

---

# Mecanismos para Contenção de Tráfego Malicioso de Saída em Honeynets

Klaus Steding-Jessen  
jessen@nic.br

Cristine Hoepers  
cristine@nic.br

Antonio Montes  
montes@lac.inpe.br

NIC BR Security Office – NBSO  
Comitê Gestor da Internet no Brasil

Lab. Associado de Computação e Matemática Aplicada  
Instituto Nacional de Pesquisas Espaciais – INPE

# Roteiro

---

## Contenção de Tráfego Malicioso de Saída em Honeynets:

- Motivação
- Técnicas de Contenção
- Construção de um Protótipo
- Conclusões

# Motivações

---

- *Honeynets*: utilizadas para acompanhar atividades dos invasores
  - todo tráfego de entrada é permitido
- Um invasor provavelmente usará a honeynet para atacar outros sistemas
  - a preocupação é o tráfego de saída
- Mecanismos de contenção: técnicas para diminuir a ameaça que uma honeynet pode representar para outras redes

# Contenção de ARP Spoofing

---

- Um invasor, forjando consultas ou respostas ARP, pode realizar:
  - ataques de *Man-in-the-Middle*
  - ataques de *Denial of Service*
  - encher o *cache* de *bridges* e *switches* (DoS? *fingerprinting*?)
- No caso de *bridges* atuando como *firewalls*
  - entradas ARP estáticas
  - barrar tráfego de endereços *ethernet* desconhecidos

# Filtragem de Pacotes

---

- Regras estáticas. Simples, mas efetivas em vários casos:
  - *Egress Filtering*
  - serviços com histórico de vulnerabilidades (111/TCP, 515/TCP, etc)
  - tráfego não usual (src port = dst port), *source routing*, etc
  - protocolos não usuais (0, 53, 77, 255, etc)
- Problema: falta de seletividade

# Limitação de Banda

---

- Em ataques de DoS, um invasor tentará usar toda a banda disponível
- Algum tipo de controle de consumo de banda é desejável, como ALTQ
- Alguns filtros de pacotes permitem associar regras de filtragem com limitação de banda
- Deve ser usado como um mecanismo adicional

# Normalização de Tráfego

---

- Um invasor pode lançar ataques e escapar da detecção de um IDS com:
  - fragmentos de pacotes IP, fragmentos duplicados, fora de ordem ou com sobreposição
  - tamanhos não usuais de pacotes, combinação inválida de *flags* TCP, etc.
- Normalizador de tráfego: elemento de rede para remover essas ambigüidades do tráfego de saída

# Limitação do Número de Sessões e Volume de Dados por Sessão

---

- Limitar o número de sessões pode conter algumas atividades, mas ainda assim permitir um certo nível de atividade
- Nem todo ataque necessita da criação de muitas sessões
  - necessidade de controle do volume de dados por sessão

# Alteração Dinâmica de Regras de Filtragem

---

- Solução para o problema de falta de seletividade dos filtros de pacotes
- Características a monitorar:
  - número de sessões de saída, por *host*
  - elevação da taxa de crescimento de sessões
  - volume máximo de dados
  - tempo máximo de conexão por sessão

# Filtragem por Conteúdo de Pacotes

---

- É possível para um invasor lançar ataques bem sucedidos sem a criação de um grande número de sessões
- Desejável descartar pacotes em função de padrões conhecidos de ataques no seu conteúdo
- Problemas deste método:
  - limitado a ataques conhecidos
  - lento (cópia entre *user* e *kernel space*)

# Outras Possibilidades

---

- Modificação de Pacotes
  - pode ser de difícil detecção
- Redirecionamento de Tráfego
- Filtragem em função do Sistema Operacional de Origem
  - *Passive Fingerprinting*

# sessionlimit

---

- Implementa algumas das técnicas descritas
  - linguagem C
  - interage com o `pf` do OpenBSD
  - Projeto Honeynet.BR
- Monitoração da Tabela de Estados
  - taxa de crescimento
  - número máximo de estados
  - número máximo de *bytes* transmitidos (ICMP)

## sessionlimit (cont)

---

- Caso um *host* satisfaça esses critérios:
  - inserção de regra de filtragem para este *host*
  - remoção das sessões de saída em andamento
- Apenas as sessões de saída são afetadas
- As regras de bloqueio expiram após um certo tempo
- Ações são registradas via `syslog`

# Conclusões

---

- Contenção de tráfego é extremamente importante em *honeynets*
- A ferramenta sessionlimit tem se mostrado eficiente na contenção de tráfego malicioso
- Limitações
  - consome muita CPU
  - planeja-se integrar filtragem por conteúdo
  - tornar a configuração mais flexível

# Referências

---

- Projeto HoneyNet.BR

<http://www.honeynet.org.br/>

- *The HoneyNet Project*

<http://project.honeynet.org/>

- *HoneyNet Research Alliance*

<http://project.honeynet.org/alliance/>