SISTEMAS DISTRIBUÍDOS E TOLERÂNCIA A FALHAS

Automatic detection of firewall misconfigurations using firewall and network routing policies

Flávio Amorim nº3409
Fábio Campos nº3481

ESTRUTURA DA APRESENTAÇÃO

- Introdução
- Contribuição
- Modelo de Rede
- Más configurações
  - Generalization and correlation
  - Shadowing
- Inconsistência entre-firewalls
- Inconsistências cross-path
- Ineficiência intra-firewall
- Método de detecção
- Descoberta do caminho
- Definição de regras
- Identificar a origem das más configurações
- Sugestões de Soluções
- detectar problemas do fluxo de danos, em uma conexão
- Detecções do Prometheus
- Conclusão

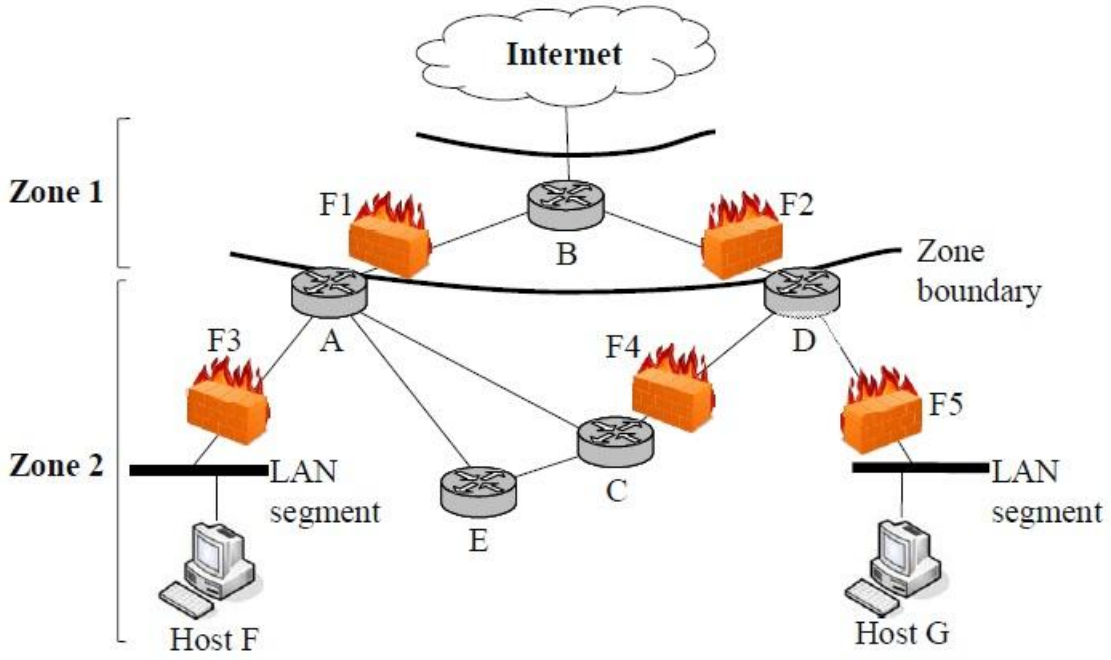
INTRODUÇÃO

- Este artigo fala sobre a importância das Firewalls nas organizações em que as suas redes crescem rapidamente.
- A importância de ter um sistema que consiga detectar problemas de configuração nas Firewalls que podem ser muito difíceis de resolver manualmente.

- Complexidade de gerir uma rede de firewalls em larga escala.

INTRODUÇÃO(CONT)

Um outro problema é o exemplo seguinte que demonstra os problemas que pode causar o routing nestes sistemas



*retirado do presente artigo

Introdução(cont)

- Existe uma grande quantidade de configurações mal feitas possíveis nas redes distribuídas.
- Apresentam a ferramenta “Prometheus” como solução para estes problemas em sistemas de larga escala
- O Prometheus constrói um modelo das politicas de firewalls , uma visão clara da topologia da rede e todos os caminhos possíveis.
- Depois cria uma relatório com as possíveis más configurações existentes.

Contribuição

- São os primeiros a incluir informação de routing dinâmico num modelo que permite descobrir problemas de configuração e mau routing de pacotes,
- Usaram uma rede de larga escala e dinâmica com cerca 30 nós e mais de 12000 regras de firewall.
- Os equipamentos são de várias marcas.

- O Prometheus detectou que as más configurações de firewalls, vem das suas próprias regras ou das outras firewalls na mesma rede.

Contribuição

- O Prometheus além de detectar os problemas ainda sugere as acções correctas a tomar para corrigir esses problemas.
- Oferece um mecanismo para testar a conectividade entre dois nós e se um determinado pacote consegue passar entre esses nós

Modelo de rede

II. Model of Network

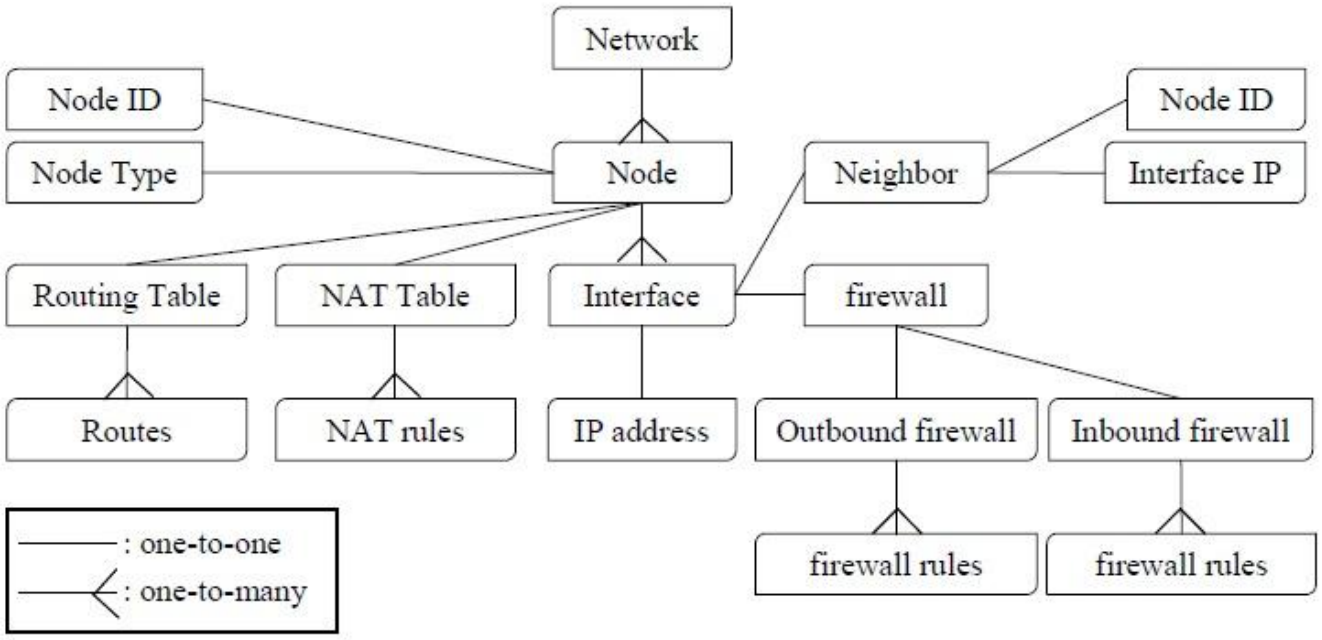


Figure 3. Network Model

***retirado do artigo**

Modelo de rede

- O modelo de rede é composto por nós que são representados por routers, switchs layer 3 e firewalls.
- Para construir modelo de rede a partir de uma grande variedade de fabricantes, foi usado o Nipper.
- Nipper: programa que faz parser das configurações a uma grande quantidade de equipamentos.

Modelo de rede

- Na tabela seguinte temos um exemplo de um conjunto de regras de uma firewall

TABLE I. Example Firewall Rule Set

Rule no.	Source		Destination		Protocol	Action
	IP	*Port*	*IP*	*Port*		
1	10.10.1.0/24	Any	10.1.0.0/24	Any	TCP	Deny
2	10.10.0.0/16	Any	10.1.0.0/24	Any	TCP	Accept
3	10.10.1.0/24	Any	10.1.0.5/32	Any	TCP	Accept
4	10.10.2.0/24	Any	10.1.0.8/32	Any	TCP	Accept
5	10.10.3.0/24	Any	10.2.0.0/25	80	TCP	Accept
6	10.10.3.0/24	Any	10.2.128.0/25	80	TCP	Accept
7	Any	Any	Any	Any	Any	Deny

***retirado do artigo**

Más configurações

- A má configuração de uma firewall surge dentro de ela própria isto quando algumas regras escondem de uma forma parcial ou totalmente outras regras.
- São detectadas como inconsistências e ineficiências do tipo:
  - intra-firewall
  - Inter-firewall
  - Cross-path
  - Ineficiência intra-firewall

Más configurações

- As inconsistências intra-firewall podem ser de três tipos:
  - Shadowing
  - Generalização
  - Correlação

Shadowing

- A regra que vamos analisar é um subconjunto da primeira regra da tabela I e as duas regras definem acções diferentes
- A regra 3 é escondida pela regra 1.
- O problema de uma regra esconder outra é criar um problema de conexão ou criar um problema de segurança.

Generalização e Correlação

- Um subconjunto de pacotes é compensado por uma regra anterior com diferentes acções.

- Se este subconjunto cobrir um conjunto completo de pacotes da regra anterior, esta regra é uma generalização da regra anterior.
- Se este subconjunto cobrir menos do que o conjunto completo da regra anterior, esta regra é uma correlação da regra anterior.
- Na tabela I, regra 2 é uma generalização da regra 1.
- Estas inconsistências são alertadas como avisos e não como erros.

Inconsistência entre-firewalls

- O Prometheus identifica inconsistências entre-firewalls ao longo de um conjunto de caminhos possíveis que são extraídos do estado routing actual.

- Ao contrário do problema anterior, o problema de shadowing só é levantado como um erro se for uma regra “accept”.
- A regra “deny” é lançada como aviso, uma regra destas ao longo do caminho da rede só faz o efeito de aumentar a segurança.

Inconsistências cross-path

- As inconsistências cross-path ocorrem quando os pacotes que esperamos que viagem por um caminho da rede são desviados para outro caminho.
- Se estes dois caminhos tiverem politicas de segurança diferentes, o novo caminho pode contradizer as politicas de segurança desejadas.

- Ver figura 1,
  - Caminho(F,A,C,D,G)
  - Caminho (F,A,B,D,G)
  - As politicas de segurança para os caminho acima são diferentes.

INEFICIÊNCIA INTRA-FIREWALL

- Prometheus identifica dois tipos ineficiência:
  - **Redundâncias:** refere a regras que quando são removidas não altera as politicas de segurança da firewall- Ver tabela I, a regra 4 é redundante porque aceita todos os pacotes que a regra 2 também aceita.
  - **Verbosities:** é um conjunto de regras que podem ser reduzidas a um conjunto pequeno de regras.
- A redundâncias e verbosities não são propriamente erros, então são lançadas como avisos pelo sistema Prometheus.

Método de detecção

- Passos para a detecção das configurações:
  1. Para cada par de nós da rede são determinados todos os caminhos possíveis .
     - Os caminhos são obtidos através da informação dos routers
     - Qualquer má configuração de routing vai influenciar este passo
  2. Para cada caminho são detectadas as más configurações das firewalls
     - Intra-Firewall
     - Inter-firewall

Descoberta do caminho

- Dado um par de nós, vai-se descobrir os caminhos alternativos possíveis
- O Prometheus usa um algoritmo parecido ao Depth-First Search o Depht-Limited Serch(DLS)
  - Está limitado a 30 saltos
- Para optimisar a procura tem um algoritmo de Overlapping Path Segments(OPS)

Definição de regras

- Nas regras são considerados os seguintes campos:
  - IP de origens
  - IP de destino
  - Porta de Origem
  - Porta de Destino
  - Protocolo
- Existem dois tipos de regras possiveis
  - Permitir o acesso
  - Bloquear o acesso
- Firewall:
  - Para cada regra são testados ao pacotes que por ela são tratados
  - De todos os pacotes que passam por uma regra existe um conjunto que é aceite e um conjunto que é descartado

Definição de regras

- Firewall:
  - Para cada regra são testados os pacotes
  - De todos os pacotes que são tratados por uma regra existe um conjunto que é aceite e um conjunto que é descartado
  - O conjunto descartado ou é analisado pela regra seguinte ou é bloqueado caso já não haja mais regras

Identificar a origem das más configurações

- Tendo em consideração os pacotes que passam pela firewall:
  - Caso seja uma regra que bloqueie o tráfego – o numero de pacotes enviados tem de ser igual ao numero de pacotes bloqueados na firewall
  - Caso que seja uma regra que permita o tráfego – o numero de pacotes aceites tem de ser igual ao numero de enviados
  - Caso não respeite um destes casos anteriores ou não receba nenhum pacote, é marcada como regra errada ou em conflito com outras regras

Sugestões de Soluções

- O prometheus não altera nem remove regras de firewalll
- A remoção ou a alteração de regra pode criar novos problemas com as regras
- Para cada erro ou aviso encontrado o Prometheus indica o erro e faz uma sugestão para a correcção do problema

Sugestoes de Soluções

- Exemplo

1) Verify the intended policy.
2) If rule 1 is the intended policy, remove rule 3.
   If rule 3 is the intended policy,
   (a) Modify rule 1 to … to remove the set that conflicts with rule 3.
   This can increase the number of rules by …
   (b) Move rule 3 to before rule 1.
   This can shadow another rule, rule x, so you need to move rule x to before rule 3 as well.

DETECTAR PROBLEMAS DO FLUXO DE DANOS, EM UMA CONEXÃO

- Quando é adicionada uma nova máquina ou serviço numa rede de grande escala

- O Prometheus permite ao ver qual são a firewall que terão que ser reconfiguradas

  - Introduzindo a portas , a maquina de origem e a máquina destino

Detecções do Prometheus

Node no.	# of firewalls	# of rules	Inconsistencies			Inefficiencies	
			S	G	C	R	V
1	161	4156	-	-	-	120	-
2	7	1566	-	-	-	52	-
3	8	678	-	-	-	3	-
4	57	3109	-	-	-	64	-
5	109	1775	-	-	-	7	-
6	6	49	-	4	-	-	-

Imagem retirado artigo - Detecções de más configurações[1]

Conclusão

- A configuração de politicas de segurança de firewalls e a sua interacção com politicas de routing tornam o processo difícil e complicado de executar correctamente. Neste artigo foi proposta a aplicação Prometheus que visa a facilitar o processo de configuração e testes das politicas.

- O Prometheus distingue-se de outras aplicações já existentes porque consegue reconhecer dinamicamente a rede, encontrar os caminhos mais viáveis entre pares de nós, e testar varias firewalls através da rede.

Conclusão(continuação)

- O Prometheus é uma aplicação com um potencial enorme em redes de maior complexidade em que possam existir vários caminhos possíveis, pois não só descreve as más configurações das firewalls como também calcula caminhos alternativos.

FIM

- Questões?????????