Estudo do protocolo SSL/TLS como Implementação de Segurança para Protocolos Inseguros Utilizados por Mecanismos de Controle de Acesso

Tiago Duarte, Mauricio Simões, Guilherme Gattino, Natan Schultz

Graduação Tecnológica em Segurança da Informação

Universidade do Vale do Rio dos Sinos (Unisinos)

São Leopoldo – RS – Brasil

{tiaguera,mauricio.nh,ggattino,natanschultz}@gmail.com

Abstract. *This paper describes a study on the SSL/TLS protocol to implement security for access control mechanisms that use insecure protocols and presents a practical experiment about it.*

Resumo. *Este artigo descreve um estudo sobre o protocolo SSL/TLS como implementação de segurança para mecanismos de controle de acesso que utilizam protocolos inseguros e que apresenta um experimento prático sobre este.*

1 Introdução

Este artigo tem como principal motivação entender o funcionamento do SSL/TLS como um complemento para mecanismo de controle de acessos, por ser uma camada de segurança adicional para diversos protocolos de comunicação em rede de computadores. Para que seja melhor compreendida a utilização do SSL/TLS nas aplicações e protocolos, será necessário que se entenda as motivações que levaram ao desenvolvimento desta camada de proteção.

A fundamentação teórica é essencial, pois com ela é possível ter uma noção básica de todas as funcionalidades do SSL/TLS, bem como entender seus métodos de segurança e autenticação. Importante também entender sobre os mecanismos de autenticação, quais as suas funções e real importância nos dias de hoje.

Com estas bases é possível iniciar o estudo mais aprofundado, acrescentando a fundamentação do SSL/TLS em cima de aplicações amplamente utilizadas no cotidiano, como HTTP, FTP, Telnet, SMTP, IMAP, POP3. O artigo abordará como que o SSL/TLS pode interagir com estas aplicações, tornando-as muito mais seguras e assegurar que os dados trafegados entre estes protocolos estejam íntegros e privados.

Para ilustrar a atuação do SSL/TLS sob uma aplicação, um estudo de caso é discutido e apresentado, mostrando os pontos que o SSL/TLS acrescenta em uma aplicação de conexão à Internet ou mesmo em rede local. Será mostrada as vantagens e as desvantagens de uso de aplicações com o SSL/TLS.

2 Fundamentação Teórica

2.1 Sobre o SSL/TLS

O *Secure Sockets Layer* - SSL foi desenvolvido pela Netscape em 1994, como resposta a uma crescente preocupação da época em ter segurança na navegação pela Internet. Foi originalmente desenvolvido para oferecer segurança para *browsers* de Internet e para servidores de comunicação. Foi desenvolvido para ser suportado por diversos protocolos, como o *File Transfer Protocol* (FTP) e o *Telnet*.

A versão 1.0 do SSL, desenvolvida em 1994, jamais foi lançado ao público. Em 1995 foi lançada a versão 2.0, que possuía inúmeras falhas de segurança. A correção destas falhas, originaram, em 1996, a última versão do SSL, a versão 3.0, que foi base para o protocolo *Transport Layer Securit*y - TLS versão 1.0.

Por ser um protocolo de aplicação independente pode ser usado com inumeros protocolos, complementando a segurança dos mesmos e proporcionando privacidade, autenticidade e a integridade de dados entre duas aplicações que se comunicam pela Internet [1]. Isto ocorre através da autenticação de ambas as partes e da encriptação dos dados trocados entre elas. A criptografia é aplicada de forma simétrica usando entre outros os métodos AES ou RC4, efetuando a troca das chaves atravéz do *handshake*, que pode ser desenhado de acordo com a aplicação pois o SSL/TLS é expansível. Toda a troca de dados é feita com verificação da integridade baseada em hash, usualmente SHA-1. O protocolo faz uso de certificados com formato X.509.

O SSL/TLS realiza a aplicação da segurança diretamente na camada de aplicação, diferenciando-se de outros protocolos de segurança, como o IPSec, por exemplo, que aplica a segurança na camada de rede, aumentando o tamanho do overhead do pacote.

2.2 Sobre mecanismos de autenticação

Os mecanismos de autenticação são maneiras de se autenticar um usuário dentro do sistema de forma lógica, seja ele distribuido ou não. É a capacidade que se tem em garantir que um usuário é quem ele diz ser. É a habilidade de segurança mais básica que um sistema deve possuir. Os sistemas de autenticação podem ser classificados em quatro categorias, de acordo com a sua forma de autenticação. A forma mais comum utilizada é a forma de usuário e senha, sendo esta a forma mais insegura de todas, entrando na categoria de “algo que você sabe”. Outros mecanismos, podem ser tokens, e até mesmo smartcards, classificados na categoria de “algo que você possui”. As impressões digitais, a biometria e análise da retina, são formas mais incomuns, mas que são as mais seguras que podem existir e estão classificadas na categoria de “algo que você é”. Há também autenticação baseada em sua localização (classificadas como “onde você está”), como autenticação por endereço do adaptador de rede, posicionamento global, etc.

3. SSL/TLS como recurso de segurança para protocolos inseguros utilizados mecanismos de autenticação

O SSL/TLS interage com os mecanismos de controle de acesso garantindo a privacidade da troca de usuário e senha da conexão a ser estabelecida, pois ele já está ativo antes mesmo de o provedor solicitar os dados do usuário pois ocorre a troca de

chaves publica no momento da conexão, independente do aplicativo usado. O SSL/TLS é um protocolo que opera na camada de aplicação encapsulando outros protocolos da camada de aplicação de forma a aumentar a segurança dos dados trafegados.

Existem RFCs, que são documentos que descrevem os padrões de cada protocolo para fim de padronização dos mesmos, que descrevem como é feita a implementação do SSL/TLS com os protocolos IMAP, POP3, ACAP [2], Kerberos [3], HTTP [4], SMTP [5], FTP [6], UDP [7], XMPP [8], EAP-TLS [9], LDAP [10], entre outros. Mas também existem outras implementações do protocolo SSL/TLS em soluções não documentadas por nenhuma RFC como o OpenVPN, solução muito difundida inclusive comercialmente, e sobre outros protocolos como por exemplo: RDP, VNC, Telnet, MS-CHAP, entre outros.

4 Estudo de casos

Será descrito a seguir como o protoloco SSL/TLS interage com os mecanismos de controle de acesso de um servidor de FTP, o VSFTPd, e com o sistema PHPmyAdmin, que serve para administrar banco de dados MySQL por intermédio de uma interface Web. Para fins didáticos utilizamos o sistema operacional Linux Ubuntu 10.04 LTS para a instalação destes softwares e todos os procedimentos serão feitos locamente, sem utilização de rede, com o usuário root.

4.1 Análise do protocolo FTP com e sem suporte a SSL/TLS

Foi instalado o servidor de FTP VSFTPd, o OpenSSL/TLS para geração de certificados, o cliente de FTP Filezilla e o analizador de tráfego de rede Wireshark em um sistema operacional Linux Ubuntu 10.04 LTS para fins de estudo de segurança do protocolo FTP sem e com o uso de SSL/TLS. Não entraremos em detalhes sobre a instalação e configuração destes softwares para não prolongar muito a extensão deste artigo sendo que iremos focar este no experimento prático em si com todos em funcionamento.

Foram gerados certificados auto-assinados no OpenSSL/TLS para utilizar este no servidor VSFTPd, sendo que este teve seu arquivo /etc/vsftpd.conf configurado para primeiramente não utilizar o suporte a SSL/TLS e posteriormente o utilizar.

O mecanismo de autenticação utilizado pelo servidor VSFTPd neste caso é feito através do PAM (Pluggable Authentication Modules) que é um módulo que permite utilizar algum mecanismo de controle de acesso que seja compatível, como por exemplo o Kerberos, mas neste caso em especial será utilizado o mecanismo de controle de acesso nativo do sistema operacional Linux.

Com o servidor VSFTPd pronto com as duas configurações, sem SSL/TLS e com SSL/TLS, salvas em arquivos de becape, iremos utilizar para realização de teste o cliente de FTP Filezilla e o analizador de tráfego de rede Wireshark no mesmo sistema operacional em questão que iremos rodar o servidor VSFTPd.

Primeiramente para iremos rodar o servidor VSFTPd sem o suporte a SSL/TLS e então rodar o Wireshark para capturar todo o trafégo local para então utilizar o cliente de FTP Filezilla que se conectará ao serivor de FTP local em operação. A figura abaixo ilustra as configurações do Filezilla para acessar um servidor de FTP sem suporte a SSL/TLS, sendo que para tal é necessário clicar no ícone “Open de Site Manager” do mesmo.

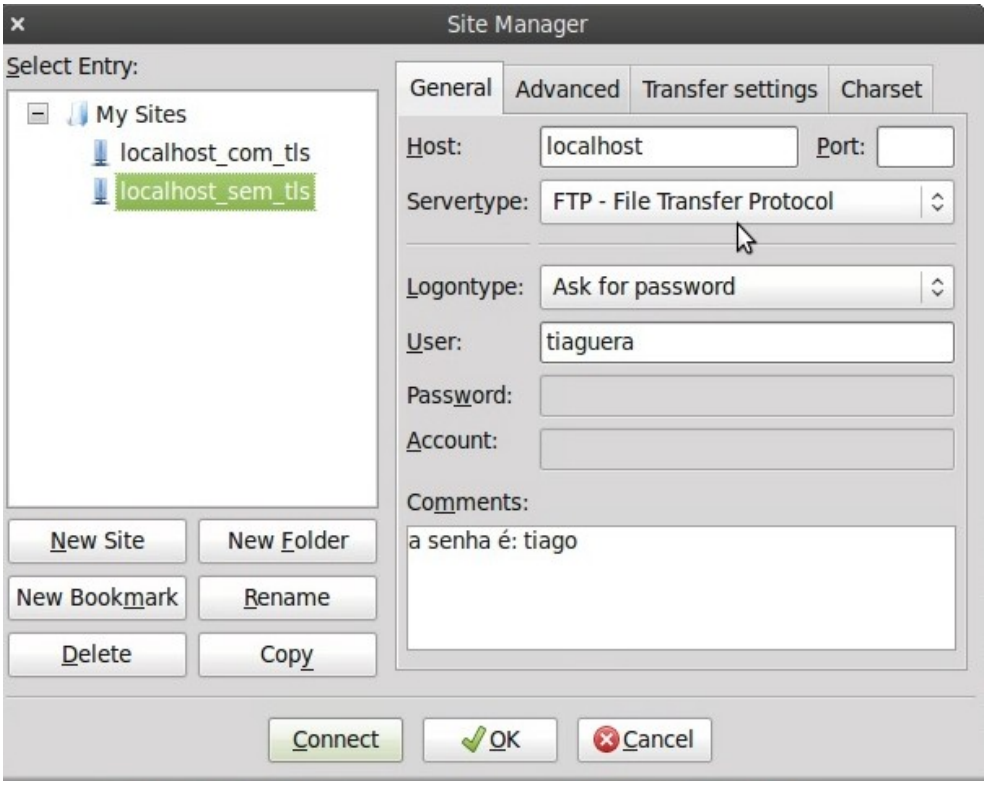

Figura 1 – Configuração do cliente de FTP Filezilla para não usar SSL/TLS

Ao analisar o tráfego no Wireshark podemos localizar através da tecla de atalho "CTRL+F" marcando a opção String e digitando no campo "Filter" a termo "pass" para podermos localizar entre os pacotes capturados o conteúdo que leve-nos a um "password" (senha). A figura abaixo exibe como é feita tal localização:

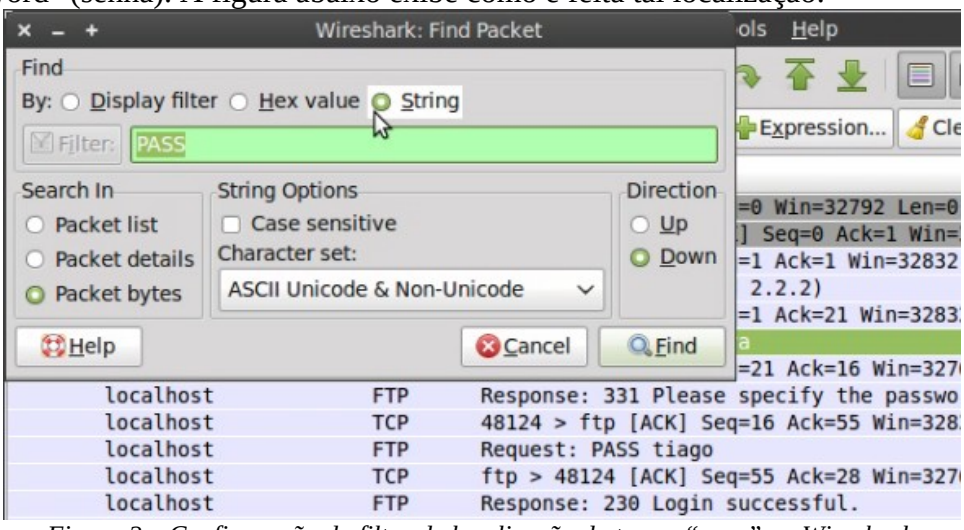

Figura 2 – Configuração do filtro de localização do termo "pass" no Wireshark

Podemos notar após a pesquisa que a senha em questão é "tiago". Isto ocorreu porque não existe nenhum tipo de criptografia sobre o protocolo FTP.

Agora podemos realizar exatamente mesmo procedimento feito com a configuração do servidor VSFTPd com o suporte a criptografia SSL/TLS utilizando a

segunda configuração apresentada do arquivo /etc/vsftpd.conf e após a alteração deste deve-se reiniciar o servidor.

O cliente de FTP Filezilla deve ter as seguintes opções descritas na figura abaixo para que possa se beneficiar do uso da criptografia do SSL/TLS:

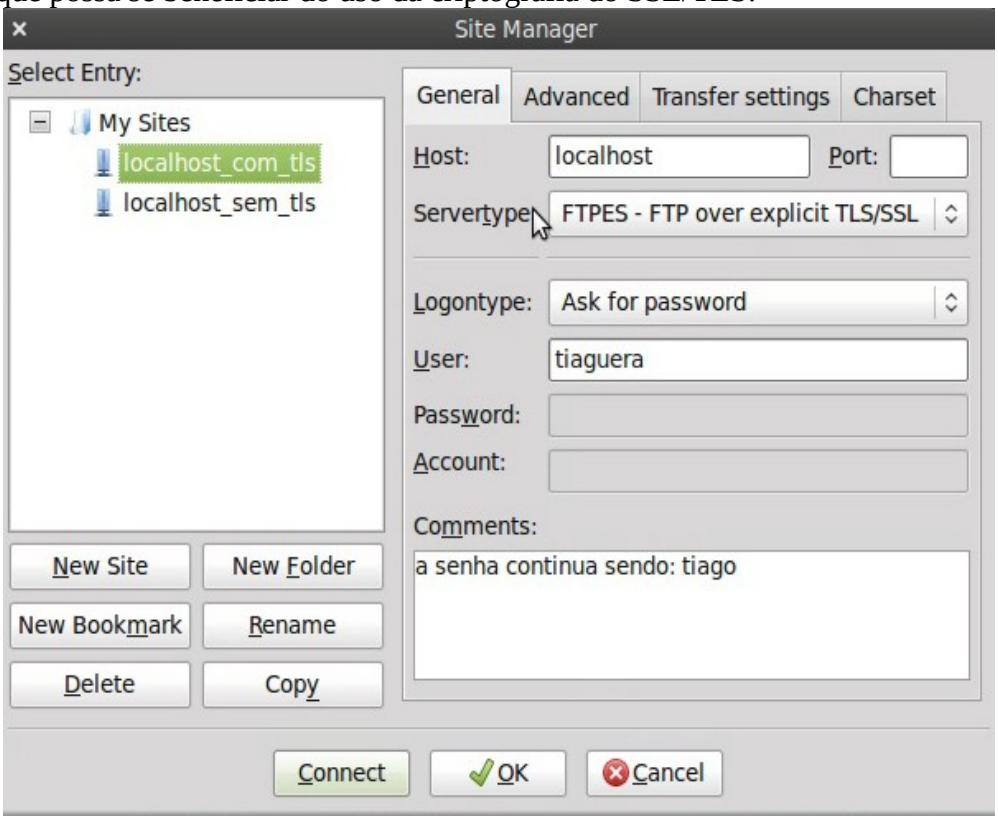

Figura 3 – Configuração do cliente de FTP Filezilla para usar SSL/TLS

Desta vez não será possível localizar a senha do usuário tiaguera da mesma forma descrita anteriormente no Wireshark, sendo que desta vez será retornado um erro como este da figura abaixo:

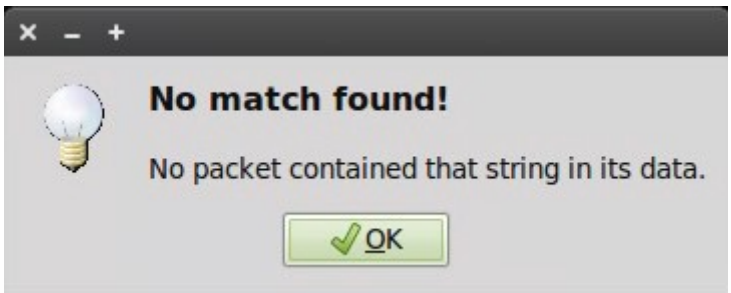

Figura 4 – Erro do Wireshark de pesquisa de string não localizada

Também foi constatado que não houve mudança de tráfego quanto ao protocolo, que foi de FTP em ambos os casos, sendo que o segundo o mesmo estava ilegivel devido a criptografia provida pelo SSL/TLS.

4.1 Análise do protocolo HTTP com e sem suporte a SSL/TLS

Foi instalado o servidor de websites Apache 2 com suporte a linguagem PHP5, o servidor de banco de dados MySQL 5, o software web de adminsitração do mesmo PHPMyAdmin e o navegador web Chromium no mesmo sistema operacional Linux utilizado no experimento anterior que já possuia o OpenSSL/TLS e o Wireshark previamente instalados. No servidor Apache 2 foi adicionado o recurso de suporte ao protocolo HTTPS através da implementação de certificado auto-assinado com o OpenSSL/TLS para o mesmo. Não entraremos novamente em detalhes sobre a instalação e configuração destes softwares para não estender demais este artigo, sendo que o foco deste é o experimento em si da análise de segurança do protocolo HTTP e HTTPS.

Para tal será analisado através do Wireshark o tráfego decorrente de uma seções de login no PHPMyAdmin através dos protocolos mencionados anteriormente.

Primeiramente foi realizado um login no PHPMyAdmin utilizando o protocolo HTTP através do endereço local http://localhost/phpmyadmin utilizando o usuário “root” e a senha “tiago” novamente, sendo esta a senha do banco de dados MySQL 5.

Através da mesma pesquisa pelo termo “pass” no Wireshark como descrito no experimento anterior, veja a figura 2, e chegamos a senha “tiago” conforme a ilusta figura abaixo:

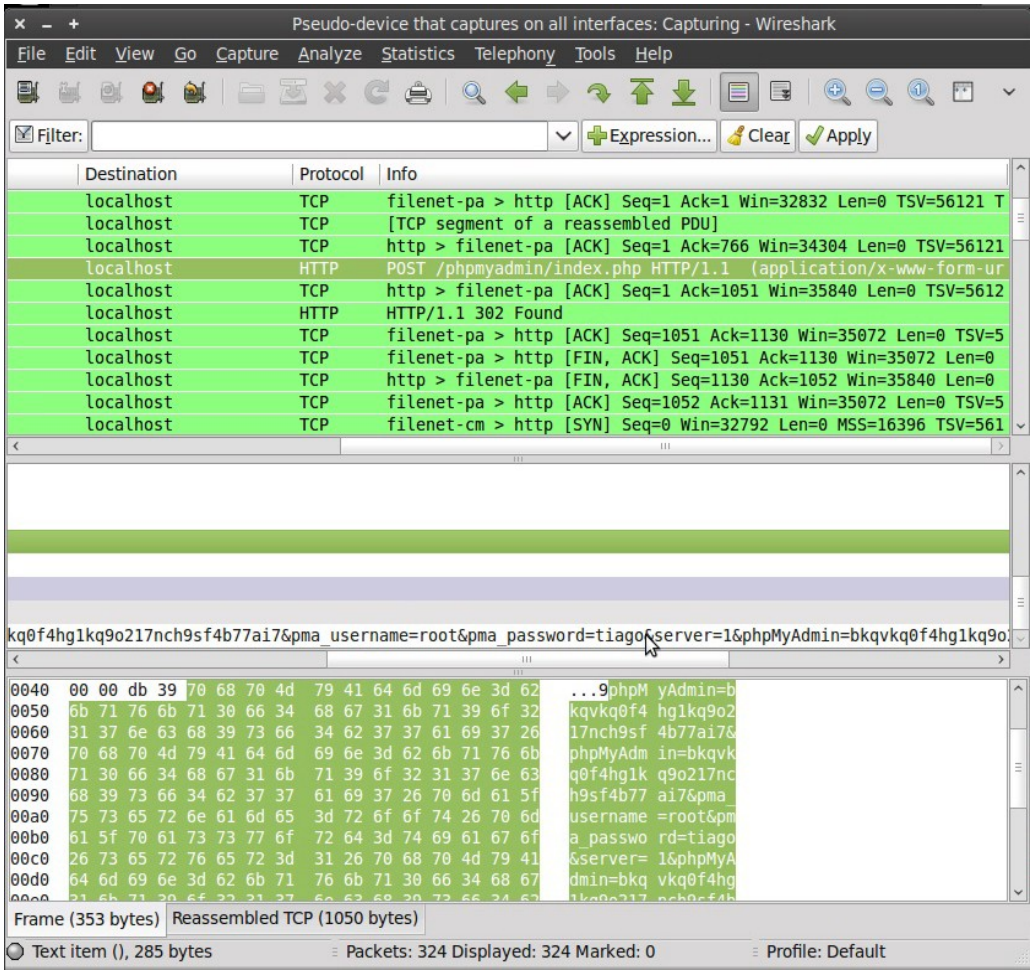

Figura 4 – Localização de senha em autenticação feita usando protocolo HTTP

Após foi efetuada novamente outra seção de login com o mesmo usuário e senha descrito anteriormente, mas desta vez acessando o endereço

https://localhost/phpmyadmin que se beneficia do uso de criptografia SSL/TLS através de certificado auto-assinado do gerado pelo OpenSSL/TLS.

Analisando este tráfego no Wireshark percebemos que houve mudança no trafégo gerado, alterando o mesmo de protocolo HTTP para SSL/TLS. Ao fazer a mesma pesquisa pelo termo “pass” no Wireshark obtivemos o mesmo erro descrito pela figura 4.

Conclusão

O protocolo SSL/TLS oferece um nível de criptografia satisfatório para protocolos considerados inseguros por natureza que são utilizados por mecanismos de controle de acesso aumentando a segurança dos mesmos viabilizando a implementação destes em redes computacionais.

Referências

[1] WAGNER, David ; SCHNEIER, Bruce - Analysis of the SSL 3.0 Protocol – Proceedings of the Second USENIX Workshop on Electronic Commerce - Oakland, California – 1996.

[2] RFC 2595: “Using TLS with IMAP, POP3 and ACAP”

[3] RFC 2712: "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)"

[2] RFC 2595: “Using TLS with IMAP, POP3 and ACAP”

[3] RFC 2712: "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)"

[4] RFC 2817: “Upgrading to TLS Within HTTP/1.1”

[5] RFC 3207: “SMTP Service Extension for Secure SMTP over Transport Layer Security”

[6] RFC 4217: “Securing FTP with TLS”.

[7] RFC 4347: “Datagram Transport Layer Security

[8] RFC 3920: "Extensible Messaging and Presence Protocol (XMPP)"

[9] RFC 2716: "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)"

[10] RFC 4513: "LDAP: Authentication Methods and Security Mechanisms"