Novo método de utensílios de flipswitch supera as defesas do kernel Linux

Data: 2025-10-01 11:34:31

Autor: Inteligência Against Invaders

Surgiu um novo método de enigma rootkit dublado Flipswitch, contornando as mais recentes

salvaguardas do Linux 6.9 Kernel e reacendendo preocupações sobre o compromisso no nível do

kernel.

Manipulando o código da máquina do novo despachante syscall, em vez do depreciado sys_call_table. kill e getdents64.

Durante anos, raízes linux como diamorfina e pumakit explorou o sys_call_table-Uma variedade simples de ponteiros de função-para redirecionar syscalls através de funções controladas por atacantes.

Ao desativar a proteção de gravação e substituir entradas específicas, um adversário pode ocultar arquivos maliciosos de ls Saídas ou tentativas de terminação do processo de frustração.

O liberar do Linux Kernel 6.9, no entanto, deu um golpe fatal a essa abordagem: o kernel substituiu a pesquisa de matriz direta

```c
// Pre-6.9: Direct array lookup sys_call_table[__NR_kill](regs);
```

com um switch-Despacho de estatura dentro x64_sys_callignorando efetivamente quaisquer modificações para sys_call_table Para o manuseio real do syscall.

Flipswitch: redescobrindo o gancho

O principal insight de Flipswitch é que a lógica original do syscall ainda existe em forma compilada por trás do switch declaração.

Em vez de adulterar com sys_call_tableFlipswitch localiza e remende o nível da máquina call instrução interior x64_sys_call Isso chama a função Syscall alvo. Este processo se desenrola em quatro etapas:

1. **Descubra o endereço de função original**
   Embora o sys_call_table Não governa mais o despacho, ele ainda mantém indicações válidas para compatibilidade. Lendo uma entrada como sys_call_table[__NR_kill]Flipswitch obtém o endereço do original sys_kill rotina.
2. **Localizar kallsyms_lookup_name**

Para encontrar símbolos do kernel programaticamente, o flipswitch usa um kprobe em um símbolo conhecido, extraindo o endereço de kallsyms_lookup_name. Isso permite a pesquisa de qualquer ponteiro de função Syscall exportado, ignorando as restrições de exportações diretas.

3. **Digitalize a instrução de chamada exclusiva**
   O x64_sys_call O código da máquina da função é pesquisado byte por byte para o código de opção de um bytes 0xe8 seguido pelo deslocamento relativo de 4 bytes precisos que se destina sys_kill. Esta assinatura singular identifica as instruções exatas para sequestrar.
4. **Patch o despachante**
   Operando no anel 0, o flipswitch desativa a proteção de gravação da CPU limpando o bit WP no CR0. Em seguida, substitui o deslocamento de 4 bytes do identificado call instrução para apontar para um malicioso fake_kill manipulador. Após a descarga do módulo, as proteções são reativadas e o deslocamento original restaurado, deixando evidências forenses mínimas.

Implicações e estratégias defensivas

O flipswitch ressalta a persistente dinâmica de gato e rato entre o endurecimento do kernel e a inovação adversária. Enquanto os desenvolvedores do Linux continuam fortalecendo a despacho do Syscall, os atacantes se adaptam ao direcionar a própria lógica de despacho compilado. Mitigações podem incluir:

- **Verificação de integridade de tempo de execução**: Hash periodicamente e validando o código da máquina de x64_sys_call Para detectar modificações não autorizadas.
- **Restrições aprimoradas de KProbe**: Limitar ou auditar ainda mais o uso de KProbes para localizar endereços de símbolos críticos.
- **Integridade do fluxo de controle (CFI)**: Empregando técnicas de CFI dentro do kernel para aplicar que todas as chamadas indiretas correspondem a metas legítimas.

Detectando flipswitch com yara

Manter a visibilidade no fluxo de controle do kernel e alavancar a detecção baseada em assinatura será essencial para ficar à frente na batalha em andamento por Linux Kernel integridade.

A detecção de rootkits no nível do kernel permanece desafiadora devido à sua operação furtiva e na memória. Para ajudar os defensores, a Elastic Security publicou uma regra Yara visando o flipswitch prova de conceito. A regra digitaliza os padrões exclusivos de código de máquina introduzidos durante o processo de patch:

```
textrule Linux_Rootkit_Flipswitch_821f3c9e { meta: author = "Elastic Secur
ity" description = "Detect FlipSwitch rootkit PoC" os = "Linux" arch = "x8
6" strings: $all_a = { FF FF 48 89 45 E8 F0 80 ?? ?? ?? } $main_b = { 41 5
4 53 E8 ?? ?? ?? ?? 48 C7 C7 ?? ?? ?? ?? } condition: #all_a >= 2 and 1 of
 ($main_b) }
```

Ao implantar essa regra em ferramentas de varredura de memória ou plataformas de proteção de terminais, as equipes de segurança podem sinalizar a presença do despachante corrigido da Flipswitch e responder antes que ocorram danos significativos.

À medida que as defesas do kernel evoluem, pesquisas como o FlipSwitch destacam a necessidade crítica de proteções em camadas e monitoramento proativo.

Siga -nos Google News**Assim,**LinkedIn**e**X**Para obter atualizações instantâneas e definir GBH como uma fonte preferida em**Google**.**