UNIVERSIDADE FEDERAL DE OURO PRETO
DEPARTAMENTO DE COMPUTAÇÃO

SISTEMAS OPERACIONAIS

PROVA 3

Aluno
Lucas de Araújo

Professor
Dr. Carlos Frederico Marcelo da Cunha Cavalcanti

Sumário

1 Introdução

Esta prova tem como objetivo apresentar aspectos de segurança envolvendo a ferramenta OpenSSL em um sistema operacional fictício chamado XSO. Dentre os diversos aspectos que envolvem a ferramenta, a a prova irá destacar como os mais importantes:

- O que é o OpenSSL e como o mesmo será útil para nós
- As aplicações do OpenSSL hoje e como aplicá-lo neste sistema operacional
- A bibliografia consultada durante a execução do trabalho

Também vale ressaltar que a apresentação em vídeo será feita seguindo o modelo de uma apresentação de negócios, assim como é solicitado pelo o enunciado da prova

2 OpenSSL

2.1 O que é o OpenSSL

Lançado em 1998 para Windows, Linux e MacOS, o OpenSSL se trada de um projeto que possui uma extensa biblioteca de criptografias capaz de manipular o protocolo de segurança TLS e permitindo aos usuários executar diversas tarefas relacionadas com o certificado digital SSL.

O projeto do OpenSSL disponibiliza uma caixa de ferramentas em código aberto que implementa o protocolo SSL e também vários algoritmos e primitivas criptográficas de uso comum envolvendo o mesmo. Dentre estas primitivas, podemos citar:

- Algoritmos de troca de chaves
- Funções de hash
- Algoritmos simétricos e assimétricos

Esta caixa de ferramentas (*toolkit*) se apresenta na forma de duas bibliotecas e um conjunto de programas que implementam as rotinas por elas disponibilizadas. Os mecanismos do SSL estão implementadas na biblioteca libssl e os outros algoritmos na biblioteca libcrypto

2.2 Utilidades do OpenSSL no Cotidiano

Como ponto forte da utilização do OpenSSL, temos a criação de certificados digitais para serviços na web, serviços estes que dependem profundamente de provar sua autenticidade por alguma forma afim de evitar fraudes. Como exemplo destes serviços, temos: Bancos e Sites do Governo

2.2.1 Certificados Digitais

Um certificado digital é utilizado como forma de provar a ligação entre uma chave pública e a instituição que a possui. Os certificados digitais destinam-se à partilha de chaves públicas a serem utilizadas para a encriptação e autenticação.

Os certificados digitais incluem em si, a chave pública a ser certificada, informações que identificam a instituição proprietária desta mesma chave, metadados relacionados ao certificado digital e uma assinatura digital da chave pública criada pelo emissor do certificado

2.2.2 Geração de Certificados Digitais pelo OpenSSL

Como citado, podemos utilizar do OpenSSL para criar certificados digitais autoassinados para nossos serviços. O procedimento é simples e se da através dos seguintes passos:

- Anote o Nome comum (CN) do Certificado SSL. O CN é o nome completo do sistema que usa o certificado. Se você estiver usando DNS dinâmico, seu CN deverá ter um curinga, por exemplo: *.api.com. Caso contrário, use o nome do host ou o endereço IP configurado no Cluster de gateway (por exemplo: 192.16.183.131 ou dp1.acme.com)

- Execute o comando OpenSSL a seguir para gerar sua chave privada e seu certificado público. Responda às perguntas e insira o Nome comum quando solicitado.

  openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem

- Revise o certificado criado:

  openssl x509 -text -noout -in certificate.pem

- Combine sua chave e o certificado em um pacote configurável PKCS12 (P12):

  openssl pkcs12 -inkey key.pem -in certificate.pem -export -out certificate.p12

- Valide seu arquivo P2.

  openssl pkcs12 -in certificate.p12 -noout -info

- Basta agora fazer upload do certificado no servidor onde se encontra hospedado o serviço e configurar o servidor para começar a fazer uso do mesmo

3 Utilidades do OpenSSL no XSO

Como mencionado, o OpenSSL é capaz de gerir muito bem chaves que envolvem criptografia e seus respectivos algoritmos. Com isto, podemos pensar pensar em algumas utilidades envolvendo o processo de criptografia dentro do sistema operacional e como isso beneficiaria o usuário final. Podemos citar:

3.1 Criptografia de Arquivos Importantes

Dentro do nosso sistema operacional, arquivos com informações sensíveis (por exemplo, senhas do usuário) não devem simplesmente estarem disponíveis para sem lidos por qualquer pessoa ou programa na maquina com facilidade, isto causaria um problema enorme de privacidade ao usuário.

Por isso, seria interessante utilizarmos alguma criptografia sobre estes arquivos e armazená-los de forma segura

Criptografando o arquivo, seria realizado somente a descriptografia do mesmo em momentos de necessidade (por exemplo, comparação de senhas) através do uso de uma palavra chave pré-determinada pelo próprio usuário

Como exemplo, poderíamos criptografar um arquivo utilizando o algoritmo AES-256 já presente na biblioteca do OpenSSL, através do seguinte comando:

openssl aes-256-cbc -in arquivoSenha.txt -out arquivoCriptografado.bin

Uma senha será solicitada e a mesma será utilizada para descriptografar o arquivo quando necessário

3.2 Autenticidade do Kernel

Uma possível ideia interessante para nosso sistema seria a criação de chaves que cuidem de provar a autenticidade do kernel presente no nosso sistema. A ideia consistem em criar uma hash ao final do processo de criação do kernel e salvá-la criptografada em um espaço seguro do hardware.

Após feita a instalações de programas e softwares que realizem algum acesso ao kernel, é feita a hash do kernel novamente e comparada a hash antiga para detectar se houve ou não alterações.

Dessa forma, garantimos uma integridade fundamental do kernel para nosso usuário, permitindo a segurança de uma parte fundamental do sistema operacional

3.3 Conexões Seguras

Como mencionado anteriormente, o OpenSSL trabalha com o Transport Layer Security (TLS) e Secure Sockets Layer (SSL). Dessa forma, podemos utilizá-lo para realizar testes que nos permitam conferir a segurança de conexões estabelecidas a serviços com nosso computador.

Por exemplo, se utilizarmos o comando:

openssl sclient -connect domain:port

Podemos conferir se a conexão a determinado domínio em determinada porta é segura ou não.

O retorno disto deve ser as chave publica e as diversas informações sobre o certificado digital envolvido no determinado serviço. Mas como isso seria útil no sistema OSX?

Simples, nosso sistema precisará sofrer atualizações ao longo do tempo para que melhorias sejam enviadas ao usuário, com isso, precisamos garantir que a conexão que está sendo realizada com o servidor responsável por fornecer as atualizações sejam autênticas.

Um exemplo disso na vida real são os famigedados *PPAs* dos sistemas linux baseados em debian, que consiste em diversos dominios de repositóris que gerem atualizações para o sistema de forma segura

4 Bibliografia

Todas as fontes foram checadas com atenção a fim de garantir que as mesmas eram confiáveis

- https://www.ssldragon.com/blog/what-is-openssl-and-how-it-works
- https://www.ibm.com/docs/pt-br/api-connect/5.0.x?topic=profiles-generating-self-signed-certificate-using-openssl
- https://www.liquidweb.com/kb/how-to-verify-a-connection-is-secure-using-openssl/
- https://www.openssl.org
- https://wiki.openssl.org
- https://www.openssl.org/docs/fips/UserGuide-2.0.pdf
- https://en.wikipedia.org/wiki/OpenSSL