O slackware é a distribuição linux mais antiga ainda em atividade. Tendo sido criada por Patrick Volkerding em 1993, a partir da SLS.

Em todos esses anos, a distro conquistou ardorosos utilizadores, principalmente graças à sua filosofia de simplicidade e estabilidade.

Um produto de extrema qualidade para usuários com esta mesma característica. E este zine é de slacker para slacker.

slackware zine

3 a 5 de Novembro de 2006 – Edição #14.5

Editorial

Mais um CONISLI e mais uma edição especial do SlackwareZine. Que agora parece estar voltando para os eixos em termos de periodicidade -:) E, a boa notícia do slackware 11.0 sempre deixa a comunidade slacker mais animada -:)

Essa edição 14.5 traz dois artigos. Um bem simples sobre como mapear teclas “invisíveis”. E outro sobre a técnica do “Port Knocking”, um artigo bem completo e com bastante conteúdo (apesar de só ter uma página).

Ou seja, dois artigos técnicos, cumprindo a proposta zine: artigos técnicos, de técnicos e para técnicos.

Boa Leitura!

Piter PUNK

PS> Espero encontrar todos vocês no II SlackShow!!

II SlackShow

18 de Novembro na FIAP

Palestras técnicas, de técnicos e para técnicos

III Encontro Nacional de Usuários Slackware

Informações: http://piterpunk.info02.com.br/evento

Mapeando Teclas “inexistentes”

Teclado de notebook é, no mínimo, algo bem original. Encaixam as teclas de qualquer jeito e em qualquer lugar. O meu tem um € e um $ em cima das setas cursoras. Mas nenhum dos dois funciona. O $ por exemplo mostra essa mensagem no dmesg:

```
atkbd.c: Unknown key released \
 (translated set 2, code 0xb4 on\
 isa0060/serio0).
atkbd.c: Use 'setkeycodes e034 \
 <keycode>' to make it known.
```

E no “xev”, não mostra absolutamente nada. Para resolver isso, vamos primeiro obedecer a mensagem no dmesg e usar o comando:

```
setkeycodes e034 201
```

Usei o 201 porque normalmente os números altos estão todos vagos. Com isso a tecla com o $ vai passar a ser reconhecida. Agora uma segunda fase:

```
echo “keycode 201 = dollar” | loadkeys
```

Isso mapeia o nosso $ no console. Para fazer o mesmo no X faça:

```
echo "keycode 169 = dollar" | xmodmap -
```

É, o número dos keycodes é diferente no X. Como agora a tecla aparece no xev, é fácil pegar o keycode dentro do X. Se quiser mapear sempre o $ de maneira automática, é só colocar uma linha com "keycode 169 = dolar" dentro do seu .Xmodmap.

Corrigir o problema do € segue o mesmo procedimento, só mudam os números. O scancode dele é e033, mapeei para 200. O nome do € é “euro” no console e “EuroSign” no X. Difícil é achar uma fonte de console que tenha o €. O princípio é o mesmo para qualquer tecla, em um teclado USB (abnt2) tive que usar essa técnica para mapear o “.” do teclado numérico.

Piter PUNK <piterpk@terra.com.br>

slack users

Knocking Ports com Iptables

Port Knocking é uma técnica de segurança baseada em obscuridade que, na prática, significa ocultar um serviço até que ele seja realmente necessário. Enquanto um super-daemon (inetd/xinetd) roda um daemon inteligente que aguarda conexões na porta de um serviço pré-configurado e só executa o serviço quando houver essa requisição, o *Port Knocking* mantém o serviço rodando porém bloqueado por uma regra de firewall que só libera uma conexão após um "*toc-toc*" em uma determinada porta, que não precisa estar aberta ou ter nada rodando nela.

A situação

Vamos usar o *Port Knocking* em um servidor que roda o SSHD na porta 2200. Essa ficará sempre bloqueada, até que o firewall capture um pacote na porta 65065 e então permita que o IP que ENVIOU o referido pacote possa conectar-se ao serviço disponibilizado na porta 2200(`SSHD`), até que a conexão termine e esse IP de origem seja novamente bloqueado, assim que o pacote de encerramento de conexão for enviado à porta 1500.

Existe um daemon chamado `knockd` que faz esse trabalho, porém a abordagem do artigo é a implementação dessa técnica usando somente o Iptables e seus módulos. No caso, o módulo que utilizaremos agora é o "`ipt_recent`", cujo suporte deve estar compilado no kernel e devemos ativá-lo com o comando:

```
# modprobe ipt_recent
```

Conhecendo e operando o módulo recent

Com o comando "`iptables -m recent -h`" iremos colher algumas informações utéis, leia com atenção a função dos parâmetros `-set`, `--rcheck`, `--remove`, `--seconds`, `--name` e `–rsource`. Iremos trabalhar com as seguintes regras de negócio:

- Nosso firewall irá bloquear conexões externas com destino ao `SSHD` (porta 2200);

Pacotes com o bit de início de conexão ativados com destino à porta 65065, terão seu IP de origem armazenados em uma lista e em seguida, serão descartados;

- Quando um pedido de conexão com destino à porta 2200 for realizado, a lista será consultada. Se o IP que solicitou a conexão estiver na lista, o processo de conexão será realizado. Senão, o pacote será descartado;

- De mesmo modo, ao fim de uma conexão, um pacote deverá ser enviado à porta 1500, a lista será novamente consultada e se o IP de origem estiver nela será removido e o pacote descartado em seguida.

Implementação

```
# iptables -A INPUT -p tcp -s 0/0 \
       --sport 1024:65535 --dport \
       65065 -m recent --rsource \
       --set --name SSHKNOCK -j DROP

# iptables -A INPUT -p tcp -s 0/0 \
       --sport 1024:65535 --dport \
       2200 -m recent --rcheck \
       --rsource --name SSHKNOCK -j \
       ACCEPT

# iptables -A INPUT -p tcp -s 0/0 \
       --sport 1024:65535 --dport \
       1500 -m recent --remove \
       --rsource --name SSHKNOCK -j \
       DROP
```

Feito! Agora para conseguirmos nos conectar ao `SSHD`, que nesse caso roda na porta 2200, precisaremos antes fazer um *Knock* na porta 65065. O ideal é que quando terminarmos a sessão, façamos um outro *Knock* na porta 1500 bloqueando desse modo novamente o `SSHD`. Lembre-se que a liberação pelo *Port Knocking* é concedida somente a um IP específico por vez e não para toda a internet.

Ao invéz de fazer um **Knock** para fechar a porta novamente, podemos fazer isso automático. Por exemplo: só liberadar uma conexão se entre o *Knock* e a solicitação de conexão o tempo não exceder 60 segundos. A regra ficaria assim:

```
# iptables -A INPUT -p tcp -s 0/0  \
       –sport 1024:65535 --dport 2200 \
       -m recent --rcheck –seconds  \
       60 --rsource --name SSHKNOCK \
       -j ACCEPT
```

Consultando o help do `recent` você será capaz de fazer várias outras combinações, como requerer 2 *Knock's* antes de liberar a conexão, ou utilizar configurações de bloqueio do tipo: 3 tentativas em 1 minuto bloqueiam o serviço por 10 minutos, muito util para servicos expostos. Tudo que eu fiz aqui na chain `INPUT` aplica-se identicamente a chain `FORWARD`.

Conclusão

A implementação dessa técnica, que não pode ser considerada uma “novidade”, é de grande valia na administração remota de servidores e dificulta a ação de ataques baseados em bruteforce. Se desenvolverem coisas legais com o `recent` ou se tiverem dúvidas, entrem em contato.

Flávio do Carmo Júnior aka blk1d
<billpp@gmail.com>