

Segurança e Auditoria de Sistemas

EC38D - Ativ. Didática Não Presencial


Aula 1

Lucas Dias Hiera Sampaio

Universidade Tecnológica Federal do Paraná (UTFPR),
Câmpus Cornélio Procopio

04 de Agosto de 2020

Roteiro

- 
- 1 Introdução
 - 2 Revisão
 - 3 Hash
 - 4 Aplicações de Hash

Introdução

O que já foi abordado

- Fundamentos de Segurança
- Criptografia Simétrica
- Criptografia Assimétrica

Introdução

Planejamento

- 04/08: Revisão + Hash, Árvore de Merkle, Assinatura Digital
- 11/08: Blockchain, Algoritmos de Consenso e Criptomoedas
- 18/08: Segurança em Redes: Camada Física e Enlace
- 25/08: Segurança em Redes: Camada de Rede e Transporte
- 01/09: Segurança em Redes: Camada de Aplicação

Introdução

Planejamento

- 08/09: Segurança em Redes: Firewalls
- 15/09: Segurança em Redes: Firewalls
- 22/09: Segurança em Redes: Sistemas de Detecção de Intrusão
- 29/09: Buffer Overflow
- 06/10: Buffer Overflow
- 13/10: Cross-Site Scripting, Injection e Malwares

Introdução

Processo de Avaliação

- Contínuo
- Auto-Regulada
- Atividades com Feedback dxs colegas e do professor
- Não há "prova" logo não deixe de fazer as atividades
- Presença: condicionada a entrega das atividades

Introdução

Processo de Avaliação

- Entrega da Atividade até o Prazo Determinado
- Recebe o questionário de avaliação (1 dia de prazo)
- Recebe o questionário de outrx alunx para avaliar (2 dias de prazo)
- Recebe o Feedback/Nota do Professor (2 dias de prazo)
- A avaliação será feita em modo *double blind review*

Introdução

Processo de Avaliação - Data de Entrega e Atividades

- 12/08: Comunicação ponto a ponto com troca de chave simétrica e assinatura digital de mensagens.
- 20/08: Leitura de Artigo e resposta de questionário.
- 31/08: Man in the Middle via ARP Poisoning
- 08/09: Gerando Certificados e Configurando o APACHE.
- 30/09: Exercícios de configuração de Firewall.
- 01/11: Buffer Overflow e Resolução de Problemas.

Introdução

Acesso ao Moodle

■ seg20

Revisão

Segurança

- Políticas de Segurança
- Modelos de Ameaça
- Mecanismos de Segurança

Revisão

Segurança

- Diretrizes de Segurança: Características e objetivos que são almejados no processo, empresa, produto, etc. em termos dos pilares da segurança da informação. Por exemplo, a autenticidade no acesso de uma plataforma.

Revisão

Segurança

- Modelos de Ameaça: São modelos computacionais, matemáticos, estatísticos e de comportamento que os possíveis atacantes e/ou usuários do sistema podem utilizar para comprometer as diretrizes de segurança. Por exemplo: usuário tentar por força bruta autenticar num sistema de login/senha.

Revisão

Segurança

- Mecanismos de Segurança: São as ferramentas, técnicas, métodos, *software*, *hardware*, etc. que são utilizados para garantir as diretrizes dado o modelo de ameaças. Por exemplo: limitar o número de tentativas por minuto que um usuário/endereço de IP pode tentar logar no sistema.

Revisão

Causos Discutidos

- Sarah Palin
- Zach Harris
- Conta @N Twitter
- Hardware Seguro
- Consultem as Notas de Aula.

Revisão

Criptografia Simétrica

- Foco inicial era a garantia da confidencialidade
- Algoritmo público
- Chave é o segredo
- A chave que criptografa a informação é a mesma que descriptografa.

Revisão

Modo de Operação

- Bloco
- Fluxo

Revisão

Modo de Operação - Fluxo

- O texto a ser cifrado é criptografado bit a bit, de forma geral, utilizando a operação XOR.

Revisão

Modo de Operação - Bloco

- *Electronic Block Code (EBC)*
- *Cipher Block Chaining (CBC)*
- *Propagating CBC (PCBC)*
- *Cipher Feedback (CFB)*
- *Output Feedback (OFB)*
- *Counter (CTR)*

Revisão

Modo de Operação - Bloco

- *Electronic Block Code (EBC)*
- *Cipher Block Chaining (CBC)*
- *Propagating CBC (PCBC)*
- *Cipher Feedback (CFB)*
- *Output Feedback (OFB)*
- *Counter (CTR)*

Revisão

	Formulas	Ciphertext
(ECB)	$Y_i = F(\text{PlainText}_i, \text{Key})$	Y_i
(CBC)	$Y_i = \text{PlainText}_i \text{ XOR } \text{Ciphertext}_{i-1}$	$F(Y, \text{Key}); \text{Ciphertext}_0 = \text{IV}$
(PCBC)	$Y_i = \text{PlainText}_i \text{ XOR } (\text{Ciphertext}_{i-1} \text{ XOR } \text{PlainText}_{i-1})$	$F(Y, \text{Key}); \text{Ciphertext}_0 = \text{IV}$
(CFB)	$Y_i = \text{Ciphertext}_{i-1}$	$\text{Plaintext XOR } F(Y, \text{Key}); \text{Ciphertext}_0 = \text{IV}$
(OFB)	$Y_i = F(Y_{i-1}, \text{Key}); Y_0 = F(\text{IV}, \text{Key})$	$\text{Plaintext XOR } Y_i$
(CTR)	$Y_i = F(\text{IV} + g(i), \text{Key}); \text{IV} = \text{token}()$	$\text{Plaintext XOR } Y_i$

Figura: Fonte: [2]

Revisão

Criptografia Simétrica

- Apenas uma chave
- Processo de criptografia é rápido
- Segurança está na chave
- Troca de chaves constantes

Revisão

Criptografia Simétrica - Exemplos

- AES (Rinjdael)
- DES
- TDES
- Cifra de César
- Substituição Simples

Revisão

Criptografia Assimétrica

- Como enviar a chave simétrica em um canal inseguro?

Revisão

Criptografia Assimétrica

- Como enviar a chave simétrica em um canal inseguro?
- Criptografia Assimétrica!

Revisão

Criptografia Assimétrica

- Duas chaves
- Processo de criptografia é lento
- Segurança está no problema matemático
- Troca de chaves não precisam ser constantes

Revisão

Criptografia Assimétrica

- Chave Pública - Conhecida por todos
- Chave Privada - Conhecida apenas pelo dono de par de chaves
- Criptografia com a chave pública: garante confidencialidade
- Criptografia com a chave privada: garante autenticidade*

Revisão

Criptografia Assimétrica - Algoritmos

- RSA
- ECC
- NTRU

Funções de Hash

Definição

Seja h uma função de Hash, então $h : \{0; 1\}^* \longrightarrow \{0; 1\}^N$, i.e.

- 1 Uma função de Hash tem entrada de tamanho arbitrário
- 2 Uma função de Hash tem saída de tamanho fixo N
- 3 O alfabeto de entrada e saída são números binários

Funções de Hash

Requisitos

- h mistura os bits de entrada de tal forma que os resultados são uniformemente distribuídos no contra domínio, i.e. no espaço de possíveis Hashs.
- h é rápida o suficiente para não ferir requisitos de disponibilidade porém não tão rápida a ponto de um ataque de força bruta ser trivial.

Funções de Hash

Requisitos

- h minimiza a ocorrência de saídas idênticas para diferentes entradas (colisões)
- dado uma saída de h não é possível construir outra entrada que dê origem a mesma saída, exceto por força bruta.

Funções de Hash

Exemplos

- Checksum: somar a quantidade de bits 1 (ou zero) em uma string.
- Bitwise XOR: dividir a string de entrada em x blocos de tamanho N e realizar a operação XOR entre todos os blocos.

Funções de Hash

Algoritmos

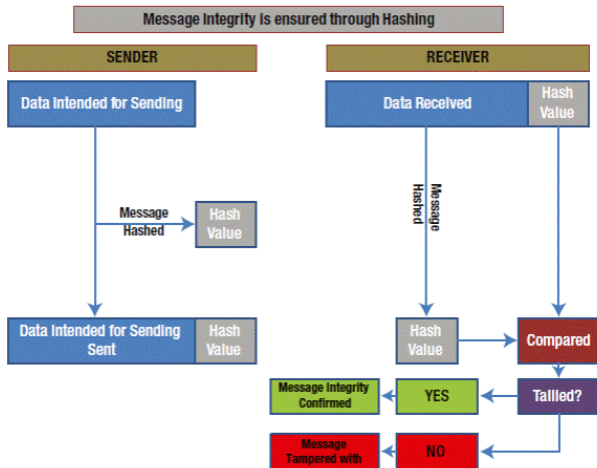
- MD5 (cuidado!) - 1992
- Família SHA (Secure Hash Algorithm) - 1993 a 2015
- SHA-0 e SHA-1
- SHA-2 e SHA-3 (ou SHA-OutputSize, e.g. SHA-256)

Funções de Hash

Uso

- Verificar a integridade de dados
- Assinatura digital (Hash Criptografado)
- Árvore de Merkle
- Senhas em bancos de dados

Verificar a integridade



Assinatura Digital

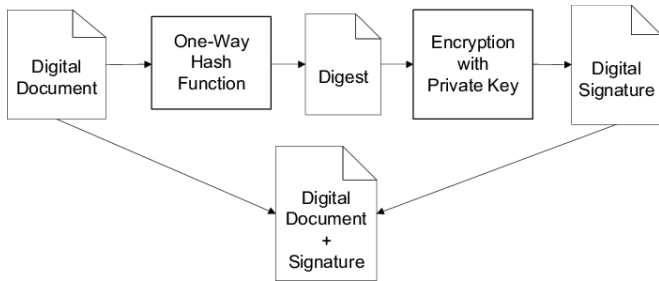


Figura: Fonte: [1]

Árvore de Merkle

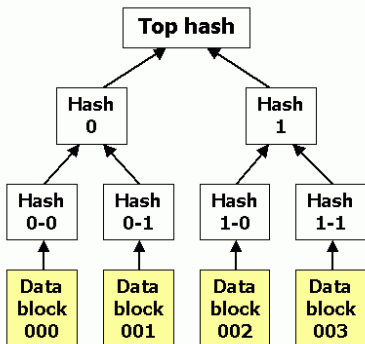


Figura: Fonte: [3]

Árvore de Merkle

- Árvore N-ária
- Permite a economia de memória ao custo de processamento

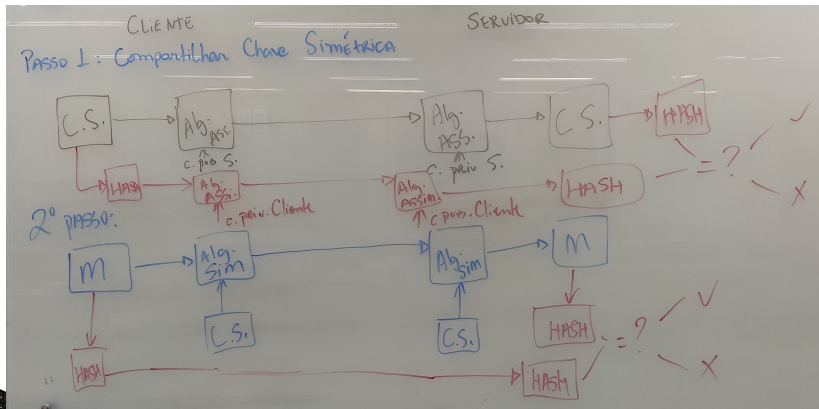
Salt

- Nunca gravar senha em texto pleno em um banco de dados
- Utilizar Salt: $\text{Hash}(\text{Senha} + \text{IV})$. Gravar o resultado do Hash e o IV no banco.

Pepper

- Utilizar Pepper: $\text{HASH}(\text{Senha} + \text{Secret})$. Gravar apenas o Hash resultante no banco. Secret deve ser gravado em outro lugar e não pode ser conhecido nem pelo usuário nem pelo Banco. O NIST recomenda o uso de Pepper para gravar senhas em bancos (Veja <https://pages.nist.gov/800-63-3/sp800-63b.html#-5112-memorized-secret-verifiers>)

Transmissão Segura



Na próxima aula:

- Blockchains, Criptomoedas e Algoritmos de Consenso

References I



R. Rudi and B. Celler.

Improving data security of home telecare systems.
08 2020.



Wikipedia.

Block cipher mode of operation, 2020.
Acesso em 28/07/2020.



Wikipedia.

Árvores de merkle, 2020.
Acesso em 28/07/2020.

