RNP

Educação, Pesquisa
e Inovação em Rede

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

cais Inteligência em Cibersegurança

TLP:CLEAR

┌──(landim㉿cais-rnp)-[~]
└─$ whoami
André Ricardo Landim

- ***Antes de tudo, sou filho, marido e pai*** *— o que, por si só, já exige habilidades avançadas de negociação, resposta a incidentes e gestão de crises em tempo real;*

- *Também* ***trabalho com Tecnologia da Informação há mais de 25 anos e os últimos 21 anos foram dedicados à Segurança*** *da Informação;*

- *Já* ***atuei em áreas como segurança de infraestrutura, Red e Blue Team, DFIR*** *e análise de malwares;*

- ***Atualmente, atuo como Especialista em Segurança no CAIS, equipe de Cibersegurança e Inteligência da RNP*** *(Rede Nacional de Ensino e Pesquisa) e que por acaso também agrega o CSIRT e o SOC;*

- *No fim das contas,* ***sigo tentando equilibrar tecnologia, segurança, família e café****, nem sempre nessa ordem;*

- *LinkedIn: https://www.linkedin.com/in/andrelandim/*
- *GitHub: https://github.com/andrelandim/*

TLP:CLEAR

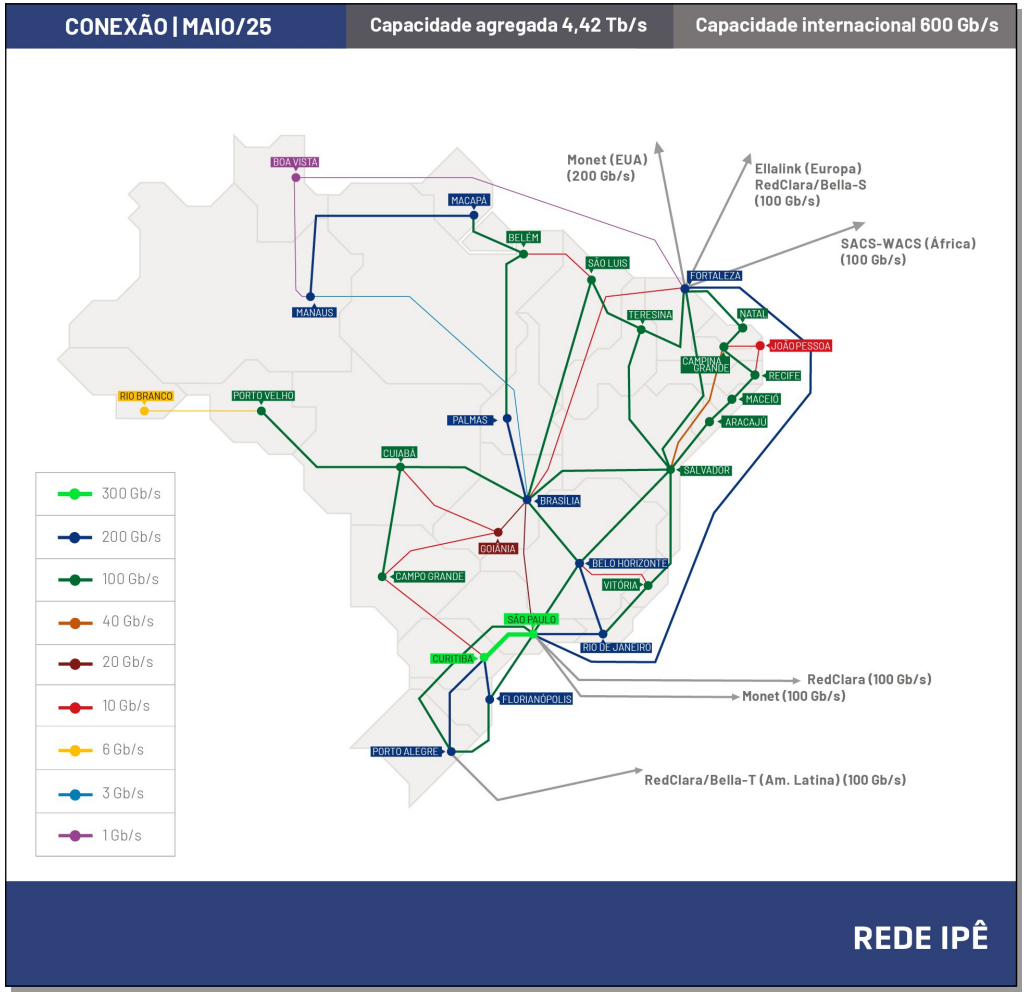
Sobre a RNP...

Conectamos universidades e instituições de ensino, pesquisa e inovação com internet e serviços seguros em uma comunidade, o Sistema RNP.

Nesse ambiente, trabalhamos em conjunto com alunos, professores e pesquisadores para desenvolver novas tecnologias com foco em inovação contínua.

É por meio dessa rede acadêmica que a ciência brasileira compartilha conhecimento e colabora entre si e com parceiros em todo o mundo.

Somos uma organização social vinculada ao governo federal e apoiamos políticas públicas de ensino e pesquisa.



TLP:CLEAR

Sobre a CAIS...

Há 26 anos, o CAIS-RNP garante a segurança em centenas de instituições de educação e pesquisa brasileiras.

A área de inteligência em cibersegurança da RNP desenvolve ações preventivas, educativas e corretivas em incidentes e ameaças na rede acadêmica, que enfrenta diariamente os desafios ligados ao aumento da complexidade de ameaças cibernéticas.

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Hardening não é uma lista infinita de configurações e parâmetros.

É a aplicação controlada de medidas que:

1. diminuem o que pode ser atacado;
2. reduzem o que um serviço pode fazer em nome de terceiros;
3. limitam quem pode administrar a infraestrutura;
4. preservam evidências úteis para detectar, responder e corrigir.

O processo de *hardening* pode ser traduzido (resumido) como:

Expor apenas o necessário, aceitar somente o esperado, registrar o relevante e validar o resultado.

TLP:CLEAR

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

O efeito prático do *hardening:*

- Menos superfície
- Menos abuso
- Mais evidência

Não é necessário percorrer dezenas de *benchmarks*. Poucos controles podem alterar o comportamento real de servidores e equipamentos de rede:

- o pacote é aceito ou descartado;
- o serviço responde ou não;
- o evento pode ou não ser reconstruído.

Situação	Sem hardening	Com controle efetivo
Serviço desnecessário	Porta e código expostos	Serviço removido ou restrito
Pacote com origem forjada	Pode atravessar a rede	Bloqueado no ponto adequado
DNS/SNMP aberto	Pode responder para qualquer origem	Somente clientes autorizados
Incidente	Horários e fontes desconexos	Linha do tempo correlacionável

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

CIS Controls & CIS Benchmarks

Referências importantes para priorizar e implementar hardening.

CIS *Controls* (priorizar o que proteger)

- Conjunto de controles prioritários de segurança.
- Ajuda a definir o que deve ser tratado primeiro.
- Orienta ações como:
  - inventário;
  - gestão de vulnerabilidades;
  - controle de privilégios;
  - *logs*;
  - proteção de serviços.
- É útil para transformar segurança em um plano de ação.

CIS *Benchmarks* (configurar efetiva proteção)

- Guias prescritivos de configuração segura.
- Mostram como endurecer sistemas, aplicações, nuvem e equipamentos.
- Trazem recomendações práticas para:
  - Linux;
  - Windows;
  - redes;
  - navegadores;
  - bancos de dados;
  - outros componentes.
- São úteis para padronizar e validar o *hardening* técnico.

Nem todo ambiente precisa aplicar tudo ao mesmo tempo. O valor do CIS está em oferecer uma referência estruturada para evoluir o hardening com critério, consistência e evidência.

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Primeiro passo: saber o que está exposto!

Perguntas operacionais

- Quais endereços, interfaces e serviços estão realmente ativos?
- Quais portas são públicas por necessidade documentada?
- Quais serviços pertencem ao plano de dados e quais ao plano de gerência?
- A exposição vista externamente coincide com o inventário?

Impacto prático

- Serviço desativado não precisa ser defendido, monitorado nem corrigido em emergência.
- Interface desligada não oferece um caminho acidental.
- Pacote removido reduz dependências, vulnerabilidades e esforço de atualização.

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Defendendo as jóias da coroa...

Proteja o plano de gerência!!!

Controles prioritários

- Rede de gerência dedicada, quando possível
  - VLANs especificas para gerencia e monitoramento (preferencialmente);
  - ACL de origem para SSH, HTTPS, SNMP e APIs, por exemplo;
  - Evitar TELNET, FTP, HTTP (protocolos "sem proteção").
- Uma conta por operador:
  - Privilégio mínimo;
  - MFA quando possível;
  - Gestão centralizada;
  - Conta de emergência controlada, não compartilhada como rotina.
- *Logs* e *backups* enviados para sistemas externos ao equipamento.

Nota importante: trocar a porta padrão reduz ruído de varreduras e bots, mas não substitui ACL, autenticação forte e segmentação.

TLP:CLEAR

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Hardening do SO: remova o que não precisa, reduzindo a superfície de ataque antes de adicionar controles mais complexos.

Prioridades simples e de alto impacto

- Desinstalar ou desabilitar serviços e pacotes sem função operacional clara.
- Verificar quais processos realmente estão escutando na rede e em quais interfaces.
- Evitar serviços administrativos ou bancos de dados expostos em "0.0.0.0/::" quando não houver necessidade.
- Manter sistema operacional, *kernel* e aplicações suportadas com atualizações de segurança em dia.
- Remover componentes legados que permanecem ativos apenas por histórico ou conveniência.

Impacto prático

- menos serviços + menos pacotes + menos *sockets* = menor superfície explorável

O serviço mais fácil de proteger é aquele que não precisava estar executando.

TLP:CLEAR

Hardening do SO (Linux): acesso e privilégio mínimo

Prioridades simples

- Evitar *login* administrativo direto como *root* (elevar privilégio somente quando necessário);
- Preferir chaves SSH e restringir autenticação por senha quando a operação permitir;
- Restringir SSH e demais interfaces administrativas às redes/origens de gerência;
- Aplicar "sudo" com o menor conjunto de comandos necessário para cada função;
- Proteger chaves privadas, credenciais, arquivos de configuração e segredos com permissões adequadas.

Resultado esperado

- comprometer uma credencial não deve significar obter automaticamente controle irrestrito do *host*;
- uma conta administrativa não precisa estar acessível de qualquer origem;
- ações privilegiadas devem ser atribuíveis a um operador.

Hardening não é apenas impedir a entrada, é limitar o que alguém consegue fazer depois de entrar.

Hardening do SO (Linux): firewall local e evidência

Dois controles simples que produzem resultado imediato

Controle	Resultado prático
Firewall local	Restrição de acesso
Logs/auditoria	Registro de eventos

Princípios simples

- Adotar uma política de entrada restritiva: publicar somente o serviço necessário.
- Não depender exclusivamente do *firewall* de perímetro para proteger cada servidor.
- Registrar tentativas de autenticação, uso de privilégio, bloqueios e erros importantes.
- Preservar ou centralizar *logs* relevantes fora do próprio *host* quando possível.
- Se a distribuição já utiliza *SELinux/AppArmor*, não desabilitá-lo por conveniência; tratar exceções de forma controlada.

O firewall local reduz quem consegue chegar ao serviço e logs mostram quem tentou, o que foi bloqueado e o que precisa ser investigado.

Hardening do SO (Windows): reduza superfície e exposição

Prioridades simples e de alto impacto

Controle	Resultado prático
Remover funções, componentes e softwares desnecessários	Menos código e serviços disponíveis para abuso
Manter *Windows* e aplicações atualizados	Reduz vulnerabilidades conhecidas
Desabilitar serviços sem uso	Menos processos, portas e dependências expostas
Manter o *Windows Defender Firewall* ativo	Restringe portas, protocolos e origens diretamente no *host*
Manter proteção antimalware/EDR ativa	Acrescenta prevenção, detecção e evidência no próprio *endpoint*
Restringir RDP, SMB, WinRM e outras interfaces administrativas	Reduz exposição do plano de gerência e movimentação lateral

Hardening do SO (Windows): privilégio mínimo e ferramentas administrativas

Prioridades simples

- Separar a conta de uso cotidiano da conta administrativa e evitar administrador local como padrão.
- Restringir RDP/WinRM e administração remota às redes ou hosts de gerência.
- Tratar *PowerShell* como ferramenta administrativa: manter quando necessário, mas restringir privilégios, controlar a administração remota e registrar sua utilização. Remover ou não instalar "wmic.exe".
- Habilitar *Script Block Logging* e, conforme o ambiente, *Module Logging*, considerando a proteção dos próprios logs.
- Usar JEA (*Just Enough Administration*) quando fizer sentido para delegar somente "cmdlets", funções e comandos necessários à função administrativa.

Regra prática para ferramentas administrativas

- Não é utilizada? Remover
- Uso muito específico? Restringir
- É necessária? Controlar & registrar

Menor privilégio + menos ferramentas desnecessárias + administração restrita + boa auditoria = menor capacidade de movimentação e persistência após um comprometimento.

IP spoofing: falsificar a origem do pacote

IP spoofing é a alteração do campo endereço IP de origem no cabeçalho do pacote. O pacote é transmitido com um endereço diferente da origem real e, se não houver validação de origem, pode atravessar roteadores e firewalls até o destino.

O que ocorre no encaminhamento:

1. O atacante cria o pacote e substitui o IP de origem real por outro endereço;
2. O roteador normalmente escolhe o próximo salto pela rota do destino;
3. Sem ACL, validação de origem ou RPF/uRPF, a incompatibilidade da origem pode não impedir o encaminhamento;
4. O destino recebe o pacote com a origem "forjada" no cabeçalho e não enxerga diretamente a real origem.

O que IP spoofing não é:

- Não significa invasão, roubo de sessão ou controle total da origem pelo atacante;
- Não garante que a resposta irá chegar ao atacante (por isso é extremamente usado em ataques (D)DoS);
- ARP e DNS *Spoofing* são técnicas diferentes

Em resumo, spoofing é a capacidade de falsificar a origem de um pacote. O impacto depende de como o pacote será utilizado e de quais controles validam essa origem.

IP spoofing: Ataques (D)DoS reflexão/amplificação

No ataque de reflexão, o atacante envia requisições a serviços de terceiros usando como origem falsificada o endereço da vítima. Os serviços respondem normalmente ao endereço informado e, sem saber, atuam como refletores.

Conceito	O que acontece
Spoofing	O endereço da vítima é colocado no campo IP de origem da requisição.
Reflexão	A resposta é enviada pelo serviço refletor para a vítima, e não para o atacante real.
Amplificação	A resposta é maior que a requisição; o volume entregue à vítima é multiplicado.
(D)DoS	O volume agregado de muitos pacotes, origens ou refletores compromete recursos ou disponibilidade.

Nem todo *spoofing* causa (D)DoS, nem toda reflexão produz amplificação. Existe inundação com origem falsificada sem refletor ((D)DoS sem *spoofing*).

Na reflexão amplificada os elementos se combinam:

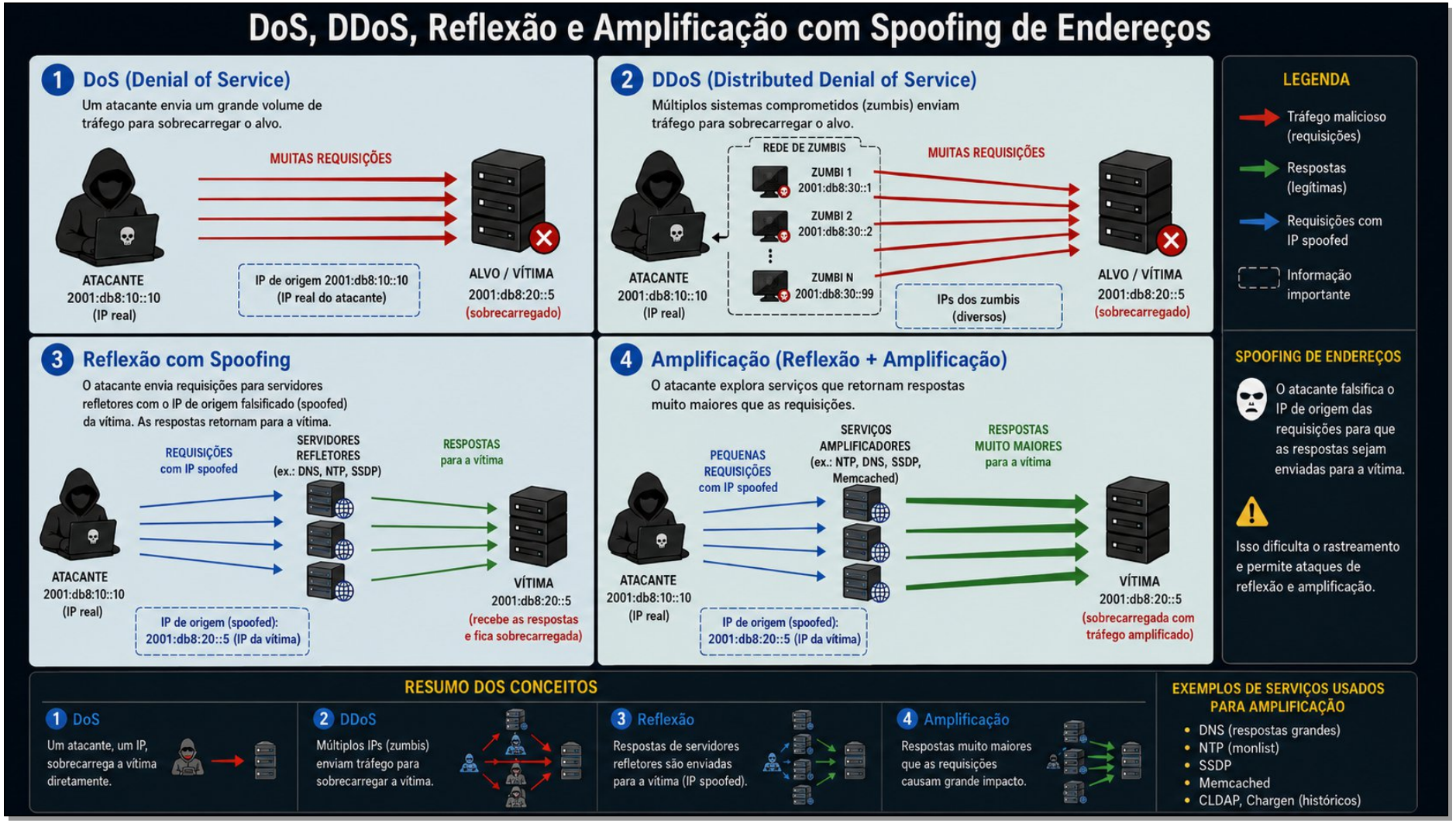
- o atacante consegue emitir pacotes com origem forjada;
- um serviço UDP responde a origens não autorizadas;
- a resposta é maior que a requisição;
- muitos pedidos ou refletores concentram as respostas na vítima.

Portanto, a mitigação é complementar: validação de origem + restrição do serviço + limitação de resposta/volume.

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

IP spoofing: Ataques (D)DoS reflexão/amplificação



TLP:CLEAR

BCP38: valide a origem o mais perto possível de onde ela nasce

A BCP38 (RFC 2827) trata da validação do endereço de origem: tráfego que entra em uma rede de provedor a partir de um cliente deve possuir endereços que possam se originar, sem truques, a partir desse cliente. BCP38 é o princípio/política. uRPF é uma das técnicas que podem implementar essa política.

Aplicação prática

- No enlace cliente > provedor, permitir como origem somente os prefixos associados ao cliente.
- O mais perto possível da origem, impedir que um *host* ou segmento envie pacotes usando endereços que não lhe pertencem.
- Na borda externa, rejeitar pacotes recebidos da Internet que tenham como origem endereços da própria organização ou outros prefixos impossíveis naquele enlace.

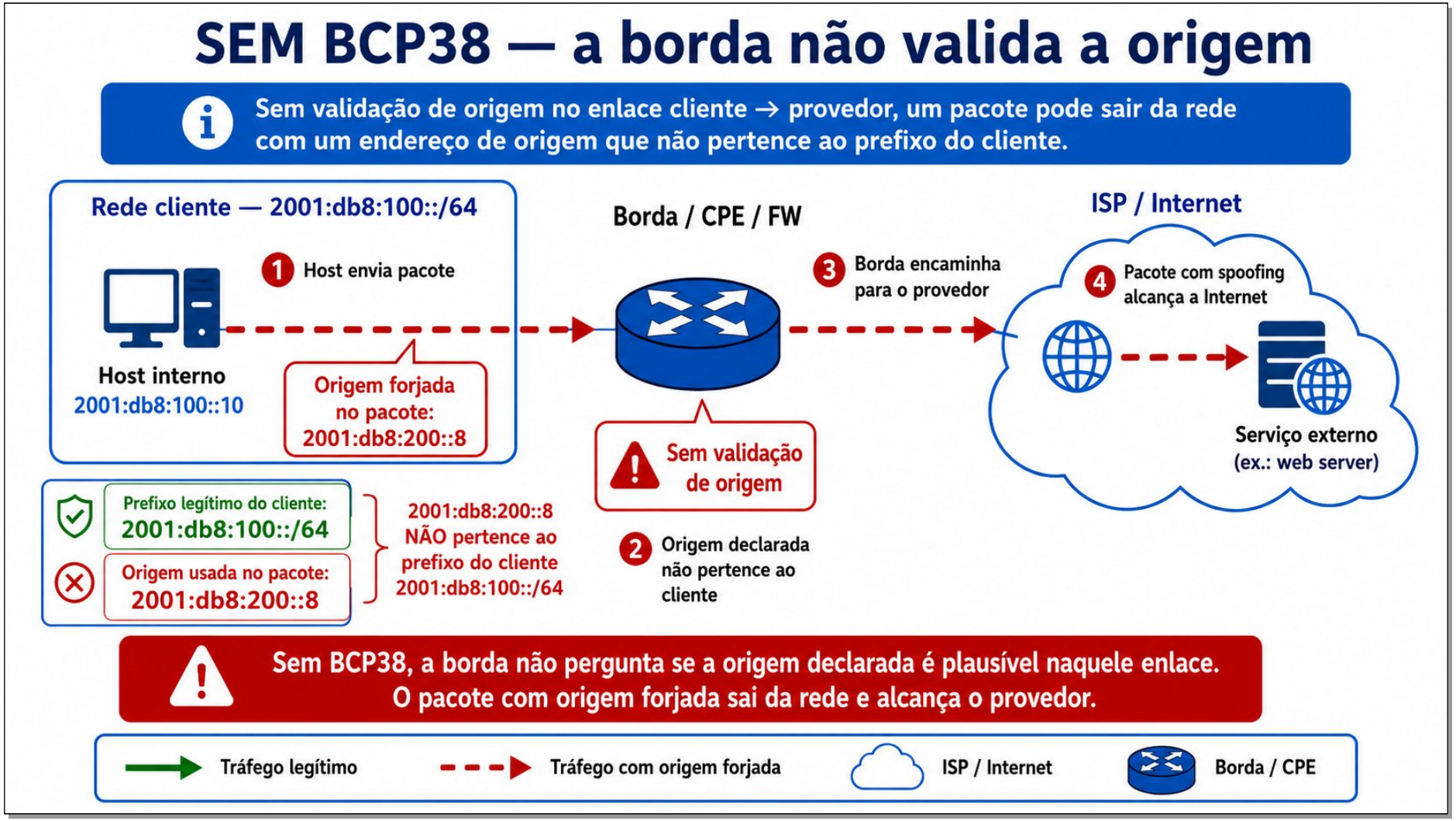
Impacto prático

- Dificulta que a própria rede seja usada como origem de ataques com endereço falsificado.
- Melhora a rastreabilidade até a rede/segmento de onde o tráfego realmente saiu.
- Evita aceitar determinadas origens impossíveis no ponto de entrada.

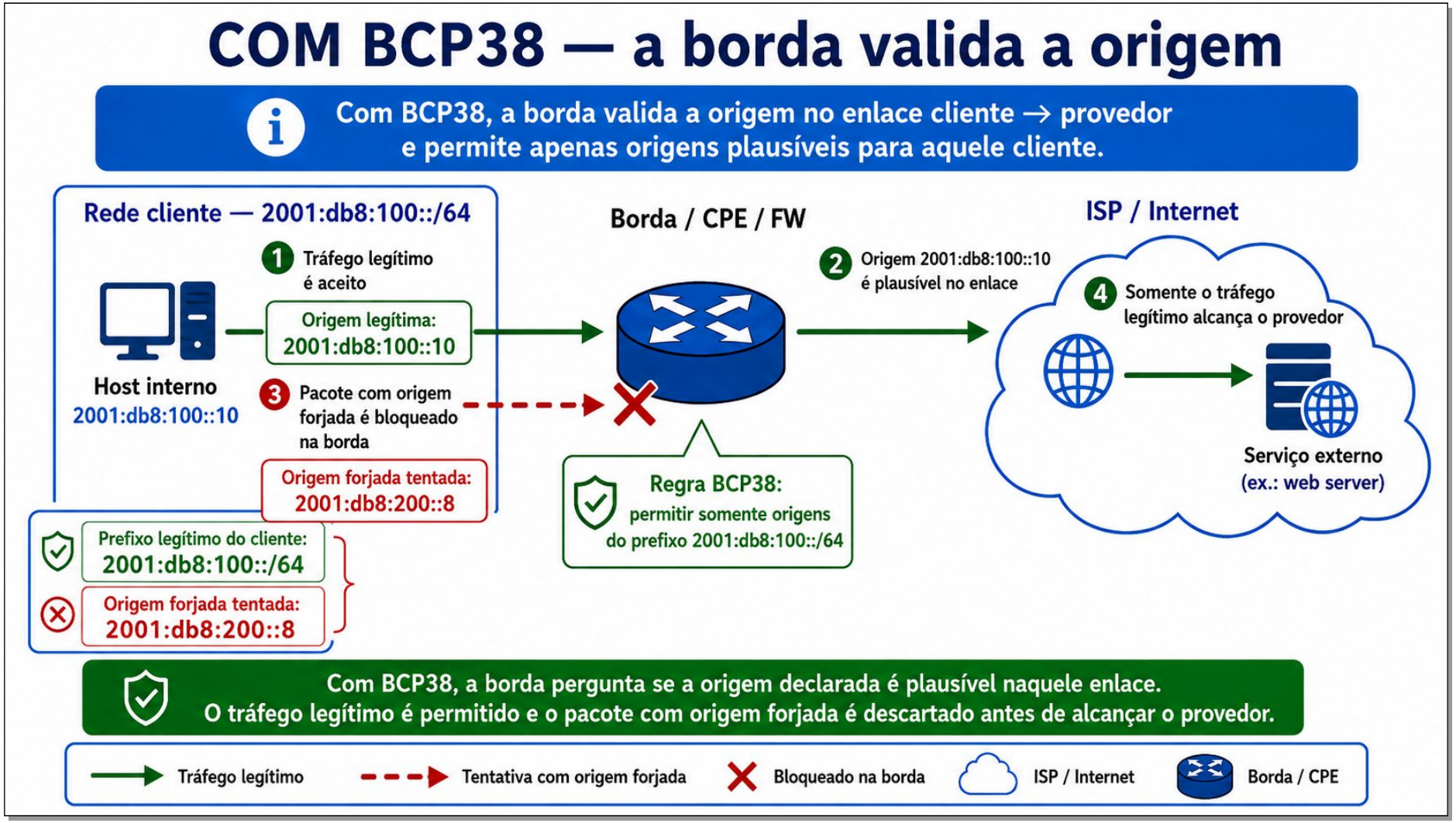
BCP 38 não pergunta se existe uma rota para o destino.

Ela pergunta se a origem esta de acordo com aquele ponto da rede.

BCP38: valide a origem o mais perto possível de onde ela nasce



BCP38: valide a origem o mais perto possível de onde ela nasce



BCP84: como validar a origem sem quebrar multihoming e assimetria

A BCP84 (RFC 3704) complementa e atualiza operacionalmente a BCP38, implementando filtragem de origem e apresenta onde o "Strict uRPF" pode ocasionar falsos positivos devido ao multihomed ou assimetria.

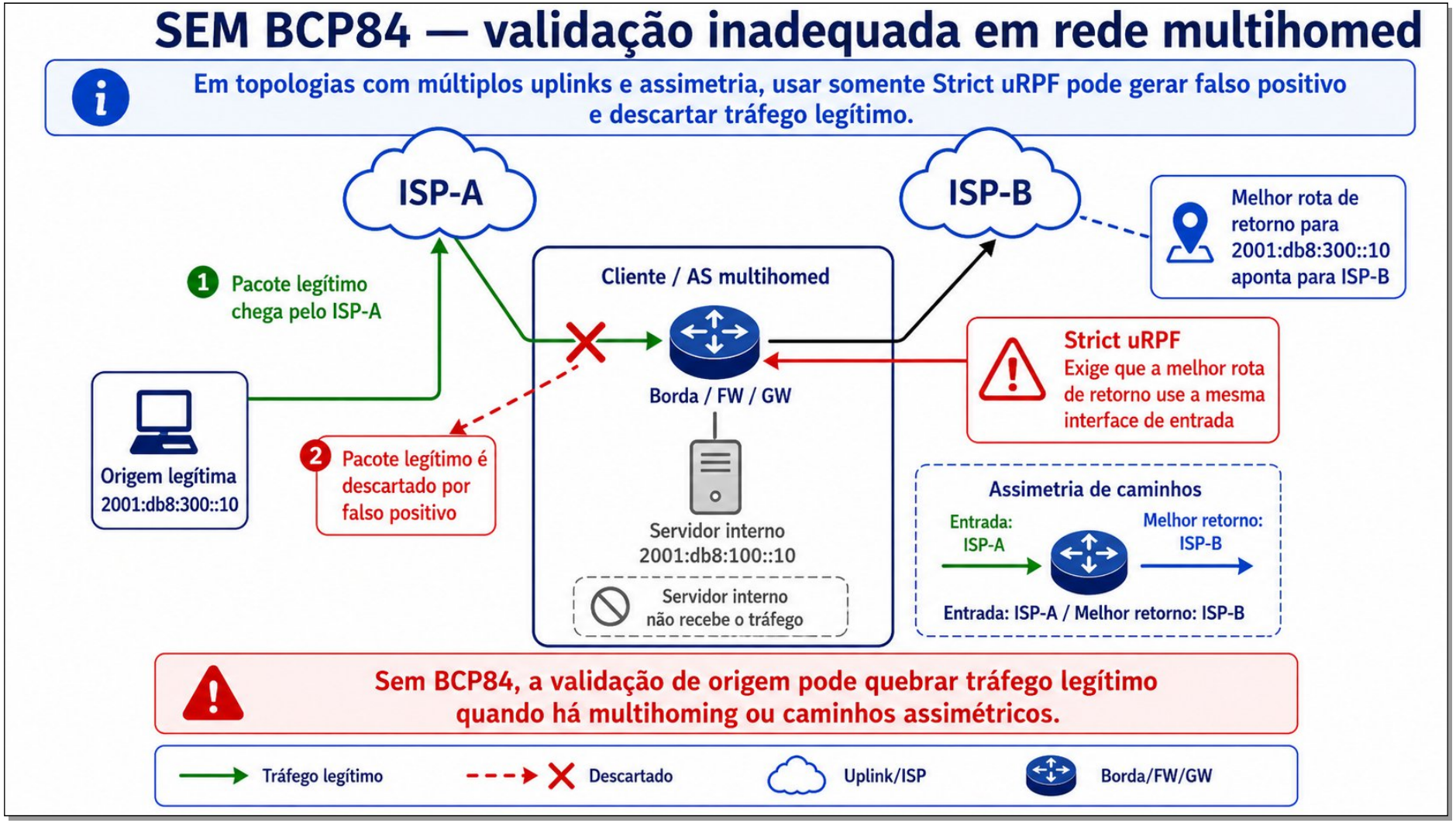
A diferença que vale fixar

- BCP38: "não aceite uma origem que não deveria vir daqui".
- BCP84: "como implementar essa validação de forma compatível com a topologia e o roteamento reais".

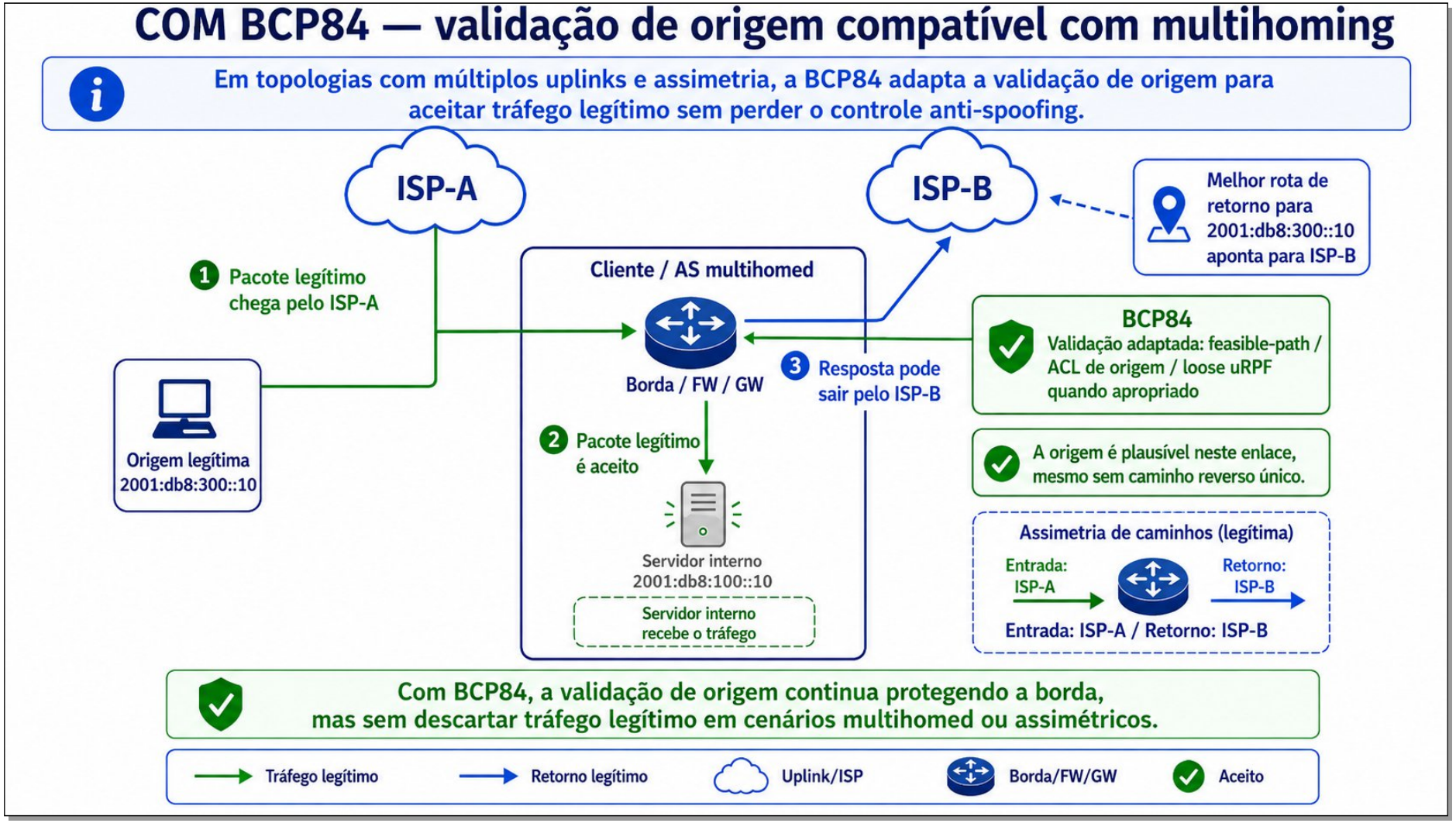
Principais mecanismos

Mecanismos	Regra simples	Onde funciona
ACL origem	A origem precisa pertencer à lista de prefixos permitidos naquele enlace	Cliente/segmento com prefixos bem conhecidos
Strict uRPF	A melhor rota de retorno para a origem deve apontar pela mesma interface por onde o pacote entrou	Topologias simples e simétricas
Loose uRPF	Basta existir rota para a origem por alguma interface	Ambientes assimétricos; valida menos a interface de entrada
Enhanced Feasible-Path uRPF (RFC8704)	Aceita a origem quando ela é alcançável por um dos caminhos considerados válidos/viáveis pelo roteamento	Operação de provedores/AS com multihoming/BGP complexo

BCP84: como validar a origem sem quebrar multihoming e assimetria



BCP84: como validar a origem sem quebrar multihoming e assimetria



Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

rp_filter no Linux (IPv4)

Valor "0" - Sem validação por *reverse path*

Valor "1" - *"Strict"* indica que a melhor rota de retorno precisa ser compatível com a interface de entrada

Valor "2" - *"Loose"*, india que a origem precisa ser alcançada por alguma interface

Os parâmetrnos "net.ipv4.conf.*.rp_filter" exemplificados são controles aplicáveis em IPv4.

A validação de origem em IPv6 deve ser implementada com os mecanismos disponíveis no firewall/roteador.

Anti-Spoofing (BCP38) – Implementação no Linux (exemplo)

```
# /etc/sysctl.d/60-network-hardening.conf
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.log_martians = 1
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0
net.ipv4.tcp_syncookies = 1
```

Anti-Spoofing (BCP38) – Implementação no Linux IPv6 com nftables (exemplo)

```
CENÁRIO DE EXEMPLO
eth0 = WAN / Internet
eth1 = LAN
Prefixo da LAN = 2001:db8:100::/64
```

```
table inet hardening {
    chain forward {
        type filter hook forward priority filter;
        policy drop;
        # 1. BCP38 – endereço da LAN chegando pela WAN = spoofing
        iifname "eth0" ip6 saddr 2001:db8:100::/64 counter log prefix "BCP38-WAN-SPOOF " drop
        # 2. BCP38 – LAN tentando usar origem que não pertence à LAN
        iifname "eth1" ip6 saddr != 2001:db8:100::/64 counter log prefix "BCP38-LAN-SPOOF " drop
        # 3. Descarta estados inválidos
        ct state invalid counter drop
        # 4. Permite respostas de conexões já autorizadas
        ct state established,related counter accept
        # 5. Preserva ICMPv6 necessário ao funcionamento do IPv6
        meta l4proto ipv6-icmp counter accept
        # 6. Permite que a LAN legítima inicie tráfego para a Internet
        iifname "eth1" oifname "eth0" ip6 saddr 2001:db8:100::/64 counter accept
    }
}
```

Anti-Spoofing (BCP38) – Implementação no MikroTik (exemplo IPv4)

```
CENÁRIO DE EXEMPLO
ether1=WAN
ether2=LAN
LAN=192.0.2.0/24
```

```
/ip settings set rp-filter=strict

/ip firewall filter add chain=forward in-interface=ether1 src-address=192.0.2.0/24 action=drop
log=yes log-prefix="BCP38-WAN-SPOOF "
/ip firewall filter add chain=forward in-interface=ether2 src-address=!192.0.2.0/24 action=drop
log=yes log-prefix="BCP38-LAN-SPOOF "

/ip firewall filter add chain=forward connection-state=invalid action=drop comment="DROP invalid"
/ip firewall filter add chain=forward connection-state=established,related action=accept
comment="ACCEPT established,related"

/ip firewall filter add chain=forward protocol=icmp action=accept comment="ACCEPT ICMP"

/ip firewall filter add chain=forward in-interface=ether2 out-interface=ether1 src-
address=192.0.2.0/24 action=accept comment="ACCEPT LAN to WAN"
/ip firewall filter add chain=forward action=drop log=yes log-prefix="FW-DROP " comment="DEFAULT
DROP"
```

Anti-Spoofing (BCP38) – Implementação no MikroTik (exemplo IPv6)

```
CENÁRIO DE EXEMPLO
ether1=WAN
ether2=LAN
LAN=2001:db8:100::/64
```

```
/ipv6 firewall filter add chain=forward in-interface=ether1 src-address=2001:db8:100::/64
action=drop log=yes log-prefix="BCP38-V6-WAN "
/ipv6 firewall filter add chain=forward in-interface=ether2 src-address=!2001:db8:100::/64
action=drop log=yes log-prefix="BCP38-V6-LAN "

/ipv6 firewall filter add chain=forward connection-state=invalid action=drop comment="DROP
invalid"
/ipv6 firewall filter add chain=forward connection-state=established,related action=accept
comment="ACCEPT established,related"

/ipv6 firewall filter add chain=forward protocol=icmpv6 action=accept comment="ACCEPT ICMPv6"

/ipv6 firewall filter add chain=forward in-interface=ether2 out-interface=ether1 src-
address=2001:db8:100::/64 action=accept comment="ACCEPT LAN to WAN"
/ipv6 firewall filter add chain=forward action=drop log=yes log-prefix="IPV6-DROP "
comment="DEFAULT DROP"
```

NTP: registrando corretamente evidências do ambiente

Sem sincronismo de tempo

- A linha do tempo de *firewall*, servidor, DNS e roteador não fecha.
- *Tokens*, certificados e autenticações sensíveis ao tempo podem falhar.
- Eventos e registros de sistemas diferentes parecem ocorrer fora de ordem.
- A resposta a incidentes perde confiança e velocidade.

Boas práticas

- Múltiplas referências confiáveis; preferir NTS quando disponível.
- *Hosts* clientes não devem responder NTP para a Internet (risco de reflexão/amplificação).
- Firewall deve permitir somente o papel necessário.
- Registrar fuso explicitamente; preferir UTC nos sistemas e converter na apresentação.
- Monitorar *offset*, *reachability* e mudanças abruptas.

Servidor NTP – Implementação Chrony (exemplo IPv6 & IPv4)

```
Servidor NTP interno
IPv6: 2001:db8:100::10 / IPv4: 192.0.2.10
Redes autorizadas:
IPv6: 2001:db8:100::/64 / IPv4: 192.0.2.0/24
```

```
# SERVIDOR NTP INTERNO - CHRONY
# ------------------------------------------------------------
# Fontes externas confiáveis / O servidor interno é CLIENTE do NTP.br e utiliza NTS
# ------------------------------------------------------------
server a.st1.ntp.br iburst nts
...
server gps.ce.ntp.br iburst nts
# ------------------------------------------------------------
# Redes autorizadas a utilizar ESTE servidor NTP
# ------------------------------------------------------------
allow 192.0.2.0/24
allow 2001:db8:100::/64
# ------------------------------------------------------------
# Endereços locais (IPv4 & IPv6) em que o Chrony atenderá NTP
# ------------------------------------------------------------
bindaddress 192.0.2.10
bindaddress 2001:db8:100::10
...
```

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

DNS de um provedor deve ser "público" para seus clientes, nao para a Internet.

Tipos de servidores DNS (básico)

Função	Quem deve consultar?	Recursão permitida?
Autoritativo público	Internet	Não
Recursivo interno	Redes autorizadas	Sim, somente para clientes
Recursivo/*Cache* interno	Redes autorizadas	Somente clientes autorizados

- Nunca oferecer recursão aberta.
- Restringir UDP e TCP/53 conforme o papel.
- Separar autoritativo e recursor sempre que viável.
- Aplicar *Response Rate Limiting*.
- Evitar divulgação desnecessária de versão e dados internos.
- Manter *software* e zonas atualizados; controlar transferências de zona.
- Monitorar aumento de *ANY*, respostas grandes, *NXDOMAIN* e picos por origem.

TLP:CLEAR

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Servidor DNS – Implementação BIND (exemplo IPv6 & IPv4)

```
/etc/bind/named.conf.options

acl "clientes-dns" {
    127.0.0.1;
    ::1;
    // Prefixos IPv4 dos clientes
    198.51.100.0/24;
    203.0.113.0/24;
    // Prefixos IPv6 dos clientes
    2001:db8:100::/48;
    2001:db8:200::/48;
};
```

```
options {
    directory "/var/cache/bind";
    // Resolver recursivo
    recursion yes;
    // SOMENTE clientes autorizados podem utilizar recursão
    allow-recursion {
        clientes-dns;
    };
    // SOMENTE clientes autorizados acessam o cache
    allow-query-cache {
        clientes-dns;
    };
    // Consultas DNS somente dos clientes autorizados
    allow-query {
        clientes-dns;
    };
    // Endereços onde o serviço DNS será oferecido
    listen-on port 53 {
        127.0.0.1;
        192.0.2.53;
    };
    listen-on-v6 port 53 {
        ::1;
        2001:db8:53::53;
    };
    // Validação DNSSEC
    dnssec-validation auto;
    // Privacidade das consultas recursivas
    qname-minimization relaxed;
    // Evitar dados adicionais desnecessários nas respostas
    minimal-responses yes;
};
```

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

SNMP: monitorar sem expor nem amplificar. O agente deve responder somente ao NMS, somente aos OIDs necessários e somente com o nível de acesso necessário.

Riscos práticos

- Comunidade padrão ou fraca permite leitura não autorizada.
- SNMPv2c trafega a comunidade sem proteção criptográfica.
- Respostas UDP podem ser maiores que as requisições e participar de reflexão/amplificação.
- Escrita SNMP amplia o impacto de uma credencial comprometida.

Prioridades

- Desabilitar SNMP se não houver monitoramento.
- Preferir SNMPv3 *authPriv*.
- Permitir UDP/161 somente dos coletores/servidores de gerência.
- Remover "*public*", impedir acesso de "0.0.0.0/0" e desabilitar escrita (*private*).
- Enviar traps apenas para destinos previstos; proteger UDP/162 conforme a arquitetura.
- Monitorar volume e padrão de consultas.

Consultas não autorizadas podem vazar informações relevantes sobre o ambiente além produzir uma resposta útil para reflexão/amplificação

Servidor SNMP – Implementação SNMPv3 (exemplo IPv6 & IPv4)

```
Ativo monitorado:
IPv4 = 192.0.2.10
IPv6 = 2001:db8:100::10
NMS autorizado:
IPv4 = 192.0.2.50
IPv6 = 2001:db8:50::50
```

```
# Criar usuário:
net-snmp-create-v3-user \
-ro -A 'SenhaAuthForte' \
-a SHA -X 'SenhaPrivForte' \
-x AES monitor

# Ajustar snmpd.conf
rouser monitor priv -V monitoring
```

```
# Escuta SNMP em IPv4 e IPv6
agentAddress udp:161,udp6:161

# View somente de leitura para monitoração
view monitoring included .1.3.6.1.2.1

# Usuário SNMPv3: somente leitura (auth e crypt)
rouser monitor priv -V monitoring

# Identificação do ativo
sysLocation "POP-CPS - Datacenter"
sysContact "NOC <noc@example.net>"
```

Hardening, Superfície de Ataque e Priorização de Vulnerabilidades

Hardening de sistemas e equipamentos de rede

Checklist geral

1. Inventariar *sockets*, serviços, pacotes, componentes e interfaces ativos.
2. Remover ou desabilitar o que não possui função operacional clara.
3. Manter sistema operacional, *kernel* e aplicações com atualizações de segurança em dia.
4. Restringir administração: contas individuais, privilégio mínimo, SSH/RDP/WinRM protegidos e plano de gerência filtrado.
5. Aplicar *firewall* local permitindo somente portas, protocolos e origens necessárias; manter proteção de *endpoint* ativa quando aplicável.
6. Limitar ferramentas administrativas: *PowerShell* controlado e auditado, JEA quando adequado e "wmic.exe" removido quando não houver dependência; preservar *logs* relevantes e manter SELinux/AppArmor quando fizer parte da plataforma.
7. Aplicar validação de origem segundo BCP38 e escolher o mecanismo compatível com a topologia segundo BCP84.
8. Padronizar NTP/NTS e monitorar *offset*.
9. Fechar recursão DNS e restringir DNS por papel.
10. Migrar SNMP para v3 *authPriv*, limitar coletores e validar externamente o resultado.

O controle só está concluído quando sabemos o estado anterior, aplicamos a mudança, testamos de fora para dentro e preservamos evidência do resultado.

TLP:CLEAR

Conclusão: Hardening que produz efeito

***Hardening* não é simplesmente ativar mais controles: é reduzir a superfície, limitar privilégios, validar origens, restringir serviços e permitir somente os fluxos realmente necessários**. Seja no Linux, Windows, firewall, BCP38/84, NTP, DNS ou SNMP, o princípio é o mesmo: quanto menos comportamento desnecessário existir, menor será a superfície de ataque.

Também **não existe controle isolado suficiente**. BCP38 reduz *spoofing*, mas não corrige um DNS aberto; SNMPv3 protege o protocolo, mas não elimina a necessidade de ACL e *firewall*; NTP confiável melhora toda a infraestrutura, mas deve atender somente quem realmente precisa. ***Hardening* eficiente é defesa em profundidade**: controles simples, complementares e aplicados no ponto correto.

Por fim, **configuração segura precisa ser verificável**. Não basta acreditar que o serviço está protegido: **teste a partir de uma origem autorizada e de outra não autorizada**, observe contadores e *logs* e confirme o comportamento real. O **controle só estabiliza quando** é possível demonstrar que o **tráfego legítimo continua e o comportamento indesejado desaparece**.

REFERÊNCIAS

- **NIC.br – Melhores Práticas de Hardening**
  - https://bcp.nic.br/i+seg/acoes/hardening/
- **NTP.br – Sincronização de Linux/BSD e recomendação do Chrony**
  - https://ntp.br/guia/linux/
- **Linux Kernel – IP sysctl**
  - https://docs.kernel.org/networking/ip-sysctl.html
- **Center For Internet Security (CIS)**
  - https://www.cisecurity.org/controls
- **IETF BCP 38 / RFC 2827 – Network Ingress Filtering**
  - https://datatracker.ietf.org/doc/html/rfc2827
- **IETF BCP 84 / RFC 3704 – Ingress Filtering for Multihomed Networks**
  - https://datatracker.ietf.org/doc/html/rfc3704
- **IETF RFC 5358 – Preventing Use of Recursive Nameservers in Reflector Attacks**
  - https://datatracker.ietf.org/doc/html/rfc5358
- **MikroTik – Securing your router**
  - https://help.mikrotik.com/docs/spaces/ROS/pages/328353/Securing+your+router
- **MikroTik – Reverse Path Filtering**
  - https://help.mikrotik.com/docs/spaces/ROS/pages/103841817/IP+Settings
- **MikroTik – DNS**
  - https://help.mikrotik.com/docs/spaces/ROS/pages/37748767/DNS
- **MikroTik – SNMP**
  - https://help.mikrotik.com/docs/spaces/ROS/pages/8978519/SNMP
- **MikroTik – Traffic Flow / NetFlow / IPFIX**
  - https://manual.mikrotik.com/docs/diagnostics-monitoring-and-troubleshooting/traffic-flow/

OBRIGADO!!!

André R. **Landim**
andre.landim@rnp.br