PIkit : A New Kernel-Independent Processor-Interconnect Rootkit

Autores do artigo:
Wonjun Song, Hyunwoo Choi, Junhong Kim, Eunsoo Kim, Yongdae Kim, John Kim
Apresentação: Vagner Kaefer Dos Santos

Introdução

O principal objetivo de um rootkit é, muitas vezes, rodar um software malicioso em uma máquina comprometida.

Como ele funciona?

- PIkit explora uma vulnerabilidade de hardware muito utilizados em datacenters e sistemas de alta performance.
- PIkit explora a tabela de endereçamento da DRAM.

Como um rootkit pode ser instalado

- Diferentes tipos de rootkits podem ser instalados em nível de aplicação, nível do kernel, boot, loader ou hypervisor.
- Também existem várias formas de detectar os rootkits.

		Malicious Payload	
		User-Level	Kernel-Level
Vulnerabilities	Software	t0rn [35], lrk5 [48], dica [23], etc.	ROR [25], DKOM [15], knark [17], etc.
	Hardware	This work (PIkit)	Cloaker [20] Shadow Walker [50]

Table 1: Classification of different rootkit attacks.

User Level

- Modificam utilitários existentes no sistema e/ou binários do sistema para habilitar códigos maliciosos.
- Ex: ls, ps, netstat...

Kernel Level

- Explora o controlo e intercepção, modificando as estruturas de dados estáticos do kernel para executar códigos maliciosos de forma indireta.

PIkit

- Uma vez instalado, permite utilizar bash shell credenciada, compartilhar bibliotecas, buffer do teclado, entre outros, utilizando somente acesso à memória em nível de usuário.

Modelo do alvo

- Atacante e vítima compartilham o mesmo servidor com multi-socket.
- Atacante não possui acesso físico.
- Necessita de vulnerabilidades ou engenharia social para ter acesso com nível de **root pelo menos uma vez.**

Depois de instalado, torna-se muito difícil de detectar ou determinar a fonte do ataque! Pois não há alterações no kernel.

Visão do processador interconectado

- Existe um algoritmo de roteamento que determina o caminho percorrido e o destino de um “pacote”.
- Normalmente isso é feito através de uma tabela, por ser flexível.
- O destino de um packet é determinado pelo seu cabeçalho.

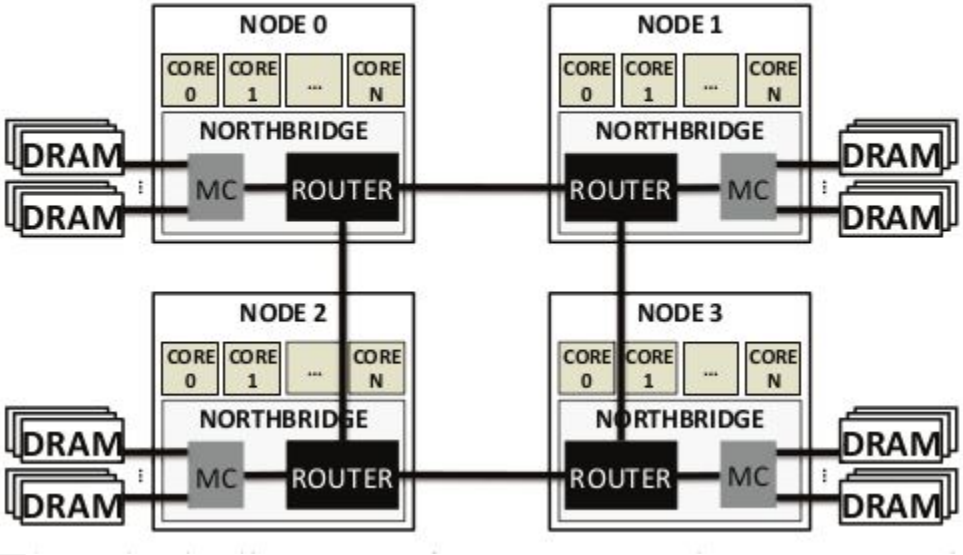
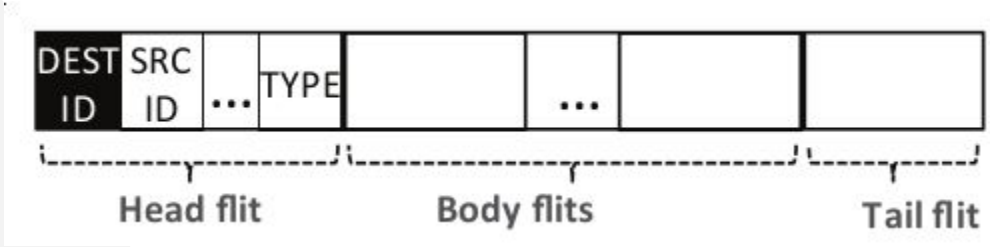


Diagrama de blocos de um processador interconectado entre 4 sockets.

Um pacote consiste em uma mensagem de alto nível enviada entre os nós.

Pode incluir requisições de memória, ler caches, mensagem de coerência, etc.



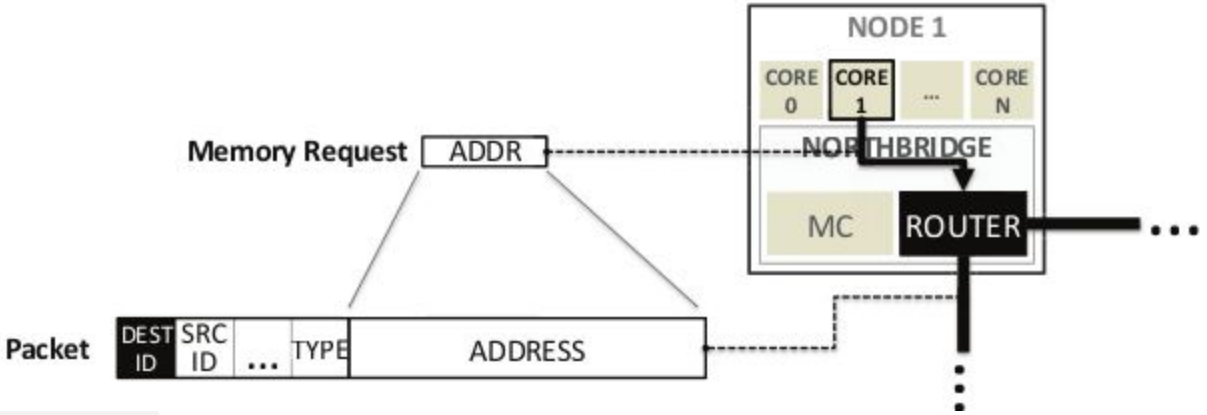
Formato do pacote em redes de interconexões.

	Base Address	Limit Address	Destination ID
0	0x0000000000	0x041F000000	0
1	0x0420000000	0x081F000000	1
2	0x0820000000	0x0C1F000000	2
3	0x0C20000000	0x101F000000	3
4	RESERVED	RESERVED	RESERVED
5	RESERVED	RESERVED	RESERVED
6	RESERVED	RESERVED	RESERVED
7	RESERVED	RESERVED	RESERVED

O mapa da DRAM é inicializado pela BIOS durante o boot.

Características de hardware vulnerável

- **Configuração**: O mapeamento da memória precisa ser configurado conforme a capacidade de memória.
- **Entradas Extras**: Devem existir entradas livres no mapeamento de memória.
- **Discrepância**: Valores da tabela podem ser diferentes dos definidos durante a inicialização.


NODE 1
CORE 0
CORE 1
...
CORE N
NORTHBRIDGE
MC
ROUTER
Memory Request
ADDR
Packet
DEST ID
SRC ID
...
TYPE
ADDRESS

Definindo o endereço da região de ataque

- O atacante precisa preparar o endereço de ataque para que somente ele tenha acesso de determinada região.
- Outras aplicações não podem tentar acessar esses endereços.

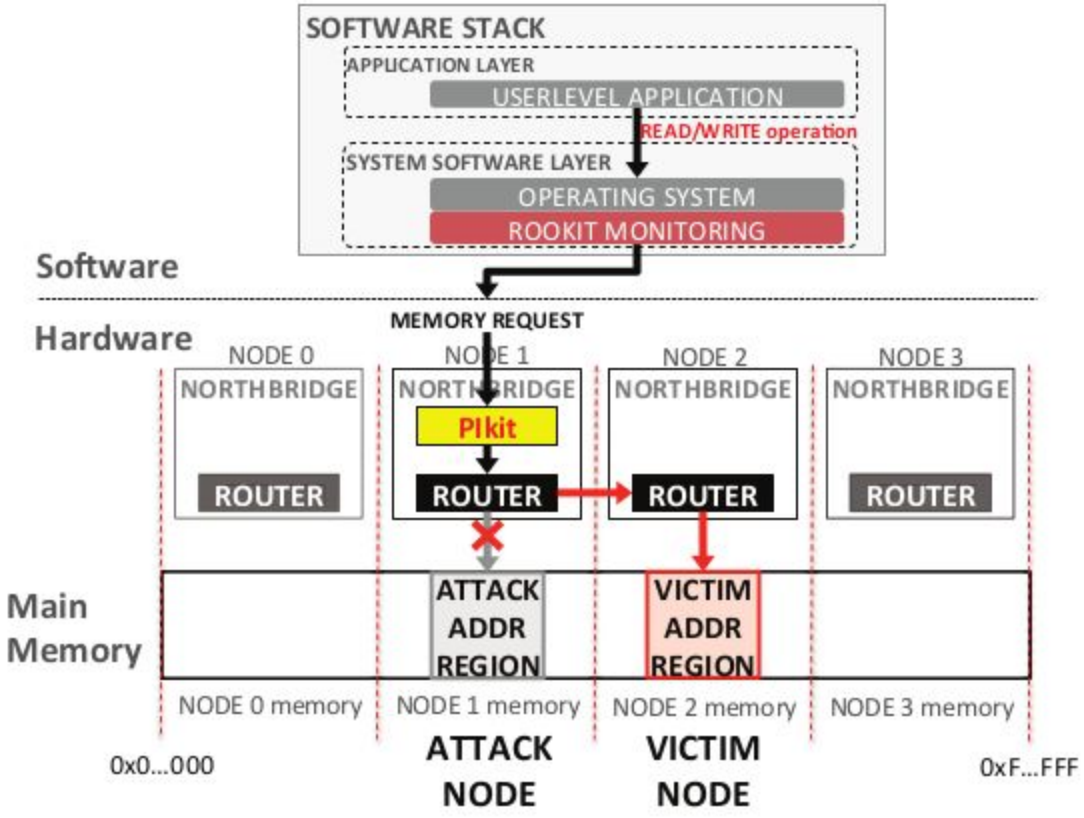
Modificando a tabela de mapeamento da DRAM

	Base Address	Limit Address	Destination ID
0	0x0000000000	0x041F000000	0
1	RESERVED	RESERVED	RESERVED
2	0x0820000000	0x0C1F000000	2
3	0x0C20000000	0x101F000000	3
4	0x0420000000	0x07BF000000	1
5	**0x07C0000000**	**0x07C1000000**	**2**
6	0x07C2000000	0x081F000000	1
7	RESERVED	RESERVED	RESERVED

Nó 1: Atacante
Nó 2: Vítima

Detalhes do novo roteamento

- As novas entradas devem ser escritas antes das velhas serem removidas.
- O endereço base deve ser escrito antes do endereço de limite.
- Para remover uma entrada, primeiro deve ser removido o endereço limite e depois o endereço base.


SOFTWARE STACK
APPLICATION LAYER
USERLEVEL APPLICATION
READ/WRITE operation
SYSTEM SOFTWARE LAYER
OPERATING SYSTEM
ROOKIT MONITORING
Software
Hardware
MEMORY REQUEST
NODE 0
NODE 1
NODE 2
NODE 3
NORTHBRIDGE
NORTHBRIDGE
NORTHBRIDGE
NORTHBRIDGE
PIkit
ROUTER
ROUTER
ROUTER
ROUTER
Main Memory
ATTACK ADDR REGION
VICTIM ADDR REGION
NODE 0 memory
NODE 1 memory
NODE 2 memory
NODE 3 memory
0x0...000
ATTACK NODE
VICTIM NODE
0xF...FFF

Exemplo

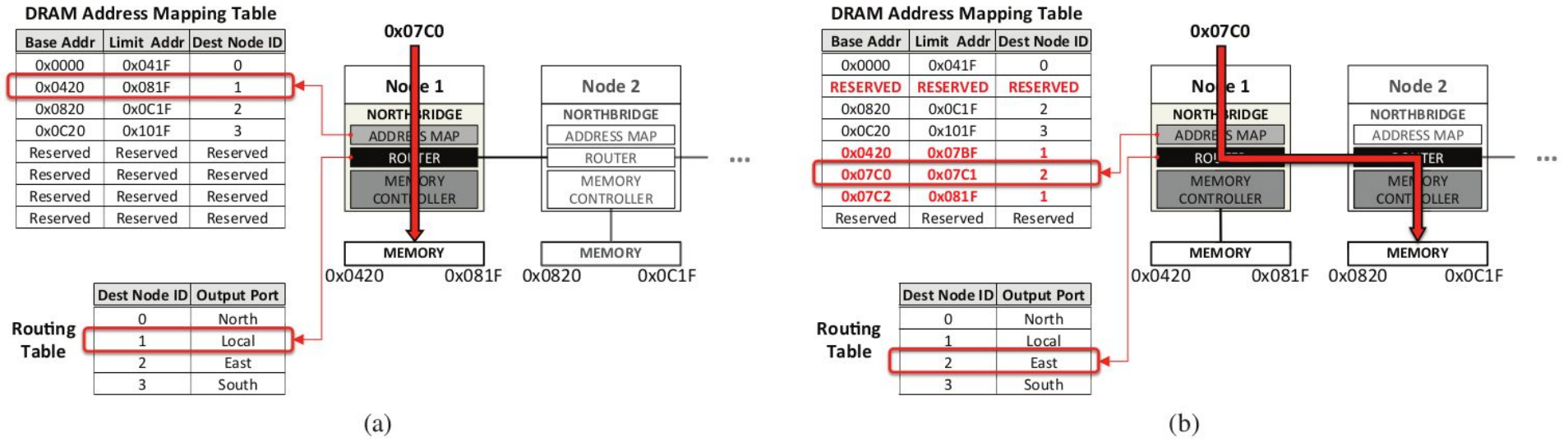
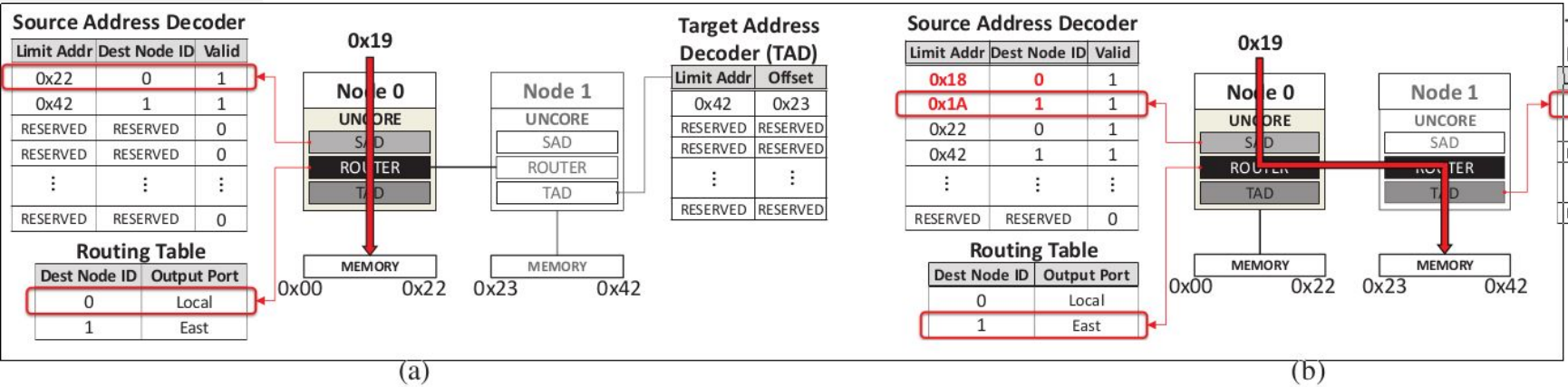


Figure 8: Example of PIkit (a) before and (b) after PIkit is installed on an AMD Opteron 6128 server.

O artigo é totalmente baseado em AMD servers, mas pode ser feito em processadores Intel de maneira semelhante!
Uma das diferenças principais é a tabela de roteamento, que possui somente o endereço limite.


Source Address Decoder
Limit Addr | Dest Node ID | Valid
0x22 | 0 | 1
0x42 | 1 | 1
RESERVED | RESERVED | 0
RESERVED | RESERVED | 0
RESERVED | RESERVED | 0
Routing Table
Dest Node ID | Output Port
0 | Local
1 | East
0x19
Node 0
UNCORE
SAD
ROUTER
TAD
MEMORY
0x00
0x22
Node 1
UNCORE
SAD
ROUTER
TAD
MEMORY
0x23
0x42
Target Address Decoder (TAD)
Limit Addr | Offset
0x42 | 0x23
RESERVED | RESERVED
RESERVED | RESERVED
RESERVED | RESERVED
(a)
Source Address Decoder
Limit Addr | Dest Node ID | Valid
0x18 | 0 | 1
0x1A | 1 | 1
0x22 | 0 | 1
0x42 | 1 | 1
RESERVED | RESERVED | 0
Routing Table
Dest Node ID | Output Port
0 | Local
1 | East
0x19
Node 0
UNCORE
SAD
TAD
MEMORY
0x00
0x22
Node 1
UNCORE
SAD
MEMORY
0x23
0x42
Target Address Decoder (TAD)
Limit Addr | Offset
0x22 | 0x0
0x42 | 0x23
RESERVED | RESERVED
RESERVED | RESERVED
(b)

Atividades maliciosas

Depois de instalado, o PIKit ativa o acesso à memória da vítima.

Vários payloads podem ser utilizados.

Payload: Bash Shell

Atividades maliciosas – Bash Shell

- Um processo é representado por um Bloco de Controle de Processos (PCB), que é uma estrutura de dados em local privilegiado da memória.
- O PCB possui informações críticas como memória, arquivos abertos, contextos de processos, prioridades, etc...

Atividades maliciosas – Bash Shell

- Se o atacante localizar o PCB na região da vítima, ele pode modificar seus valores com o PIKit.
- Nesse exemplo, o PIKit será utilizado para modificar o UID ou o EUID de um processo Shell Bash, resultando em acesso com nível de root.

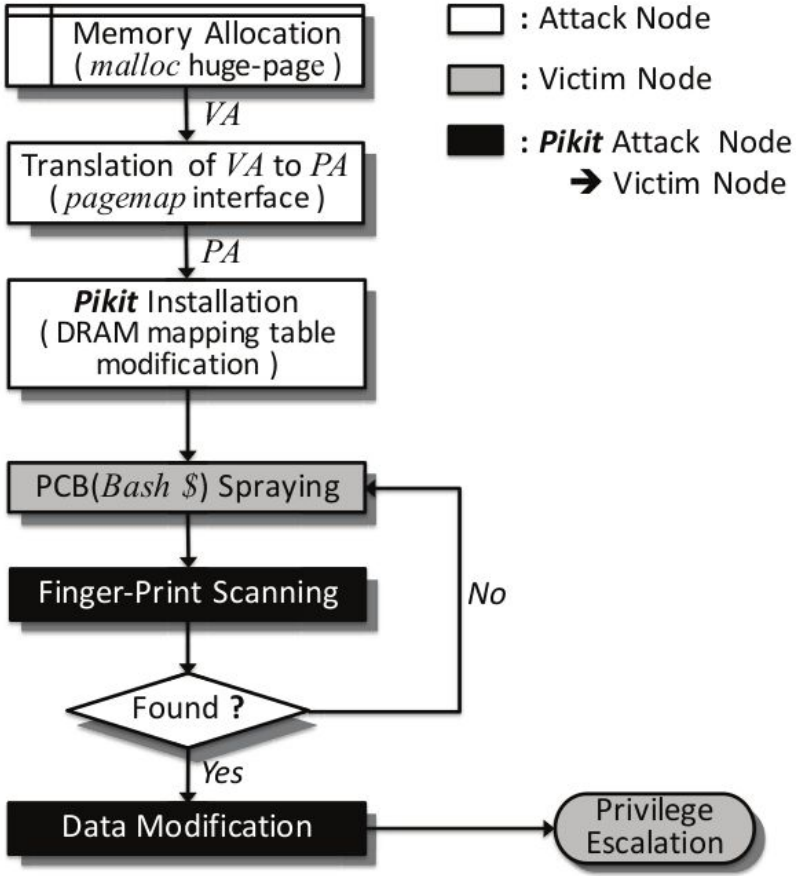


Figure 12: High-level overview of the attack with PIkit for privilege escalation.

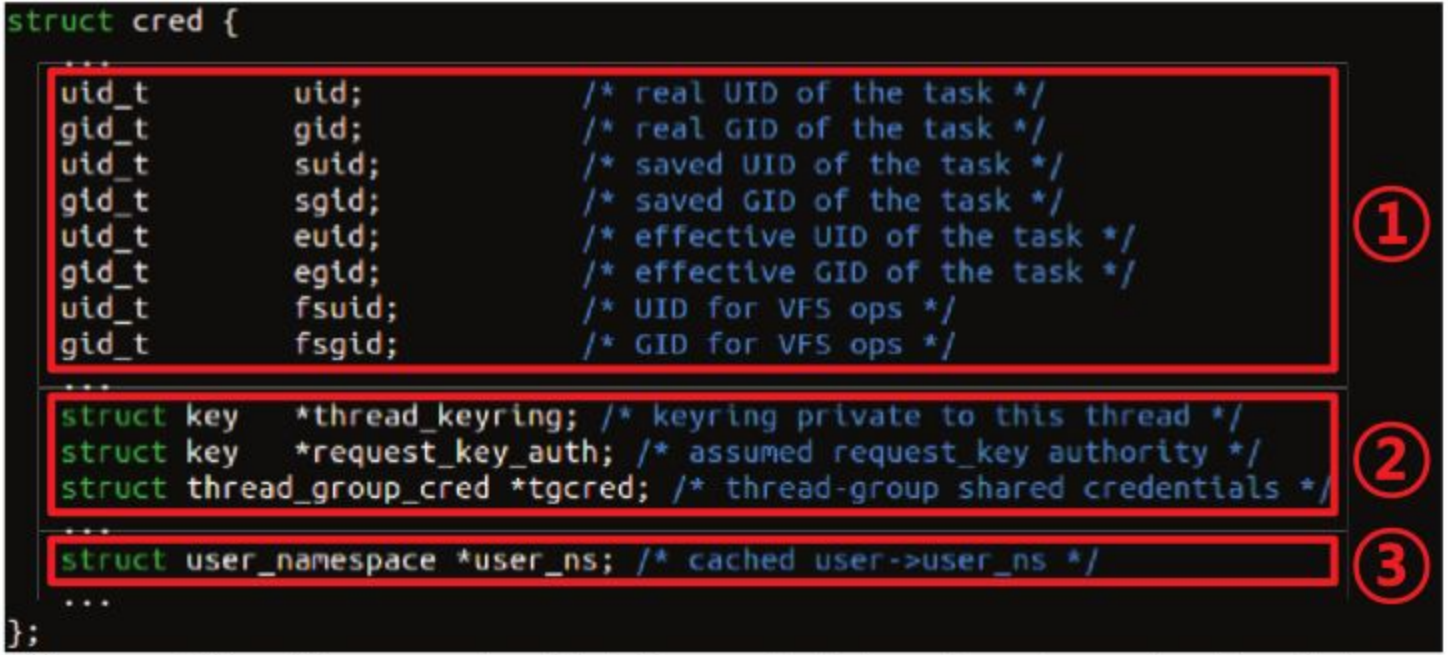
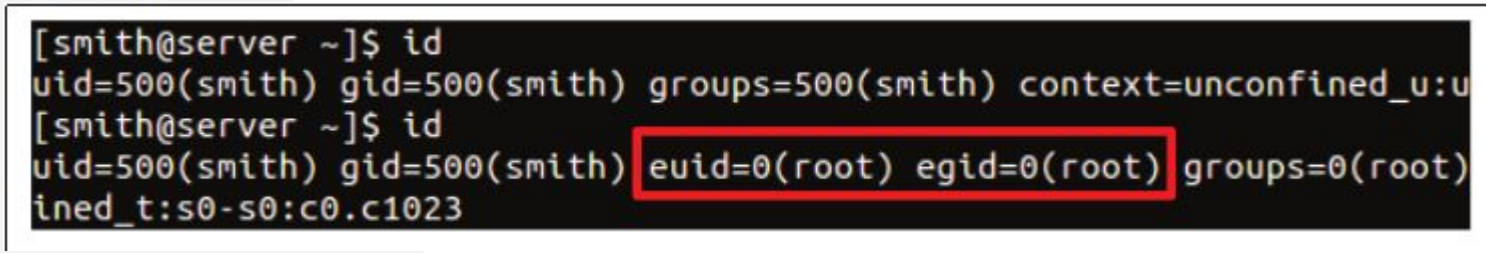


Figure 13: The credential kernel data structure in the Linux kernel 3.6.0 and the fingerprint that we exploit in this work, with the fingerprint highlighted with a rectangle.

Atividades maliciosas – Bash Shell

- Depois que o scan encontrar os dados, o trabalho se resume em escrever na memória:

  movnti $0, (Virtual Address)



Atividades maliciosas – Bash Shell

- O PCB pode estar fora da região escolhida no início do processo.
- Para aumentar a probabilidade do PCB estar na região de ataque, são executados múltiplos processos no nó da vítima.

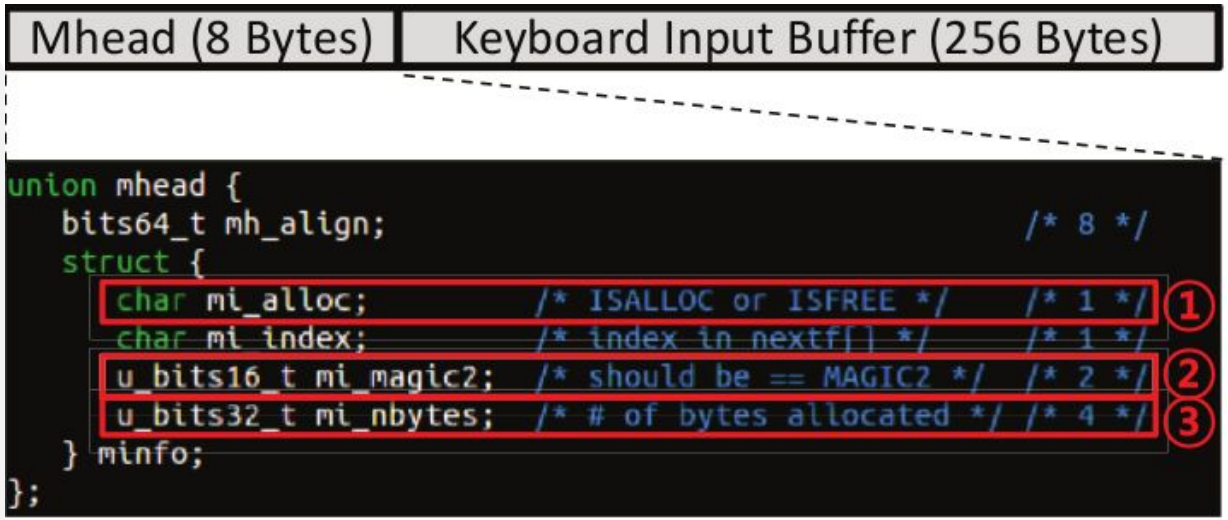
Payload: Bash Keyboard Buffer

Atividades maliciosas – Bash Keyboard Buffer Attack

- Esse ataque consiste em monitorar a memória da vítima. Nenhuma alteração de memória é necessária, ou seja, o ataque consiste somente em leitura de memória.
- Quando um usuário digitar qualquer palavra em seu prompt, todos os caracteres são armazenados em um buffer de teclado na memória, o dado não é criptografado.

Atividades maliciosas – Bash Keyboard Buffer Attack

- No Bash Shell (V4.3) o buffer é representado por uma estrutura chamada mhead:



Atividades maliciosas – Bash Keyboard Buffer Attack

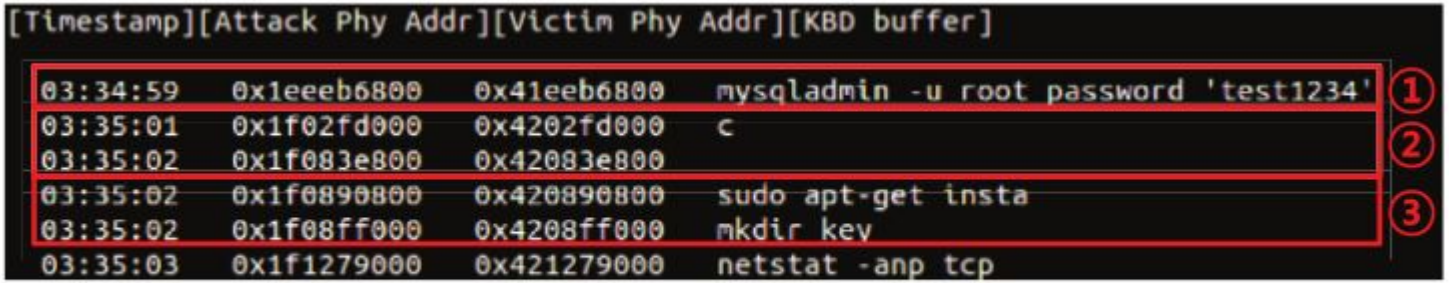


Figure 16: Snapshot of bash keyboard buffer monitoring.

Payload: Shared Library

Atividades maliciosas - Shared Library Attack

- Bibliotecas compartilhadas são alocadas na memória uma única vez e compartilhada por múltiplos usuários.
- Esse ataque irá modificar estas bibliotecas para executar códigos maliciosos.
- A biblioteca utilizada foi a libc

Atividades maliciosas – Shared Library Attack

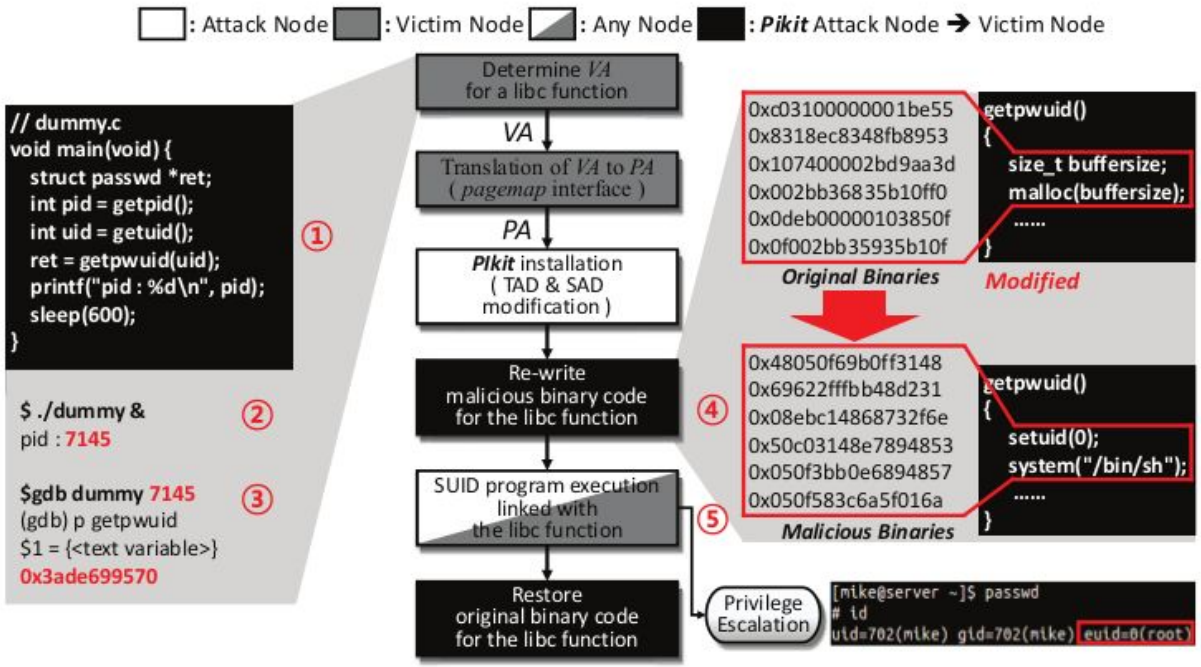


Figure 17: Shared library (`libc`) Attack with PIkit.

Discussão

Como se proteger do PIKit?

- O mais simples: Não deixar ele ser instalado! Mas como podem surgir falhas que permitem acesso privilegiado para a instalação, podemos acompanhar a tabela de roteamento e detectar modificações.

Como se proteger do PIKit? – Soluções existentes

- Os AMDs recentes não são vulneráveis, não por implementações de segurança, mas por causa da economia de energia.
- Sempre habilitar o LockDramCfg.
- Intel também suporta o LockDramCfg.

Como se proteger do PIKit? – Soluções baseadas em Software

- Um programa pode ficar rodando para detectar alterações na tabela de roteamento da DRAM.
- Esse programa pode ser protegido para que o PIKit não interfira em seu funcionamento.
- Ele teve um impacto de 2% nas máquinas.

Como se proteger do PIKit?

Algorithm 1: PIkit monitor to detect modification to the DRAM address mapping table.

```
Input : monitoring
begin
    while monitoring do
        – Get DRAM information from SPD
        – Calculate Valid_address_ranges of installed
          DRAM from information
        A ← Valid_address_ranges
        – Get Current_address_ranges from DRAM
          address mapping table
        B ← Current_address_ranges
        if A != B then
            PIkit detected
return
```

Como se proteger do PIKit? – Soluções baseadas em Hardware

- Uma solução de hardware seria restringir o número de entradas da tabela de roteamento para o número de nós do sistema. Isso iria minimizar o ataque.
- Outra solução seria não permitir a alteração da tabela de roteamento.

Como se proteger do PIKit? – Soluções baseadas em Hardware

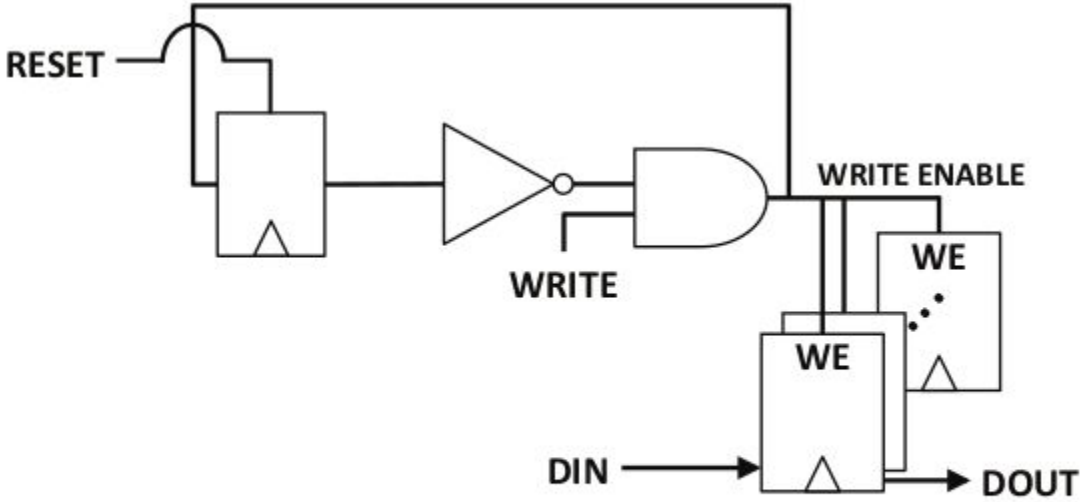


Figure 19: A PIkit hardware solution by creating a write-once register for the DRAM address mapping table.

Como se proteger do PIKit? – Soluções baseadas em Hardware

- A segunda solução bloqueia a utilização de CPU hotplug e Memory hotplug.
- Mais uma entrada para detectar o evento do plug. O que também poderia habilitar o ataque.

Limitações

Limitações

- **Granularidade (condição de ser regular)**
  O atacante precisa conseguir um tamanho específico de páginas de memória para não ter problemas com outros usuários tentando acessar aquela faixa de endereço, o que iria gerar problemas e erros.

Limitações

- **Cobertura da DRAM:**
  Um dos desafios do PIKit é conhecer o mapeamento físico da DRAM. Essas configurações são alteradas conforme o fabricante.

Conclusão

Conclusão

- Foi descrito um novo tipo de rootkit, que utiliza falhas de hardware para realizar atividades maliciosas com níveis de acesso de usuários comuns.
- A dificuldade de detectar esse tipo de ataque.

Referências

SONG et al; **PIkit : A New Kernel-Independent Processor-Interconnect Rootkit.** USENIX Security Symposium. 2016. Página 37-51.

Obrigado!

Alguma dúvida/comentário?