinar os recursos esotéricos do kernel incorporados nas interfaces do sistema central para evitar explorações semelhantes.