Uma solução de Autenticação Fim a Fim para o LDP (Label Distribution Protocol)

Morvan D. Müller, Carlos B. Westphall, Carla Westphall

Laboratório de Redes e Gerência (LRG) - Universidade Federal de Santa Catarina (UFSC) - Tel: +55.48.3317559, Caixa Postal 476, CEP 88040-970, Florianópolis, SC, Brasil

{morvan,westphal,carla}@lrg.ufsc.br

Abstract. *This works propose a solution for the LDP (Label Distribution Protocol) protocol from the MPLS architecture. The objective is authenticate in a trust way, on an end to end basis, the establishment of an LSP (Label Switching Path) between the Ingress LSR (Label Switching Router) and its Egress. It is intended to supply the LDP protocol deficiency, that doesn't have one end to end authentication mechanism defined for non-adjacent LSRs. (Key-words. LDP, security, MPLS).*

Resumo. *Este trabalho propõe uma solução de autenticação para o protocolo LDP (Label Distribution Protocol) da arquitetura MPLS. O objetivo é autenticar de forma confiável, em um escopo fim a fim, o estabelecimento de um LSP (Label Switching Path) entre um LSR (Label Switching Router) de Ingresso e o seu respectivo LSR de Egresso. Pretende-se suprir a deficiência do protocolo LDP de não possuir um mecanismo de autenticação fim a fim definido para LSRs não-adjacentes.(Palavras-chave. LDP, segurança, MPLS).*

1. Introdução

O MPLS *(Multiprotocol Label Switching*), RFC 3031 [ROSEN, 2001], é uma técnica de comutação de pacotes baseada em etiquetas (*labels*). O protocolo LDP (*Label Distribution Protocol*) é responsável pela distribuição dessas etiquetas e pelo estabelecimento dos caminhos lógicos, LSPs (*Label Switched Paths)* no MPLS. Uma lacuna na segurança do LDP pode comprometer todo o ambiente MPLS, pois a distribuição das etiquetas realizada pelo LDP é o que determina quem pode participar ou não do domínio MPLS. Existe uma autenticação definida para o LDP, RFC 3036 [ANDERSSON, 2001], baseada em TCP/MD5 [HEFFERNAN, 1998], porém a mesma é restrita a LSRs adjacentes pois depende de uma conexão TCP estabelecida entre os LSRs envolvidos. No caso de LSPs entre LSRs não-adjacentes, especialmente durante o estabelecimento do primeiro LSP, não existe uma conexão TCP fim a fim entre estes LSRs.

1.1. Trabalhos Correlatos

[DE CLERCQ, 2001], descreve uma proposta de autenticação fim a fim para o protocolo LDP, sugerida em forma de *draft* (atualmente expirada) para o IETF. Através de uma análise desta proposta conclui-se que a mesma apresenta um erro arquitetural, fato reconhecido pelos seus autores, por considerar erroneamente que ao enviar uma mensagem LDP solicitando um LSP para uma determinada FEC, o LSR de origem

(ingresso) sabe qual é o LSR de destino (egresso) que vai processar a requisição. Na maioria dos casos isso não é uma verdade dentro da forma padrão de operação do protocolo LDP.

[BUDA, 2001] aborda a segurança do MPLS e levanta a problemática da autenticação fim a fim no estabelecimento de LSPs pelo LDP. [WU, 2000] descreve uma solução que depende da confiabilidade dos LSPs criados entre LSRs não-adjacentes, pelo LDP.

2. A Solução de Autenticação Fim a Fim para o LDP

A solução deste trabalho faz uso de um mecanismo de autenticação baseado em criptografia assimétrica (chave pública e privada), anexado as mensagens LDP, o que possibilita ao LSR receptor verificar e autenticar o emissor das mensagens. Provê integridade às informações através de um mecanismo de resumo de mensagens (*hash*) e adicionalmente protege contra ataques de repetição através da inserção de um *nonce* as mensagens LDP. A solução não prove confidencialidade aos dados e foi planejada para ambientes onde LSPs atravessam múltiplos domínios externos, não confiáveis entre si, que por esse motivo necessitam de uma forma para autenticar as extremidades do LSP durante o seu estabelecimento. Como requisito a solução exige que o LDP esteja operando no Modo de Controle Ordenado e quanto aos modos de distribuição do LDP, "Sob Demanda" e "Não Solicitado", ambos são compatíveis com a solução proposta, a qual pode adicionalmente ser aplicada ao protocolo CR-LDP (*Constrained-Based Routing Protocol*) [JAMOUSSI, 2002] pelo fato do mesmo ser baseado no LDP.

2.1. TLVs e Tipos Definidos ao LDP pela Solução de Autenticação

Foram definidos dois novos TLVs (*Type-Length-Value*) ao LDP para prover a autenticação: "TLV de Nonce" e "TLV de Hash", e um novo "Código de Status" com o valor "Authentication Failed" para o TLV de Status do LDP, usado nas mensagens LDP Notification para anunciar que uma mensagem Label Mapping ou Label Request falhou na autenticação. As mensagens LDP envolvidas no processo de autenticação são: LABEL REQUEST, LABEL MAPPING e LDP NOTIFICATION. Os TLVs da autenticação são inseridos (no envio de mensagens LDP) e processados (no recebimento de mensagens LDP) em mensagens LABEL MAPPING, LABEL REQUEST e LDP NOTIFICATION, somente se o LSR se encontrar na condição de EGRESSO ou INGRESSO para a(s) FEC(s) em questão na mensagem LDP. Nos LSRs INTERMEDIÁRIOS os TLVs da autenticação são apenas repassados ao próximo LSR do caminho (*next hop*), mantendo a mesma ordem da mensagem original.

1.2.1 TLV de Hash

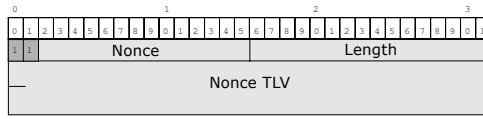

Figura 1. TLV de Hash

Este TLV transporta um resumo de mensagem (*hash)* cifrado. ***U-bit e F-bit*****:** (1 bit cada) Atribuído em "1" indica ao LDP que ignore este TLV se o mesmo não for reconhecido e

o repasse para o próximo LSR do caminho. ***Hash*:** (14 bits) Este campo define o tipo do TLV, "TLV de Hash". ***Length*:** (2 bytes) indica o tamanho total em bytes dos seguintes campos: ***LSR Identifier*:** (6 bytes) identifica o LSR que originou a mensagem LDP, composto pelo LSR-ID (Identificador do LSR) e pelo espaço de etiquetas em uso pelo LSR. ***Hash Digest*:** (20 bytes) contém um valor Hash gerado a partir de uma mensagem LDP cifrado com a chave privada do LSR remetente. Na definição do tamanho deste campo foram considerados os algoritmos de *hash* (sha-1/160 bits) e de criptografia assimétrica "Curvas Elípticas", discutido na seção 0.

2.2.1 TLV de Nonce

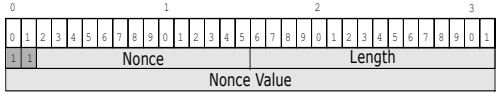

Figura 2. TLV de Nonce

Foi definido um novo TLV para transportar um valor de *nonce*. ***U-bit e F-bit*** seguem a mesma descrição do item 1.2.1. ***Nonce:*** (14 bits) Este campo define o tipo do TLV, "TLV de Nonce". ***Length*:** (2 bytes) indica o tamanho em bytes do campo "Nonce Value". ***Nonce Value*:** (8 bytes) armazena um valor nonce, de natureza incremental usado para detectar ataques de repetição.

2.2. O Modelo de Autenticação Proposto

Considerando que o LERA deseja estabelecer um LSP para uma FEC 10.1.0.0/8, prefixo IP o qual conhece via informações do seu roteamento IP. A Figura 3 ilustra o cenário onde o LERB (EGRESSO) autentica positivamente a mensagem de requisição LDP (*Label Request*) enviada pelo LERA (INGRESSO) e retorna uma mensagem LDP (*Label Mapping*) autenticada ao LERA, utilizando a solução de autenticação fim a fim que será descrita passo a passo nesta seção.

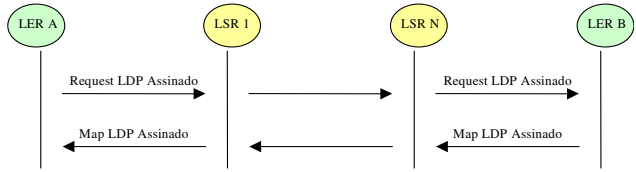

Figura 3. Diagrama da autenticação fim a fim.

Considere que o LDP está operando no modo de distribuição "Sob Demanda" e modo de controle "Ordenado" nos LSRs do ambiente e que os relógios do LERA e LERB estão sincronizados. Para SOLICITAR o LSP aplicando a solução de autenticação fim a fim, o LERA executa os seguintes passos:

a) codifica uma mensagem LDP LABEL REQUEST solicitando uma etiqueta para a FEC 10.1.0.0/8, cuja rota de destino (*next hop*) é conhecida via seu roteamento IP;

b) gera um valor nonce, codifica os campos do TLV de Nonce (Figura 2) e anexa o mesmo ao final da mensagem LDP.

c) codifica o TLV de HASH (Figura 1) baseado no conteúdo da mensagem LDP. No campo "LSR Identifier", o LERA insere o seu LSR-ID e o espaço de etiquetas (*labels*) que está usando. Codifica o campo “Hash Digest” que depende do tipo da mensagem LDP. Para mensagens LABEL REQUEST e LABEL MAPPING a entrada de dados é formada por um string de bytes conforme a Figura 4 e para para mensagens LDP NOTIFICATION uma string de bytes conforme a Figura 5:

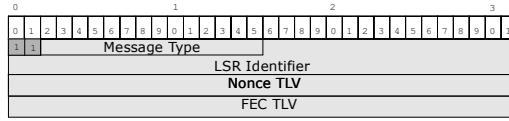

Figura 4. Entradas para mensagens LDP LABEL REQUEST e LABEL MAPPING

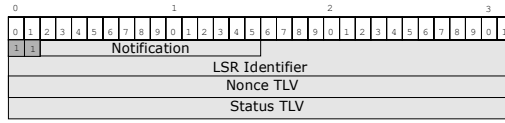

Figura 5. Entrada para mensagens LDP NOTIFICATION

Como o LERA esta enviando uma mensagem LABEL REQUEST, forma um string conforme a Figura 4 e sobre esta aplica uma função hash (esta proposta considera o algoritmo “sha-1/160 bits”). Sobre o resultado da função *hash* aplica um algoritmo de criptografia assimétrica (esta proposta considera o algoritmo de “Curvas Elípticas”) usando a sua chave privada (LERA) para a cifragem. Os resultados destas operações formam o campo “Hash Digest”. Então o LERA anexa o TLV de Hash ao final da mensagem LDP e a envia ao próximo LSR do caminho LSR1 (*next hop*), descoberto via informações do seu roteamento IP;

Os LSRs INTERMEDIÁRIOS do caminho (LSR1 e LSRN) não podem atender a requisição solicitada (pois não possuem uma etiqueta para a FEC e não estão na condição de Ingresso/Egresso para a FEC), assim repassam os campos da autenticação e ao mesmo tempo mantém um status de requisição pendente em relação a FEC a cada hop até a requisção alcançar o LER B, que ao receber a requisição LDP verifica que é o EGRESSO para a FEC então processa a autenticação.

Para PROCESSAR os Tlvs da autenticação fim a fim no recebimento da mensagem, o LERB executa os seguintes passos:
a) identifica que o LERA é o emissor pelo campo "LSR Identifier" do TLV de Hash;
b) verifica em sua configuração local se este LSR (LERA) está autorizado a estabelecer LSPs (**controle de autorização**), e em caso positivo seleciona a chave pública do LERA. Cada LSR que implementa a autenticação fim a fim possui em sua configuração local uma lista de controle de acesso a qual contém o “LSR-ID” e a chave pública correspondente dos LSRs autorizados a estabelecer LSPs com este LSR. Estas entradas são informadas manualmente via configuração local dos LSRs.
c) decifra/valida o campo "Hash Digest", do TLV de Hash recebido, usando a chave pública do remetente (LERA). Caso obter sucesso significa que o remetente é AUTÊNTICO (**autenticação da origem**);

d) nos mesmos moldes do LERA (Figura 4), o LERB gera um Hash sobre a mensagem recebida e compara com o valor do campo "Hash Digest" do TLV de Hash recebido, assim pode verificar se mensagem recebida está ÍNTEGRA (**controle de integridade**);
e) gera um nonce local e compara este valor com o nonce recebido do LERA no campo "Nonce Value" do TLV de Nonce, aplicando algum mecanismo de verificação de nonce, [ABADI, 1996] descreve vários mecanismos de nonce, esta proposta não define um mecanismo de nonce específico. Se a verificação falhar a mensagem deve ser descartada. (**controle contra ataques de repetição**).

Se a autenticação ocorrer com sucesso, o LERB gera uma mensagem LDP LABEL MAPPING de resposta, gera/atribui uma etiqueta MPLS e executa os mesmos passos que o LERA: gera um *hash* da mensagem, cifra com a sua chave privada (LERB) e envia a resposta. A resposta é encaminhada através dos LSRs intermediários, geram um par de etiquetas (entrada/saída) correspondentes a cada hop até alcançar o LERA. Este por sua vez detecta que é o LSR de INGRESSO para FEC (10.1.0.0/8) e procede a autenticação do LERB. Para isso verifica se o LERB está autorizado a estabelecer LSPs, verifica sua autenticidade e a integridade da mensagem recebida. Baseado no resultado da autenticação executa ou não o estabelecimento do LSP com o LERB

Se a autenticação falhar, uma mensagem de notificação (LDP NOTIFICATION) com o código de status "*Authentication Failed*" será enviada em resposta para reportar a falha de autenticação.

Observe que ao solicitar o LSP o LERA conhece apenas a FEC (10.1.0.0/8) e o endereço IP do LSR1 (*next hop* para esta FEC). Ele não sabe quem será o LSR de egresso no domínio MPLS para esta FEC. Observe também que as mensagens Label Request e Label Mapping que criam o LSP, são trocadas via roteamento IP e apenas após o estabelecimento do LSP que os pacotes subseqüentes serão roteados via MPLS, através das etiquetas geradas e autenticadas pelo LDP.

Os mecanismos adotados para prover a solução de autenticação foram salientados em negrito no texto acima: controle de autorização, autenticação da origem, controle de integridade e controle contra ataques de repetição.

Quanto a distribuição de chaves, cada LSR envolvido na autenticação precisa gerar e conhecer seu próprio par de chaves (pública e privada) e ambas as entidades LSR das extremidades do LSP (Ingresso e Egresso) devem conhecer a chave pública do LSR da extremidade oposta e inseri-la em sua lista de controle de autorização dessa forma autorizando que este LSR possa estabelecer LSPs. Sugere-se duas alternativas para distribuir as chaves públicas no ambiente: a) distribuição manual: informar as chaves públicas dos LSRs autorizados manualmente na configuração local dos LSRs. b) distribuição usando Certificação Digital: a solução está descrita em [MÜLLER, 2002].

2.3. Discussão sobre o Método de Autenticação Adotado

O método de autenticação adotado foi baseado em criptografia de chave pública por dois motivos: a) ao solicitar um LSP para uma FEC o LSR requisitante não sabe quem será o Ingresso (em caso de Label Mapping) ou Egresso (em no caso de Label Request), ou seja, não conhece o destinatário final.. A criptografia de chave pública resolveu essa problemática pois de posse da chave pública do LSRs autorizados, mantida na lista de controle de autorização em cada LSR, o LSR receptor pode verificar a assinatura do

LSR remetente. b) como o MPLS objetiva fast switching e alto desempenho, procurou-se evitar uma troca de mensagens adicional/inicial apenas para negociação de chaves de sessão e algoritmos para a autenticação, o que faria dobrar o tempo de estabelecimento do LSP. A solução adotada utiliza campos de controle ("TLV de Hash" e "TLV de Nonce") "de carona" nas mensagens Label Request ou Label Mapping que o LDP usa para criar o LSP, assim não exige mensagens adicionais.

2.4. Algoritmos de Função Hash e Criptografia Assimétrica Recomendados

Para função *Hash*, sugere-se o algoritmo SHA-1 (*Secure Hash Algorithm*) [STALLINGS, 1999], com *digest* de 160 bits o qual gera como saída uma string com 20 bytes de tamanho. Para as funções de criptografia de chave pública sugere-se o algoritmo "Curvas Elípticas" [STALLINGS, 1999], que é rápido, trabalha com blocos pequenos e não acrescenta *overhead* a criptografia, ou seja, o hash cifrado se mantém com 20 bytes. O objetivo é gerar o mínimo *overhead* possível ao LDP.

2.5. Implementação do Protótipo em Linux

Foi realizada a implementação de um protótipo da solução descrita neste trabalho, utilizando um projeto de código fonte aberto que implementa o LDP e MPLS na plataforma linux. O projeto utilizado foi o "*MPLS for Linux*" (http://sourceforge.net/projects/mpls-linux/) [LEU, 2000] o qual está vinculado ao grupo "source forge" (http://sourceforge.net). O código está dividido em dois módulos principais MPLS-LINUX, que implementa o MPLS no kernel do linux, e LDP-PORTABLE, que implementa as funcionalidades do LDP. Alteramos o módulo LDP-PORTABLE, inserido o código necessário para implementar as funcionalidades da autenticação fim a fim apresentada neste trabalho. A linguagem utilizada foi o "C ANSI" (compilador gcc) e manteve-se a interface original do LDP-PORTABLE. As versões das ferramentas utilizadas na implementação do protótipo foram: Linux RedHat 7.2, linux kernel 2.4.19, LDP-PORTABLE versão 0.200 (http://prdownloads.sourceforge.net/mpls-linux/ldp-portable-0.200.tar.gz?download), MPLS-LINUX versão 1.170 (http://prdownloads.sourceforge.net/mpls-linux/mpls-linux-1.170.tar.gz?download) e ZEBRA versão 0.96 (http://www.zebra.org). Maiores detalhes a respeito da implementação estão descritos em [MÜLLER, 2002].

3. Conclusão

A solução apresentada traz incrementos importantes com relação à segurança do LDP. Uma forma de autenticação fim a fim, para viabilizar a autenticação mútua entre os LSRs de Ingresso e Egresso durante o estabelecimento de um novo LSP é de fundamental importância para a segurança LDP, principalmente em ambientes MPLS multi-domínio onde os domínios não são confiáveis entre si. Um exemplo clássico de ambiente multi-domínio MPLS é o provimento de VPNs baseadas em BGP/MPLS onde vários provedores VPN fornecem o serviço VPN a um cliente baseados em acordos (SLA's) que possuem entre si [ROSEN, 1999]. A solução apresentada possui um escopo de aplicação genérico e abrangente dentro do LDP, ou seja, se aplica a todas as situações de estabelecimento de LSPs, inclusive entre LSRs adjacentes e pode adicionalmente ser aplicada ao protocolo CR-LDP. Como perspectivas futuras, sugere se avaliar os pós e contras de prover confidencialidade as informações transportadas pelo protocolo LDP.

4. Referências Bibliográficas

ABADI, M.; NEEDHAM, R. "Prudent Engineering. Practice for Cryptographic Protocols" (1996). IEEE Transactions on Software Engineering, v. 22, n. 1, p. 6-15. (Disponível http://www.cs.virginia.edu/~survive/DOCS/prudent.ps. Acesso em 10 set. 2002.)

ANDERSSON, L.; Doolan, P., Feldman, N., et al. (2001) "LDP Specification". RFC 3036, Janeiro. (Disponível por http://www.ietf.org/rfc/rfc3036.txt. Acesso em 20 nov. 2001).

BUDA, G.; CHOI, D.; et. al. (2001) "Security Standards for the Global Information Grid". IFIP/IEEE International Symposium on Integrated Network Management, Seatle, Maio.

DE CLERCQ, J.; PARIDAENS, O.; T'JOENSET Y., SCHRIJVER, P. (2001) "End to End Authentication for LDP". Draft-schrijvp-mpls-ldp-end-to-end-auth-03.txt. jeremy.de_clercq@alcatel.be, fevereiro. (Contato com o autor em 10 jan. 2002. Cópia disponível por http://www.lrg.ufsc.br/~morvan/draft-schrijvp-mpls-ldp-end-to-end-auth-03.txt. A *draft* no IETF expirou cf. http://www.ietf.org/internet-drafts/draft-schrijvp-mpls-ldp-end-to-end-auth-04.txt).

HEFFERNAN, A. (1998) "Protection of BGP Sessions via the TCP MD5 Signature Option". RFC 2385, Agosto. (Disponível por http://www.ietf.org/rfc/rfc2385.txt. Acesso em 25 jan 2002).

JAMOUSSI, B, et al. (2002) "Constraint-Based LSP Setup using LDP". RFC 3212, Janeiro. (Disponível por http://www.ietf.org/rfc/rfc3212.txt. Acesso 12 fev. 2002).

LEU, J; et. al. (2000) "Project: MPLS for Linux". Grupo Source Forge, Novembro. (Disponível por http://sourceforge.net/projects/mpls-linux. Acesso em 06 fev. 2002). (Trabalho em progresso).

MÜLLER, M. (2002) "Uma Solução de Autenticação Fim a Fim para o LDP (Label Distribution Protocol)". Dissertação de Mestrado, Universidade Federal de Santa Catarina (UFSC), Centro Tecnológico (CTC), Florianópolis-SC, Brasil, Dezembro. (Disponível por http://www.lrg.ufsc.br/~morvan/dissert-ldpauth.pdf).

NIST - *National Institute for Standards and Technology.*(2000) "Descriptions of SHA-256, SHA-384, and SHA-512". Outubro. (Disponível por http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf. Acesso em 20 mai. 2002).

ROSEN, E.; REKHTER, Y. (1999) "BGP/MPLS VPNs". RFC 2547, Março. (Disponível por http://www.ietf.org/rfc/rfc2547.txt. Acesso em 13 set. 2001).

ROSEN, E; VISWANATHAN, A.; CALLON, R. (2001) "Multiprotocol Label Switching Architecture". RFC 3031, Janeiro. (Disponível por http://www.ietf.org/rfc/rfc3031.txt. Acesso em 26 jul. 2001).

STALLINGS, William. (1999) "Cryptography and Network Security: Principles and Pratice". New Jersey, editora Prentice-Hall, Inc., 2ª ed.