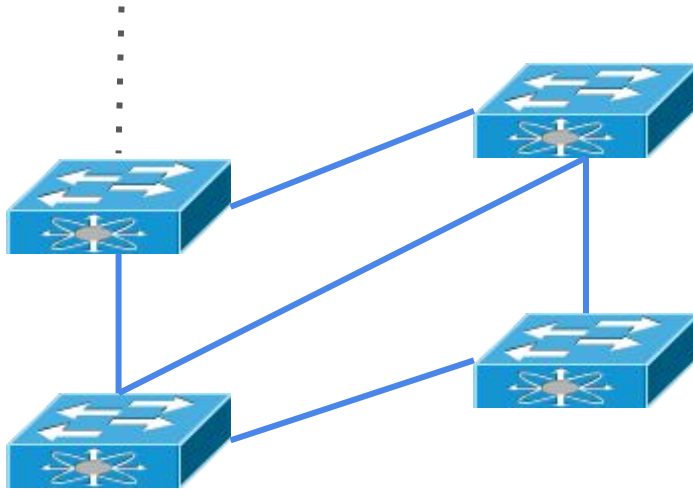


Políticas de Segurança de Firewall em Redes Híbridas

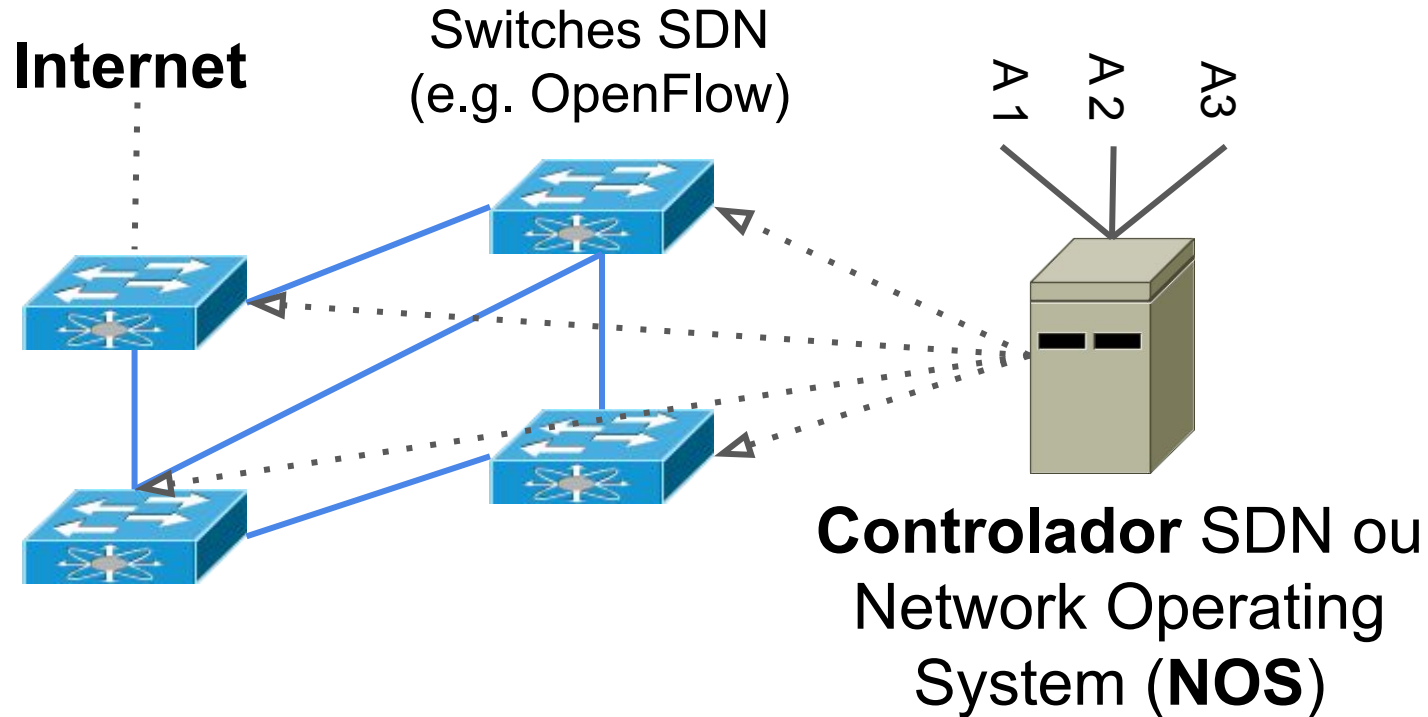
Maurício M. Fiorenza
Orientação: Diego Kreutz
Coorientação: Rodrigo Mansilha

Redes convencionais

Internet

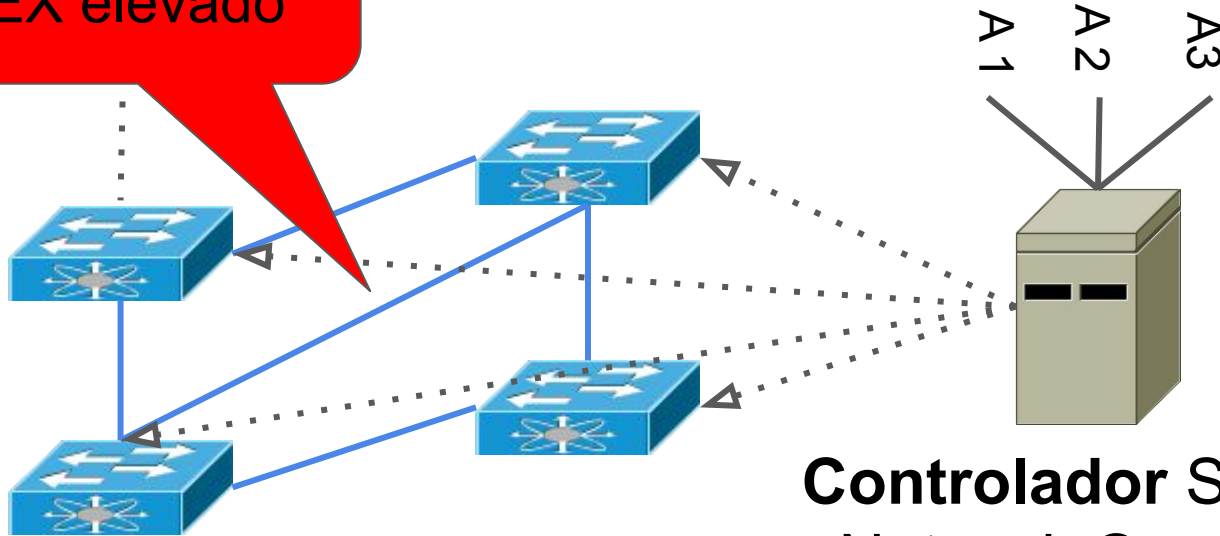


Redes SDN (Software-Defined Networking)



Migração de redes convencionais para SDN

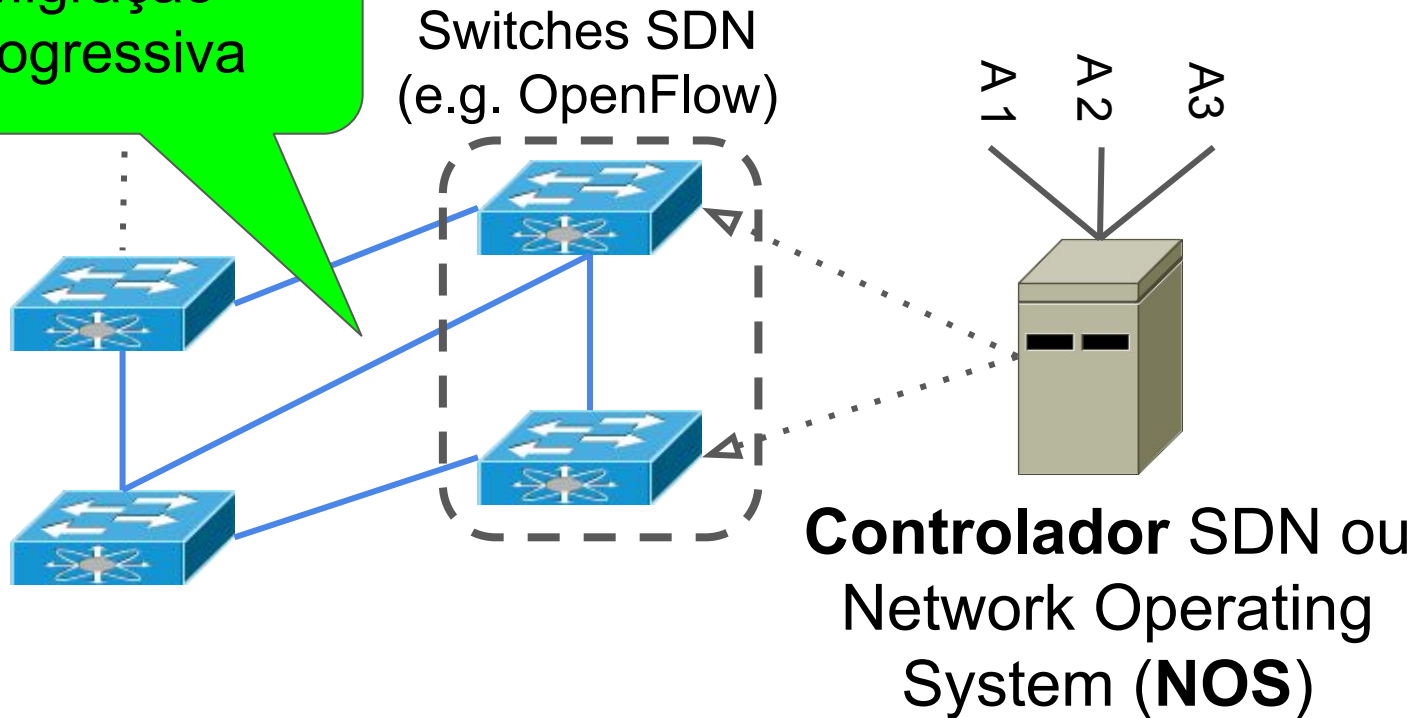
Custo CAPEX e
OPEX elevado



**Controlador SDN ou
Network Operating
System (NOS)**

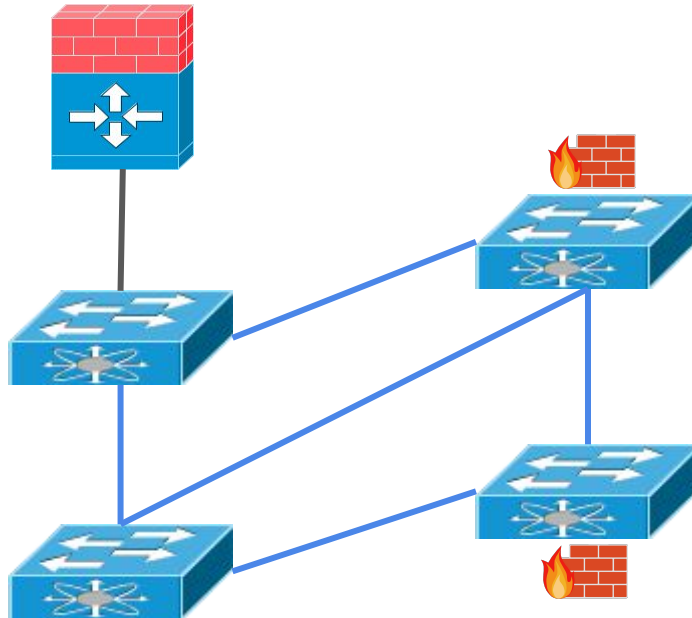
Rede híbrida: convencional + SDN

Migração
progressiva

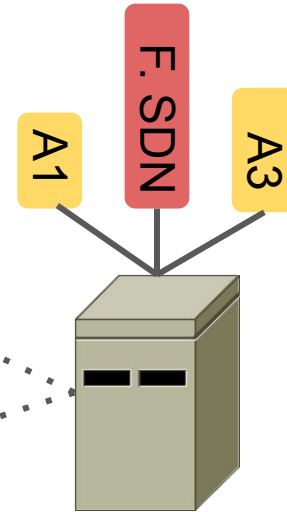


Rede híbrida: Gerenciamento de Firewalls

Firewall Cisco

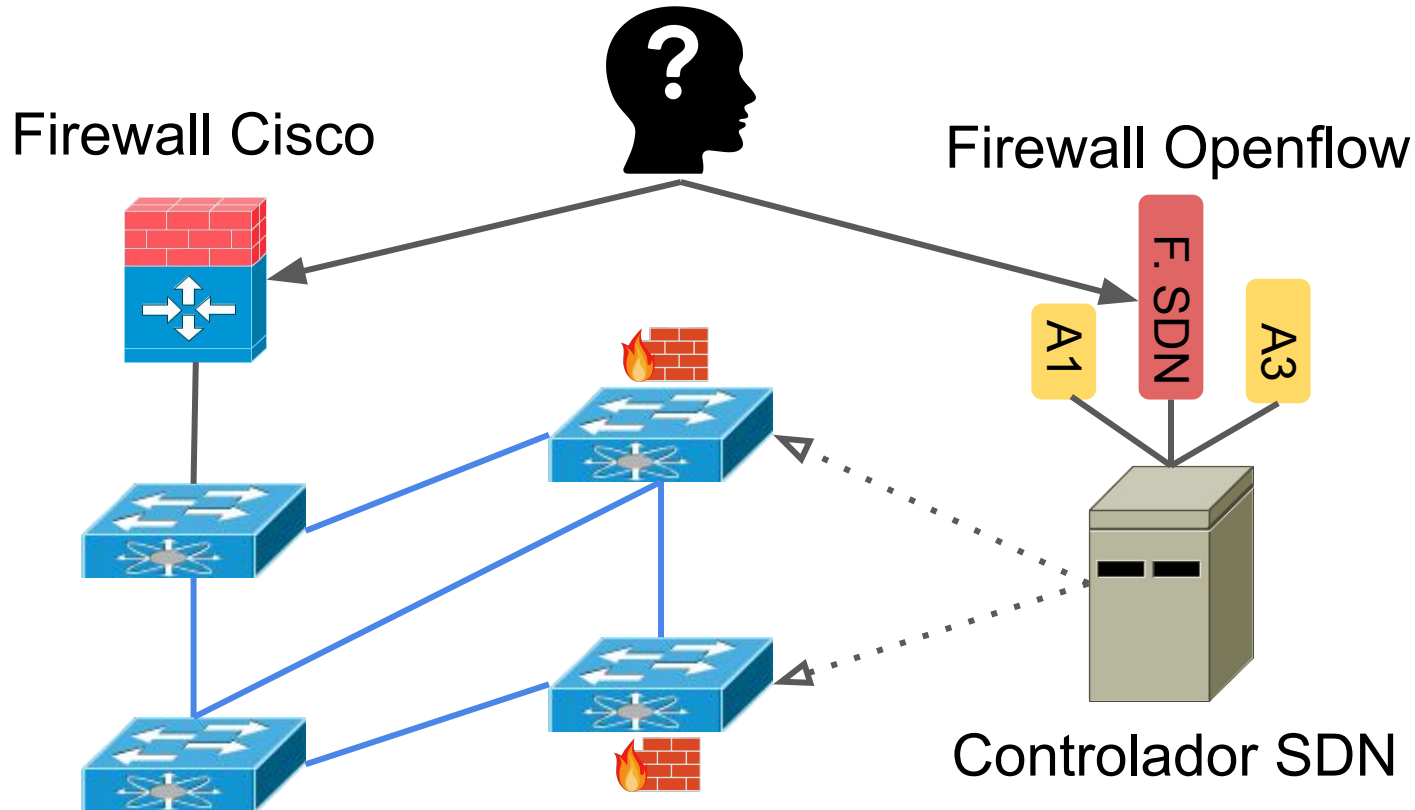


Firewall Openflow



Controlador SDN

Redes Híbridas: 1º Desafío



Desafio 1: Diversidade de Soluções

FORTINET.



FortiManager



Gufw

Fireflow



Cisco ASDM-IDM Launcher



P4Guard

Desafio 1: Diversidade de Soluções

FORTINET.



FortiManager



Cisco ASDM-IDM Launcher



Gufw

Fireflow



P4Guard

Desafio 1: Diversidade de Soluções

FORTINET.



FortiManager



Cisco ASDM-IDM Launcher



Gufw



Fireflow

P4Guard

Desafio 1: Diversidade de Soluções

FORTINET.



FortiManager



Gufw



Cisco ASDM-IDM Launcher



Fireflow

P4Guard

Desafio 1: Diversidade de Soluções

Solução	Arquitetura	Código fonte	Escopo
Cisco ASDM	Monolítica	Proprietário	Cisco
FortiManager	Monolítica	Proprietário	Fortinet
Gufw	Monolítica	Livre/Disponível	IPTables
Mignis	Monolítica	Livre/Disponível	IPTables
pfSense	Monolítica	Livre/Disponível	pfSense
Firewall for POX	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fireflow	Monolítica	Não disponibilizado	SDN (OpenFlow)
REFLO	Monolítica	Não disponibilizado	SDN (OpenFlow)
FlowTracker	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fortress	Monolítica	Não disponibilizado	SDN (OpenFlow)
AI-SDNF	Monolítica	Não disponibilizado	SDN (OpenFlow)
SMPU-P4	Monolítica	Não disponibilizado	SDN (P4)
P4GUARD	Monolítica	Não disponibilizado	SDN (P4)
FWunify	Em camadas e modular	Livre/Disponível	Todos

Desafio 1: Diversidade de Soluções

Solução	Arquitetura	Código fonte	Escopo
Cisco ASDM	Monolítica	Proprietário	Cisco
FortiManager	Monolítica	Proprietário	Fortinet
Gufw	Monolítica	Livre/Disponível	IPTables
Mignis	Monolítica	Livre/Disponível	IPTables
pfSense	Monolítica	Livre/Disponível	pfSense
Firewall for POX	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fireflow	Monolítica	Não disponibilizado	SDN (OpenFlow)
REFLO	Monolítica	Não disponibilizado	SDN (OpenFlow)
FlowTracker	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fortress	Monolítica	Não disponibilizado	SDN (OpenFlow)
AI-SDNF	Monolítica	Não disponibilizado	SDN (OpenFlow)
SMPU-P4	Monolítica	Não disponibilizado	SDN (P4)
P4GUARD	Monolítica	Não disponibilizado	SDN (P4)
FWunify	Em camadas e modular	Livre/Disponível	Todos

Desafio 1: Diversidade de Soluções

Solução	Arquitetura	Código fonte	Escopo
Cisco ASDM	Monolítica	Proprietário	Cisco
FortiManager	Monolítica	Proprietário	Fortinet
Gufw	Monolítica	Livre/Disponível	IPTables
Mignis	Monolítica	Livre/Disponível	IPTables
pfSense	Monolítica	Livre/Disponível	pfSense
Firewall for POX	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fireflow	Monolítica	Não disponibilizado	SDN (OpenFlow)
REFLO	Monolítica	Não disponibilizado	SDN (OpenFlow)
FlowTracker	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fortress	Monolítica	Não disponibilizado	SDN (OpenFlow)
AI-SDNF	Monolítica	Não disponibilizado	SDN (OpenFlow)
SMPU-P4	Monolítica	Não disponibilizado	SDN (P4)
P4GUARD	Monolítica	Não disponibilizado	SDN (P4)
FWunify	Em camadas e modular	Livre/Disponível	Todos

Desafio 1: Diversidade de Soluções

Solução	Arquitetura	Código fonte	Escopo
Cisco ASDM	Monolítica	Proprietário	Cisco
FortiManager	Monolítica	Proprietário	Fortinet
Gufw	Monolítica	Livre/Disponível	IPTables
Mignis	Monolítica	Livre/Disponível	IPTables
pfSense	Monolítica	Livre/Disponível	pfSense
Firewall for POX	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fireflow	Monolítica	Não disponibilizado	SDN (OpenFlow)
REFLO	Monolítica	Não disponibilizado	SDN (OpenFlow)
FlowTracker	Monolítica	Não disponibilizado	SDN (OpenFlow)
Fortress	Monolítica	Não disponibilizado	SDN (OpenFlow)
AI-SDNF	Monolítica	Não disponibilizado	SDN (OpenFlow)
SMPU-P4	Monolítica	Não disponibilizado	SDN (P4)
P4GUARD	Monolítica	Não disponibilizado	SDN (P4)
FWunify	Em camadas e modular	Livre/Disponível	Todos

Desafio 2: Diversidade de Sintaxes

Sintaxe Cisco ASA:

access-list **inside_access_in** *line 1 extended* **permit tcp 10.0.0.0 255.255.255.0 any eq http**

Sintaxe OpenFlow:

ovs-ofctl add-flow **br0** *dl_type=0x800,priority=65535,*
nw_src=10.0.0.0/255.255.255.0,nw_dst=0.0.0.0/0.0.0.0,nw_proto=6,tcp_dst=80,
action=normal

Desafio 2: Diversidade de Sintaxes

Sintaxe Check Point:

mgmt_cli add access-rule layer "inside" position 1 name "permit-http" source "10.0.0.0/24" destination "any" service "HTTP" action "accept"

Sintaxe Palo Alto:

set rulebase security rule permit-http from inside to outside source 10.0.0.0 255.255.255.0 destination any service http action allow before all

Sintaxe IPTables:

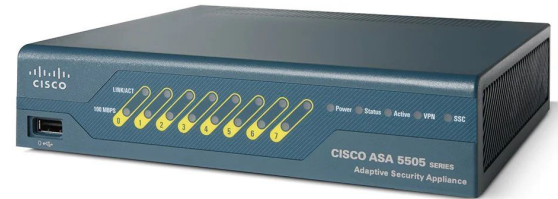
iptables -I FORWARD 1 -s 10.0.0.0/255.255.255.0 -d 0.0.0.0/0.0.0.0 -p tcp --dport 80 -j ACCEPT

Desafio 2: Diversidade de Sintaxes

Cisco PIX



Cisco ASA



Desafio 2: Diversidade de Sintaxes

Cisco PIX:

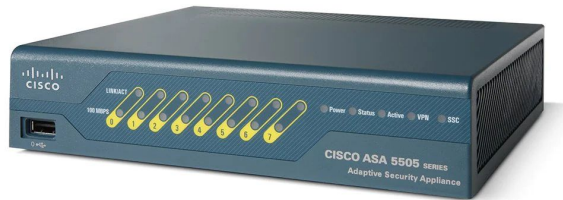
access-list 1 permit tcp 10.0.0.0 255.255.255.0 any eq http

Cisco ASA:

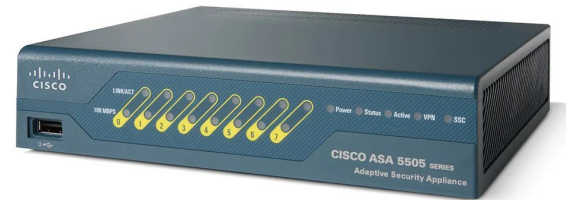
access-list inside_access_in line 1 extended permit tcp 10.0.0.0
255.255.255.0 any eq http

Desafio 2: Diversidade de Sintaxes

Cisco ASA 8.2



Cisco ASA 8.4



Desafio 2: Diversidade de Sintaxes

Sintaxe Cisco ASA 8.2:

```
static (inside,outside) tcp 200.19.0.30 90 10.0.0.30 80 netmask  
255.255.255.255 tcp 0 0 udp 0
```

Sintaxe Cisco ASA 8.4:

```
object network obj-10.0.0.30  
host 10.0.0.30  
nat static 200.19.0.30 service tcp 80 90
```

Desafio 2: Diversidade de Sintaxes

	Políticas utilizadas em firewalls modernos							
Linguagem	ACL	NAT 1to1	NAT Nto1	Traffic Shaping	URL Filter	Roteamento estático	Baseada em Intenções	Domínio de Aplicação
RichLanguage	✓	✓	✓	✗	✗	✗	✗	IPTables
PDLz	✗	✗	✗	✗	✗	✓	✗	IPTables
LAI	✓	✗	✗	✗	✗	✗	✓	WAN
NILE	✓	✗	✗	✓	✗	✗	✓	NFVs
FLIP	✓	✗	✗	✗	✗	✗	✗	Qualquer
FWS	✓	✓	✓	✗	✗	✗	✗	Qualquer
FIRMATO	✓	✗	✗	✗	✗	✗	✗	Qualquer
AFPL2	✓	✓	✓	✗	✗	✗	✗	Qualquer
FWLang	✓	✓	✓	✓	✓	✓	✓	Qualquer

Desafio 2: Diversidade de Sintaxes

Linguagem	Políticas utilizadas em firewalls modernos						Baseada em Intenções	Domínio de Aplicação
	ACL	NAT 1to1	NAT Nto1	Traffic Shaping	URL Filter	Roteamento estático		
RichLanguage	✓	✓	✓	✗	✗	✗	✗	IPTables
PDLz	✗	✗	✗	✗	✗	✓	✗	IPTables
LAI	✓	✗	✗	✗	✗	✗	✓	WAN
NILE	✓	✗	✗	✓	✗	✗	✓	NFVs
FLIP	✓	✗	✗	✗	✗	✗	✗	Qualquer
FWS	✓	✓	✓	✗	✗	✗	✗	Qualquer
FIRMATO	✓	✗	✗	✗	✗	✗	✗	Qualquer
AFPL2	✓	✓	✓	✗	✗	✗	✗	Qualquer
FWLang	✓	✓	✓	✓	✓	✓	✓	Qualquer

Desafio 2: Diversidade de Sintaxes

Linguagem	Políticas utilizadas em firewalls modernos						Baseada em Intenções	Domínio de Aplicação
	ACL	NAT 1to1	NAT Nto1	Traffic Shaping	URL Filter	Roteamento estático		
RichLanguage	✓	✓	✓	✗	✗	✗	✗	IPTables
PDLz	✗	✗	✗	✗	✗	✓	✗	IPTables
LAI	✓	✗	✗	✗	✗	✗	✓	WAN
NILE	✓	✗	✗	✓	✗	✗	✓	NFVs
FLIP	✓	✗	✗	✗	✗	✗	✗	Qualquer
FWS	✓	✓	✓	✗	✗	✗	✗	Qualquer
FIRMATO	✓	✗	✗	✗	✗	✗	✗	Qualquer
AFPL2	✓	✓	✓	✗	✗	✗	✗	Qualquer
FWLang	✓	✓	✓	✓	✓	✓	✓	Qualquer

Desafio 2: Diversidade de Sintaxes

Linguagem	Políticas utilizadas em firewalls modernos						Baseada em Intenções	Domínio de Aplicação
	ACL	NAT 1to1	NAT Nto1	Traffic Shaping	URL Filter	Roteamento estático		
RichLanguage	✓	✓	✓	✗	✗	✗	✗	IPTables
PDLz	✗	✗	✗	✗	✗	✓	✗	IPTables
LAI	✓	✗	✗	✗	✗	✗	✓	WAN
NILE	✓	✗	✗	✓	✗	✗	✓	NFVs
FLIP	✓	✗	✗	✗	✗	✗	✗	Qualquer
FWS	✓	✓	✓	✗	✗	✗	✗	Qualquer
FIRMATO	✓	✗	✗	✗	✗	✗	✗	Qualquer
AFPL2	✓	✓	✓	✗	✗	✗	✗	Qualquer
FWLang	✓	✓	✓	✓	✓	✓	✓	Qualquer

Contribuições

1. **FWunify**: uma arquitetura em camadas, modular e extensível, para o gerenciamento de firewalls em redes híbridas;
2. **FWlang**: uma linguagem para definição de regras utilizadas em firewalls, baseada em intenções;
3. Uma avaliação experimental da FWunify e da FWlang.

FWunify

FWlang

Implementação

Avaliação

FWunify

FWlang

Implementação

Avaliação

FWunify - Visão de Camadas e Módulos

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify - Arquitetura

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

OpenFlow

P4

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



P4



FWunify

FWlang

Implementação

Avaliação

FWlang - Objetivos

- Representação de políticas utilizadas em firewalls;
- Alto nível de abstração, sem ligação com sintaxes específicas.

FWlang - Baseada em Intenções

- Intenção (ou “Intent”) é uma evolução do conceito de política (“policy”);

FWlang - Baseada em Intenções

- Intenção (ou “Intent”) é uma evolução do conceito de política (“policy”);

```
rule family="ipv4" source address="10.0.0.10" drop
```

FWlang - Baseada em Intenções

- Intenção (ou “Intent”) é uma evolução do conceito de política (“policy”);

```
rule family="ipv4" source address="10.0.0.10" drop
```

```
define intent acl:  
  from      endpoint('10.0.0.10')  
  block     traffic('all')
```

FWlang - Levantamento de políticas

- Políticas utilizadas em ambientes reais;
- Documentação oficial dos fabricantes;
- Literatura científica.

FWlang - Levantamento de políticas

- ACL
- Traffic Shaping
- Filtros de URL
- NAT 1to1
- Roteamento Estático
- NAT Nto1

FWlang - Levantamento de políticas

- ACL



- Traffic Shaping



- Filtros de URL



- NAT 1to1



- Roteamento Estático



- NAT Nto1



FWlang - Levantamento de políticas

- ACL



- Filtros de URL



- Roteamento Estático



- Traffic Shaping



- NAT 1to1



- NAT Nto1



FWlang - Levantamento de políticas

- ACL



- Traffic Shaping



- Filtros de URL



- NAT 1to1



- Roteamento Estático




- NAT Nto1



FWlang - Requisitos das políticas

Política	Requisitos	Política	Requisitos
ACL	<i>Origem</i>	Traffic Shaping	<i>Origem</i>
	<i>Destino</i>		<i>Destino</i>
	<i>Tráfego</i>		<i>Tráfego</i>
	<i>Ação</i>		<i>Largura de banda</i>
	<i>Prioridade</i>		<i>Prioridade</i>
	Serviço de logs		Serviço de logs
	Intervalo de tempo		Intervalo de tempo
	Descrição		Descrição
Filtro de URL	<i>Origem</i>	NAT Nto1	<i>Origem</i>
	<i>Destino</i>		<i>Destino</i>
	<i>Tráfego</i>		Serviço de logs
	<i>Ação/Largura de banda</i>		Descrição
	<i>Prioridade</i>	NAT 1to1	<i>Origem</i>
	Serviço de logs		<i>Destino</i>
	Intervalo de tempo		<i>Tráfego</i>
	Descrição		Serviço de logs
Roteamento estático	<i>Origem/destino</i>		Descrição
	<i>Gateway</i>		
	Intervalo de tempo		
	Descrição		

FWlang - Requisitos das políticas

Política	ACL
ACL 	Origem
	Destino
	Tráfego
Filtro de URL	Ação
	Prioridade
	Serviço de logs
Roteamento estático	Intervalo de tempo
	Descrição

FWlang - Gramática

- Gramática base: linguagem NILE:
 - Representação de intenções;
 - Gerenciamento de serviços em NFVs;
 - Reutilização de operadores básicos.

FWlang - Gramática

```
define intent acl:  
name          text('netAhttp')  
from         range('10.0.0.0/24')  
to           endpoint('200.19.0.100')  
block        traffic('http')  
order        before('all')  
description  text('all')  
del         middlebox('cisco-1')
```

FWunify

FWlang

Implementação

Avaliação

FWunify e FWlang - Implementação

Aplicações de Gerenciamento

Editor de texto

Interface gráfica

Chatbot

Linguagem universal (FWlang)

Interface Norte

API REST

API IPC

API Web Sockets

Controle de Acesso

Mecanismos de controle de acesso

Resolução de Conflitos

Algoritmos para detectar e resolver conflitos

Microserviços de Tradução

Cisco M1

IPTables

Palo Alto

Openflow

Interface Sul

SSH

SNMP

OpenFlow

Netconf

Dispositivos e Serviços

Firewall Cisco



OpenFlow



Palo Alto



FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask	Serviço REST	
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

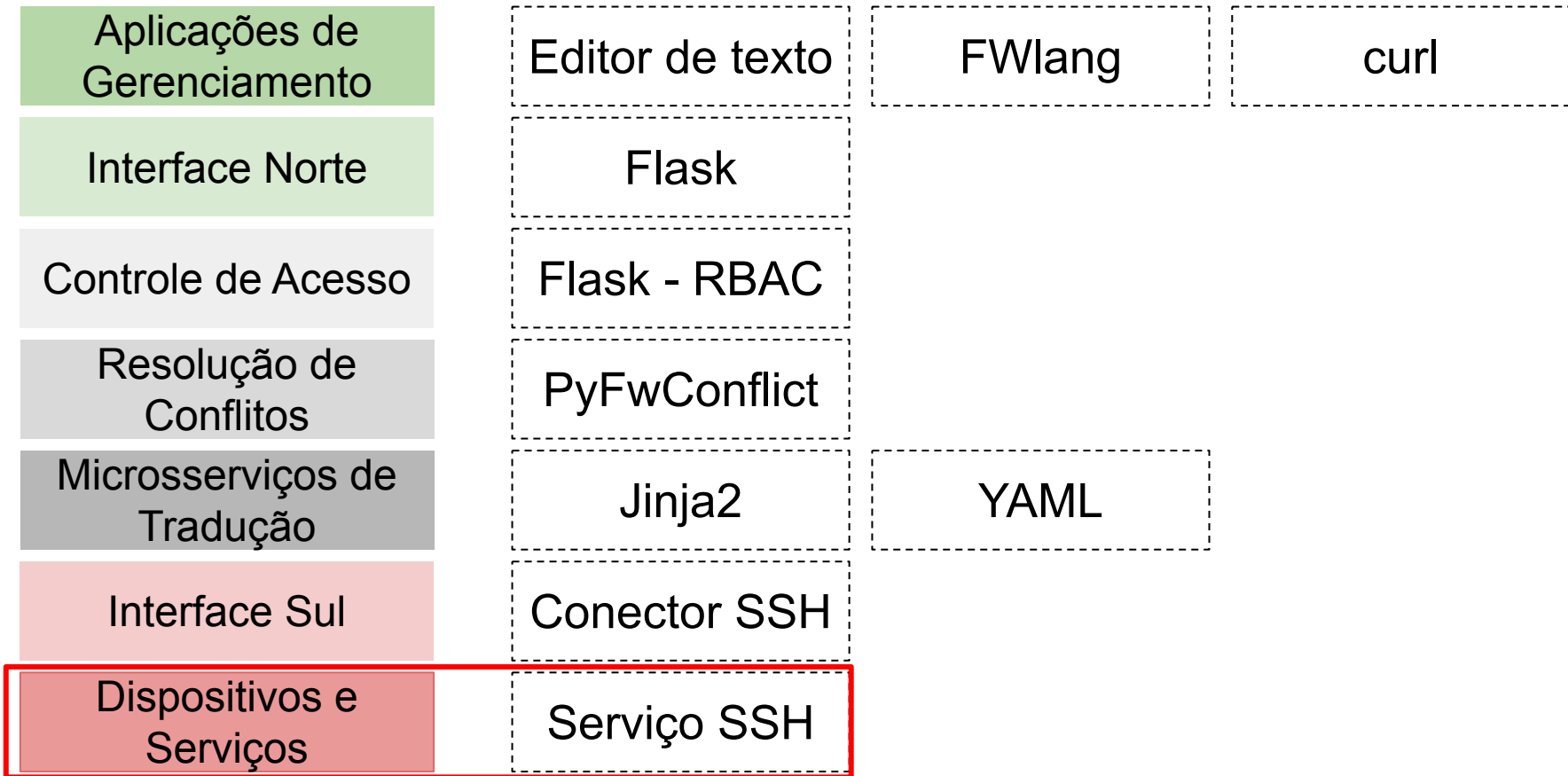
FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação

Aplicações de Gerenciamento	Editor de texto	FWlang	curl
Interface Norte	Flask		
Controle de Acesso	Flask - RBAC		
Resolução de Conflitos	PyFwConflict		
Microserviços de Tradução	Jinja2	YAML	
Interface Sul	Conector SSH		
Dispositivos e Serviços	Serviço SSH		

FWunify e FWlang - Implementação



FWunify e FWlang - Implementação

- Utilização prática com alunos de 3 disciplinas:
 - Coleta de feedbacks;
 - Atualizações e evoluções na implementação.
- <https://github.com/mmfiorenza/fwunify>

FWunify

FWlang

Implementação

Avaliação

Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

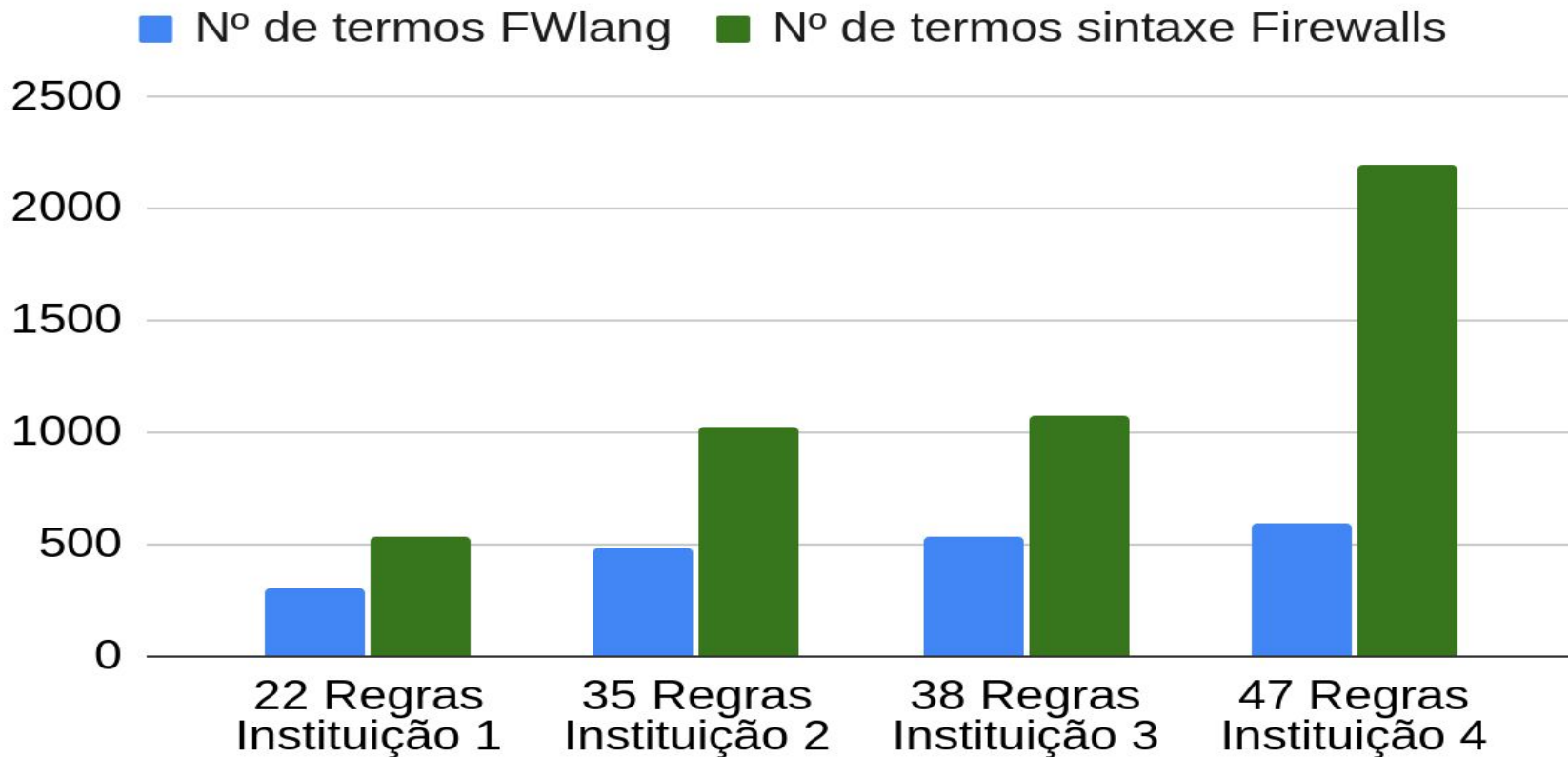
Avaliação FWlang - Número de termos

- 4 conjuntos de regras reais:
 - 22 políticas ACL;
 - 35 políticas ACL;
 - 38 políticas ACL;
 - 8 políticas ACL + 39 políticas NAT 1to1.

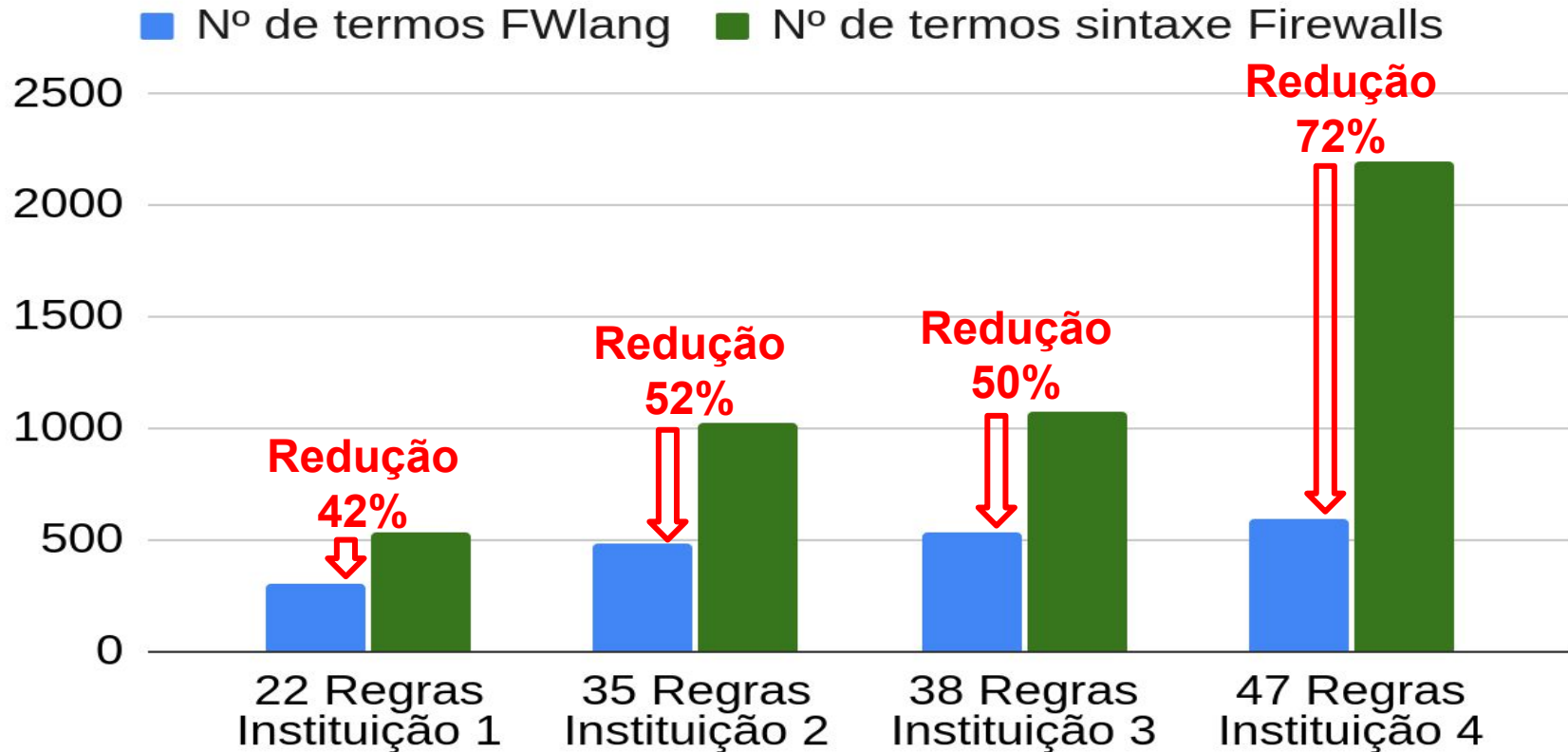
Avaliação FWlang - Número de termos

- Regras representadas utilizando:
 - FWlang;
 - Sintaxe dos firewalls Cisco, IPTables e Openflow.
- Contagem manual dos termos estáticos.

Avaliação FWlang - Número de termos



Avaliação FWlang - Número de termos



Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

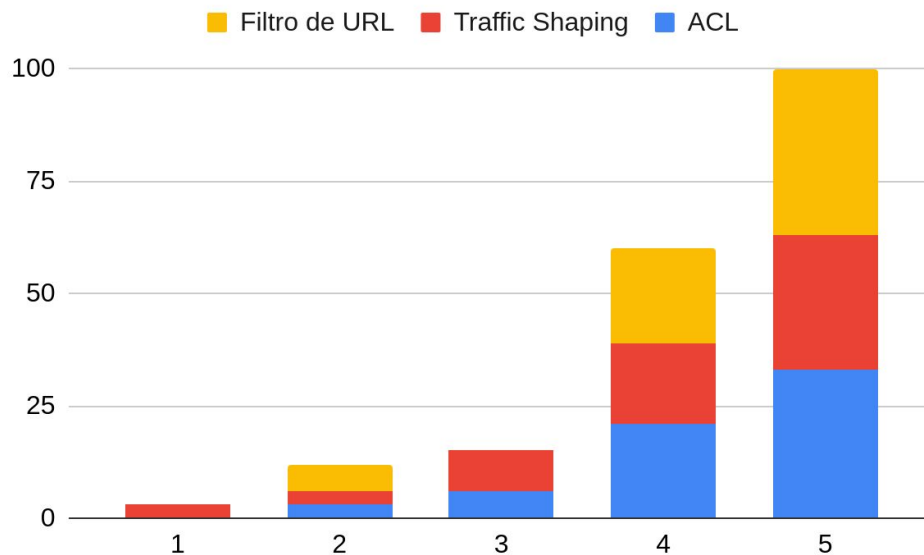
FWunify - Aplicação de múltiplas intenções

FWlang - Intuitividade X Complexidade

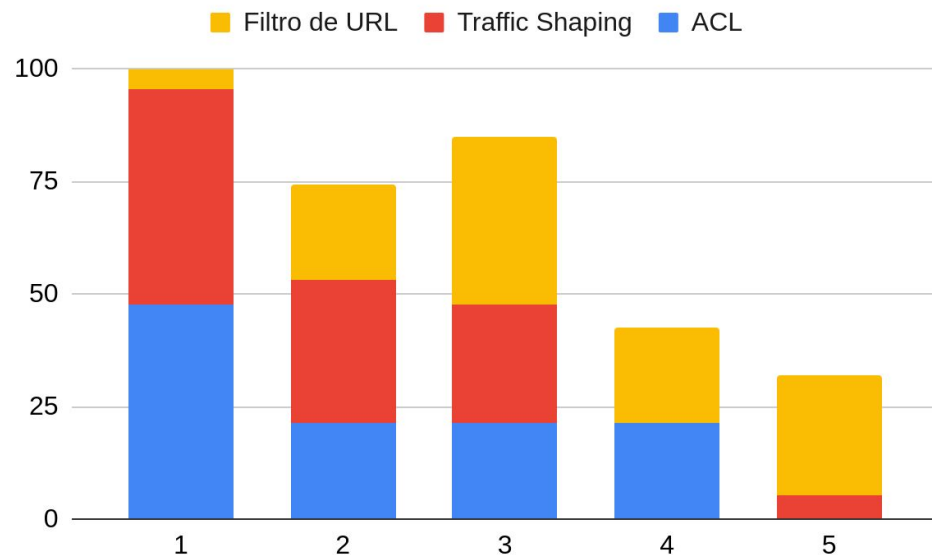
- Pesquisa realizada com administradores de sistemas:
 - <https://forms.gle/iSQt27j26Xcnp2hT6>
 - 21 participantes.
- 3 políticas (filtro de URL, *traffic shaping* e ACL):
 - em FWlang;
 - em outras sintaxes dos firewalls.

FWlang - Intuitividade

FWlang



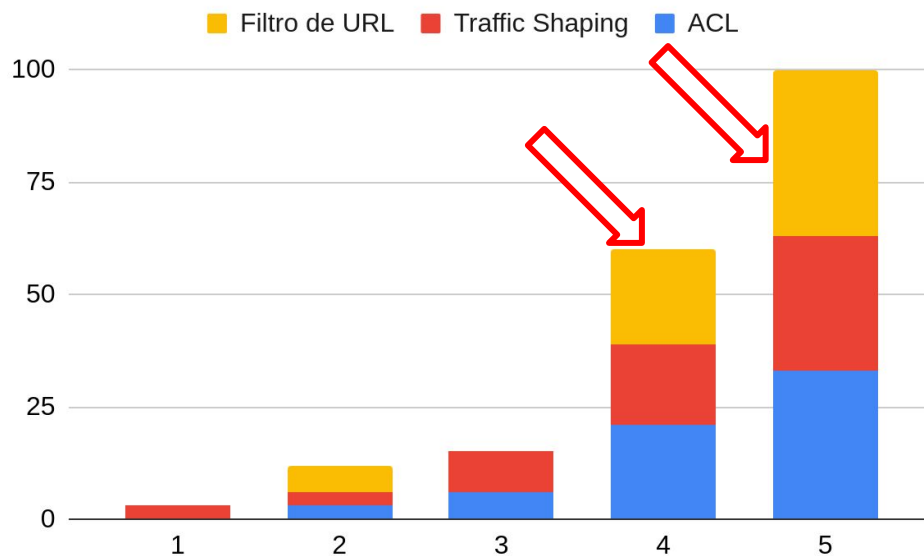
Sintaxes firewalls



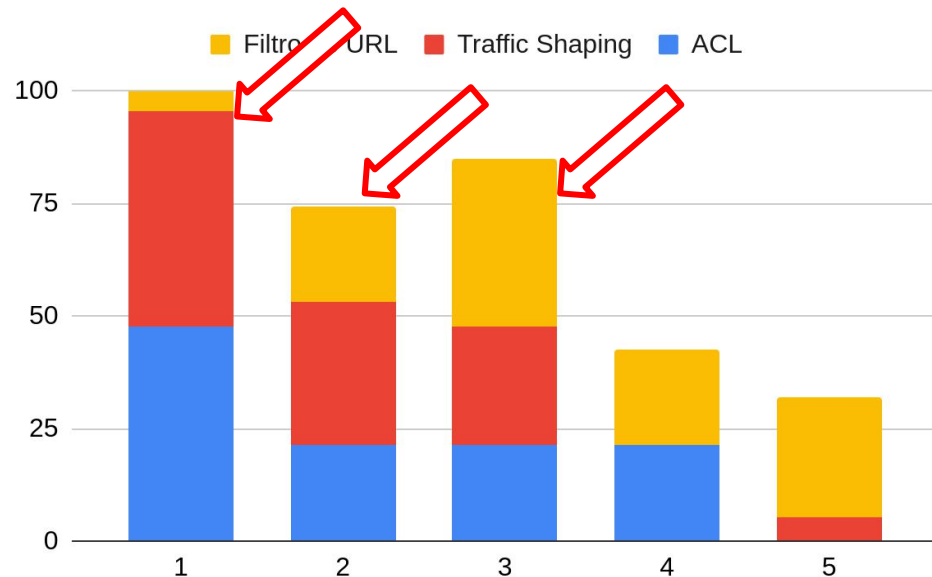
1 significa pouco intuitiva e 5 significa muito intuitiva

FWlang - Intuitividade

FWlang



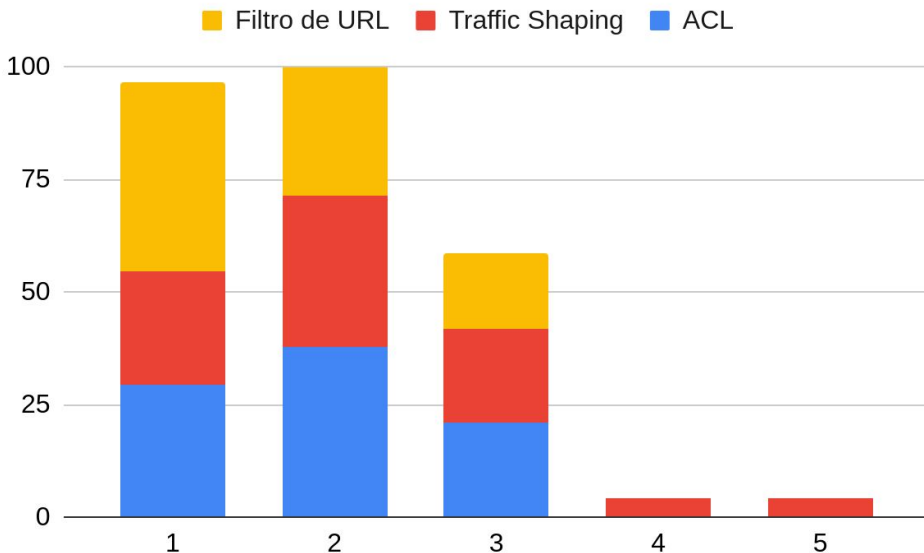
Sintaxes firewalls



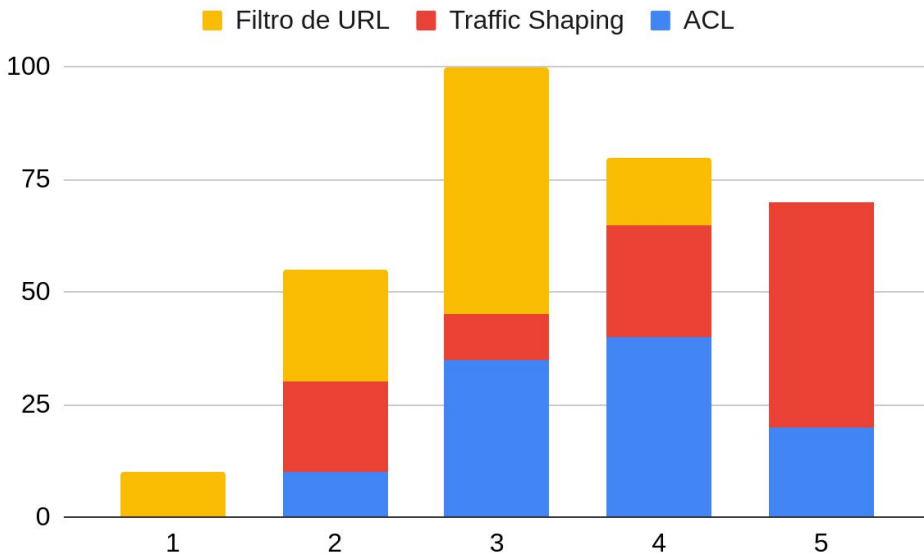
1 significa pouco intuitiva e 5 significa muito intuitiva

FWlang - Complexidade

FWlang



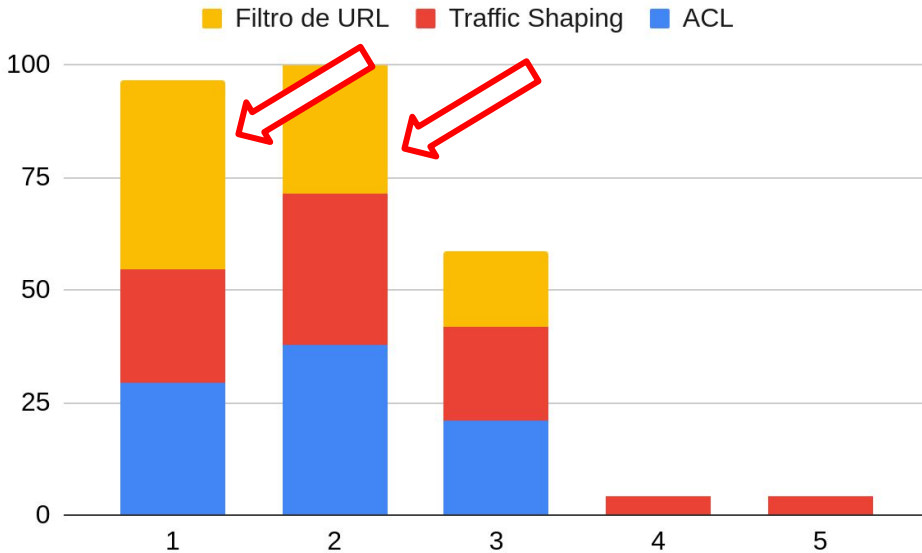
Sintaxes firewalls



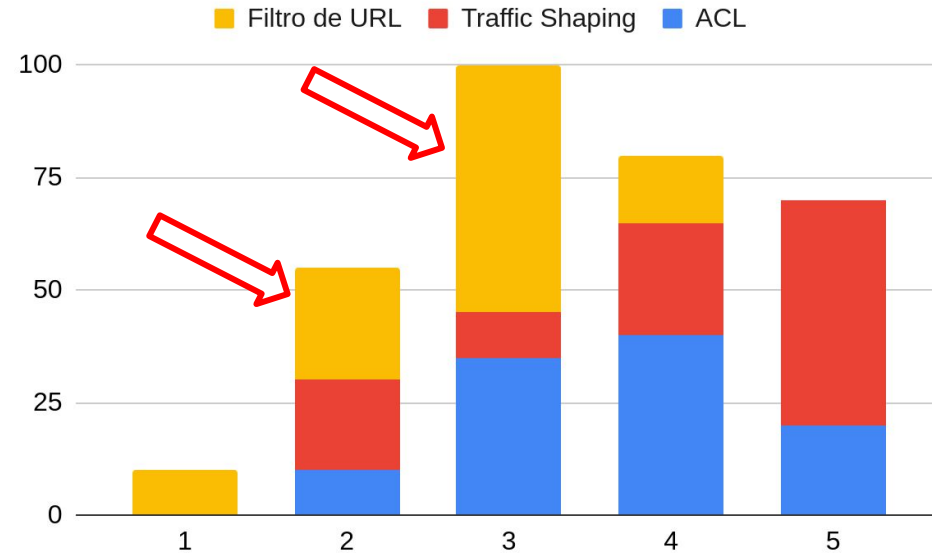
1 significa pouco complexa e 5 significa muito complexa

FWlang - Complexidade

FWlang



Sintaxes firewalls



1 significa pouco complexa e 5 significa muito complexa

Avaliação

FWlang - Complexidade e propensão a erros

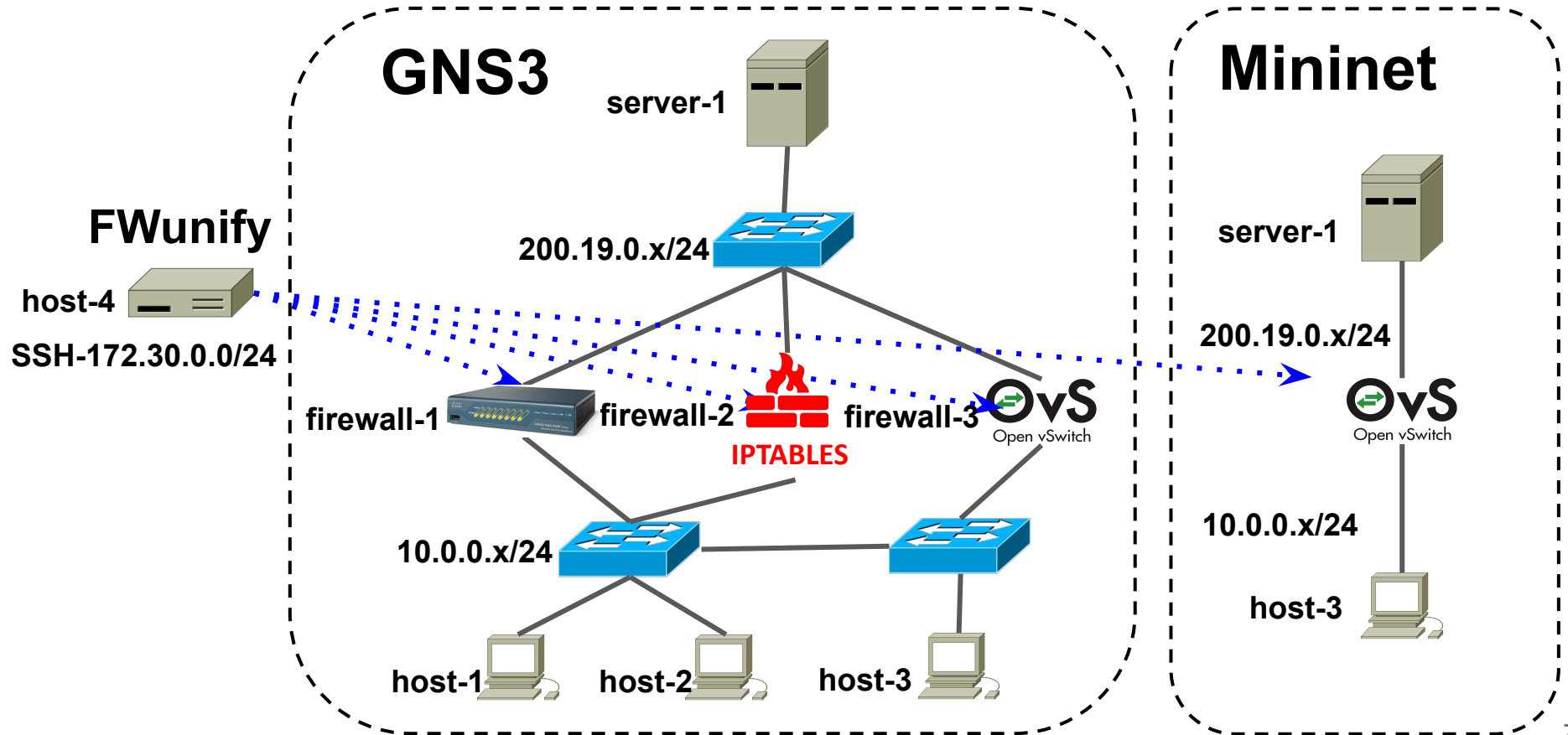
FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

FWunify - Ambientes de testes



Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

FWunify - Corretude da Tradução

- Definidas 3 políticas a serem traduzidas:
 - ACL;
 - NAT 1to1;
 - Traffic Shaping.
- Resultados das traduções comparados com as sintaxes esperadas.

Avaliação da Tradução - NAT 1to1

```
define intent nat_1to1:  
from      endpoint('200.19.0.50')  
to        endpoint('10.0.0.50')  
for       port('protocol:tcp|src_port:80|dst_port:90')  
del       middlebox('cisco-1')
```

Avaliação da Tradução - NAT 1to1

```
define intent nat_1to1:  
from      endpoint('200.19.0.50')  
to        endpoint('10.0.0.50')  
for       port('protocol:tcp|src_port:80|dst_port:90')  
del       middlebox('cisco-1')
```

Sintaxe do comando esperado:

object network “IP público”

no nat static “IP interno” service

“protocolo” “porta origem” “porta destino”

no object network “IP válido”

no object network “IP interno”

Avaliação da Tradução - NAT 1to1

```
define intent nat_1to1:  
from      endpoint('200.19.0.50')  
to        endpoint('10.0.0.50')  
for       port('protocol:tcp|src_port:80|dst_port:90')  
del       middlebox('cisco-1')
```

Sintaxe do comando esperado:

```
object network "IP público"  
no nat static "IP interno" service  
"protocolo" "porta origem" "porta destino"  
no object network "IP válido"  
no object network "IP interno"
```

Comando gerado:

```
object network 200.19.0.50  
no nat static 10.0.0.50 service tcp 80 90  
no object network 200.19.0.50  
no object network 10.0.0.50
```

Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

FWunify - Eficácia das políticas aplicadas

- Definição de 2 políticas de segurança:
 - ACL, aplicada em todos os firewalls;
 - *Traffic Shaping*, aplicada ao firewall OpenFlow.

ACL Bloqueio HTTP (Intent)

define intent acl:

name **text('deny-netA-h100-http')**

from **range('10.0.0.0/24')**

to **endpoint('200.19.0.100')**

block **traffic('http')**

order **before('all')**

add/del **middlebox('cisco-1'),middlebox('iptables-1'),**
 middlebox('openflow-1')

ACL Bloqueio HTTP (resultado)

```
user@host-1: ~  
Arquivo Editar Abas Ajuda  
user@host-1:~$ bash port-response.sh  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
^C  
user@host-1:~$
```

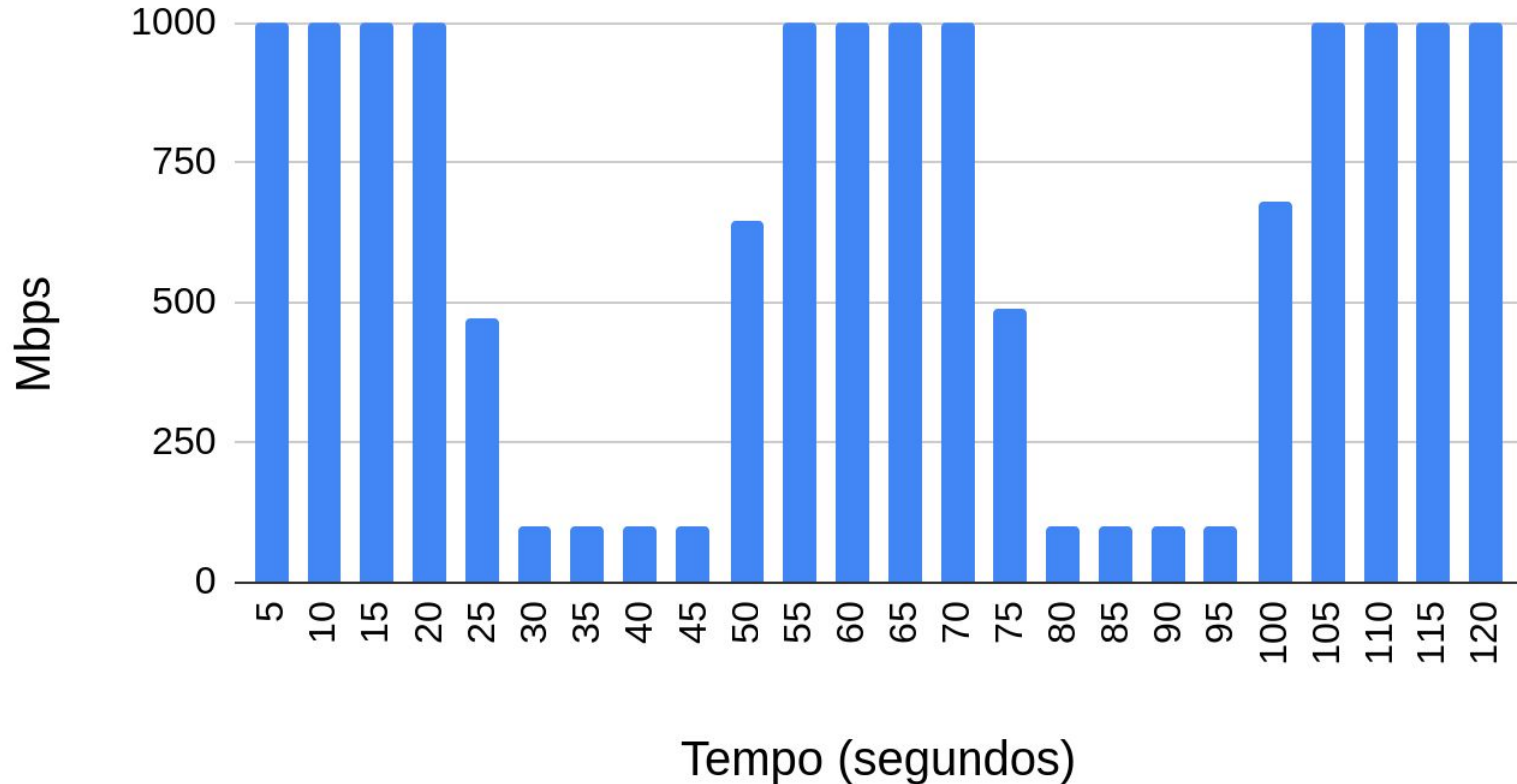
```
user@host-2: ~  
Arquivo Editar Abas Ajuda  
user@host-2:~$ bash port-resonse.sh  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
^C  
user@host-2:~$
```

```
user@host-3: ~  
Arquivo Editar Abas Ajuda  
user@host-3:~$ bash port-response.sh  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 failed.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
Connection to 200.19.0.100:80 successful.  
^C  
user@host-3:~$
```

ACL Traffic Shaping (Intent)

```
define intent acl:  
name      text('limit_net-h100-100m')  
from      range('10.0.0.0/24')  
to        endpoint('200.19.0.100')  
order     before('all')  
for       traffic('udp/5555')  
block     throughput('100Mbps')  
add/del   middlebox('openflow-1')
```


ACL Traffic Shaping (resultado)



Avaliação

FWlang - Complexidade e propensão a erros

FWlang - Intuitividade e complexidade

FWunify - Corretude da tradução

FWunify - Eficácia das políticas

FWunify - Aplicação de múltiplas intenções

FWunify - Aplicação de Múltiplas Intenções

- Definidas 6 intenções do tipo ACL:
 - Inspiradas em regras reais de uma instituição;
 - Adicionadas em uma ordem estabelecida;
 - Obedecendo a ordem de prioridade definida.

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all
- permit-net-all-http

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all
- permit-net-all-http
- permit-net-all-https

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all
- permit-h10-h20-mysql
- permit-net-all-http
- permit-net-all-https

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all
- permit-net-all-http
- permit-net-all-https
- permit-h10-h20-mysql
- drop-net-h20-mysql

FWunify - Aplicação de Múltiplas Intenções

- drop-all-all
- permit-net-all-http
- permit-net-all-https
- permit-h10-h20-mysql
- drop-net-h20-mysql
- drop-incident-h21

FWunify - Aplicação de Múltiplas Intenções

- Comparação das regras adicionadas:
 - Manualmente;
 - Via FWunify.

FWunify - Inserção Manual Cisco

Cisco:

- *access-list inside_access_in extended deny ip any any*
- *access-list inside_access_in line 1 extended permit tcp 10.0.0.0 255.255.255.0 any eq 80*
- *access-list inside_access_in line 1 extended permit tcp 10.0.0.0 255.255.255.0 any eq 443*
- *access-list inside_access_in line 3 extended permit tcp host 10.0.0.10 host 200.19.0.10 eq 3306*
- *access-list inside_access_in line 4 extended deny tcp 10.0.0.0 255.255.255.0 host 200.19.0.10 eq 3306*
- *access-list inside_access_in line 1 extended deny ip any host 200.19.0.20*

FWunify - Intenções em FWlang

define intent acl:

name **text**('permit-h10-h20-mysql')

from **endpoint**('10.0.0.10')

to **endpoint**('200.19.0.10')

allow **traffic**('tcp/3306')

order **before**('drop-all-all')

add **middlebox**('cisco-1'),**middlebox**('iptables-1'),**middlebox**('openflow-1')

FWunify - Intenções em FWlang

define intent acl:

name text('permit-h10-h20-mysql')

from endpoint('10.0.0.10')

to endpoint('200.19.0.10')

allow traffic('tcp/3306')


order before('drop-all-all')

add middlebox('cisco-1'),middlebox('iptables-1'),middlebox('openflow-1')

FWunify - Intenções em FWlang

define intent acl:

```
name    text('permit-h10-h20-mysql')
from    endpoint('10.0.0.10')
to      endpoint('200.19.0.10')
allow   traffic('tcp/3306')
order   before('drop-all-all')
add     middlebox('cisco-1'),middlebox('iptables-1'),middlebox('openflow-1')
```



define intent acl:

```
name    text('drop-net-h20-mysql')
from    range('10.0.0.0/24')
to      endpoint('200.19.0.10')
block   traffic('tcp/3306')
order   after('permit-h10-h20-mysql')
add     middlebox('cisco-1'),middlebox('iptables-1'),middlebox('openflow-1')
```

FWunify - Resultados (Manual e FWunify)

```
firewall-1-cisco# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
    alert-interval 300
access-list inside_access_in; 6 elements; name hash: 0x433a1af1
access-list inside_access_in line 1 extended deny ip any host 200.19.0.20 (hitcnt=0) 0xfdb26f1a
access-list inside_access_in line 2 extended permit tcp 10.0.0.0 255.255.255.0 any eq https (hitcnt=0) 0xec5c9adf
access-list inside_access_in line 3 extended permit tcp 10.0.0.0 255.255.255.0 any eq www (hitcnt=0) 0xa26ab2db
access-list inside_access_in line 4 extended permit tcp host 10.0.0.10 host 200.19.0.10 eq 3306 (hitcnt=0) 0xd683cdb3
access-list inside_access_in line 5 extended deny tcp 10.0.0.0 255.255.255.0 host 200.19.0.10 eq 3306 (hitcnt=0) 0x16c38c22
access-list inside_access_in line 6 extended deny ip any any (hitcnt=0) 0xbe9efe96
```

FWunify - Resultados (Manual e FWunify)

```
firewall-1-cisco# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
    alert-interval 300
access-list inside_access_in; 6 elements; name hash: 0x433a1af1
access-list inside_access_in line 1 extended deny ip any host 200.19.0.20 (hitcnt=0) 0xfdb26f1a
access-list inside_access_in line 2 extended permit tcp 10.0.0.0 255.255.255.0 any eq https (hitcnt=0) 0xec5c9adf
access-list inside_access_in line 3 extended permit tcp 10.0.0.0 255.255.255.0 any eq www (hitcnt=0) 0xa26ab2db
access-list inside_access_in line 4 extended permit tcp host 10.0.0.10 host 200.19.0.10 eq 3306 (hitcnt=0) 0xd683cdb3
access-list inside_access_in line 5 extended deny tcp 10.0.0.0 255.255.255.0 host 200.19.0.10 eq 3306 (hitcnt=0) 0x16c38c22
access-list inside_access_in line 6 extended deny ip any any (hitcnt=0) 0xbe9efe96
```

```
firewall-1-cisco# show access-list
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096)
    alert-interval 300
access-list inside_access_in; 6 elements; name hash: 0x433a1af1
access-list inside_access_in line 1 extended deny ip any host 200.19.0.20 (hitcnt=0) 0xfdb26f1a
access-list inside_access_in line 2 extended permit tcp 10.0.0.0 255.255.255.0 any eq https (hitcnt=0) 0xec5c9adf
access-list inside_access_in line 3 extended permit tcp 10.0.0.0 255.255.255.0 any eq www (hitcnt=0) 0xa26ab2db
access-list inside_access_in line 4 extended permit tcp host 10.0.0.10 host 200.19.0.10 eq 3306 (hitcnt=0) 0xd683cdb3
access-list inside_access_in line 5 extended deny tcp 10.0.0.0 255.255.255.0 host 200.19.0.10 eq 3306 (hitcnt=0) 0x16c38c22
access-list inside_access_in line 6 extended deny ip any any (hitcnt=0) 0xbe9efe96
```


Considerações Finais - Contribuições

- Arquitetura FWunify
- Linguagem FWlang
- Avaliação experimental da arquitetura e linguagem

Considerações Finais - Trabalhos Futuros

- Evolução do FWunify para uma solução IBF (Intent-Based Firewalling)
- Incluir outros mecanismos e técnicas a camada de resolução de conflitos
- Extensão da FWlang para outros tipos de políticas não mapeadas

Considerações Finais - Trabalhos Futuros

Linguagem Natural

Monitoramento e inteligência



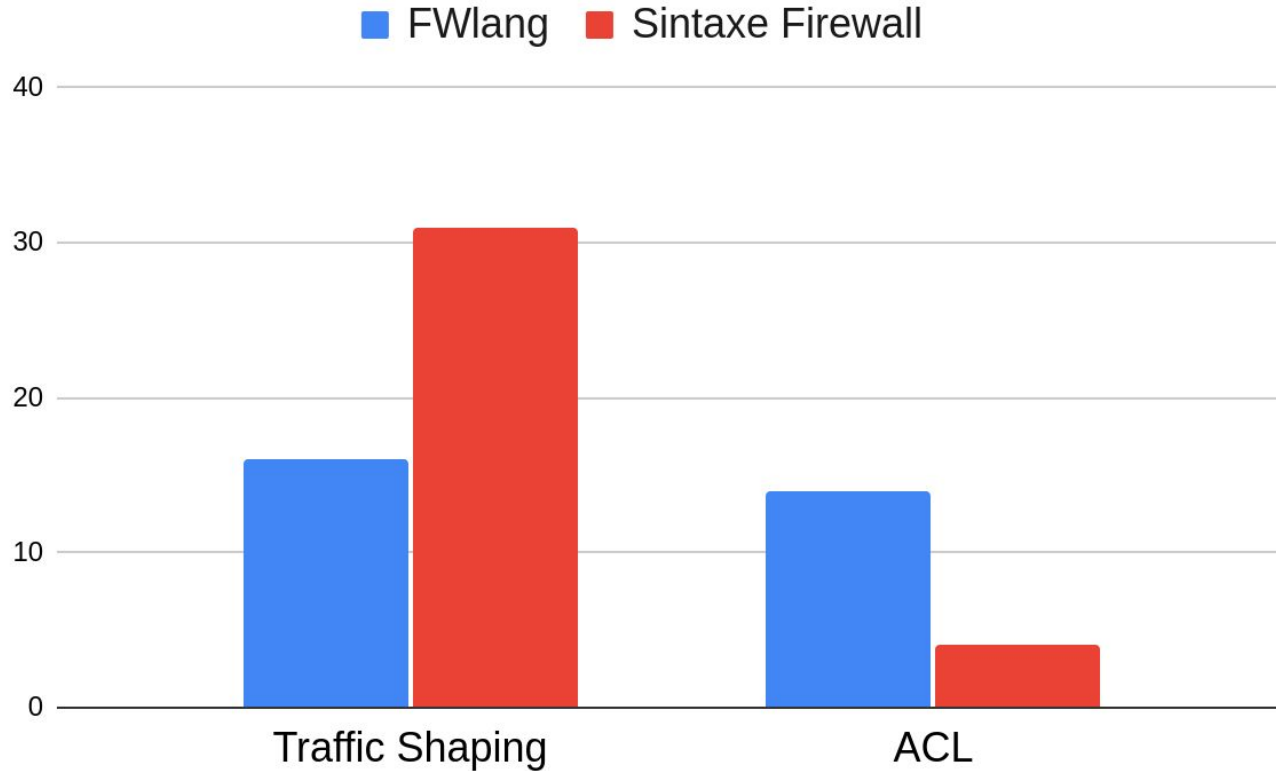
Considerações Finais - Publicações

- Maurício Fiorenza, et. al. **“Firewalls em Redes Definidas por Software: Estado da Arte”**. ERRRC 2019.
- Maurício Fiorenza, et. al. **“Gerenciamento de Firewalls em Redes Híbridas”**. SBSeg 2020. (prêmio de melhor artigo curto)
- Maurício Fiorenza, et. al. **“Representação e Aplicação de Políticas de Segurança em Redes Híbridas”**. SBRC, 2021. (Previsto para submissão em breve)

Obrigado!

mauriciofiorenza@unipampa.edu.br

FWlang X Sintaxe firewall Cisco



Levantamento políticas para FWlang

- ACLs
 - Regras reais
 - Documentação dos fabricantes
 - Literatura científica
- Traffic Shaping e Filtros de URL
 - Documentação do fabricante
- NATs
 - Regras reais
 - Documentação dos fabricantes

FWunify e FWlang X boas práticas de firewalls

- Documentação das regras de firewall
 - Marcador 'description' da FWlang
- Automação da aplicação de políticas
 - Um dos focos do FWunify
- Organização das regras para maximizar o desempenho
 - Soluções deste tipo podem ser adicionadas ao FWunify

Tipos de firewalls

1. Filtros de pacotes
2. Firewalls stateful
3. Firewalls de aplicação
4. Firewall de nova geração

Avaliação da Tradução - ACL

define intent acl:

name	text('rule-acl-1')
from	range('10.0.0.0/24')
to	range('200.19.0.0/24')
order	before('all')
block	traffic('icmp')
add	middlebox('cisco-1')

Avaliação da Tradução - ACL

Sintaxe do comando esperado:

*access-list “**interface**”_access_in line “**posição**” extended
“**permitir/bloquear**” “**protocolo**” “**IP/rede de origem**” “**IP/rede de destino**”*

Avaliação da Tradução - ACL

Sintaxe do comando esperado:

*access-list “**interface**”_access_in line “**posição**” extended “**permitir/bloquear**” “**protocolo**” “**IP/rede de origem**” “**IP/rede de destino**”*

Comando gerado:

*access-list **inside**_access_in line **1** extended **deny icmp 10.0.0.0 255.255.255.0 200.19.0.0 255.255.255.0***

Avaliação da Tradução - Traffic Shaping

```
define intent traffic_shaping:  
name      text('rule-ts-1')  
from      range('10.0.0.0/24')  
to        endpoint('200.19.0.100')  
order     before('all')  
for       traffic('ftp')  
with      throughput('30Mbps')  
add       middlebox('cisco-1')
```

Avaliação da Tradução - Traffic Shaping

Sintaxe do comando esperado:

access-list global_mpc line "posição" extended permit "protocolo" "IP/rede de origem" "IP/Rede de destino" eq "porta"

access-list global_mpc line "posição" extended permit "protocolo" "IP/rede de destino" "IP/Rede de origem" eq "porta"

class-map "nome class-map"

match access-list global_mpc

policy-map global-policy

class "nome class-map"

police input "largura de banda" "burst size" conform-action transmit exceed-action drop

police output "largura de banda" "burst size" conform-action transmit exceed-action drop

Avaliação da Tradução - Traffic Shaping

Comando gerado:

```
access-list global_mpc line 1 extended permit tcp 10.0.0.0 255.255.255.0 host 200.19.0.100 eq 21
access-list global_mpc line 2 extended permit tcp host 200.19.0.100 10.0.0.0 255.255.255.0 eq 21
class-map global-class-rule-ts-1
  match access-list global_mpc
policy-map global-policy
  class global-class-rule-ts-1
    police input 30000000 15000 conform-action transmit exceed-action drop
    police output 30000000 15000 conform-action transmit exceed-action drop
```


FWunify - Intenção 1

```
define intent acl:  
name    text('drop-all-all')  
from    endpoint('all')  
to       endpoint('all')  
block   traffic('all')  
order   after('all')  
add      middlebox('cisco-1'),middlebox('iptables-1'),middlebox('openflow-1')
```

FWunify - Intenção 2

define intent acl:

name **text**('permit-net-all-http')

from **range**('10.0.0.0/24')

to **endpoint**('all')

allow **traffic**('http')

order **before**('all')

add **middlebox**('cisco-1'),**middlebox**('iptables-1'),**middlebox**('openflow-1')

FWunify - Intenção 3

define intent acl:

name **text('permit-net-all-https')**

from **range('10.0.0.0/24')**

to **endpoint('all')**

allow **traffic('https')**

order **before('all')**

add **middlebox('cisco-1'),middlebox('iptables-1'),middlebox('openflow-1')**

FWunify - Intenção 6

define intent acl:

name **text**('drop-incident-h21')

from **endpoint**('all')

to **endpoint**('200.19.0.20')

block **traffic**('all')

order **before**('all')

add **middlebox**('cisco-1'),**middlebox**('iptables-1'),**middlebox**('openflow-1')