SSH Private Key Files

Chaves criptográficas para uso pessoal com SSH e muitas outras ferramentas.

Autor
Ricardo Striquer Soares

Ricardo Striquer Soares

SSH Personal Key Files

Chaves criptográficas para uso pessoal com SSH

Crash Course RSS Series

1a.Edição

Curitiba

Edição do Autor

2020

Sobre Direitos Autorais

Fotos e ilustrações

Ricardo Striquer Soares

Macrovector - http://macrovector.com/

freepik: https://www.freepik.com/free-photos-vectors/business

Chamada para “Colaboração”

https://www.freepik.com/free-vector/public-contribution-concept_2448537.htm#page=1&query=contribution&position=4

Ícone em “Dica”

https://www.freepik.com/free-vector/great-idea-retro_749597.htm#page=1&query=lamp&position=30

Ícone em “Importante”

https://www.freepik.com/free-vector/warning-pop-up-template-with-flat-design_2611512.htm#page=3&query=exclamation&position=44

Ícone em “Notas do Autor”

https://pixabay.com/illustrations/note-icon-symbol-sign-design-2389227/

Ícone em “Curiosidade”

https://pixabay.com/vectors/magnifying-glass-unknown-search-2831367/

Ícone de Chave

https://www.freepik.com/free-vector/set-key-silhouettes_1544035.htm#page=1&query=keys%20flat&position=9

Ícone de Engrenagens

https://www.freepik.com/free-vector/illustration-gear-icon_3232943.htm#page=1&query=gears&position=2

Ícone de Documento

https://www.freepik.com/free-vector/work-office-elements_715490.htm#page=1&query=document%20flat&position=27

Símbolos

https://www.freepik.com/free-vector/memphis-pattern_1177561.htm#query=simbols&position=3

Selo de Aprovado

https://www.freepik.com/free-vector/three-rubber-stamps_1055199.htm#page=1&query=approve&position=1

Imagem de Introdução

https://www.freepik.com/free-vector/colorful-name-tag-template-collection_2994922.htm#page=1&query=introduction&position=11

Imagem de Conhecimentos Elementares

https://www.freepik.com/free-vector/modern-education-concept-with-flat-design_2916225.htm#page=1&query=knowledge&position=0

Imagem de Boas Práticas

https://www.freepik.com/free-vector/public-approval-concept-illustration_6543305.htm#page=3&query=thumbs+up&position=21

Aviso

Esse livro é distribuído como está, sem garantia de qualquer forma ou tipo, seja expressa ou implícita.

Marcas Registradas

Várias marcas e imagens aparecem ao longo deste livro. A equipe de criação deste livro declara estar utilizando tais marcas para fins editoriais, em benefício único do proprietário destas, sem intenção de infringir regras e leis de sua utilização, propriedade ou qualquer outra.

Sumário

“De boas intenções o inferno está cheio, mas sem elas o céu seria um deserto. Sem boas intenções o coração não é feliz e a vida não vale nada!”

Ricardo Soares

Agradecimentos

A toda a equipe do Google Docs por me disponibilizar um ótimo editor de textos para escrever este documento.

Dedicatória

A Dennis MacAlistair Ritchie, que juntamente com Ken Thompson criou a linguagem C e que em conjunto com Brian Wilson Kernighan criou o primeiro livro sobre a linguagem C, que por sua vez facilitou o desenvolvimento de tecnologias como o SSL e permitiu todo um universo de novidades que hoje vemos ao toque de nossos dedos.

Sobre o Autor

Ricardo Striquer Soares nasceu no interior do Paraná. Ainda muito jovem seus pais o mandaram para viver com alguns de seus irmãos mais velho em uma cidade maior, onde teve a oportunidade de uma educação mais aprimorada. Em seu entendimento, nascer no interior e morar em grandes centros urbanos o ajudaram a construir uma visão diferenciada de vida.

Como Engenheiro de Programação desenvolveu vários sistemas e atendeu vários tipos de instituições, governo, indústria, comércio e serviço. Participou de diversos projetos de alto nível de dificuldade, alguns com inteligência artificial e segurança da informação. É formado em Processamento de Dados no Colégio Dominus, cursou Bacharel em Analise de Sistemas, Gestão Estratégica de Vendas e Gestão Financeira.

Hoje ele se considera um Geeker e empreendedor, adora andar de moto, gosta tanto de história que às vezes pensa que é um viajante do tempo! É fã de Star Trek (Jornada nas Estrelas) e teve a sorte de se casar com o amor de sua vida.

“Acredito que ajudarmos ao próximo é o motivo de estarmos nesta vida! Por isto comecei a escrever, transmitindo assim aquilo que para mim é de maior valor, meus pensamentos!”

Caso deseje você pode entrar em contato diretamente com o autor, ele certamente estará muito feliz em discutir assuntos sobre esse e outros livros que publica.

Twitter: https://twitter.com/rstriquer

LinkedIn: https://www.linkedin.com/in/rstriquer

email: rstriquer@gmail.com

Lista de colaboradores

No momento a única pessoa que colaborou para a construção do livro é o próprio autor, todavia havendo pessoas entrando em contato para agregar mais conteúdos estas pessoas ganharão espaço nesta parte do livro para se apresentarem a comunidade.

Sobre Livros

Platão não gostava de livros, dizem que o comparava a um “mestre que fala e não houve”, completamente avesso aos ideais contemporâneos de um professor que deve entender o aluno para inspirar nele a demanda de conhecer. Por isto ele preferia a dialética. Entretanto mesmo nos dias de hoje não temos uma plataforma em que alguém possa fomentar seu conhecimento de forma que os demais o consomem de maneira extremamente personalizada, similar a um diálogo direto entre dois indivíduos, talvez por isso Platão cuidava muito bem de escrever seus versos em seus conhecidos livros que construíram o que veio a ser os alicerces do mundo ocidental.

Escrever um livro, ou fazer uso de outros meios terá um resultado similar naquele que faz uso do recurso para transmitir a informação, ao passo que a absorção do conhecimento irá variar de acordo com as preferências do receptor. Isto é o que dizem vários comunicadores, ávidos a vender sua própria opinião! Não há dúvidas de que um vídeo tem um impacto diferente de um livro, no entanto, por experiência própria e por estudos sei que, quando o assunto é complexo como o que aqui iremos tratar, o mais recomendado para a verdadeira abstração do conhecimento ainda é o livro, com ele você terá tempo para interpretar a informação fazendo com que você realmente entenda o assunto sendo apresentado.

Por séculos seguidos o livro é comprovadamente uma maneira eficaz de transmitir o conhecimento e o livro como o conhecemos acredito que é uma das verdadeiras invenções do homem, seja ele digital, papel impresso, ou mesmo no formato mais antigo como em bambu ou em tábuas de barro. É o que possibilitou a sociedade como conhecemos hoje, não apenas no quesito tecnológico como em qualquer âmbito da vida humana. Por este motivo resolvi transmitir o conhecimento por meio do livro. Você pode fazer uso do conteúdo dele para construir vídeos e

áudios, mas peço que ao faze-lo recomende a leitura do mesmo, e que recomende também que seus telespectadores e ouvintes leiam mais. Até o fim da II Guerra o mundo consumia muitos livros, recentemente estamos em um declínio do consumo deste conteúdo, precisamos incentivar todos a aprender e para isto é preciso que consumam não apenas áudios e vídeos, que são de suma importância, mas em verdade são ferramentas complementares! É importante a leitura de livros!

Se você acredita que não tem aptidão para leitura de livros saiba que você está errado! Todos possuem aptidão para a leitura, é o mesmo que dizer que não tem aptidão para assistir ao vídeo ou ouvir música só por ter tido uma péssima experiência nos primeiros contatos com a mídia. Quando eu era criança um irmão conversando com uma psicóloga pediu recomendações de livros para me presentear, sabendo minha idade, gênero e outros fatores ela o recomendou uma edição de “O menino maluquinho” de Ziraldo. Aquele livro abriu minhas portas para a leitura e para o que hoje me tornei! Se você gosta de esportes pegue um livro sobre o tema, considere não apenas o conteúdo como também a disposição do livro, letras grandes podem ajudar ao iniciar o hábito e conversar sobre leitura com amigos também pode ser uma ótima oportunidade de descobrir itens interessantes para explorar neste novo mundo!

Vá a biblioteca pública de sua cidade, toda a biblioteca é em realidade um local de oportunidades gratuitas para aprender qualquer assunto de seu interesse! E nestas bibliotecas sempre haverá um guarda livros ou assistente disposto a lhe falar sobre livros, técnicas de leituras, que vai poder lhe indicar uma leitura boa, algo que seja de seu agrado. Só não compre a ideia de outros, que falam que ler não é bom, que ler é algo do passado! Não siga exemplos de pessoas que dizem que não lêem por serem preguiçosos, que leitura é difícil e sem graça, existem exceções, mas em geral ou é uma pessoa que não teve a orientação das pessoas certas, ou pior ainda, é uma pessoa que realmente não quer o seu bem pois, como vários, falam para você que ler não é bom, mas possuem horário marcado para leitura diária. Não escute péssimos conselhos, como o de Platão, saiba diferenciar o que é bom nas pessoas e o que é

ruim e abstrair o que é bom! No caso de Platão temos ótimas histórias para crescermos como indivíduos, como a metáfora da caverna, ou os aprendizados da Apologia de Sócrates, mas por melhores que sejam suas ideias, por mais importante que seja a pessoa, não siga o que você sabe não ser um bom conselho, como o de ficar sem ler livros!

Ler faz bem a alma e ao corpo! Ler é vida!

Prefácio

Gostaria de começar o livro fazendo uma recomendação! Não fique tão “neurótico” com o assunto de segurança! Não me entenda mal, segurança é de suma importância! A questão é que não adianta guardar um copo de café dentro de um cofre biométrico com várias camadas de aço blindado sendo que em algumas horas o café estará frio! Enquanto você estiver se preocupando com segurança você também estará deixando de desfrutar do sabor e do aroma! Vou retornar a esse assunto posteriormente pois é algo delicado de explicar, no setor de segurança, os verdadeiros profissionais de segurança entendem o conceito, segurança custa recursos valiosos, seja no âmbito financeiro, seja no âmbito tempo, mas profissionais supostamente da área de segurança vão falar que você deve implementar segurança e se não o fizer pode correr riscos extremos! Eles muitas vezes são vendedores enrustidos de profissionais de segurança. Como se sua vida dependesse disto! O que quero dizer é que existem situações e situações, guarde sob sete chaves o que realmente é importante e deixe o que não precisa ser segredo acessível de forma mais simples. O americano Dan Farmer tem uma frase que gosto: “Se segurança fosse tudo o que importa, computadores nunca seriam ligados, quanto mais plugados a uma rede com literalmente milhões de potenciais invasores”. Dito isto vamos ao livro ...

Inicialmente este livro se chamava SSH Key Files Crash Course, mas quando comecei a escrever a parte de descrição do que é um certificado e a história de como chegamos neles percebi que ele ficaria mais interessante se explicasse um pouco mais. Você pode usá-lo como um guia de referência, acessar o “como fazer” e executar, ou você pode utilizá-lo para entender o como fazer e ir além do simples executar, incorporando os conceitos aqui descritos em seu dia-a-dia, seu workflow e nos produtos que construir.

O mundo dos certificados digitais é amplo e cresce a cada dia, não existe hoje um advogado, ou um contador no Brasil que não precise fazer uso de certificados digitais, o mesmo acontece para os empresários de pequenas e médias empresas, imagino que alguns tipos de empresas, como as as MEI (Microempreendedor individual) não sejam obrigadas a fazer uso dos certificados digitais, assim como médicos que em sua maioria ainda não faz uso de certificados, mas em breve esta demanda também vai se estender para estes posto que esta tecnologia é um caminho sem volta e só tende a evoluir.

No mundo da programação e da tecnologia da informação os certificados digitais são uma demanda, usamos continuamente, extensivamente e diariamente certificados digitais, tabmeḿ por isto resolvi mudar este livro, na intenção de explicar um pouco mais os arquivos de certificados digitais voltados para identificação do programador, porém aproveitando o assunto resolvi dar “uma palhinha a mais” e explicar um pouco sobre certificados em um modo geral, desta forma o programador vai poder entender melhor sobre o assunto, também irei agregar alguns comandos de referência para uso diário, com isto espero que o livro se torne um ponto de referência, sempre que o leitor vir a precisar de algum comando ou esclarecimento sobre esses arquivos que tanto nós são importantes.

Introdução

Nesta parte explico conceitos para que o entendimento seja mais completo, além de explicações de como utilizar o livro e alguns sinais ilustrativos utilizados.

Qual a proposta do livro?

Aprender o funcionamento de key files é crucial para a utilização de aplicativos como o github, SFTP e outros e podem fazer uma grande diferença no gerenciamento de servidores com aplicativos como apache e SSH Server, além de serem hoje indispensáveis para administração de sistemas em redes como a da Amazon AWS que força o uso de arquivos de Senha.

O livro se propõem a explicar SSH Personal Key Files, que são arquivos de Chaves Assimétricas utilizados por pessoas para acessar recursos como shell remotos. Considerando que este livro é focado em desenvolvedores de aplicativos de diversas plataformas assim como a importância que aplicativos como Apache, nginx e IIS possuem na vida destes desenvolvedores, espero a posteriori agregar na parte teórica algumas pequenas explicações sobre outros tipos de arquivos de Chaves Assimétricas, como os arquivos utilizados para a geração e instalação de Chaves de SSL em servidores de internet.

A ideia do livro é lhe passar algumas bases para entender como funciona um Key File e deixar uma cola para lembrar dos comandos sempre que for necessário. Se você quer resolver algum problema de imediato vá para o capítulo “CheatSheet e dicas”, se você quer melhorar sua segurança veja o capítulo de “Boas Práticas”, se você quer saber como funciona Chaves Assimétricas e outras tecnologias envolvidas veja o “Conhecimentos Elementares” e dependendo do que está procurando recomendo dar uma olhada no “Links para evolução de estudo”.

Contudo e sobretudo, espero que goste da leitura!

Roteiro de criptomoedas

Eu tenho um outro livro que está no forno, vai demorar um pouco ainda para lançá-lo, mas será sobre criptomoedas. Se você chegou a esta obra referenciado a partir daquele livro este item foi escrito especialmente para você.

Entendo que você quer o caminho curto e direto, então se você quer entender um pouco sobre criptografia para melhorar seus entendimentos sobre criptomoedas recomendo no mínimo a leitura dos capítulos “Conhecimentos Elementares” e do capítulo “Boas Práticas”, serão de grande ajuda para entender o contexto de criptografia, entretanto a leitura da obra como um todo lhe concederá maior conhecimento sobre Chaves Assimétricas como um todo, sem lhe fazer mal nenhum!

Sobre este livro e a coleção

O livro foi criado a partir de meus conhecimentos sobre a ferramenta, de pesquisas que fiz na internet sobre os diversos assuntos aqui tratados e também sobre estudos que fiz com a ferramenta.

O livro faz parte de uma série de livros que pretendo lançar intitulados “Crash Course RSS Series”, uma coleção inspirada na série “Para Leigos” e que objetiva apresentar o mais simples possível e de forma mais completa possível assuntos da área de TI como de várias outras áreas, sempre com o menor custo possível.

O livro estará disponível nos formatos digitais PDF e EPUB, alguns serão totalmente gratuitos outros terão custos de aquisição. Se você teve acesso a este livro de forma gratuíta e ele lhe for útil de alguma forma peço que colabore com a continuidade de meus serviços fazendo alguma doação para qualquer entidade beneficente, sem fins lucrativos e de sua região. Como a ideia dos livros gratuítos é fazer o bem, ao realizar uma doação você me ajudará a realizar este objetivo.

Independente do livro digital ser ou não gratuito ele também estará disponível no formato impresso por meio de parceiros e livrarias online, porém a demanda de impressão irá gerar custos, mesmo a versão online sendo gratuita, os custos de impressão, entrega e outros de qualquer natureza são de responsabilidade da pessoa que adquiriu o livro.

Super usuário e operador

O livro contém ilustrações e exemplos diversos sobre execuções em linhas de comandos, na grande maioria das vezes tais execuções foram construídas em um ambiente Linux, por isso, sempre que ver no livro uma sequência de comandos apresentada a frente do caractere “#” (sustenido)

saíba que aquela sequência de comandos foi realizada como super-usuário, ou seja, quem executou aqueles comandos foi o root do Linux e você precisará deste nível de execução para realizar o procedimento, no caso do Windows você precisará ser administrador no caso do mac precisará ter acesso de usuário raiz. Da mesma forma, se a sequência se inicializar como o caractere “$” (cifrão) quer dizer que tais comandos foram realizados por um usuário comum.

Ícones utilizados no livro

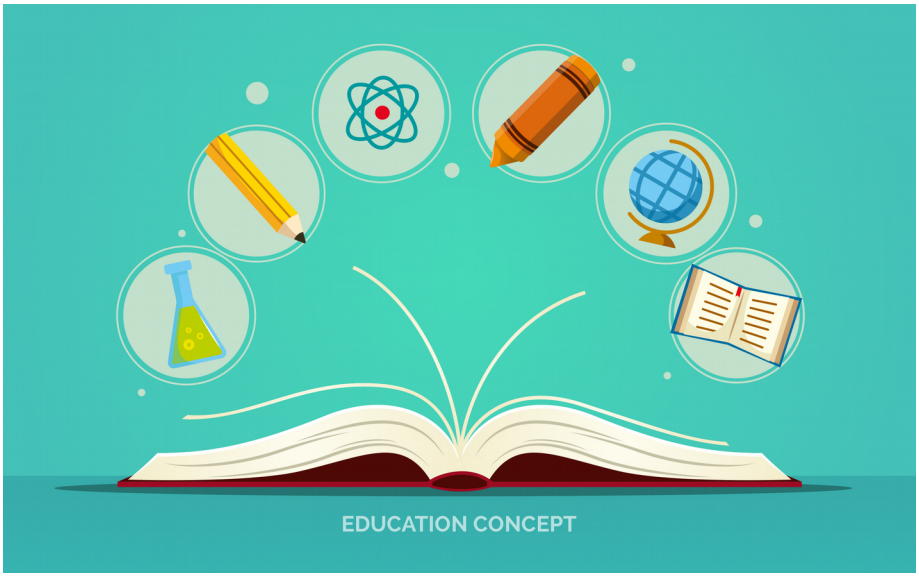
Dica: São pontos de vista que geram um melhor aproveitamento da ferramenta ou uma melhor compreensão do elemento sendo apresentado.

IMPORTANTE: Conterá elementos importantes de lembrar, geralmente essenciais para o uso da ferramenta.

Nota do autor: É utilizado quando o autor faz uma observação e quer agregar algum destaque a esta

Curiosidade: Comentário que não agrega muito mas que agrega alguma informação interessante para o contexto.

Conhecimentos Elementares



Antes de explicar sobre SSH Persoal Key File quero explicar o que são estes arquivos, com isto espero que você tenha mais confiança ao falar sobre o assunto quando precisar tanto aplicar em sua programação quanto explicar para outro amigo. Para isto vamos começar bem lá do começo!

O que é uma Senha?

Chaves de criptografia em um primeira instância nada mais são além de uma Senha, inclusive “Senha” é diretamente entendida e comparada como Chave no âmbito da criptografia, simples assim! Tento não exemplificar casos tão simples, mas dizem que quanto mais diversificado a explicação melhor, considerando que o livro é didático então acredito ser importante colocar exemplos um pouco diferentes, entretanto se você não entender os outros dois parágrafos basta lembrar o que leu no início deste e pular para o item seguinte “Chave criptográfica”, onde o conceito de senha é melhor explicado.

Sua Senha para acessar seu email é sua Chave de Criptografia para acessar o aplicativo de emails. Muitos autores explicam Chaves de Criptografia desta forma pois facilita a compreensão para não iniciados, mas vou fazer o máximo para que entenda como se fosse iniciado nessa magia obscura da criptografia! Enfim, preciso que você devaneie comigo por algumas estórias, imagine que no mesmo conceito da Senha do seu email seu CPF pode ser considerado como uma Senha, é uma sequência de caracteres que passa por um algoritmo de construção para agregar a ele mesmo um numeral que serve de validação e que está associado unicamente e exclusivamente ao seu nome, o numeral não é associado a seu nome por uma elaboração matemática, mas sim por um registro do governo, ainda assim é associado com seu nome. Ocorre que ninguém considera o CPF como Senha pois é uma informação relativamente pública e é neste entendimento que uma Senha começa a se distanciar conceitualmente de uma Chave de Criptografia.

Criptografia é uma técnica utilizada para cifrar um texto, quando você informa ao gmail sua Senha ele utiliza uma técnica criptográfica para cifrar sua Senha e comparar o texto resultante com um texto que ele previamente possui, com isso ele válida que você é você mesmo, concedendo assim acesso a suas informações, que por sua vez não estão encriptadas com sua Senha, e sim com outras técnicas e Senhas do próprio Google, ou seja a Senha não foi utilizada como Chave de

Criptografia, ela é apenas um texto que é criptografado e a validação é feita com uma funcionalidade de comparação dos textos criptografados. Se sua criptografia resultar no mesmo texto que já existia então entende-se que a Senha está válida. No caso do CPF, se você possuir apenas os primeiros 9 dígitos você poderá aplicar a técnica de construção para gerar os últimos 2 dígitos do seu CPF, no exemplo se fosse prática fornecer apenas os 9 primeiros dígitos do CPF seria uma forma muito arcaica, mas ainda assim mais funcional de validar se uma pessoa é quem diz ser. Se você notar você não tem como encontrar os nove primeiros dígitos do CPF a partir dos últimos 2 dígitos, o mesmo funciona na maioria dos mecanismos de Senha utilizados, isso é entendido como criptografia de mão única (one way cryptography, também explicado no item a seguir Chave de Criptografia).

PS: Agora esqueça o CPF, ele não deve ser entendido como uma Senha, ele só foi utilizado por ser um elemento popular de vasto conhecimento público para ilustrar uma Senha de mão única.

Nota do autor: No Brasil o ICP-Brasil (falo mais sobre ele na parte “Para que servem?” em “O que é Symmetric Keys”) entende vários mecanismos de Senhas como sendo uma “Assinatura Eletrônica” que é diferente de uma “Assinatura Digital”, uma Assinatura Digital faz uso de Chaves Assimétricas, qualquer outro mecanismo de identificação é conhecido como “Assinatura Eletrônica”.

O que é Chave criptográfica?

Uma Senha, como na ilustração do aplicativo de email, é considerada uma chave, pois é utilizada para “abrir” recursos da ferramenta, ou seja, para ter acesso a ferramenta, mas por natureza uma Chave Criptográfica é utilizada para criptografar um contexto que será utilizado, no caso do aplicativo de email o contexto não é utilizado, a Senha é a mensagem em

sí. No caso da Senha de aplicações em geral, como a da ilustração do gmail, é utilizado um processo chamado função de hash criptografado que é uma funcionalidade de mão única, após gerar a frase resultante não há como ter a frase construtora, por isso chamamos de “Encriptação de Mão Única”, ou em inglês “One-way Encryption”, a única forma de obter isso é por força bruta, no modelo “tentativa e erro”, fazendo uso da mesma funcionalidade de criptografia, até que se encontre a frase original. por isso difere-se de uma Chave Criptográfica, ou seja, uma Senha sempre é uma Chave, pode até ser chamada de Chave “DE” Criptografia, mas nem sempre é uma Chave Criptográfica.

Nota do autor: Falo muito neste livro sobre “força bruta”, mas entenda que existem várias formas de colocar um pouco mais de inteligência na força, no caso de Chaves Assimétricas é a fatoração de módulos. Quando falamos de Chaves Simétricas estamos falando de Sequência Numeral (1, depois 2, depois 3 e assim por diante), quando falamos de Chaves Assimétricas falamos de Fatoração de Módulos de Números Inteiros. Comento um pouco mais sobre isso no item “Quanto Maior Melhor, só que não!” no capítulo “Boas Práticas”

Em termos jurídicos brasileiros temos o que conhecemos como Assinatura Eletrônica que é um termo que qualifica vários mecanismos de autenticação que geram documentos virtuais com validade jurídica, “Assinatura Eletrônica” é o gênero, é algo amplo que identifica Chaves de Criptografia e Chaves Criptográficas, já uma Assinatura Digital por sua vez é um termo que envolve uma Chave Criptográfica.

Mas se a Senha não é uma Chave Criptográfica, o que é uma Chave Criptográfica? Uma Chave Criptográfica é uma Senha utilizada em conjunto com um texto para cifrar ou decifrar tal texto e o texto por sua vez é utilizado também como parte desta Senha. Vamos elaborar um exemplo prático para tentar elucidar, digamos que você queria criptografar a frase “Ola Mundo” com a chave “MinhaChave”, você iria combinar

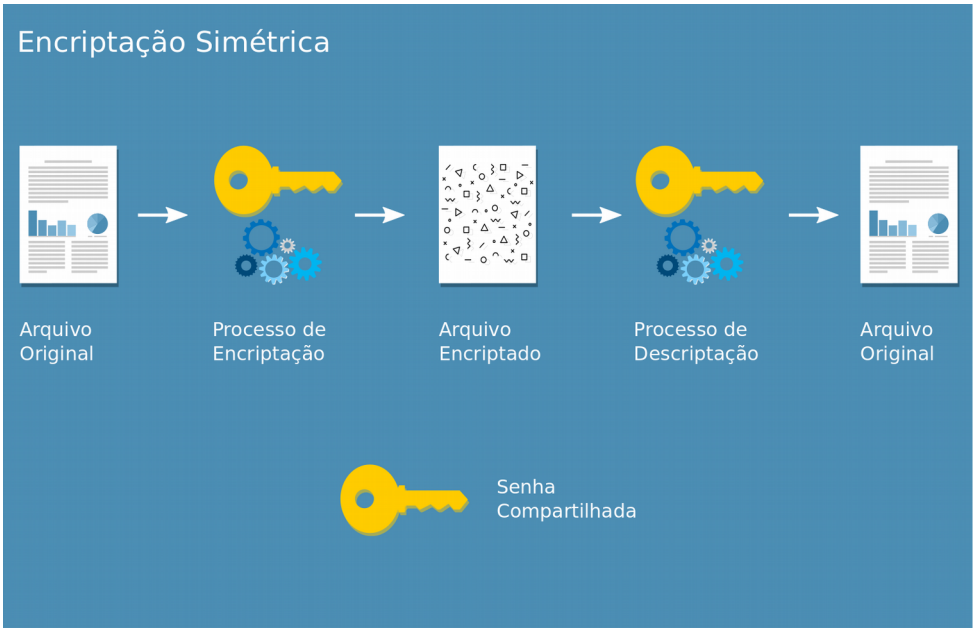
ambas as frases “MinhaChave” e “Ola Mundo” em um único texto, que seria “MinhaChaveOla Mundo”, este texto por sua vez seria utilizado em um algoritmo de encriptação para gerar um texto encriptado, digamos que nosso algoritmo pegue cada uma das letras e troque ela pela letra seguinte no alfabeto, então “MinhaChaveOla Mundo” geraria a frase “NjoibDibxfPmb Nvoep” desse modo se você fornecer qualquer outra Senha a decifração geraria um texto completamente diferente.

Na vida real o processo é um pouco mais complexo que isso, poderíamos fazer um algoritmo mais útil fazendo algum processo diferente, como p. ex., as letras da Senha encontradas no contexto sendo encriptado fazem o numeral ser acrescido, já as letras que não estão na Senha são decrescida, fazendo o “a” do “MinhaChave” ser um “z” sempre que encontrado em “Olź Mundo”, mas isto faria a explicação extrapolar em muito o que queremos explicar aqui que é apenas o conceito base, porém dentro deste contexto sua Senha é utilizada para gerar um hash, um texto intermediário, também conhecido como Impressão Digital (fingerprint, ver mais sobre no item “Para verificar o fingerprint” neste mesmo capítulo) e essa Impressão Digital por sua vez passa por um algoritmo matemático para criptografar ou descriptografar a mensagem.

A Chave Criptográfica é utilizada para criptografar e descriptografar um contexto e é composta pela Senha e pelo Algoritmo utilizado para gerar o Hash e com todos esses conceitos em mente podemos finalmente chegar ao mecanismo que faz uso de Chaves Pessoais de Criptografia!

IMPORTANTE: Uma das grandes diferenças entre “Encriptação” e “Hashing” é que a encriptação sempre é “two-ways”, ou seja, o que é criptografado pode ser descriptografado, já um hash é sempre “one-way”, ou seja, após a criação de um hash o mesmo não pode retornar a frase original. Também é comum obter o mesmo hash a partir de frases diferentes, isto é conhecido como “colisão” (collision).

O que é Symmetric Keys



Como Chaves Simétricas funcionam

Apenas por diversão resolvi descrever melhor o que é uma Chave Simétrica, porém ao faze-lo percebi que facilitaria o entendimento sobre Chaves Assimétricas, por isto recomendo ler esta parte do livro se você está tentando entender Chaves de Criptografia.

Entende-se como Chave Simétrica pois o Arquivo de Chave utilizado para criptografar é o mesma utilizado para descriptografar um contexto.

Como dito, uma Chave Simétrica é um texto utilizado para encriptar outro texto que passa a ser entendido como “texto criptografado”, porém diferente de uma Senha simples que é criada de forma aleatória e pode ter seu tamanho variado, uma Chave Simétrica faz uso de um algoritmo matemático que constroi uma sequências alfanumérica. Existem vários algoritmos que geram vários tipos de sequências alfanuméricas e tais

sequências tem seu tamanho medido em bits, não de bytes, por isto são sempre múltiplos de oito. Estas sequências alfanuméricas podem ter diversos tamanhos, geralmente a menor sequência é composta por 8 caracteres (64 bits), porém também é comum sequência de 16 caracteres (128 bits), 32 caracteres (256 bits), 64 ou mesmo 512 caracteres (4096 bits). Tais sequência são conhecidas como hashes (no plural e “hash” no singular).

Nota do Autor: É possível a construção de um algoritmo que gere saída a com a quantidade de bits diferente de 8 bits, porém por diversos motivos isso não é prático para o ambiente onde o número 8 é considerado fator de armazenamento, ou seja, se o arquivo resultante tiver menos que 8 bits tais bits serão utilizados de qualquer forma, se tiver mais que 8 e menos que 16, 16 bits serão sempre utilizados para armazenar o arquivo de saída, então é pouco prático a construção de tais algoritmos.

Dica: Vale dizer que os hashes são um mecanismo utilizado para muitas coisas além de encriptar contextos, como p. ex., detecção de erros em comunicação de redes ou detecção de alteração de dados em processamento de alta escala, mas no contexto de criptografia um hash pode vir a fazer o papel de uma Senha.

Algoritmos de Chaves Simétricas.

A partir do momento em que passamos a utilizar Chaves Criptográficas nós deparamos com vários algoritmos e passamos a conhecê-los melhor, para adiantar o assunto achei interessante deixar aqui uma lista de exemplos de algoritmos de criptografia simétrica:

- AES (Advanced Encryption Standard): Um padrão utilizado por muitas pessoas e empresas em todo o mundo, foi selecionado em

um concurso após identificar que o DES era muito vagaroso em algumas situações e muito vulnerável a ataques de força bruta;

- DES (Data Encryption Standard): Muito utilizado na década de 70, considerado inseguro para os padrões modernos e tido como descontinuado (deprecated) em 2003, porém ainda utilizado em várias situações;
- IDEA (International Data Encryption Algorithm): Algoritmo utilizado por muitas tecnologias, como a tecnologia de PGP (Pretty Good Privacy) e que se tornou público a partir de 2012;
- Blowfish: Considerado por alguns como um dos mais seguros algoritmos para encriptação de contextos. É relativamente novo, por este motivo poucas aplicações o utilizam e por consequência não tem tantos estudos avaliando a qualidade efetiva de sua segurança;
- Twofish: Foi criado considerando algumas técnicas do Blowfish porém implementando algumas estruturas utilizadas em algoritmos como o DES;

Curiosidade: Teoricamente computadores são incapazes de implementar aleatoriedade, por isto em computação o termo pseudo-randômico é mais apropriado. Um número “aleatório” é sempre gerado a partir de uma data (dia, mês, ano, dia, hora, segundo, milissegundo e quantidade de ciclos) ou algo do tipo, existem vários algoritmos para implementar aleatoriedade, porém sempre existe alguma previsibilidade de qual número será o resultado da aleatoriedade.

Como Chaves Simétricas funcionam?

Os algoritmos de Chave Simétrica fazem uso de valores de entrada,

muitas vezes estes valores de entrada são Senhas criadas por humanos, outras vezes fazem uso de elementos aleatórios, como data hora, segundo, milissegundo e vários outros elementos. A partir destes valores de entrada então os algoritmos criam as Chaves Simétricas que por sua vez pode ser utilizada de forma similar a uma Senha para encriptar um contexto.

Também é comum vermos pessoas não técnicas (ou mesmo técnicos) referenciando Arquivos de Chave simplesmente como Senha, vemos em filmes referências como “faz uso de uma Senha de 512 bytes”, esse tipo de situação certamente faz uso de uma Chave Simétrica. O ser humano é capaz de lembrar quantidades altas de frases, a última “Nota do Autor” acima, p. ex., foi construída de forma a possuir exatamente 512 caracteres para exemplificar (entre o “N” de Nota do Autor até o ponto ao final da frase, considerado este inclusive). Não é impossível memorizar exatamente cada palavra, cada espaço, pontuação e caixa alta (formato da letra, se está em maiúscula ou minúscula), porém é preciso um certo tempo e arquivos garantem não apenas velocidade de disponibilidade como também certeza de que tal Senha não seja esquecida. Enfim, este é o motivo de que muitas vezes Chaves Simétricas são referenciadas simplesmente como Senhas.

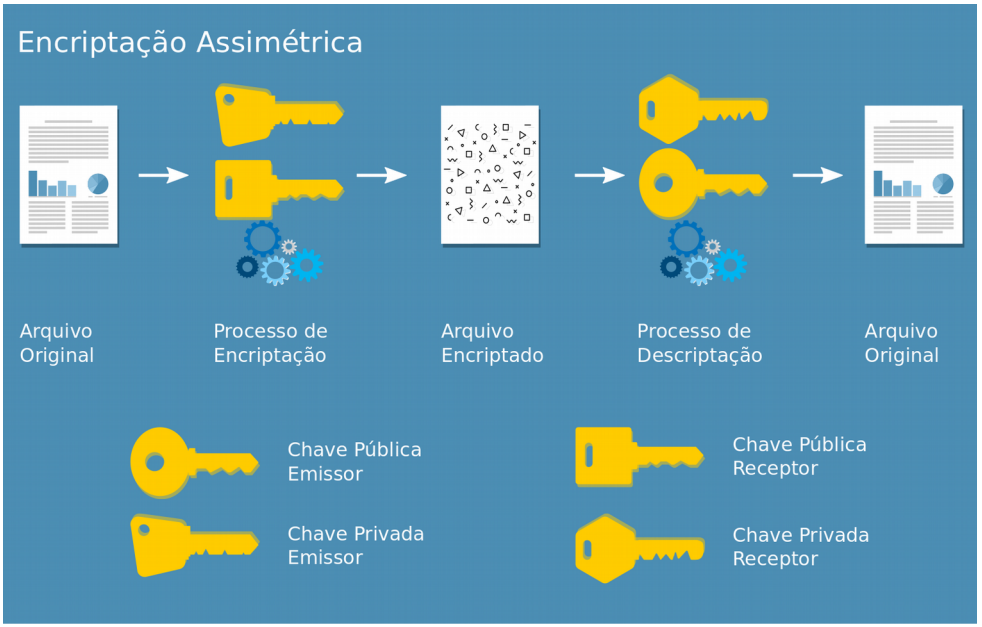
Para que servem?

Chaves simétricas são muito úteis quando o computador precisa de algum nível de automatização ao implementar criptografia, técnicas onde a demanda de um ser humano não é prática. Digo isso pois os cálculos para gerar uma Chave Simétrica são relativamente caros de serem implementados (consomem muitos ciclos do processador), de forma que se é possível receber os caracteres de um ser humano muitas vezes o uso de simples Senhas disponibilizadas por estes são mais práticas, entretanto na eventualidade ou demanda de velocidade, ou mesmo em momentos em que se espera levar mais conforto ao usuário sem que este precise fornecer uma Senha a situação faz com que o uso de Chaves Simétricas seja mais prático.

Algoritmos de Chaves Simétricas são também muito utilizadas para encriptar Senhas, evitando que elas sejam armazenadas ou transmitidas abertamente de forma que qualquer pessoa possa visualizar tal Senha, contudo Chaves Simétricas não impedem ou dificultam ataques conhecidos como “homem no meio” (simplesmente MITM, ou ainda “man in the middle” attacks) pois, digamos que alguém tenha acesso a sua Senha encriptada fazendo uso de algoritmos de Chaves Simétricas, o atacante pode passar a utilizar apenas a Senha no formato encriptado para realizar o ataque, sem mesmo conhecer sua Senha. Espero que esteja claro que a explicação está abstrata demais e o cenário mais uma vez simplificado demais, porém ainda assim é uma situação possível na vida real.

Recentemente o mecanismo de Chaves Simétricas foram utilizados para construir o que chamamos de “blockchain” que é a arquitetura utilizada para a construção de um “open ledger” que por sua vez é um mecanismo de autenticação em cadeia utilizado pela grande maioria das criptomoedas atualmente disponíveis (como o bitcoin) para garantir sua integridade de valor.

O que é Asymmetric Keys



Como Chaves Assimétricas funcionam

Finalmente chegamos ao entendimento necessário para compreender arquivos de Chave de Criptografia!

Chaves Assimétricas é quando você possui uma Chave Pública e uma Chave Privada, elas possuem uma correlação matemática, você consegue obter uma Chave Pública a partir de uma privada, mas você não consegue obter uma Chave Privada a partir de uma Chave Pública, por isto é algo tão legal! É como se você apresentasse sua assinatura de punho a todos, mas só você sabe reproduzir ela efetivamente, que basicamente é como funciona uma assinatura de punho, vale aqui observar que assim como na assinatura de punho é possível duplicar uma assinatura de Chave Assimétrica. Não obstante, como na assinatura de punho, geralmente ainda é preciso muito esforço para que isso aconteça,

de forma que a probabilidade da ocorrência é muito pequena, e quase nula se utilizado em conjunto com outras técnicas de segurança. Em outras palavras, na grande maioria das vezes o uso de Chaves Assimétricas é suficiente para resguardar a integridade e/ou ocultação do contexto.

Curiosidade: Como uma Chave Pública pode ser obtida a partir de uma privada é comum ouvir dizerem que a Chave Pública está dentro da Chave Privada, isto é uma afirmação incorreta, apesar de "matematicamente coerente", sendo que você obtém uma Chave Pública a partir de uma Chave Privada por meio de sua correlação matemática.

Chaves assimétricas também são conhecidas simplesmente como Chaves Públicas posto que geralmente quando encaminhamos uma Chave Assimétrica encaminhamos justamente a Chave Pública, este termos se refere justamente a Chaves Assimétricas em especial quanto fazendo uso de seu plural, "Chaves Públicas".

Muito dos conceitos implementados em Chaves Simétricas são herdados e implementados de forma similar e a grande vantagem é que Chaves Simétricas possibilitam com que você não compartilhe sua Senha com os demais. Lembrando o cenário onde um ataque de homem no meio ocorre, o uso de Chaves Assimétricas se torna mais indicado pelo simples fato de que o atacante precisará inicialmente saber de quem se origina e se destina o contexto capturado no meio do caminho para só depois passar a tentar quebrar as Chaves Privadas, mas conceitualmente não existe outra forma de quebrar uma Chave Assimétrica a não ser por ataque de força bruta, o que torna a ação virtualmente impossível.

Algoritmos de Chaves Simétricas.

Quando falamos de Chaves "Simétricas" é mais fácil falar sobre um algoritmo, Chaves Assimétricas é um pouco diferente pois cada um dos arquivos usa um tipo de algoritmo e existem várias formas de implementar uma Chave Pública a partir de uma Chave Privada. É

possível, por exemplo, a criação de um arquivo padrão RSA a partir de algoritmos DES ou AES privado. Com o propósito de simplificar faremos o mesmo que geralmente ocorre na internet, ao falarmos de algoritmos de Chaves Assimétricas iremos comentar apenas o algoritmo de geração da Chave Privada e neste caso temos as seguintes opções abaixo como sendo as mais populares:

- **RSA**: É um dos algoritmos mais antigos e diz-se que todos os clientes e servidores de SSH reconhecem este algoritmo, por isto seu uso é amplamente difundido e muito aconselhado na internet, todavia sempre seguido pela frase “use frases de 4096 bits” pois também entende-se que sua fragilidade aumenta com o passar dos tempos e espera-se para qualquer momento uma notícia do tipo “encontrado algoritmo de quebra do algoritmo RSA”;
- **DSA**: Não é recomendado seu uso no formato original, infelizmente decorrente desta recomendação poucas pessoas o usam nos dias de hoje, entretanto o formato original não é muito implementado em sistemas modernos. Se você não souber a diferença recomendo uso de outro algoritmo;
- **ECDSA**: (Elliptic Curve Digital Signature Algorithm) Mais moderno que o DSA, sendo considerado como uma variante do mesmo e também desenvolvido pelo governo americano para implementar assinaturas digitais. Nos dias de hoje a maioria dos sistemas já possibilita seu uso. Sempre que utilizando este algoritmo recomenda-se o uso da implementação de 521 bits;
- **ed25519**: Um algoritmo proposto pelos desenvolvedores do OpenSSH quando liberaram a versão 2 do aplicativo principal do grupo de aplicativos, por isto se tornou rápidamente objeto de estudo por muitos, como a versão 2 do aplicativo é uma das mais utilizadas na internet o uso do ed25519 é muito popular. Também é entendido como sendo mais complexo que o RSA e por isto é mais eficiente que ele, ao menos quando comparável a uma chave de aproximadamente 3000 bits de RSA;

Curiosidade: Teoricamente falando a computação quântica torna qualquer método de criptografia digital descritos neste livro ultrapassada, ainda vai demorar muito para que computadores quânticos possam ser utilizados para funcionalidades como quebra de Chaves Assimétricas, porém muitos acreditam que isto deva acontecer em um futuro relativamente próximo (dentro dos próximos 50 anos), mas no momento apenas força bruta quebra algoritmos aqui citados, por isso a recomendação “quanto maior melhor”! (ver o capítulo de Boas Práticas)

Como Chaves Assimétricas funcionam?

Digamos que você irá encaminhar uma frase para seu amigo, seu amigo compartilha com você a Chave Pública dele e você encaminha a ela sua Chave Pública. Você pega a frase que quer encaminhar, cria uma mensagem encriptando ela fazendo uso de sua Chave Privada e da Chave Pública dele e encaminha tal mensagem a ele. Seu amigo ao receber a mensagem irá utilizar a sua Chave Pública e a Chave Privada dele para descriptografar a mensagem e ter acesso a frase. Nenhuma outra composição irá possibilitar a descriptografia da mensagem, apenas a junção de sua Chave Pública com a Chave Privada dele. Desta forma nenhuma Chave Simétrica é trocada e a Chave Pública pode ser encaminhada de forma aberta através de conexões simples.

Nota do autor: Alguns consideram que o uso de Chaves Assimétricas tem um elemento negativo, uma vez encriptada a mensagem só pode ser descriptografada fazendo uso da Chave Privada de seu destinatário, ou seja, no nosso exemplo mais recente você não poderia desfazer a criptografia criada ao preparar a mensagem para encaminhar ao seu amigo após fazer uso da Chave Pública dele, apenas ele poderá abrir a mensagem, sendo este o justo propósito de Chaves Assimétricas isso não é

exatamente entendido como um elemento negativo, apenas como um fato.

Quando você navega na internet em um ambiente encriptado, como um protocolo HTTPS ou SFTP seu computador solicita informações do servidor que lhe entrega um arquivo de Certificado Digital contendo a Parte Pública da Chave Assimétrica dele, assim como uma outra Chave Assimétrica de uma Autoridade Certificadora que emitiu o certificado, O certificado da Autoridade Certificadora é validado a partir de vários certificados que você possui em seu computador, seu computador pode vir a solicitar informações diretamente a Autoridade Certificadora para validar seu Certificado Digital, mas geralmente eles simplesmente aceitam que seu certificado está correto e completam o que chamamos de handshake! Handshake (“Aperto de Mão”) é justamente o processo que acabei de descrever, acontece quando duas ou mais máquinas se comunicam, trocam assinaturas e a partir disto passam a se perceber como originais (ser quem se diz ser) e verdadeiras (estar a disposição naquele momento). O handshake pode ser feito de diversas outras formas, mas ao utilizarmos Chaves Assimétricas agregamos mais segurança na validação da originalidade das máquinas, principalmente quando fazemos uso de um CA (Autoridade Certificadora).

Caso queira entender melhor esta parte de comunicação entre servidores e o protocolos HTTPS veja o capítulo “Certificados para Servidores de Página” onde explico um pouco mais sobre como instalar e como funcionam Certificados de Segurança em servidores de internet.

Para que servem?

Sendo direto Chaves Assimétricas tem as seguintes funções: Autenticação ou identificação de indivíduo ou máquina (ex: uso de chave de segurança quando utilizamos ferramentas como o github.com); Garantir privacidade de comunicação entre um ou mais pares (ex: quando fazemos uso de SSH ou SFTP); Encriptar contextos (ex: Utilizado aplicativos como o PGP Zip); e Garantir a integridade de um contexto (ex: Assinando digitalmente um documento).

As Chaves Públicas são utilizadas por diversas entidades e sistemas. No Brasil temos, p. ex., as assinaturas digitais normatizadas por mecanismos de leis do governo federal conhecidos como “lei do ICP-Brasil” (Infra-estrutura de Chaves Públicas do Brasil), tal lei determina estrutura de aceitação e uso dos arquivos assim como modelos de arquivos específicos para pessoas físicas e jurídicas. O uso de Chaves Assimétricas foi escolhido (como em vários outros países) pois você pode marcar o arquivo gerado com um hash construído a partir de seu conteúdo binário com sua Chave Privada e agregar neste sua Chave Pública. O ICP-Brasil determina 2 séries de certificados digitais e 4 tipos de assinaturas para cada série, sendo “Série S” e “Série A” onde temos A1, S1, A2 e S2 com a geração de arquivos com 1024 bits similares aos que estamos estudando aqui neste livro, e os A3, S3, A4 e S4 que são dependentes de hardware para seu uso.

Nota do Autor: Nos dias de hoje implementar assinatura digital não é tão incomum ou difícil, existem várias bibliotecas específicas para isto, formatos como PDF e DOF, p. ex., possuem até campos específicos para armazenar a assinatura digital. O que o pessoal faz é executar a funcionalidade de assinatura e agregar na parte visível do documento algum selo indicando que este foi assinado. Se precisar fazer algo assim recomendo que disponibilize no selo o fingerprint do arquivo utilizado para a assinatura, com isto a pessoa de posse do documento pode verificar de forma mais simples a assinatura do documento.

Exemplo de selo de certificado da empresa Comodo

Como já explicado a tecnologia de Chaves Assimétricas possibilita com que ataques de “man in the middle” se tornem obsoletas, mesmo que o atacante tenha acesso a sua Chave Pública ele não pode decifrar as mensagens posto que ele não possui a Chave Privada de nenhuma das partes, por este motivo Chaves Assimétricas são utilizadas em tunelamento de SSL por meio de protocolos como os conhecidos HTTPS, SSH e SFTP. Ou seja, nos dias de hoje esta tecnologia é o que basicamente mantém a internet funcionando com segurança e no formato que conhecemos!

No caso de um programador o arquivo serve para lhe identificar no github, lhe conceder acesso a algum recurso como um shell de uma máquina virtual, no computador de seu amigo, ou em algum servidor.

Quando configuramos uma Chave Pública no servidor nós evitamos de ficar digitando Senhas toda hora que vamos acessar algum recurso naquela máquina, a primeira vez que tive acesso a este recurso achei desnecessário pois não percebia quanto tempo perdia digitando Senhas, hoje em dia simplesmente entraria em desespero vendo minha produtividade caso precisasse abrir mão da tecnologia de Chaves Assimétricas.

Vamos entender um pouco mais como funciona o processo. Considerando o acesso a um servidor por meio de SSH nós salvamos no

arquivo “authorized_keys” do usuário que desejamos acessar a Parte Pública da Chave Assimétrica no servidor destino, o arquivo authorized_keys deve estar dentro de um diretório com nome “.ssh” diretamente no home do usuário, caso o usuário não possua o diretório você pode cria-lo normalmente, ele só precisa obrigatoriamente ter nível de acesso “0700” para que o acesso funcione, o arquivo authorized_keys também é um arquivo simples que pode ser criado em um editor qualquer como o VIM, este por sua vez deve ter autoridade “0600” para que tudo ocorra normalmente. Após disponibilizar a Chave Pública no arquivo basta solicitar o acesso por meio de um cliente de SSH ao servidor apontando no cliente de SSH sua Chave Privada, o servidor irá validar o conteúdo de sua Chave Privada com o conteúdo público em seus arquivos, ele não irá pedir a Senha do usuário apenas lhe concederá acesso a seus recursos, desta forma você pode distribuir este arquivo para diversos servidores e usuário, evitando assim a demanda de conhecer as Senhas de cada um dos usuários envolvidos. O arquivo aqui envolvido pode ou não ter Senha, porém quem irá lhe perguntar a Senha é seu computador local, antes mesmo de encaminhar ela ao servidor, contudo você irá guardar apenas uma Senha.

Dica: Existe um comando para copiar os arquivos públicos no authorized_keys, veja no item “Como copiar o Arquivo de Chaves para o servidor” do caítulo “CheatSheet e dicas” como fazer.

Considerando a Chave Assimétrica de nome “teste” vemos logo abaixo o processo como sendo executado no servidor, diretamente no usuário “sample”. Este procedimento é conhecido como “instalação da Chave Pública no servidor”. No exemplo acessamos a máquina uma primeira vez utilizando a Senha do usuário “sample” para isto.

Algo importante de acrescentar ao entendimento do funcionamento é o arquivo “known_hosts” que guarda a Parte Pública da chave dos

servidores que você já visitou no passado, esse arquivo é utilizado para saber se uma chave sendo recebida é a mesma que foi recebida no passado. Para saber um pouco mais sobre known_hosts veja “Como saber se já acessei um servidor” e “Como apagar a Chave Pública de um servidor que visitei?” no capítulo “CheatSheet e dicas”.

Note que os comandos utilizados são todos básicos e sempre utilizados com caracteres simples, sem grandes elaborações, este é outra grande diferencial de Chaves Assimétricas, são complexas de serem quebradas, mas seu uso no dia-a-dia é algo extremamente simples e direto para qualquer nível de usuário.

```
$ ssh sample@servidor.com.br
sample@servidor.com.br's password:
Welcome to Linux Mint 18.1 Serena (GNU/Linux 4.4.0-165-generic x86_64)

 * Documentation:  https://www.linuxmint.com
Last login: Wed Sep 25 10:58:31 2019
$ cd
$ mkdir .ssh
$ chmod 0700 .ssh
$ cd .ssh
$ echo "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIFq3CZplP7ocPYvmKUg1wW5j/1G+rXFq
DzOLUjESagjb ricardo@ferro-n2" > authorized_keys
$ chmod 0600 authorized_keys
$ ls -lah
total 12K
drwx------  2 sample sample 4,0K Oct 16 22:09 .
drwxr-xr-x 83 sample sample 4,0K Oct 16 22:08 ..
-rw-------  1 sample sample   98 Oct 16 22:09 authorized_keys
```

Nota do Autor: Note no exemplo acima o uso de “>” (um símbolo de maior que), caso não saiba isso cria ou sobrescreve o arquivo, ou seja, tome cuidado para não sobrescrever Chaves antigas sem querer.

Ok, agora que nossa Chave Pública está no servidor basta acessarmos ele fazendo uso de nosso aplicativo de SSH, diferente da interação acima aqui iremos disponibilizar a Senha da Chave Assimétrica, não do usuário. Abaixo sequência ilustrativa.

```
$ ssh sample@servidor.com.br -i ~/.ssh/teste
Enter passphrase for key '/home/SeuUsuario/.ssh/teste':
Welcome to Linux Mint 18.1 Serena (GNU/Linux 4.4.0-165-generic x86_64)

 * Documentation:  https://www.linuxmint.com
Last login: Wed Sep 25 10:50:21 2019
$ logout
Connection to servidor closed.
```

Em um último exemplo vemos que ao demandarmos em um terceiro momento a comunicação com o servidor2.com.br em um usuário em que a Chave Pública já estava previamente instalada a Senha nem será mais solicitada pois, se seu shell estiver corretamente configurado, a Senha só será solicitada novamente caso você reinicie sua máquina ou efetue logout de seu ambiente.

```
$ ssh sample@servidor2.com.br -i ~/.ssh/teste
Welcome to Linux Mint 18.1 Serena (GNU/Linux 4.4.0-165-generic x86_64)

 * Documentation:  https://www.linuxmint.com
Last login: Wed Sep 25 10:58:31 2019
```

```
$ logout
Connection to servidor2 closed.
```

Imagine quantas vezes você digita Senhas no decorrer de um dia, não apenas para acessar servidores como também recursos como o github e outros. Agora perceba que você pode encaminhar o arquivo de Chave Privada para um colega e conceder acesso a ele fazendo uso de uma Senha, caso você queira trocar a Senha do servidor você pode encaminhar para ele outro arquivo de Chave Privada, configurar no servidor uma nova Chave Pública descartando a antiga e manter a mesma Senha para ele. Você pode distribuir um arquivo de Chaves para uma equipe e todos fazerem uso dela, quando um integrante sair da equipe basta alterar e distribuir uma nova chave, sem precisar notificar nova Senha.

O uso de Chaves Assimétricas é tão comum que o termo “SSH Personal Key Files” (ou key pairs) foi cunhado, se tornou comum em buscas do google e originou o título deste livro, “SSH Personal Key Files” são Chaves Assimétricas criadas para uso particular ou por um grupo de pessoas para recursos como o SSH, outro termo “abrasileirado” seria “Chaves de Segurança para Uso Particular”.

O que são Arquivos de Chaves? (Extensões de arquivos e seus significados)

Conheço dois tipos de Arquivos de Chaves, Chaves SSH e Certificados SSL.

Quando falamos de “Chaves SSH” estamos falando de arquivos de Chaves Assimétricas geradas para uso em processos de autenticação por meio do protocolo SSH. São arquivos mais simples que fazem uso dos menos mecanismos de autenticação do que um Certificado SSL, porém quanto mais o tempo passa mais mecanismos utilizados exclusivamente em Certificados SSL são incorporados para uso em Chaves SSH, como é o exemplo do X.509 que era exclusivamente utilizado em Certificados

SSL e que nas versões mais modernas de programas já são incorporados em Chaves SSH.

Quando falamos de servidores e nos referimos a Arquivos de Chaves estamos falando de Certificados SSL, que costumeiramente ouvimos falar que deveriam ser chamados de Certificados X.509, mas que nos dias atuais já seria um termo confuso posto que Chaves SSH já fazem também uso do padrão de armazenamento X.509.

SSL quer dizer “Secure Socket Layer” (Camada de Soquete Seguro) e determina uma série de regras para assegurar que uma comunicação entre um cliente e um servidor é segura. Nos dias de hoje, apesar do nome continuar sendo amplamente utilizado o SSL é raramente implementado, seu sucessor, o TLS (Transport Layer Security, ou Segurança de Camada de Transporte) é quem efetivamente está em operação na grande maioria dos servidores, no entanto ainda hoje o termo “Certificados SSL” são utilizados para determinar os Arquivos de Chaves X.509 utilizados nas comunicações de TLS.

Os Arquivos de Chaves, seja ele uma Chave de SSH ou um Certificado SSL, nada mais são além de simples arquivos que possuem conteúdo variado. Alguns são arquivos binários, como os arquivos pfx, mas na maioria os arquivos são de texto simples, como os arquivos .pub, alguns possuem algum tipo de codificação, como o pem que faz uso de base64, e ainda assim são de texto simples contendo apenas números, letras e alguns sinais como o “=” (igual).

Existem vários tipos de arquivos de certificados, é praticamente padrão inclusive que um arquivo de Chave Privada em um ambiente Unix-Like não possua nenhuma extensão. É importante dizer que quando falamos de Arquivos de Chaves as extensões de arquivos são praticamente que ignoradas pois o importante é o conteúdo e não seu nome. Entretanto nós como bons usuários que somos sempre recorremos a subterfúgios como extensões de arquivos para facilitar nossa vida, por isso abaixo uma lista de extensões de arquivos importante para o contexto que acredito ser bem consistente, completa e útil, especialmente para novatos no assunto.

- **pub**: O nome da extensão é diminuitivo da palavra “público”. Geralmente conterá a Parte Pública de uma Chave Assimétrica;
- **pem** (**P**rivacy **E**nhanced **M**ail): Pode conter apenas uma Chave Pública ou um certificado completo incluindo as Chaves Públicas e certificados de Autoridades Certificadoras. Contém uma chave de certificado codificada em uma string de base64. É o formato que a Amazon AWS disponibiliza para download quando você cria um “key-pair”; Chaves do tipo PEM também são utilizadas pelo cliente de FTP Filezilla;
- **ppk** (Putty Private Key): Existe um programa Windows muito popular conhecido como PuTTY (putty), ele é um cliente de SSH, junto com este programa existe o PuTTYgen que gera arquivos ppk que é um formato proprietário deste programa, diferente do padrão Unix-Like este arquivo contém tanto a Chave Pública quanto a Chave Privada. Chaves do tipo PPK são utilizadas por ele e podem ser convertidas para uso em outros programas;
- **key**: Pode ser qualquer tipo de chave, mas geralmente é uma Chave Privada. Pode conter tanto uma chave no formato base64 quanto no formato binário;
- **csr** (Certificate Signing Request): É um arquivo utilizado para solicitar a uma Autoridade Certificadora (CA - Certificate Authority) a criação de um arquivo de certificado. Contém várias informações encapsuladas, dentre elas o conteúdo de uma Chave Pública. O tamanho do arquivo CSR também varia de acordo com o algoritmo utilizado, neste caso a maioria das Autoridades Certificadoras solicitam que o arquivo seja criado no formato RSA de no mínimo 2048 bits, eu mais uma vez recomendo o uso de 4096 bits;
- **crt**: Esta extensão ganhou este nome pois CRT é o diminutivo de “certificate”, o arquivo é gerado por uma CA (Autoridade Certificadora) e contém o certificado emitido pela CA sobre os dados recebidos no CSR. É comum que este tipo de extensão

seja trocada por “.cert” ou “.cer”, em geral é muito similar a “.pem”, todavia, além de arquivos de extensão “.crt” serem entendidos pelo ambiente Windows como arquivos de certificados este geralmente é a extensão utilizadas por empresas como certisign na geração de seus arquivos de certificado para servidores de HTTPS. Este arquivo é utilizado para disponibilizar a funcionalidade HTTPS em servidores como Apache, nginx e ISS. Também são conhecidos como “chained” pois está “chaveado” diretamente a Autoridade Certificadora;

- **ca-bundle**: Também disponibilizado pela CA (Autoridade Certificadora), contém certificados intermediários e raiz da autoridade certificadora utilizada para assinar o CSR e gerar o CRT. Este arquivo deve ser concatenado ao CRT ao configurar um servidor de HTTPS;
- **ca** (Certificate Authority): Contém o arquivo de certificado de uma Autoridade Certificadora. Atualmente todos os clientes de HTTPS já vem com uma quantidade alta de arquivos de CA, porém vez por outra alguma entidade gera seu próprio CA, como p. ex., a Visa que gera um CA próprio para alguns produtos e vez por outra precisamos atualizar em nossos sistemas, entretanto se você utiliza o webservice este procedimento não é necessário pois neste caso a Visa faz uso de um certificado previamente instalado tanto em seu servidor quanto em seus clientes. Alguns possuem o hábito de agregar o certificado da CA junto com o CRT e o bundle ao configurar um servidor de páginas, contudo isto é improdutivo posto que tais certificados já estão nos clientes, apenas aumenta o tamanho do arquivo a ser servido ao cliente;
- **pfx (ou p12)**: Considerando que esta extensão de arquivos é o formato mais utilizado para a emissão de Certificados Digitais brasileiros padrão A1 achei uma boa adicioná-lo nesta lista. Este formato também é utilizado por máquinas Windows para

exportar certificados entre máquinas. É um arquivo binário que pode contér diversas informações e múltiplas Chaves em um único arquivo. Irei elaborar mais sobre o formato no item “Certificado de Assinatura de Objetos” em “Quais os tipos de assinatura” logo a seguir;

Existem outros tipos de arquivos que não estão listados acima, como o p7b, p7c e outros.

Dica: É comum encontrarmos arquivos que não possuem extensão nenhuma, em especial no ambiente Unix-Like, se fossemos aplicar extensões em tais arquivos é provável que a mais apropriada seria a extensão “.key”, entretanto este tipo de extensão chama a atenção, por este motivo é comum mantermos tais arquivos sem extensão nenhuma.

Autoridades Certificadoras são de suma importância para Arquivos SSL pois garantem a confiabilidade, já para arquivos de Chaves Assimétricas para uso em ferramentas como SSH e github.com elas não são uitilizadas.

No geral, considerando apenas SSH Personal Key Files as extensões realmente importantes quando estamos em um ambiente Unix-Like (Linux, FreeBSD e MacOS) seria os arquivos sem extensões e a extensão “.pub”, já em ambientes Windows-Like (Microsoft Windows, FreeDOS e outros) as extensões “.key” e “.pem” são as que você irá utilizar.

Quais os tipos de assinaturas

Além das extensões de arquivos também vale entender os tipos de assinatura. No mundo da criptografia existe o que conhecemos como "Public Key Cryptography Standards" (PKCS, ou Padronização de Criptografia de Chaves Públicas), que são regras determinadas pela RSA Security LLC, uma das empresas mais importantes quando se fala de

criptografia pois detém várias patentes de algoritmos e padrões de segurança em uso desde a década de 70.

Curiosidade: RSA vem de Rivest–Shamir–Adleman que é as iniciais de três técnicos que inventaram um dos primeiros algoritmos de criptografia de Chave Assimétrica, “Ron Rivest”, “Adi Shamir” e “Leonard Adleman”. Também são técnicos altamente competentes e conhecer a história de vida deles é conhecer um pouco mais sobre criptografia.

Os PKCSs surgiram na década de 90 para promover o uso dos produtos da RSA, hoje são 15 e não são de uso público, mas como são amplamente utilizados estão sendo considerados como um padrão de internet, por isto empresas como a IETF (Internet Engineering Task Force) e o grupo de trabalho PKIX (Public-Key Infrastructure X.509) são obrigados a emitir documentações muito similares que por sua vez são de uso público.

Eles são de suma importância pois determinam os formatos dos arquivos de Chaves Assimétricas que vimos anteriormente e a partir destes formatos surgem sua aplicação, abaixo os exemplos mais populares.

Certificado de Autoridades de Certificação

Este é um pouco mais conhecido, são os certificados que garantem que as entidades emissoras de certificados são quem dizem ser. São conhecidos como CA Certificates (extensão de arquivo “.ca” descrita)

Como dito nos dias de hoje os computadores já vem com vários certificados de autoridades certificadoras, cada aplicativo possui sua lista própria e seu computador, dependendo da instalação do sistema operacional, também possui uma que deve variar por volta de 150 entidades.

Apesar de serem bem conhecidas são poucas pessoas que precisam de lidar com estes arquivos, eu tomei conhecimento ao lidar com um bug de

configuração de um antigo parceiro que não configurou corretamente o servidor e precisamos forçar o aplicativo a carregar o certificado do servidor por programação, no decorrer dos estudos tive que entender e garantir que o problema não era com o certificado do cliente.

Certificado SSL de Servidor

Do termo inglês “Server SSL Certificate”, são os certificados gerados pelas CAs para validar a identidade de um servidor, por isto ao gerar um certificado SSL você precisa fornecer a CA um arquivo CSR ao pagar por seu certificado. Com o CSR a CA possui os dados do servidor e de seu proprietário, como localidade do responsável pelo domínio, nome da pessoa física ou jurídica responsável pelo domínio, enfim, todas as informações, e com elas gerar o CRT e o CA-Bundle, também explicados anteriormente.

Esses arquivos devem ser copiados para o servidor e instalados no aplicativo que deseja utilizá-lo, como p. ex., apache, nginx e outros, eles não são arquivos que disponibilizam apenas uma comunicação sigilosa mas também possibilitam com que o servidor seja validado como sendo verdadeiro. Explico mais sobre isso no capítulo “Certificados para Servidores de Página”

Certificados SSL de Cliente

Possuem uma estrutura de arquivos muito similar aos certificados de servidores, todavia são utilizados arquivos “.pfx” para instalação nos navegadores. Quando navegamos em um site por meio do protocolo HTTPS geralmente nosso navegador gera automaticamente um par de Chaves Assimétricas (uma pública e uma privada), segundo o que sei é gerado uma chave para cada site a partir da Chave Pública do Certificado SSL do Servidor, entretanto é possível fazer uso de um par de Chaves Assimétricas previamente criados, com isto é possível, p. ex., criar certificados para um servidor local que só aceitaria a navegação por meio de um cliente que possua um certificado previamente conhecido por ela.

PS: Para desenvolvedores que fazem uso de protocolos HTTPS para navegar no localhost é mais fácil configurar o navegador para aceitar certificados provenientes de localhost.

Certificados S/MIME

De “Secure/Multipurpose Internet Mail Extensions”.

São os certificados utilizados para validar e transferir emails, ou seja, além de garantir que a mensagem foi encaminhada em sigilo, que seu conteúdo chegou por completo e corretamente também garante a pessoa que emitiu a mensagem.

Antigamente era pouco utilizado, mas nos dias de hoje vários servidores já implementam a tecnologia como padrão. Alguns chegaram a acreditar que esta tecnologia seria responsável pelo fim dos spammers, é improvável que ela consiga um dia fazer isso pois spam é um email indesejado e ela não impede que você receba os emails daquele parente chato, só vai garantir que a chatice tem nome!

Certificados de Assinatura de Objetos

Do inglês Object-Signing certificates, usado para assinar objetos. No Brasil os certificados digitais do tipo A1 para emissão de Notas Fiscais Eletrônicas são Certificados de Assinatura de Objetos que fazem uso do PKCS#12.

Os Certificados de Assinatura de Objetos são amplamente utilizados em várias funcionalidades, tanto no Windows quanto no Linux e geralmente possuem a extensão “pfx”.

Certificados de Usuário

São arquivos de certificados utilizados para identificar usuários, podem ser utilizados para vários outros objetivos, mas geralmente são aplicados apenas para identificação.

Certificados de Usuários são os arquivos de Chaves Assimétricas

utilizados como “SSH Personal Key Files” para se identificar ao commitar alterações em repositórios remotos de servidores git, ou para realizar fast login em servidores SSH, ou mesmo melhorar o upload por meio do protocolo FTP fazendo uso de um aplicativo SFTP.

Como programador basicamente “Certificados de Usuário” são seus Arquivos de Chave que você precisa criar para fazer uso no seu dia-a-dia. Você pode criar quantos quiser e associar estes de diversas formas. Você pode ter um arquivo para o github.com e outro para o bitbuket.com, ou usar o mesmo em ambos os ambientes. Você pode criar um arquivo e utilizar este para a vida inteira (o que obviamente não é recomendável, o ideal é recriar todos os seus certificados no mínimo uma vez por ano), ou criar uma tarefa periódica de renovação dos certificados (o que é o correto a ser feito). Entretanto lembre-se, uma vez criado o certificado evite perde-lo pois ele é como uma Senha, mesmo que pense que não o utilize mais, vez por outra pode vir a precisar “lembrar” daquela Senha!

Um pouco de história

Neste momento fica mais fácil explicar o motivo de ter elucidado tantos elementos, Chaves Assimétricas é algo que vai além do simples conceito de Chaves Públicas e seu entendimento a nível mais técnico precisa desse embasamento. Historicamente falando criptografia é um organismo vivo que nos remete ao princípio da história da civilização contemporânea. Sabemos de casos em tempos antigos de situações menos conhecidas como uma que descreve o possível primeiro uso da criptografia, ao menos o primeiro que temos conhecimento, outros incorporadas em pleno conhecimento popular, em documentos como o Kama Sutra ou em encenações e elaborações criadas a partir dos trabalhos de Leonardo Da Vinci através de filmes e livros, ou dos casos, alguns verdadeiros, da máquina Enigma da Alemanha Nazista na Segunda Guerra. A criptografia não é apenas parte de nossa história mas um pilar importante para formar nossa civilização, por isso tantos estudos e investimentos nesta área.

Neste contexto histórico a criptografia gerada a partir do conceito de

Chaves Assimétricas é considerado como uma verdadeira revolução, um marco que ocorreu na década de 70 do século XX (1970) e que muda e molda nossas vidas constantemente.

O conceito de Chaves Assimétricas surgiu para o mundo no texto intitulado “New Directions in Cryptography” (Novas direções em Criptografia) no volume 22 da revista “Transactions on Information Theory”, periódico da instituição IEEE (Institute of Electrical and Electronics Engineers) em 1976 onde os americanos Whitfield Diffie e Martin E. Hellman discutem o desenvolvimento da criptografia moderna que minimiza a demanda de canais seguros para a distribuição de Chaves e Senhas, todavia, sabemos hoje que o governo britânico por meio de sua instituição de inteligência, a GCHQ (Government Communications Headquarters) já possuia desde a década de 60 documentos mencionando o conceito de Chaves Assimétricas e possui desde 1973 um mecanismo que faz uso de algoritmos similares aos apresentados em 1977 por Ron Rivest, Adi Shamir e Leonard Adleman descrevendo o que hoje conhecemos como o algoritmo RSA.

Qual a diferença entre SSH e SSL?

Um SSH Private Key é uma Chave Assimétrica utilizada por pessoas físicas ou robôs para receber acesso a um shell de comandos, seja ele um shell de sistema operacional ou um shell de algum outro aplicativo. O SSH, no caso, disponibiliza acesso ao shell de um sistema operacional, pode ser um Bash, pode ser um sh, ou algum aplicativo que esteja configurado para ser sua porta de entrada ao ambiente do servidor, como o MySQL, o SSH não se limita ao shell do sistema operacional. SSH Private Keys podem ser utilizadas para encriptar uma conexão, mas em geral são utilizadas para identificar o usuário e conceder acesso a informação. O SSL por outro lado é utilizado para confirmar a autenticidade de uma máquina e encriptar uma conexão. O como é feito o uso do SSL será melhor esclarecido no capítulo “Certificados para Servidores de Página”. Já certificados do governo é uma classe diferente de certificado, mais parecido com o SSL, justamente por estes concederem autoridade a seus portadores, todavia é um tipo de

certificado feito para uso pessoal.

Evoluindo nosso descritivo histórico, saindo dos conceitos de criptografia e entrando em ferramentas importantes para nosso estudo temos a criação do protocolo SSH (“Secure Shell”, “Shell Seguro”) que representa a adição de uma camada de segurança extra ocorrida em 1997 no protocolo Telnet originalmente incorporado por meio do SSL (“Secure Sockets Layer”, “Camada de Soquete Seguro”) e atualmente pelo TLS (“Transport Layer Security”, “Segurança de Camada de Transporte”) que tomou mais força no século XXI e hoje é mais popular.

Outro protocolo bastante conhecido de nosso dia-a-dia é o HTTPS, e assim como o SSH que substituiu completamente o Telnet, o HTTPS vem substituindo gradativamente o HTTP por um protocolo que permite uma comunicação segura entre os pares. A diferença é que o SSH substitui um ambiente tido como restrito, já o HTTPS terá uma dificuldade maior para ser substituindo posto que muitos arquitetos de informação não se importam como a distribuição de informações públicas ocorrem. No entanto é importante entender que apesar da informação ser pública o conhecimento de quem consumiu esta informação não deveria ser e é por isto que instituições como o Google fazem um esforço para que todos implementem o HTTPS e deixem de utilizar o HTTP.

Um elemento importante de comentar sobre o HTTPS é que ele não permite cache nem local nem intermediário, ao menos não antes de 2010. Quando você solicita uma página HTTP o servidor recebe a solicitação e começa o streaming do arquivo, alguns hardwares como switches e mais especificamente routers fazem cache do conteúdo de seu arquivo para evitar ficar solicitando ele a toda hora para o servidor, porém em uma conexão encriptada, por mais que ele queira fazer cache, o conteúdo é conceitualmente e efetivamente especial para você, não faz sentido o router cachear o conteúdo para outro possível usuário, por isso caches intermediários são completamente inexistentes e o cache local, dependendo do seu software de cliente, também não é existente posto que o arquivo é exclusivo para você alguns não armazenam uma cópia por um curto espaço de tempo, levando o cliente a fazer uma nova busca

completa pelo conteúdo do arquivo sempre que precisar apresentar ele em tela, isto vem mudando desde 2010 quando houve uma alteração na forma com que clientes consideram arquivos HTTPS, os caches locais existem, mas os intermediários não. Você como programador pode e deve fazer caches internos da aplicação, mas saiba que seu servidor terá uma carga extra justamente por esta alteração na forma com que o cache é percebido pela rede entre você e seu cliente.

Ainda falando de segurança podemos também citar o protocolo SFTP, todavia este é tido como descontinuado, ainda hoje é muito popular e útil quando falamos de distribuição de informações, mas possivelmente em um futuro ele não mais exista, ao menos não como o conhecemos hoje.

Outro dia conversando com um amigo soube que ele chegou a trabalhar com máquinas de perfuração de cartão, um antigo equipamento de armazenamento de informação. Em brincadeiras sempre que ele me chamava de velho eu respondia que “ao menos não cheguei a trabalhar com máquina de cartão!”, indicando que apesar de ter mais idade que ele sou mais jovem também. Porém ao escrever esse livro me vi pensando que um dia vão me dizer algo do tipo “ao menos não cheguei a trabalhar com telnet, ou sftp!”. A substituição do uso do Telnet pelo SSH é relativamente recente, mas altamente importante para o dia-a-dia de nossas vidas, apesar de ser incorporado em vários programas, um dos quais possui exatamente o mesmo nome que o protocolo, por isso essa dificuldade de entender o que são os elementos, mas “uma coisa é uma coisa” e “outra coisa é outra coisa”!

Existem softwares e ferramentas, como o SSH do Linux, mas estes são apenas aplicativos que possuem o mesmo nome do protocolo que fazem uso. Além das ferramentas também existem pacotes e bibliotecas de sistemas que também levam o nome destes protocolos, como o OpenSSL, também do Linux, que incorpora no sistema operacional chamadas do sistema para a configuração e manipulação de outros softwares, habilitando assim a máquina a trabalhar com os protocolos SSL ou o TLS, permitindo assim por sua vez que softwares como o SSH sejam utilizados nestas máquinas. Cada sistema operacional possui sua

própria arquitetura e sei que a maioria não vai entender minha tentativa de simplificação, mas falando de forma popular e sendo direto e mais claro que água, "*O 'OpenSSL' é um módulo do 'Kernel' que possibilita com que o 'Sistema Operacional' possa implementar os protocolos 'TLS' ou o 'SSL' diretamente na camada de 'Transporte' da arquitetura 'TCP/IP' para que aplicativos como 'Apache', 'nginx', 'SSH', 'MySQL', 'MS-SQL' e vários outros, implementem comunicação por meio da camada 'Aplicação' do 'TCP/IP' também por meio de protocolos, como o 'SFTP', o 'SSH' e o 'HTTPS'*".

Quem utiliza

Quando estava escrevendo este livro e cheguei neste item pensei em desfazer tudo, a ideia que tenho na criação da série Crash Course de livros é ter uma estrutura padronizada, daí entendi que não poderia seguir exatamente a mesma estrutura para todos pois não existe demanda de ter um item do tipo "quem usa isso" pois a resposta seria efetivamente "todo mundo", até você, mesmo sem saber. Nos dias modernos mesmo quem nunca ouviu falar do assunto SSH Personal Keys utiliza essa tecnologia, só não sabe disso.

No entanto, tentando encaixar um texto para responder a pergunta de "quem utiliza" e considerando apenas "SSH Personal Key Files" no contexto de desenvolvedores eu diria que seriam programadores, gestores de projetos, analistas de testes e qualquer pessoa que precise baixar o código fonte para fazer testes ou demonstrações locais, dependendo do projeto até um vendedor pode ter um SSH Personal Key File para se identificar e baixar o projeto. Contudo este item na estrutura da série seria para apresentar grandes empresas que fazem uso da tecnologia, neste contexto então a resposta é justamente "Todo mundo", desde o Google ao fazer deploy de alguma aplicação até o jose da esquina vendendo um picolé ao declarar imposto de sua empresa para o governo brasileiro.

Quem é o mantenedor

O grupo conhecido como IETF (Internet Engineering Task Force) é tido como o mantenedor da maioria dos protocolos envolvidos, ela é a instituição responsável pelos populares RFCs e é a partir deles que a maioria das empresas se baseia para a criação de softwares e sistemas. Porém aqui também poderíamos dizer “todo mundo” sendo que diversas empresas implementam os protocolos e softwares aqui descritos. Praticamente podemos dizer que a IETF é responsável em propor um padrão que pode ou não ser seguido pelas empresas ao criar uma aplicação, mas objetivando uma maior amplitude de clientes as empresas acabam incorporando tais protocolos em suas aplicações, passando assim a serem mantenedores indiretos de tais padronizações.

Já o OpenSSH, um dos pacotes mais importantes quando o assunto é Chaves Assimétricas, é uma biblioteca open source, foi inspirado no formato de projeto do OpenBSD e inicialmente desenvolvido por alguns dos desenvolvedores originais do sistema operacional.

Formato da chave e criação da chave.

De agora em diante iremos falar sobre Arquivos de Chaves pessoais, as Personal Key Files são arquivos do tipo texto simples, então não fique com medo, eles não mordem, para ver seu conteúdo basta utilizar comandos simples como o cat, more (funcional tanto em Windows como em Unix-Like) e less diretamente em seu shell, ou mesmo com abrir eles com seu editor de textos preferido.

Na grande maioria das vezes os arquivos de Chaves Privadas gerados não possuem extensão, no entanto podem ter as extensões “.key”, “.pem” ou “.ppk”, mas em geral as Chaves Públicas relacionadas a estes Arquivos de Chaves possuem a extensão “.pub”.

Caso deseje criar uma Chave Pessoal no Windows você precisará instalar o aplicativo PuTTYgen, caso esteja com uma versão de Windows 10 ou posterior você pode optar por utilizar o pacote OpenSSH da

Microsoft, caso esteja realizando este procedimento no Linux deve utilizar o pacote OpenSSL. Como já dito os comandos aqui ilustrados serão apresentados conforme o linux, mas os conceitos são aplicáveis e independentes da plataforma, seja FreeBSD, Linux, Mac (versão antiga), iOS, Windows (antigos e novos). Entretanto em ambientes com interfaces gráficas existam outras formas de gerar os arquivos, como é o caso do Windows, que a geração é feita por meio de uma interface gráfica.

No linux (e imagino que a grande maioria dos Unix-Like, se é que não é a totalidade) fazemos uso do ssh-keygen. Abaixo um exemplo da execução deste comando. O sistema irá gerar um arquivo com o algorítimo RSA e com a palavra “Comentário” ao final do arquivo de nome “NomeArquivo.pub” que será a Chave Pública do arquivo de Chave Priada de nome “NomeArquivo” (sem extensão).

```
$ ssh-keygen -t rsa -C "Comentário" -f NomeArquivo
```

Dica: A palavra “Comentário” nada mais é além de um comentário, você pode agregar este comentário em suas Chaves Públicas para facilitar o controle destas nos servidores em que as instalarem.

O comentário vai para o final do Arquivo Público, por isso recomendamos sempre, no mínimo, colocar um email para identificar o responsável ou o criador dos arquivos de Chaves. Obviamente tudo tem limite nessa vida, mas nunca encontrei o limite de caracteres de um comentário em uma Chave Pública. Segundo o que sei alguns sistemas limitam as linhas nos arquivos authorized_keys mas podemos colocar uma frase com data limite de uso do arquivo sem problemas. O comentário também não faz parte da Chave Privada e pode ser totalmente diferente para cada Chave Pública que você criar.

Exemplos de conteúdos de arquivos

Enfatizando, todos os arquivos relacionados a Chaves Assimétricas de

uso pessoal são arquivos de caracteres simples, texto puro. Em geral não é necessário se preocupar com a tabela de caracteres de armazenamento (ex: se o arquivo é UTF-8 ou ISO8859-1), isso por que os caracteres no arquivo não possuem caracteres acentuados nem cê-cedilha. Você pode agregar esses caracteres nos comentários, situações em que pode gerar algum tipo de corrosão, mas esta corrosão não interferiria no conteúdo da chave, apenas nos comentários. Entretanto nunca tentei algo parecido, mas digo “em geral” pois é bem possível que haja algum problema ao tentar salvar o arquivo em tabelas de caracteres muito diferentes dos ocidentais como o cirílico, ou o han (cantonense).

O arquivo de Chave Pública, quando gerado com o algoritmo RSA, que é o caso da maioria dos comandos apresentados neste livro, sempre será iniciado com a string ssh-rsa, terá um hash e será seguido do comentário que você incluiu quando criou o arquivo, caso você o tenha fornecido.

O arquivo de Chave Privada contém mais caracteres dispostos em várias linhas, no final ele é bem maior que o apresentado aqui onde fazemos apenas a representação visual para a apresentação, mas ele não contém o comentário, como ocorre no Arquivo Público.

```
$ cat NomeArquivo.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDCEoqZvsymWoVrGW6n6h+p
uf58g80a6nz9xArADXoUJWhXzzjWOP1R94wti492gi0N4yCF6zWd99ortvbf
dedxwwsxxcfBSqd57MbyJY3p+UGREJ+olWU4uyHaHyfXZZijjHEeN+3Idp
m5SkwjRm+xPwPTxwkYfefJpDMobc1OJFJmb8JPP sample@mail.com
```

```
$ cat NomeArquivo
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2BDE9039AB55B23D4285C00498081543

0+/tBiWI0DuenyER7183j/TOYHlXkcOvIheWcQ6YWjlS9rAx9BVo4N7HxIk5
PXo0
```

```
SQXH/QUF1FRgeVJ3XDcSIQSmUlU521oTddj8vjGB0BAZB9+0PqVhNl3iS
hs9ASnM
6FboEDNfmLSUbBpjFLQYSgs3GoQ7+7oSr3l1ntlSO3OaQtQILNG2psS1IZ
kwJzko
Cw714Ck/pEx6Z629uwLp3oc6YD8G7PoipTF3IKuNKB3lchYo78rk0lhoqL0
Zoeqf
(...)
P0sLJIg6aGXQbnY8kO724OVGhPUSF73O5c1m1ChO6N1OY5U9SxqW8
0wFrLtT1QYP
AgcpkLQEeqpCHOAGttD7WsPYbV/0xoZLojnE5A2ep+th2iPNCIMRMS+Sq
l9tAy0b
NifbGHljgF+qUP4u7d9W/22RLvHAKTqv+HOGhVVVULQcKed1A7JoOouN
rBqBBn7J
-----END RSA PRIVATE KEY-----
```

Nível de acesso e configuração de seu ambiente

Objetivando mais dinamismo o cliente de SSH sempre que for chamado irá buscar um diretório de configurações pessoais de nome “.ssh” diretamente no home de seu usuário para carregar diversas informações e identificar se você possui arquivos de Chaves para serem oferecidas quando estiver tentando acesso a algum servidor.

Aqui iremos detalhar como proceder sobre o ambiente Linux, o detalhamento serve para qualquer ambiente Unix-Like, como IBM-AIX, FreeBSD e o iOS da Apple. Não tenho muita experiência em ambientes windows com estes arquivos, não lembro de nenhuma demanda para configurar o nível de acesso dos arquivos quando fiz uso destes, mas é possível que na maioria das vezes você nem precise se preocupar com esta questão, especialmente se estiver utilizando o Putty, entretanto se você quiser manter um ambiente windows similar aos ambientes Unix-Like você provavelmente precisará ter um Windows 10 ou superior pois anteriormente o windows não permitia diretórios iniciados com “.”, que seria o caso do “.ssh” e este diretório deve ser criado em “C:\Users\<NomeDeSeuUsuário>\.ssh”.

Esse “nível de acesso” aqui detalhado é obrigatório quando configurando os arquivos no diretório “/.ssh” existente diretamente no home do usuário,

porém é bem possível que algo semelhante seja necessário caso queira fazer uso dos arquivos a partir de outro diretório, o que acontece é que a demanda de nível de segurança para os arquivos é do programa cliente de SSH e não do sistema operacional e o mesmo é quem indica onde o diretório ".ssh" está instalado.

Caso seu usuário não possua o diretório "/.ssh" no seu home basta cria-lo normalmente com o comando "mkdir" (ou com o Windows Explorer se estiver utilizando o Windows 10), o diretório é apenas um diretório, as particularidades dele são:

1. O nome obrigatoriamente deve ser ".ssh";
2. Ele deve estar disponível no home do usuário;
3. (Em Unix-Like) Os arquivos devem estar no charset UTF-8;
4. (Em Unix-Like) O usuário proprietário do diretório ".ssh" deve ser o mesmo usuário proprietário do diretório onde o ".ssh" se localiza; e ...
5. (Em Unix-Like) O diretório "~/.ssh" deve ser sempre "0700" (Leitura, gravação e execução para o usuário) e os arquivos internos, todos, devem ter o nível "0600" (apenas leitura e escrita, sem execução e apenas para o proprietário)! Bem, nem todos os arquivos precisam ser "0600", mas nesta situação é mais fácil manter tudo em "0600" do que ficar se lembrando dos detalhes posto que nunca precisei de configuração diferente disto.

Nota do autor: Como o nível de autoridade tanto do diretório quanto dos arquivos exclui autoridades ao grupo e a outros o grupo para o qual o diretório e os arquivos estão associados não é relevante, ao menos eu nunca tive problemas ou li algo indicando problemas quanto a isto.

Existem dois arquivos de configuração, o do servidor e o do cliente, iremos abordar aqui apenas o do cliente, que é o necessário para entender o funcionamento do key files.

Em ambientes Unix-Like para configuração do usuário existe também o arquivo "/etc/ssh/ssh_config" que normatiza o padrão do ambiente de todos os usuários é genérico de sua máquina e restrito para manutenção exclusiva do administrador de sistemas, porém é muito pouco comum

alterarmos esse arquivo, o que acontece é que geralmente colocamos no arquivo “~/.ssh/config” as preferências de nosso usuário. Abaixo exemplo do conteúdo do arquivo onde forçamos o arquivo de chave “MinhaChavePublica” para ser utilizado no github e o arquivo “OutraChavePublica” para ser utilizado quando chamamos o site teste.outrosite.com.br

```
ForwardAgent yes
IdentityFile /var/www/.ssh/MinhaChavePublica
IdentityFile /var/www/.ssh/OutraChavePublica

# linha de comentário
Host gh github.com
        Hostname github.com
        IdentityFile /var/www/.ssh/MinhaChavePublica

Host 666.113.174.666 teste.outrosite.com.br
        Hostname teste.outrosite.com.br
        IdentityFile /var/www/.ssh/OutraChavePublica
```

Curiosidade: Caso tenha interesse pode buscar informações referente ao arquivo de configuração de um servidor de SSH veja o arquivo /etc/ssh/sshd_config.

Após sua criação poderá observar que a autoridade dos arquivos é “0700” para diretórios e “0600” para arquivos e que o usuário e o grupo é o mesmo do usuário que irá interagir com os arquivos.

```
$ ls -la
total 28
drwx------ 2 userName groupName 4096 May 28 17:13 .
drwxr-xr-x 9 userName groupName 4096 May 28 17:13 ..
-rwx------ 1 userName groupName  789 Apr  2  2012 authorized_keys
-rw------- 1 userName groupName 1679 May 28 17:09 NomeArquivo
```

```
-rw------- 1 userName groupName  406 May 28 17:09 NomeArquivo.pub
-rw------- 1 userName groupName   74 May 28 17:13 config
-rw-r--r-- 1 userName groupName  407 May 28 17:06 known_hosts
```

Dica: No exemplo acima o “.” é um link para o diretório “~/.ssh” e por isto a importância de apresentamos aqui o nível de autoridade do mesmo

Em ambientes Linux um erro de permissão irá ser apresentado quando um arquivo de Chave Pública estiver com a permissão de acesso diferente de “0600”, neste caso basta ajustar a chave para um nível de permissão adequado, como no exemplo abaixo que ilustra uma primeira tentativa apresentando o erro, a alteração da autoridade e uma segunda tentativa bem sucedida de interação.

```
$ ssh-keygen -y -f mykey.key
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'mykey.key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "mykey.key": bad permissions

$ chmod 600 mykey.key

$ ssh-keygen -y -f mykey.key
Enter passphrase:
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAIAQDEgvL(..)ymxiQ==
```

Como criar Arquivos de Chave

A criação do Arquivo de Chaves pode ser feita por algumas variações do comando de criação. O comando base é o abaixo indicado.

```
$ ssh-keygen -t rsa -C "Comentário" -f <private key file>
```

Veja abaixo a funcionalidade dos parâmetros acima:

- "**t**": Indica o tipo de algoritmo de encriptação a ser utilizado na criação de Chaves. O tipo selecionado é importante pois se o servidor que você estiver tentando se conectar não tiver suporte ao tipo selecionado você pode não ter acesso a aquele servidor. Vale lembrar que você não precisa se preocupar com um tipo específico, você pode ter um arquivo para um servidor específico e outro para outra máquina e em ambos os arquivos utilizar a mesma Senha para acessar os servidores diferenciados. Veja mais abaixo exemplos de criação de arquivos com vários tipos e algumas observações sobre eles.
- "**C**": O comando acima irá criar um Arquivo Privado e outro público, o que for disponibilizado após o parâmetro "C" será agregado como comentário ao final da linha do Arquivo Público. É útil para facilitar "lembrar" o motivo do arquivo ter sido criado. Se você participa de uma equipe de programadores recomendo sempre utilizar seu email como comentário, desta forma facilita para o administrador de sistemas saber de quem é as Chaves cadastradas no servidor. Caso o comentário não seja fornecido o sistema irá criar uma string a partir do conteúdo das seguintes variáveis: "$USER\@$HOSTNAME", caso realmente queira deixar o parâmetro vazio deve informar "" (espaço vazio) como parâmetro.
- "**f**": Indica o nome do arquivo de Chave Privada, por consequência é criado também um mesmo arquivo com a extensão ".pub" onde

é salvo a Parte Pública da chave. Vale observar que se você agregar a extensão neste parâmetro a mesma será utilizada para criar o Arquivo Público, p. ex., se você criar uma chave com o nome “PrimeiraChave.key” o comando irá gerar um arquivo “PrimeiraChave.key.pub” contendo a Chave Pública.

Dica: Para saber mais sobre os tipos de algoritmos disponíveis veja o item “Algoritmos de Chaves Simétricas” em “O que é Asymmetric Keys” no capítulo “Conhecimentos Elementares” deste livro.

Ao executar o comando acima irá notar que o sistema irá solicitar a você uma Senha, todos os sistemas irão lhe autorizar a criar Arquivos de Chaves sem Senha, mas em alguns (provavelmente todos os modernos) irão solicitar que escreva uma Senha com uma quantidade mínima de caracteres, acredito que o número 5 seja o padrão. Ele também irá perguntar se deseja sobrescrever o arquivo, caso tenha fornecido um nome de arquivo que já exista no filesystem.

Alguns exemplos de comandos para a criação de arquivos

```
$ ssh-keygen -t rsa -b 4096 -C "Comentário" -f <private key file>
$ ssh-keygen -t dsa -C "Comentário" -f <private key file>
$ ssh-keygen -t ecdsa -b 521 -C "Comentário" -f <private key file>
$ ssh-keygen -t ed25519 -C "Comentário" -f <private key file>
```

Para o uso convencional de desenvolvedores como git, acesso ao Shell e funcionalidades similares sempre recomendava o primeiro exemplo dos acima, é o que gera o Arquivo de Chaves mais longo e como usa o algoritmo RSA então é amplamente utilizado em vários servidores, entretanto a recomendação mais aprimorada e atualizada seria o uso do último exemplo com Chaves no formato ed25519 que atualmente também é amplamente utilizado em servidores de internet. Minha recomendação

atual é faça uso do ed25519, se o servidor envolvido em sua demanda não suportar então utilize RSA com no mínimo 4096 bits.

Nota do autor: Um arquivo pessoal pode ser gerado por uma Autoridade Certificadora, como são os casos de Certificados A1 do governo Brasileiro, entretanto no caso de Arquivos de SSH não é necessário que uma autoridade os crie, qualquer um pode criar seu próprio Arquivo de Chave.

Para verificar o fingerprint da chave

Fingerprint é uma palavra em inglês que quer dizer “digital” (a impressão digital de nossos dedos). O termo “assinatura digital” referência uma impressão digital que é uma assinatura em um conteúdo, seja ele digital ou analógica. No caso de Chaves Assimétricas todo arquivo possui uma assinatura digital que é utilizado para identificá-lo, neste caso a tal da fingerprint.

É comum a apresentação desta assinatura em arquivos assinados digitalmente. As assinaturas digitais também servem para identificar qual Parte Pública está associada a arquivos de autorização de acesso.

No Linux para ver a assinatura de um arquivo basta usar o comando ssh-keygen com o parâmetro l (L minúsculo) como apresentado abaixo.

```
$ ssh-keygen -lf <private key file>
```

Abaixo exemplo de saída do comando. Note que o fingerprint é idêntico para as partes pública a privada da mesma chave.

```
$ ssh-keygen -lf exemplo1
2048 SHA256:vGQ147cq7KfvZeF3+bzoCrLFNr/iYqjKOWh0hq8w8qE
```

```
sample@mail.com (RSA)

$ ssh-keygen -lf exemplo1.pub
2048 SHA256:vGQ147cq7KfvZeF3+bzoCrLFNr/iYqjKOWh0hq8w8qE
sample@mail.com (RSA)

$ ssh-keygen -lf exemplo2
16384
SHA256:70S/yEPUWcOAM9C7K8gH3Wja+LBK9EJSUsCfV1xOF4Q
rstriquer.pub (RSA)

$ ssh-keygen -lf exemplo2.pub
16384
SHA256:70S/yEPUWcOAM9C7K8gH3Wja+LBK9EJSUsCfV1xOF4Q no
comment (RSA)
```

Abaixo outro exemplo onde executamos o comando em um arquivo authorize_keys, isso irá gerar o fingerprint de todas as Chaves autorizadas a acessar sua máquina. Note que a primeira saída é um fingerprint pequeno e o segundo é bem maior, isto acontece pois os algoritmos de geração de ambos os arquivos são diferentes, o primeiro seria ser proveniente de uma chave pequena, já o segundo vem de uma chave com 4096 bits de tamanho.

Junto com a saída do comando acima e com a apresentação do comando abaixo podemos notar a linha onde apresenta-se o email “sample@mail.com”, note a assinatura “vGQ147cq7KfvZeF3+bzoCrLFNr/iYqjKOWh0hq8w8qE”, com isto podemos determinar que o arquivo de Chave Privada exemplo1 possui permissão para acessar o shell do servidor em que arquivo “authorize_keys” do exemplo abaixo está localizado.

```
$ ssh-keygen -lf authorized_keys
```

```
2048 SHA256:vGQ147cq7KfvZeF3+bzoCrLFNr/iYqjKOWh0hq8w8qE
sample@mail.com (RSA)
16384
SHA256:70S/yEPUWcOAM9C7K8gH3Wja+LBK9EJSUsCfV1xOF4Q
54gtqTBWMvP6zFxd4Oo5ras+sNovZ+hlfgOpI1H7rR3+y6I76RGMEie9YG
LF5XgZIw/Wg1TFVqosMx0/ghCWzoipeVDMLa6GLEX7ALqCk3Pxj9c/TfJF
zSXl4cm5v1qcGjSxammRR131noiySZooCQlk0wxmeQFGU+UVIFShJ8QV
Wy8gm9cimLFQIF05YNIINEjpIj1X14nsS4Oql6EBrBKZIOtP4z3jvS+YRLqt5
DQDPHIs8CBl2uV/QHzxRuTbZy0+4H0AEP+XapvRf7sAkuBe4ILnk6g/RdN
fHeVi5lzfh/mgA1dQgxNeFkPxhJB5fvvesiThRXQMAqCXtwzNJYdag+XS5R
6OkfpNHdcg5F35wyoJreUCz4NaQEKm (... conteúdo diminuido para efeito
de simplificar, eliminando dados não necessários para o exemplo ..)
809l/UGITcxcyTteCf1SKT5w2hbLpPozkZXB5uYKZ08seBpTqjZNslbOjekV
G3xBjtx8Rr4CLwOqXKKROP8X1MMwGZHNcJkKEmVgVHPa0eV2+1TEl
MOfBuuQ74ibP9Uk7HNPw4V4wPU4OPGXCMySX8QyuEQnJjrec7Fj+6Je
bN0dN9q4ajh1fCXqPOxGbB7qWttTFwsJy3lzudyxqNC7wHxaNL2SG2oLh
2Xr67E8neWRP7pVXZXG2xF67uTBhpXjEhMnd+rxT1qbuDblh7sNOCMn2
4gzOQjalyOwVQEvqs73ccuk7vHQi55rTPCza4nPgT+MiVnmNh0ZkjMXRm
BxC6gSEZE4dD70Vf/yr11h3IsL54A6RR32om2VKdE6ALKe4nymxiQ==
(RSA)
```

Nota do autor: Para efetivamente acessar o shell existem outras configurações necessárias para validar, mas aqui quis dizer que o ssh terá autorização de levar o usuário ao próximo passo no fluxo de liberação deste para acesso ao shell.

Boas Práticas

Boas práticas são técnicas popularmente reconhecidas como as melhores formas de trabalho para se executar um grupo de tarefas. Aqui detalho alguns elementos que acredito serem de grande importância para o leitor por em prática e se você acredita que esqueci de alguma peço que entre em contato e me fale de sua recomendação! Quem sabe ela não sai na próxima edição do livro!

Utilizar múltiplas Chaves

Você pode utilizar quantos Arquivos de Chaves for de seu interesse, particularmente utilizo muitos arquivos, geralmente mantenho por volta de 30 arquivos de Senhas e é comum fazer um Arquivo de Senha por projeto.

Esta ‘boa prática” é algo simples de explicar, não precisa descritivo além do parágrafo acima, mas gostaria de comentar sobre os Arquivos de Chave em um servidor, mais uma vez, o livro não se destina a administradores de sistemas, mas vale explicar um pouco mais para entendermos o funcionamento da ferramenta como um todo.

Como expliquei no capítulo sobre o funcionamento de Chaves Assimétricas o servidor também terá arquivos de Chaves Públicas e Chaves Privadas, elas são criadas quando o servidor de SSH é instalado no servidor. Porém diferente de Chaves Particulares as Chaves Privadas de um servidor não devem ser criadas periodicamente! Você pode recriar os Arquivos de Chave do servidor quantas vezes quiser, ocorre que quando uma nova chave é criada todos os usuários que acessaram aquele servidor no passado irão ser solicitados para aceitar uma nova chave e instruir seus usuários a aceitar novas Chaves é um pouco perigoso, eles podem ser levados a aceitar uma chave incorreta, por isto não existem recomendação para recriar Chaves de servidores periodicamente, porém você pode fazer uso de uma renovação de IP para recriar estas Chaves posto que ele receberá uma demanda de validação de qualquer forma.

O servidor de SSH possui um Arquivo de Chave para cada algoritmo que o servidor aceita e os Arquivos de Chaves do servidor ficam disponíveis no diretório “/etc/ssh/”. Veja abaixo um exemplo de estrutura deste diretório.

```
$ sudo ls -l /etc/ssh/
total 336
-rw-r--r-- 1 root root 300261 Aug 11  2016 moduli
-rw-r--r-- 1 root root   1782 Mar 30  2017 ssh_config
-rw-r--r-- 1 root root   2559 Nov 24 22:21 sshd_config
-rw------- 1 root root    668 Mar 28  2017 ssh_host_dsa_key
```

```
-rw-r--r-- 1 root root   603 Mar 28  2017 ssh_host_dsa_key.pub
-rw------- 1 root root   227 Mar 28  2017 ssh_host_ecdsa_key
-rw-r--r-- 1 root root   175 Mar 28  2017 ssh_host_ecdsa_key.pub
-rw------- 1 root root   399 Mar 28  2017 ssh_host_ed25519_key
-rw-r--r-- 1 root root    95 Mar 28  2017 ssh_host_ed25519_key.pub
-rw------- 1 root root  1675 Mar 28  2017 ssh_host_rsa_key
-rw-r--r-- 1 root root   395 Mar 28  2017 ssh_host_rsa_key.pub
```

Quanto Maior Melhor - “The bigger the better”

Imagine que você construiu um algoritmo para quebrar Senhas fazendo uso da técnica de força bruta, para testar você manda ele encontrar a Senha “amor”. Ele começa avaliando se a Senha é “a” depois ele muda para “b”, até chegar no “z”, dai ele começa com “A” e assim vai até terminar todas as opções de apenas um caractere, em seguida ele vai para “aa” e segue até encontrar sua Senha, para identificar sua Senha o computador precisou fazer várias tentativas e fica fácil entender que a palavra “amores” seria mais difícil de quebrar do que a palavra “amor” pelo simples fato de ela conter mais caracteres. Neste mesmo entendimento sabemos que quanto maior o hash do Arquivo de Chaves, maior a dificuldade de um computador de reproduzir este hash.

Recentemente participei de um evento onde o palestrante reportou que em seus estudos ele recomendava as pessoas a utilizarem palavras ou frases como Senhas particulares, não precisando fazer uso de caracteres especiais, agregando letras maiúsculas/minúsculas e numerais nas Senhas, ele indicava que era mais forte uma Senha com 12 caracteres contendo apenas letras do que uma Senha de 6 ou 8 caracteres, mesmo estas contendo caracteres diferenciados.

No entanto quando falamos de Chaves Assimétricas não estamos falando da Senha e sim do hash que o algoritmo de geração dos Arquivos de Chaves irá utilizar. Procure utilizar algoritmos que lhe propicie um hash grande, quanto maior melhor. A recomendação que vemos na internet é de no mínimo 2024 bits de tamanho, particularmente tenho utilizado e recomendado o uso de 4096 bits de tamanho.

Em contrapartida vale observar que, quanto maior o arquivo maior o decaimento de performance com a velocidade na avaliação deste,

contudo posso afirmar por experiência própria que o decaimento com Chaves de 4096 bits de tamanho não é tão impactante, ao menos não para utilizar em ferramentas como o github e similares. Imagino que possa hver algum tipod e dificuldade de performance em situações extremas, um servidor HTTPS, p. ex., que sirva milhares de páginas simultaneamente, todavia em arquiteturas como esta geralmente separa-se o servidor e múltiplas máquinas, o que de certa forma resolve a questão de performance.

Quanto Maior Melhor, só que não!

Quando falamos de Chaves Simétricas o fator tamanho é sempre muito importante, quanto maior o tamanho maior a dificuldade de se identificar a Chave. Considerando uma chave apenas com letras e caracteres acessíveis no teclado, que em geral seria 130 caracteres passíveis de serem utilizados, então para quebrarmos uma Senha fazendo uso de brute force precisaríamos de 130^n (130 elevado a “n”), onde “n” é a quantidade de caracteres da Senha, para encontrarmos a Senha. Porém diferente do que a maioria dos profissionais de segurança acreditam, quando falamos de Chaves Assimétricas nem sempre o tamanho da frase importa.

Em Chaves Assimétricas a Senha é um objeto matemático que contém uma estrutura interna específica, conceitualmente falando quebrar a Senha consiste em revelar a estrutura interna do numeral. Fazemos isso através da “fatoração de módulos de números inteiros”, o que aumenta em muito a dificuldade de se encontrar a resposta.

Na comparação de um algoritmo RSA de 4098 bits com um ed25519 que sempre gera um resultado de 258 bits ambos possuem um resultado muito parecido, é consenso inclusive que um RSA maior que 3072 é algo desapropriado e para quebrar esse tipo de Senha o poder computacional seria muito alto, tanto que apenas bancos e entidades governamentais teriam capacidades para isso (ou empresas como Google, Amazon AWS, Microsoft, dentre várias outras).

Particularmente não sai construir esses cálculos, nunca tentei construir

um algoritmo para quebra dessas Senhas, apenas tenho superficialmente o conhecimento de como é o algoritmo para a construção dessas Senhas, o que me leva a ter o conhecimento teórico de que o uso de Senhas maiores que 3072 tem um impacto mais psicológico que prático, por isso sempre falo para usarem RSA com no mínimo 4096 bits, mas se for para escolher uma alternativa o ed25519 é uma boa alternativa.

Altere Chaves periodicamente

Essa regra também é simples de explicar, a ideia é constantemente trocar os arquivos de Chaves. A probabilidade de alguém quebrar seu Arquivo de Senhas é muito pequena, mesmo arquivos com Chaves pequenas e algoritmos antigos, todavia ocorre que existe a possibilidade de alguém ter tido acesso a seu computador e furtado os arquivos de Senha sem você ter percebido essa invasão, por este motivo existe a boa prática de renovar arquivos de Chaves periodicamente, cada situação é uma situação, mas sempre recomendo faze-lo no mínimo uma vez por ano.

Existe um dizer popular sobre Senhas que acaba sendo algo também aplicável a Chaves Privadas. “Senhas são como roupas íntimas”, são pessoais, cada um tem uma preferência, você até pode mostrar para a pessoa que ama, mas não saia em público desfilando com ela a mostra e sobre tudo! Troque periodicamente!”

Essa “boa prática” acaba sendo uma das vantagens de se utilizar Chaves Privadas, ocorre que você pode alterar sua chave periodicamente, mas manter a mesma Senha que complementa a Chave Privada. Claro que o ideal é sempre alterar ambos, entretanto mais uma vez falo que cada situação é uma situação! Por isto vai da importância do que você está tentando resguardar, se for algo de grande importância recomendo alterar também a Senha de acesso.

Atualize os programas servidores e clientes

Parece besteira lembrar as pessoas da importância de se atualizar softwares, mas sou uma prova viva de que é importante lembrar os amigos, sempre que possível!

Por volta de 2014 eu tive uma das últimas dores de cabeça com segurança, o nome dessa “dor de cabeça” era heartbleed, foi uma grande

falha de segurança que ocorreu na biblioteca OpenSSL e que permitia as pessoas a terem acesso a dumps de memória do servidor, sem ter acesso ao servidor em sí. Por situações como esta que evangelizo a importância de manter seus sistemas atualizados, mesmo softwares altamente estáveis e bem testados como o OpenSSL devem sempre serem atualizados. No caso tanto os softwares clientes quanto servidores devem ser sempre atualizados.

Sempre utilize Senhas

Quando na geração de Arquivos de Chaves temos a opção de não fornecer Senhas, com isto o arquivo é gerado e sempre que ele for utilizado o sistema não solicita Senhas, apenas realiza a tarefa. Este formato é o padrão quando geramos um Arquivo de Chave para servidores de página, não fornecemos Senhas, a não ser que queira que todos os seus visitantes precisam fornecer a mesma Senha sempre que uma página for solicitada. Porém quando se tratando de arquivos de Chaves para uso particular (Personal Key Files) então Senhas devem sempre ser utilizadas.

Existem sim exceções a regra! Digamos que você automatizou a interação entre dois computadores, você tem a opção de gerar um Arquivo de Chave sem Senha para ambos, com isto não é necessário agregar uma camada extra de programação para fornecer Senhas ou disponibilizar um intermediário humano para fornecer Senhas, basta fazer uso dos arquivos de Chave Pública e Privada que a comunicação fluirá normalmente, entretanto posso facilmente afirmar que esta situação onde a Senha pode ser ignorada é muito pouco ocorrente, em 20 anos de experiência com Arquivos de Chaves, dos quais 10 de uso praticamente diário, tive apenas um cenário onde isso foi recomendado.

Sempre utilize Senhas com arquivos de Chave Particulares!

CheatSheet e dicas

Este livro originalmente nasceu de uma apresentação de slydes que fazia para novos funcionários que não tinham costume de utilizar Arquivos de Chaves, eu acabava explicando bem menos sobre os conceitos de Arquivos de Chaves e ia direto para a parte de “como fazer”, que é atualmente este “consulta rápida” (cheatsheet)! Por isto digo que o livro se desenvolveu ao redor desta sessão.

Esta seção é construída no formato “perguntas e respostas”.

Se você acredita que uma pergunta deveria estar nesta seção mais uma vez lhe encorajo a entrar em contato e me falar sobre suas observações, terei grande alegria em escutar e se for interessante agregar sua colaboração neste livro, agregando seu nome na lista de colaboradores!

Como copiar o Arquivo de Chaves para o servidor

O procedimento é simples, basta copiar o conteúdo do arquivo da Chave Pública diretamente na última linha do arquivo authorized_keys no home do usuário que está querendo associar a chave. Para entender um pouco mais sobre o arquivo authorized_keys você pode ver o artigo “Para que server” na explicação sobre “O que é Symmetric Keys” no capítulo “Conhecimentos Elementares” deste livro.

Porém existe outra forma de fazer esse “copiar-colar”, você ainda precisará da Senha do usuário que está acessando, mas poderá utilizar o comando ssh-copy-id para que a chave seja adicionada ao arquivo de forma automática, simples e direta! Sem precisar se preocupar com nível de acesso ao arquivo e outros detalhes. Abaixo exemplo de uso do comando.

```
$ ssh-copy-id -i ~/.ssh/arquivo1 usuario@dominio.com
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"./arquivo1.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
usuario@dominio.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'usuario@dominio.com'"
and check to make sure that only the key(s) you wanted were added.
```

Note que o arquivo referenciado foi o arquivo de Chave Privada, porém ao executar o comando acima é necessário que a Parte Pública do comando esteja disponível, do contrário o sistema irá gerar um erro dizendo que o arquivo não possui Chave Pública disponível, como apresentado abaixo.

```
$ ssh-copy-id -i ./arq1 usuario@dominio.com

/usr/bin/ssh-copy-id: ERROR: failed to open ID file './arq1.pub': No such file
```

Como saber se já acessei um servidor

Você pode utilizar o arquivo “known_hosts” para saber se já acessou no passado algum servidor, basta utilizar da sintaxe apresentada abaixo alterando o “dominio.com.br” para o domínio que tem interesse em validar.

```
$ ssh-keygen -F www.dominio.com.br -f ~/.ssh/known_hosts

$ ssh-keygen -F dominio.com.br -f ~/.ssh/known_hosts
# Host dominio.com.br found: line 44
dominio.com.br ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDQ
hdLBufdwhRkW21+qi4wF1VXTPdAXk2hikdSLxy8ANMTn9X8dEdT1l0hvnJ
se7ljHpA+Sjoby3TGtUmKaXBpw=
```

Note que a url www.dominio.com.br não foi identificada, isso acontece pois o arquivo registra o nome do domínio conforme você escreve ele no momento de acessar o mesmo, se utilizar um IP então o IP será associado a chave, ao invés do domínio.

Ou seja, se precisar verificar busque por todas as variações que conhece, ou utilize o comando de shell “grep” para buscar de forma mais dinâmica. Veja o segundo exemplo com o grep no próximo item “Como apagar a Chave Pública de um servidor que visitei?” para exemplo.

Como apagar a Chave Pública de um servidor que visitei?

Se o servidor alterar alguma das configurações, como seu IP, o seu cliente de SSH vai pedir para você confirmar que está tudo bem

solicitando para você excluir o registro do servidor previamente existente em seu arquivo de known_hosts.

Para eliminar o registro do servidor no arquivo known_hosts no cliente você pode simplesmente acessar o arquivo, encontrar a linha relacionada ao servidor que deseja apagar e salvar o arquivo, faço muito isso quando não lembro quantas variações de um domínio tenho registrada em meu known_hosts, mas você também pode fazer uso do comando ssh-keygen com a opção -R para executar o comando, veja abaixo o exemplo de uso do comando:

```
$ ssh-keygen -f ~/.ssh/known_hosts -R dominio.com.br
# Host dominio.com.br found: line 107
# Host dominio.com.br found: line 114
# Host dominio.com.br found: line 115
/home/usuario/.ssh/known_hosts updated.
Original contents retained as /home/usuario/.ssh/known_hosts.old
```

Para encontrar as variações no arquivo faço uso do grep na linha de comando, veja abaixo exemplo de uso com o comando grep.

```
$ grep dominio ~/.ssh/known_hosts
www.dominio.com.br ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDIUuqDiNe7Toqr1jSl0XtRCdJu
4q5xGylBx2b5Q3w+FM4crxx3wpS/OjrzpeBPucmtgMI6sY8VZD0DSDV2Dz
L/k9fs5CMUVcLi6BbGiINQwBiB7hkBB0Pw1Jzt82BE3oXzksbB2arfUrGqv9
NhFKSV1diac0H+8a5x1DxFxwYlQVWqoYsJKf6uJyBYglfyNqJTytE7caS1V
E3fGEu8tkRYIYu9bWb9yo1o9SebhxU4nq08GfUltoFxTfxTwSY6q5aLodzu
GOhzX8BYGzWg6OfhnXAOah3RVmDSWmN+vWeG3+279oNJ+sEgcyqtu
NohXZf9n2yFo38PcQNL6hJ0tQvT
controle.dominio.com.br ecdsa-sha2-nistp256
AAAAE2VjZHNhkzNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDQ
hdLBufdwhRkW21+qi4wF1VXTPdAXk2hikdSLxy8ANMTn9X8dEdT1l0hvnJ
se7ljHpA+Sjoby3TGtUmKaXBpw=
database.dominio.com.br ecdsa-sha2-nistp256
AAAAE2VjZHNhLxNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDQ
```

```
hdLBufdwhRkW21+qi4wF1VXTPdAXk2hikdSLxy8ANMTn9X8dEdT1l0hvnJ
se7ljHpA+Sjoby3TGtUmKaXBpw=
database.dominio.com.br ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC6YNuAY/KhjUs5ydBHCau0Y
1UBpMZGEfKZ2Br522Ou4sCLArRiDL+IKFtiRQYlcGfLx8Od00rYNlNHRzq
mKrEA5n8bxHE6DXrcYCOwVx7/lo1iGOcSFZZ9dTlqFHLd4a4wBFUWW9
5hzkI2JHrG8vz6P7T2T2bQAngw8BjVFN+SsJEtw1KU7aoPh3QtEXNLBu9
O7zIUV6tel9EnJDTctXHTAhASlsprDt4Zk1BFxl5UyefLCHyWyMP8kqO6/4Li
w7+Tr6/2QADiAfoY1mO7lw7noVP8yJFeMoJfhH9BUB8PrfqWJWLAHvvc9j
W8Hr3wvoXzFKzaa2AeFFy6iq0zGhEV
```

Note que o domínio database.dominio.com.br possui várias entradas, isto acontece pois o sistema recebe uma entrada para cada Chave Pública recebida e no exemplo acima o domínio database.dominio.com.br encaminhou uma Chave Pública no formato RSA e outra no formato ECDSA.

Você também pode usar o “grep -e” ou o “grep -P” (Pê maiúsculo, que pode ser utilizado para buscar fazendo uso de expressões regulares em Perl, mas é mais complexo, vou me limitar aqui a explicar o uso do parâmetro “e”) para encontrar múltiplas entradas ao mesmo tempo, veja abaixo exemplo de uso do parâmetro “e” onde busco uma alternativa de IP do servidor, juntamente com seu nome de domínio.

```
$ grep -e 10.4.1.33 -e dominio ~/.ssh/known_hosts
www.dominio.com.br ssh-rsa AAAAB3.....6hJ0tQvT
controle.dominio.com.br ecdsa-sha2-nistp256 AAAAE2Vj...Bpw=
database.dominio.com.br ecdsa-sha2-nistp256 AAAAE2V..pw=
database.dominio.com.br ssh-rsa AAAAB3...EV
10.4.1.33 AAAAB3.....6hJ0tQvT
```

Para encurtar a saída e facilitar o entendimento eu recortei boa parte das Chaves que eram apresentadas na saída de tela do exemplo acima, mas gostaria de observar que neste caso tanto a entrada www.dominio.com.br quanto o IP 10.4.1.33 compartilham da mesma chave pois são o mesmo

servidor.

Perdi minha Chave Pública

Perder a Chave Pública não é problema, basta realizar o comando de geração da Chave Pública novamente, você pode realizar o comando quantas vezes quiser e sobrepor Arquivos de Chaves antigos sem problema algum.

Abaixo o comando de geração de novas Chaves.

```
$ ssh-keygen -y -f <private key file>
```

Como saber se é a chave

Utilizamos fingerprint, que é literalmente a digital do arquivo. A grande maioria das ferramentas que fazem uso de Chaves Assimétricas apresentam o fingerprint do arquivo cadastrado, teoricamente é melhor visualizar o fingerprint do que o conteúdo do Arquivo Público, p. ex., primeiro pelo fato de que o fingerprint é menor que o Arquivo Público, ou seja, mais fácil de comparar, segundo pois mesmo o arquivo sendo público não precisa imprimir cartaz e sair mostrando ele a todos os cantos!

Para gerar o fingerprint de seu arquivo basta utilizar o comando do exemplo abaixo.

```
$ ssh-keygen -lf <private key file>
2048 b7:62:f2:5c:e6:87:79:5f:03:f3:bc:bf:b8:bf:39:b1  <private key file>
(RSA)
```

Apenas para deixar bem claro devo agregar que, o finguerprint no exemplo acima é a string “2048 b7:62:f2:5c:e6:87:79:5f:03:f3:bc:bf:b8:bf:39:b1”, sendo as demais saídas

de tela apenas notificações extras.

Para gerar a Parte Pública de uma Chave Privada.

Caso você venha a perder a Parte Pública de uma Chave Assíncrona você pode recriá-la com o comando abaixo a qualquer momento.

```
$ ssh-keygen -y -f [private key file] > keyfile.pub
```

Teoricamente você não precisa manter em seu computador um arquivo de Chave Pública, as Chaves Públicas em geral são necessárias apenas no momento de descriptografar a informação, quando recebemos a comunicação no servidor ou quando encaminhamos algum conteúdo encriptado, e esta demanda acontece no sistema destino de nossa mensagem, entretanto é prática comum manter um arquivo ".pub" contendo a Parte Pública de uma Chave Assíncrona para encaminharmos esta sempre que necessário pois o tamanho do arquivo é tão pequeno que não compensa ficar criando ele a toda hora.

Como criar um arquivo .ppk a partir da chave

Este procedimento é a conversão de uma chave no formato ".key" (ou formato sem extensão), para o formato ".ppk" (o qual encripta o conteúdo do arquivo com o algoritmo base64) o qual pode ser utilizado pelo programa PuTTY no windows.

```
$puttygen arquivo1 -o arquivo1.ppk
```

A chave sendo convertida deve estar no formato "RSA Private Key", outros formatos é possível que não sejam aceitos.

Como converter arquivo para .pem

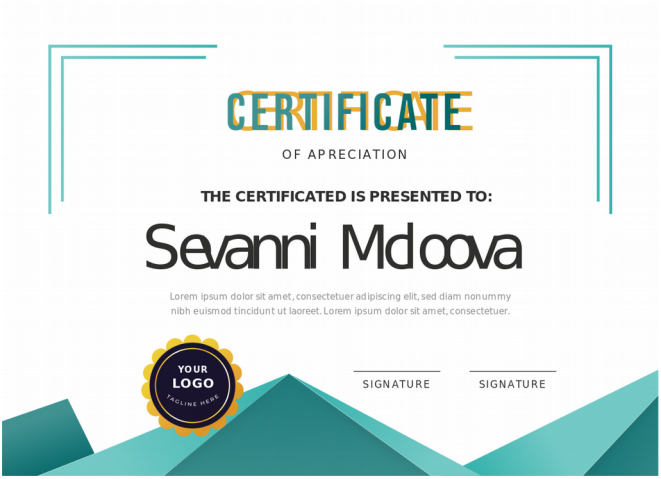
Apesar de antigo o formato ".pem" ainda é bem utilizado, principalmente por programas de email, para transformar um arquivo de Chave Privada em ".pem" primeiramente você deve fazer uso do comando anterior para

gerar um arquivo “.ppk” e em seguida você poderá converter o “.ppk” para um formato “.pem” como apresentado abaixo.

```
$ puttygen ppkkey.ppk -O private-openssh -o pemkey.pem
```

IMPORTANTE: Para o aplicativo de ftp Filezilla mais atual não é necessário converter o arquivo, ele mesmo já converte o arquivo para você.

Certificados SSL



Quando precisei lidar com a situação de criar um certificado de segurança pela primeira vez não havia tanta documentação sobre o assunto, ok, até tinha muita documentação, tudo em inglês, ok eu sabia ler em inglês, o problema é que eu não sabia nem por onde começar e quando comecei, segundo o cliente, eu já devia ter terminado! Então tive que “ir como foi” e “resolver como se resolve”, de forma que tirando o que deu errado, tudo deu certo! Mas falando sério, deu tudo certo! Ocorre que pensando nisso resolvi agregar no livro este elemento sobre como configurar um servidor de página, espero que o ajude a entender o funcionamento do protocolo HTTPS e de sistemas de servidores de página, como o NGINX e o Apache. Espero também que entendendo o funcionamento de um Certificado de Servidor facilite seu entendimento de um Certificado Pessoal e o faça entender um pouco mais sobre Certificados Digitais de uma forma generalizada.

Entendendo certificados de Servidores de Página

O Certificado de um Servidor de Página agrega duas funções, ele afere autenticidade ao site e possibilita com que as informações trocadas entre o servidor e o cliente sejam restritas, ou seja, inacessível por qualquer outra pessoa que não seja nem o cliente nem o servidor.

Os Certificados em um Servidor de Página são arquivos simples, no formato texto livre e muito similares a Chaves Pessoais, a principal diferença é que um Certificado de um Servidor de Página é emitido e chancelado por uma Autoridade Certificadora. Para iniciar recomendo a procurar na internet para selecionar qual empresa irá ser sua Autoridade Certificadora, após isso basta começar a sequência de execução abaixo. Na sequência faço uma explicação genérica de fluxo, entretanto, como cada empresa cria sua própria interface de geração de Chaves, sempre recomendo buscar junto a sua Autoridade Certificadora o passo-a-passo gerado por ela para não errar o caminho e causar transtornos.

Abaixo descrevo como criar um certificado. Considerando que você já tem uma certa afinidade com a terminologia que descrevi neste livro, recomendo a leitura da descrição abaixo para em seguida realizar a leitura recomendada por seu parceiro.

IMPORTANTE: O fluxo de trabalho é independente da Autoridade Certificadora, ou seja, você terá que criar os arquivos que vou descrever abaixo, não importa de qual empresa está adquirindo o certificado.

O primeiro passo é gerar um Arquivo de Chave. Como já falado algumas vezes, o formato mais indicado para isto é o RSA. Observou que falo muito que o mais indicado é o RSA? Isso é por conta do fato de que muitas vezes administradores de sistemas se referem a este passo simplesmente como “gerar o RSA keypair”. Para Certificados de Servidores outros formatos de chave geralmente são ignorados, tanto que nem me lembro de referências em tutoriais e documentações de empresas como Comodo, Thawte e Symantec a possibilidade de usar outro formato para esta função. Acredito que até seja possível fazer uso de outros formatos, mas nunca tentei, nunca lí nada a respeito e não sei se seria funcional, salvo casos onde o servidor é para uma função muito

específica o mais indicado é o RSA justamente por sua amplitude de compatibilidade com diversos clientes.

Geralmente também para estes casos os administradores nomeiam o arquivo como ".key" para facilitar "se encontrar" nos arquivos que são gerados no procedimento, mas como dito, você não precisa deixar este arquivo com esta extensão e também como dito anteriormente é boa prática manter o arquivo sem extensão, entretanto para facilitar a explicação iremos nos referenciar aqui ao primeiro arquivo sendo gerado como "KEY", também iremos nos referenciar ao segundo arquivo como CSR e aos demais arquivos como CRT, BUNDLE e CHAINED.

Outra prática comum neste cenário é gerar um arquivo KEY único e todo ano criar um novo CSR a partir de um mesmo KEY, o que é algo também não recomendável! A arquitetura de Chaves Assimétricas é efetivamente segura, a probabilidade de alguém quebrar uma chave é muito pequena, mas nada impede do Arquivo de Chave ser roubada de sua máquina. O arquivo KEY simplesmente também não "faz verão" Gerar um arquivo KEY é muito fácil e a partir disto você diminui a probabilidade de alguém ter copiado seu KEY e se passar por você.

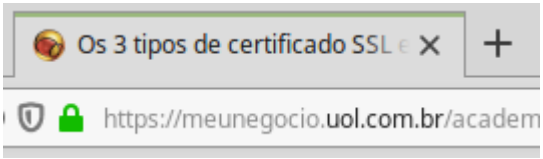
IMPORTANTE: Chaves menores que 2048 eram comuns até o final de 2013, mas por volta de 2010 iniciou-se uma campanha para aumentar esse número e desde 2013 Autoridades Certificadoras estão forçando a criação para ser no mínimo 2048 bits, número que espera-se ser seguro para até o ano de 2030.

A partir do KEY você gera um arquivo CSR o qual é entregue a Autoridade Certificadora, ao receber o arquivo as Autoridades Certificadoras irão fazer algum processo automatizado para validar se você é proprietário do domínio que está sendo tratado, eles irão solicitar para você fazer algum teste como colocar um arquivo disponível online ou adicionar alguma linha em seu DNS. Alguns disponibilizam validação através de mensagens de email. Geralmente faço uso do arquivo por achar isso mais prático pois não envolve outra tecnologia (não preciso, p. ex., solicitar a adição de comandos DNS em outro equipamento ou receber um email que muitas vezes não possuo acesso a conta). Após confirmar sua autoridade sobre o domínio a Autoridade Certificadora irá

gerar um arquivo BUNDLE e outro CHAINED, estes são mesclados em um único arquivo CRT que por sua vez é instalado no sistema de Servidor de Página, p. ex., Apache ou Nginx.

No momento existem três modelos de certificação:

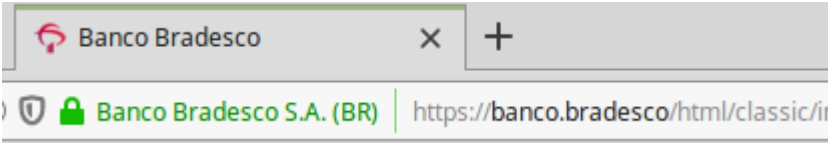
Validador de Domínio: É o mais simples, de menor custo e maior simplicidade de geração. Ele basicamente diz que a pessoa que gerou o Certificado tem autoridade sobre o domínio ao qual o certificado é relacionado. Foi a primeira forma de aferir autoridade sobre o domínio e já estava em uso desde os primórdios do protocolo SSL! Quando você cria um domínio uma entidade como a Registro.BR irá lhe autorgar propriedade sobre a URL, apenas pessoas com nível de acesso podem manipular informações nela, um Certificado de SSL confirma que a pessoa que criou o certificado tem tal autoridade e o certificado só fica ativo quando navegando sobre o mesmo domínio para o qual o certificado foi gerado! Ele até encripta a comunicação entre o servidor e o cliente quando não estiver navegando pelo domínio ao qual ele foi criado, mas sempre apresentará ao cliente uma mensagem do tipo “o certificado de segurança não é para este domínio”, de maneira que a maioria dos clientes não se sentirá seguro em situações como esta, por isto é sempre importante utilizar ele nos limites do domínio para o qual for criado. Geralmente este é o tipo de certificado utilizado pelas antigas empresas de hospedagens que ofereciam certificados compartilhados, é o tipo que apresenta o cadeado fechado na barra de endereços dos navegadores. Este tipo de certificado é o tipo instalado em Blogs, Sites Institucionais e sites genéricos de distribuição de conteúdos. A liberação para emissão do certificado depende apenas do pagamento de taxas e ocorre alguns minutos depois da confirmação do pagamento, se for por cartão de crédito é praticamente instantânea;



Exemplo do certificado Validador de Domínio

Validador de Negócio: Este certificado já começa a ser um pouco mais elaborado. Geralmente para ser emitido é necessário que alguém na Autoridade Certificadora entre em contato direto por telefone em uma linha fixa divulgada pela empresa, fale com alguém que a empresa entenda ser responsável pela empresa emitindo o certificado. Com ele você também pode indicar o endereço da sede fixa da empresa e alguns detalhes corporativos de forma mais segura pois a informação é checada pela Autoridade Certificadora no momento da emissão. Este é o certificado utilizado por várias empresas de comércio eletrônico e entidades bancárias. Quando ativo apresenta o nome da instituição (geralmente associado de alguma forma com a cor verde) na barra de navegação dos servidores. Visualmente ele era parecido com o que hoje são os Certificados Validadores de Organização, entretanto nos dias de hoje ambos tem a mesma apresentação visual e a diferença acaba sendo nas garantias que eles implementam. Em um validador de negócios, após o pagamento das taxas e demais configurações para gerar um certificado ele só acontece um ou dois dias após a confirmação de contato por telefone.

Validador de Organização: Este é o mais completo de todos, quando falamos em certificado de entidade com um Certificado Validador de Organização além de validar a existência por ligações a Autoridade Certificadora ainda valida se a empresa que adquiriu o domínio está ativa no país de origem, aqui no Brasil, p. ex., eles checam se a empresa está com CNPJ válido, se a pessoa que adquiriu o domínio realmente autorizou a compra e funções similares. Nunca tive necessidade de emissão de um Certificado de Organização, mas segundo meu conhecimento a liberação só acontece alguns dias depois do contato telefônico, após a Autoridade Certificadora validar documentos que você encaminha para ela para comprovar a existência de sua empresa.



Exemplo do certificado Validador de Organização

Algo que vale observar é que uma coisa é a aquisição do certificado, outra coisa é a emissão deste, não posso afirmar que é um padrão, mas todas as Autoridades Certificadoras que conheço separam o processo de forma que a pessoa que adquire o certificado e paga por ele não necessariamente é a mesma pessoa que realiza a geração e a configuração dos certificados. Quando digo pessoa não estou falando que você pode comprar o certificado para outra pessoa, é que seu financeiro, administrativo ou contabilidade pode criar uma conta na Autoridade Certificadora, selecionar um pacote e efetuar o pagamento, em seguida ele pode encaminhar a você o usuário e Senha que acessa o mesmo ambiente e realiza a configuração dos certificados.

Dentro destes três modelos de certificação existem três tipos de “domínios” de um certificado! O domínio, neste caso, é o escopo de utilização do certificado.

- Certificados de “**Domínio Simples**” são disponibilizados para serem configurados em apenas um domínio, algumas Autoridades Certificadoras possibilitam com que o mesmo certificado seja utilizado tanto para o domínio principal quanto para o domínio “www”, já outras impedem que o certificado seja utilizado desta forma forçando com que o mesmo seja configurado apenas e simplesmente para um único domínio, seja este domínio o principal ou o subdomínio;
- Certificados de “**Múltiplos Domínios**” (também conhecidos por SAN certificates, de “Subject Alternative Name”, ou em Português “Nome Alternativo de Assunto”) por sua vez lhe permite configurar outros domínios, além do domínio principal, ou seja, ele permite efetivamente configurar o mesmo certificado para múltiplos

domínios, geralmente 3 domínios por certificados, ex: dominio.com.br, dominio.net e dominio.com passariam a responder com o mesmo certificado. A vantagem deste certificado é justamente que o mesmo certificado é utilizado por múltiplos domínios, para os seres humanos isso não é tão relevante, é muito difícil nos dias de hoje as pessoas checarem esse nível de informação, apesar de acontecer, o legal é que entidades como google conseguem agrupar domínios de conteúdos muito diversos em um mesmo contexto;

- E por fim um certificado do tipo "**Mascara**", mais popularmente conhecido como "wildcard", muito utilizado por corporações, com ele você pode configurar uma quantidade irrestrita de subdomínios para uso em um mesmo domínio.

Curiosidade: Hoje ainda existe ainda o certificado do tipo "Unified Communications" ou "Exchange SSL Certificates" que são utilizados diretamente em servidores Microsoft Exchange 2010/2013! Certificados de Segurança são um produto e como todo produto evoluem constantemente, se você está querendo se aprimorar neste mercado recomendo visitar constantemente os sites de fornecedores destes produtos para se manter atualizado.

Vale observar que um certificado também é muito útil para mecanismos de busca entenderem a construção de um site e como dito a disponibilidade dele em vários domínios, sejam eles domínios e subdomínios ou domínios completamente diferentes.

Outro ponto interessante é que se você precisa criar um domínio por um curto período de tempo, digamos uma semana ou 30 dias, existem várias entidades hoje que disponibilizam o certificado gratuitamente, até mesmo a Comodo, uma das mais reconhecidas e recomendadas Autoridades Certificadoras disponibiliza este tipo de serviço. Nestes casos recomendo fazer uma busca no Google pelo termo "free ssl certificate", ou "certificado ssl gratuito" que encontrará muitas entidades e muitas opções de

promoções de primeiro uso. Também é bem comum o primeiro certificado ter um valor bem acessível, é comum p. ex., o primeiro certificado ser US$10 e o do ano seguinte ser R$50 ou mais. Apenas para conhecimento, acredito que o mais popular desses serviços é o “Let’s Encript” (https://letsencrypt.org) basta dar uma olhada.

Nota do autor: Temos ainda opções como a Dreamhost.com, uma das mais antigas e conhecidas empresas de hospedagem (foram fundados em 1997), que fornecem um serviço integrado de emissão de Certificados SSL facilitando sua manipulação e gerenciamento de atualização dispondo inclusive de opções gratuitas de certificados. Perfeita para quem não compreende os melindros de arquiteturas complexas de Cloud.

Ok, estou fazendo um certo merchandize aqui, mas trabalho com eles a muito tempo e se você entende de hospedagem e servidores de domínios verá que eles são uma equipe eficiente, atenciosa e com diversas opções de produtos para hospedagem simples e de baixo custo.

Caso queira pode utilizar meu link de revenda:

https://www.dreamhost.com/r.cgi?242492/hosting/shared/

Para concluir vamos explicar o que são os arquivos Chained e Bundle, muito importantes para entender como um Certificado SSL funciona.

Nota do autor: É comum acontecer de uma pessoa não ter vivência com a área de segurança e os nomes utilizados dificultam o entendimento inicial, até hoje me confundo com os nomes, por isto as vezes um arquivo Chained é chamado de Bundle e vice versa, mas com o tempo você "se acostuma" a entender o que a pessoa está falando ao buscar auxílio para resolver algum problema que possa vir a ter. Independente do nome concedido pela Autoridade Certificadora eu sempre olhava o tamanho do arquivo, o que for menor era o arquivo com o Certificado Final e o maior é o arquivo contendo os demais certificados. Caso seu fornecedor tenha lhe entregue mais de dois arquivos saiba que esta regra vale para arquivos no formato CSR, que no caso de Certificados SSL de Domínio Simples são sempre entregues pela Autoridade Certificadora.

Bundle

O "Bundle" (do inglês "agrupado") representa a corrente de Autoridades de Certificação que validam o certificado e garantem a origem da geração! A nível de segurança, para que a encriptação ocorra, o arquivo Bundle não é necessário, entretanto se você não utilizar ele para configurar seu servidor ele possuirá apenas a chave de segurança, apenas a parte da Chave Assimétrica que possibilita a encriptação, ou seja, o servidor só terá a Chave Pública para informar para seus clientes, o que nem sempre resulta em uma conexão encriptada, "nem sempre" pois muitos programas clientes irão descartar a Chave e não fechar uma conexão segura, outros até irão seguir com uma comunicação encriptada, mas não irão apresentar ao cliente que estão fazendo uso do fluxo encriptado, sendo assim o entendimento será de uma conexão insegura ao entendimento do cliente. Nesta situação isso acontece pois o cliente recebe um arquivo que diz ser gerado por uma Autoridade Certificadora, porém não disponibiliza ao cliente detalhes desta autoridade, por isto é comum que os navegadores apresentassem algum tipo de mensagem

indicando que o site não garante o certificado ou algo do tipo. Ou seja, ele cumpriu a primeira função que é encriptar mas não cumpriu a segunda que é confirmar sua personalidade.

Um Certificado SSL é constituído por outras partes além da Chave Pública, estas outras partes são o que chamamos de Certificado Raiz (Root Certificate), um ou mais Certificados Intermediários (Intermediate Certificates) e o Certificado Final (End-user Certificate), estes certificados constituem o que conhecemos como Infraestrutura de Chaves Pública, ou PKI (acrónimo do inglês Public Key Infraestructure), que é uma gama de políticas, padrões e procedimentos que geram a confiança na comunicação por meio de inter-dependência entre as partes. Neste caso eu conheço a PKI e confio nela, se você me fornecer uma Chave Pública assinada por uma Autoridade Certificadora que faz parte do PKI por consequência irá ter minha confiança! Esta é a forma que a PKI funciona. Calma, não é tão simples de entender, vamos explicar os outros termos!

Um **Certificado Raiz** é um arquivo assinado pela própria Autoridade Certificadora (Self Signed Certificate). É como se eu criasse um certificado impresso, um documento, dizendo que eu sou eu! Que por pura e simples fé confirmo que eu sou eu mesmo e o assinasse para apresentar ao mundo! É algo que parece inocente, e até o seria, se este tipo de certificado fosse o único utilizado na corrente de certificados, porém um Certificado Raiz precisa de um ou mais Certificados Intermediários para que a confiança ocorra.

Dizemos um ou mais **Certificados Intermediários** pois os Certificados Intermediários são utilizados para confirmar os certificados anteriores dentro de corrente de certificados e por isso podem ser utilizados inclusive para indicar o tipo de produto que seu certificado foi construído. Digamos que você venha a contratar um certificado de Domínio Simples na namecheap, a namecheap obteve seu certificado diretamente da “COMODO” que é, neste caso, a Autoridade Certificadora do Certificado Raiz aqui utilizado. A namecheap, que agregou seu próprio certificado e aqui considerado como um Certificado Intermediário pode gerar um outro Certificado Intermediário a partir de seu certificado para indicar que o

Certificado Final, que será o certificado instalado no site, foi gerado a partir do produto Domínio Simples e a partir deste certificado gerar o seu certificado. Apesar disto ser possível e vez por outra acontecer esta estrutura é muito incomum nos dias de hoje.

O **Certificado Final** por sua vez é o certificado que realiza a encriptação, é a Chave Assimetria efetiva utilizada na relação de comunicação entre você e seu cliente, é uma Chave Pública que carrega com sigo uma assinatura das demais Autoridades Certificadoras. Quando um cliente recebe uma Chave Pública ele pode, a partir dos demais arquivos recebidos, identificar quem gerou o arquivo e confirmar com seu banco de dados local que o arquivo recebido realmente foi emitido pela Autoridade Certificadora.

A confiança estabelecida em uma relação de Certificados SSL é percebida como uma corrente de relações, uma corrente sem pontas onde o primeiro elo é unido ao último elo da sequência. Apesar de ser uma corrente “circular” para facilitar o entendimento pense em uma corrente não circular, onde o primeiro elo não toca o último elo. O primeiro elo mais a esquerda é entendido como Certificado Raiz, é o primeiro elemento que segura a corrente, os elos seguintes, antes da última e da penúltima pontas são os Certificados Intermediários, um assinando e certificando o elo consecutivo na cadeia de elos, até chegar no penúltimo elo, que é o certificado de seu servidor, que se comunica com o último elo que é o cliente, que conhece e confira no elo inicial. Por isto entende-se que é uma corrente, o último elo não tem ligação direta com o elo inicial, mas por possuir informações oriundas deste confia no Certificado Final.

Chamamos de “arquivo Bundle” um arquivo que possui as Chaves Públicas do Certificado Raiz e de todos os Certificados Intermediários utilizados para gerar o Certificado Final.

Chained

O “Chained” (do inglês “acorrentado”) é a Chave Pública propriamente dita. É o penúltimo elo na corrente de confiança, o Certificado Final que

será utilizado para criar a comunicação entre seu servidor e seu cliente.

Note que o Chained é um documento que foi gerado a partir da união de um Certificado Intermediário e de seu CSR, que também é um documento, assinado por sua Chave Privada contendo informações sobre seu domínio. A Chave Privada é o arquivo KEY gerado no início do processo. Para quebrar uma comunicação um atacante precisaria de sua KEY além do CSR, porém nunca ví ninguém export sua CSR publicamente. Guarde tanto o CSR e seu KEY a sete chaves.

Como dito, o Chained é obrigatório, mas para uma instalação bem feita o Bundle também é necessário.

Curiosidade: Um Certificado SSL é uma “Chain Certificate” (do inglês Corrente de Certificados) e uso técnicas de blockchain para construir a corrente.

Como criar um certificado para Servidor de Páginas

Me perdoe se pareço muito repetitivo em alguns casos, como no momento de utilização de Senha ou no correto descritivo do domínio, é que são elementos importantes, por isto a lembrança a todo momento.

Caso você não tenha lido o artigo anterior “Entendendo certificados de Servidores de Página” recomendo faze-lo pois explica um pouco sobre o processo aqui descrito, sendo este artigo sequencia do anterior. Ok, então vamos começar as dicas para a geração do certificado.

IMPORTANTE: Sem pressão, mas siga com atenção os passos ao gerar um Certificado de Segurança em sua Autoridade Certificadora. Na maioria das vezes não existe “undo” e se você subir um arquivo com informação incorreta ou seguir adiante erroneamente é muito provável que acabe precisando de adquirir um novo certificado.

Por demanda das tecnologias algumas Autoridades Certificadoras sempre perguntam para qual tipo de tecnologia você está gerando o certificado, se é para servidores IIS e Java Tomcat, ou para outros tipos de servidores. Vamos seguir explicando a criação de um certificado de domínio, o mais simples disponível para facilitar o entendimento básico para servidores como Apache e nginx, porém vale observar que a geração de outros modelos de certificados não são tão diferentes, geralmente se diferenciam no momento de fornecer o nome do domínio, por esta razão é sempre importante buscar a leitura do tutorial fornecido por sua Autoridade Certificadora, para garantir que o processo seja realizado corretamente. Aqui segue um descritivo genérico e com intuito meramente didático.

O primeiro passo é gerar um Arquivo de Chave. Geralmente quando começo a execução crio um diretório para armazenar os arquivos. Abaixo os comandos de ilustração da execução. Observe que no exemplo abaixo utilizei a extensão “.KEY”, que como dito várias vezes não é obrigatório, é apenas por questões organizacionais, você poderia ter criado um arquivo com o nome “domaincombr” apenas.

```
$ cd ~
$ mkdir ssl_files
$ cd ssl_files
$ openssl genrsa -out domaincombr.key 4096
```

IMPORTANTE: Gere o SSL com RSA! O SSH comporta outros algoritmos, mas no caso de Certificados SSL o padrão de mercado é o algoritmo RSA.

O passo seguinte é a geração do CSR, abaixo o comando ilustrativo da sequência iniciada logo acima. O programa irá lhe questionar várias perguntas a respeito do domínio a ser autenticado, forneça as informações. Provavelmente seu comando deva estar em inglês, nos dias de hoje é possível que esteja em português, porém caso tenha dificuldades tudo se resolve com a ajuda de um Google Tradutor e verá que as perguntas são simples de responder, ainda assim abaixo exemplo onde crio um arquivo CSR para uma empresa de Osasco no estado de São Paulo. Observe também que não forneci Senha, também comentado anteriormente, a Senha para servidores públicos de página, na grande maioria das vezes não é recomendada.

```
$ openssl req -new -key domaincombr.key -out domaincombr.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:São Paulo
Locality Name (eg, city) []:Osasco
Organization Name (eg, company) [Internet Widgits Pty Ltd]:INet Ltda.
Organizational Unit Name (eg, section) []:Marketing
Common Name (e.g. server FQDN or YOUR name) []:domain.com.br
```

```
Email Address []:contato@domain.com.br

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:INet, internet para todos!
```

Abaixo um pouco mais sobre as perguntas realizadas.

- **Country Name (2 letter code) [AU]:BR** É simples, é o código do país seguindo a “ISO 3166-1 alpha-2” que é o nome do pais representado por duas letras;
- **State or Province Name (full name) [Some-State]:São Paulo** Estado onde a entidade representada se situa. Pode ser um nome descritivo, como o que fiz, ou um nome simplificado, “SP” apenas. Você é livre para descrever o estado como quiser. P. ex., fazem vários anos que não tenho problema em utilizar letras acentuadas no nome do estado, mas ainda hoje existem pessoas que preferem utilizar termos não acentuados na geração de certificados;
- **Locality Name (eg, city) []:Osasco** Nome da cidade em que a entidade se situa, similar ao estado, você é livre para descrever a cidade como melhor lhe convier;
- **Organization Name (eg, company) [Internet Widgits Pty Ltd]:INet Ltda.** No brasil, ao gerar um certificado para uma empresa, é comum fornecermos neste campo a Razão Social da entidade, se for blog ou um site pessoal você pode utilizar seu nome completo nesta parte, todavia também é livre e pode utilizar o termos que desejar.
- **Organizational Unit Name (eg, section) []:Marketing** Aqui geralmente coloco o departamento que responde pelo site, geralmente disponibilizo marketing. Também livre para descrever como desejar e livre para deixar sem conteúdos, vazio;

- **Common Name (e.g. server FQDN or YOUR name) []:domain.com.br** Esta parte é importante, aqui você deve saber se sua Autoridade Certificadora lhe concederá o uso do “www” ou não pois este campo será utilizado pela autoridade para identificar a qual domínio se destina o certificado, deve conter o FQDN (do inglês, “Fully Qualified Domain Name”, que traduzimos para o Português como, “Nome de Domínio Completamente Qualificado”) do domínio corretamente;
- **Email Address []:contato@domain.com.br** Caso não queira fornecer pode deixar sem conteúdos, entretanto é sempre válido fornecer um email com o qual empresas possam entrar em contato com o responsável pelo domínio. Acredito também que este campo possa vir a ser utilizado por alguma Autoridade Certificadora para validar o certificado, todavia nos momentos em que tive a oportunidade nunca consegui validar que realmente este era o campo que utilizavam para validar o domínio ao encaminhar emails no fluxo da geração do certificado;
- **A challenge password []**: IMPORTANTE! Se você está construindo um domínio aberto para a internet, onde as pessoas não irão se identificar, ou precisem ser bloqueadas para terem acesso ao conteúdo do site então você deve DEIXAR ESSE CONTEÚDO VAZIO, do contrário seu site irá solicitar Senhas aos navegadores, se você está pensando em utilizar este tipo de método para restringir acesso ao seu site, por favor, não o faça! O uso de Senhas neste momento é muito específico, caso queira fazer uso de Senhas em seu site de forma altamente simples recomendo o uso de Senhas do tipo “AuthType Basic”, basta procurar na internet descritivos como “htpasswd” que entenderá o processo (caso esteja utilizando nginx procure por “htpasswd nginx” e o uso de pacotes “apache2-utils” para distribuições padrão Debian or “httpd-tools” para distribuições derivadas do RedHat);
- **An optional company name []:** Aqui temos a oportunidade de deixar o Nome Fantasia da entidade, ou o apelido da pessoa, também livre para descrever abertamente ou deixar em branco.

Dica: Ao realizar uma renovação de certificado é importante utilizar os mesmos termos do ano anterior e caso você tenha “esquecido” os conteúdos que forneceu no ano anterior, como nome da entidade, domínio e etc, você pode utilizar o comando “**openssl req -noout -text -in 2018-dominiocombr.csr | grep 'Subject: '**” para visualizar estes conteúdos a partir do arquivo CSR gerado no ano anterior.

Agora que entendeu que são dois arquivos e a base de ambos podemos descrever como fazemos na vida real ao fazer uso de um único comando para gerar os arquivos.

```
$ openssl req -new -newkey rsa:4096 -nodes -keyout 2019-dominiocombr
-out 2019-dominiocombr.csr
Generating a 2048 bit RSA private key
.....................................................++++
.............++++
writing new private key to '2019-dominiocombr'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:Paraná
Locality Name (eg, city) []:Curitiba
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:Dominio.com.br
Organizational Unit Name (eg, section) []:Dominio.com.br
Common Name (e.g. server FQDN or YOUR name) []:*.dominio.com.br
Email Address []:suporte@dominio.com.br

Please enter the following 'extra' attributes
```

```
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

IMPORTANTE: Sempre que gerando um certificado para domínios como www.dominio.com descreva o domínio como sendo *.dominio.com (com o asterísco, indicando que é para vários subdomínios), se sua Autoridade Certificadora reclamar então gere um novo sem o “*”.

Após criar o KEY e o CSR gere o CRT que é o arquivo que efetivamente é utilizado para entregar a sua Autoridade Certificadora.

```
$ openssl x509 -req -days 365 -in 2019-dominiocombr.csr -signkey 2019-
dominiocombr.csr -out 2019-dominiocombr.crt
Signature ok
subject=C = BR, ST = Paran\C3\83\C2\A1, L = Curitiba, O =
Dominio.com.br, OU = Dominio.com.br, CN = *.dominio.com.br,
emailAddress = suporte@dominio.com.br
Getting Private key
unable to load Private key
22222223795712:error:99999996C:PEM routines:get_name:no start
line:../crypto/pem/pem_lib.c:745:Expecting: ANY PRIVATE KEY
```

Pronto, a parte de geração dos arquivos está concluída, basta agora disponibilizar o arquivo CRT para sua Autoridade Certificadora que irá solicitar uma validação e instalar o arquivo em seu domínio. Vou descrever abaixo o processo de algumas entidades e em seguida como proceder para instalar o arquivo final do servidor, assim como descrever um processo para você poder checar se o arquivo foi corretamente instalado.

Como proceder com as Autoridades Certificadoras

Não vou explicar como fazer a aquisição do certificado, entendo que você

é do ramo de TI ou tem intenção de entrar no ramo, imagino que tenha um entendimento mínimo de como compra e venda de produtos online funciona e por isto não precisa de apoio para adquirir o produto. Apesar de detalhar mais o fluxo na namecheap saiba que, apesar de ser muito similar, cada uma faz de um jeito. Entretanto saiba que sempre o procedimento é muito simples. Basicamente “clicar botões”! Você deve ter a mão o arquivo CSR gerado e atentar no momento de confirmação de geração do certificado se as informações contidas no arquivo CSR carregado no domínio da Autoridade Certificadora é do domínio que você intenciona gerar o certificado, você também terá um trabalho para realizar o procedimento de validação efetivamente, no mais o procedimento é do tipo “next”. E caso você tenha dificuldades quando efetivamente estiver realizando o procedimento tenha calma e entre em contato com o fornecedor, ele certamente irá lhe auxiliar no procedimento.

O que ocorre de “diferente” é que a maioria das Autoridades Certificadoras irá solicitar que você informe apenas o arquivo de CSR, no formato CSR mesmo, entretanto algumas irão solicitar os arquivos no formato “.PEM”, por isso abaixo sequência de comandos ilustrando o procedimento de conversão, tanto de uma rquivo “KEY” no formato “RSA”, quanto de um arquivo CSR.

```
$ openssl rsa -in dominiocombr.key -text > dominiocombr-private.pem

$ openssl x509 -inform PEM -in 2018-dominiocombr-chained.crt > 2018-dominiocombr-chained.pem
```

IMPORTANTE: O certificado sendo gerado deve ser do tipo Validador de Domínio para um único domínio, ou seja, Domínio Simples.

O que gostaria de falar um pouco neste item é de algumas das opções de como um Certificado de SSL do tipo Domínio Simple é validado que seria

as opções que as Autoridades Certificadoras disponibilizam para realizar a validação efetivamente. Este tipo de certificado é o “pão quente” das empresas, algumas chegam a gerar dezenas de centenas de certificados diariamente, por isto eles precisam de processos altamente automatizados, vou falar um pouco sobre estes processos abaixo.

Acredito que nenhuma force você a realizar todas as validações necessárias, sempre fui obrigado a selecionar uma entre as três abaixo.

Vale observar que o passo que estamos falando no momento é entendido pelas Autoridades Certificadoras como “Validação”, ou “Ativação” do certificado. Abaixo as opções citadas.

- Email: A Autoridade Certificadora encaminha um email para a conta de email detalhada no momento da geração do CSR para validar se o email informado é válido e se você tem acesso a esta caixa postal. Se você que estiver lidando com a validação não tiver acesso a este email ou precisar de agilidade não recomendo utilizar este procedimento, vez por outra é demorado e a pessoa que possuir acesso a caixa pode não saber lidar bem com emails, mesmo hoje ainda é comum a situação da pessoa não observar que existe uma caixa de spam, porém se você estiver realizando o procedimento para um cliente externo ele é interessante pois faz o cliente lembrar que você está realizando um procedimento. Caso queira optar por este procedimento para interagir com seu cliente recomendo que ligue para a Autoridade Certificadora pois algumas encaminham um link no email outras solicitam que o email seja respondido com o “from” proveniente da conta relacionada no CSR, caso a Autoridade solicite uma resposta com o “from”, p. ex., você não pode criar uma conta de encaminhamento para receber o email junto com seu cliente pois na maioria das arquiteturas de email uma conta de encaminhamento não possibilita o envio de mensagens, apenas a recepção e distribuição;
- DNS: Você terá que gerar um registro do tipo TXT em seu DNS, é fácil e rápido, todavia, apesar de cada vez ocorrer menos ainda é muito comum em domínios hospedados em ambientes

compartilhados onde você não tem acesso ao DNS, ou a pessoa responsável por manter o DNS não ter conhecimento efetivo sobre o funcionamento de DNS. Por isto antes de selecionar esta opção você deve garantir que tem acesso ao DNS e que a pessoa que irá realizar o procedimento sabe o que está fazendo, faz algum tempo, mas tive vários problemas onde eu encaminhava ao cliente a demanda de alteração no DNS, o gestor de DNS dele alterava outros registros sem relação com a validação e causava grandes problemas. O problema não era relacionado a mim, mas causava um desgaste desnecessário no cliente, o que obviamente nunca é bom. Recomendo utilizar este procedimento se você entende da ferramenta de manutenção de seu DNS e tem acesso a esta ferramenta;

- Arquivo: Este é procedimento mais simples e direto, se você não sabe como “subir um arquivo” é bem provável que deveria estar lendo outro livro, mas de todas as opções acima a metodologia do arquivo é a mais comum e provavelmente a mais selecionada. Você receberá uma string de aproximadamente 32 caracteres (o tamanho varia, mas geralmente é um arquivo bem pequeno) que deve ser colada em uma localização específica em seu servidor, o nome do arquivo desta string geralmente também é o mesmo conteúdo da string, o que varia é o diretório onde este arquivo deve ser disponibilizado. A única obrigação é que o arquivo deve estar acessível por meio de navegador e a única recomendação que posso dar é que você garanta que o arquivo esteja acessível por meio de algum navegador convencional antes de solicitar a validação do arquivo, por mais que você saiba o que está fazendo se você fizer qualquer erro, como autoridade incorreta do arquivo, você terá que aguardar o cache no servidor da Autoridade Certificadora renovar e vez por outra isso demora alguns minutos. Também é uma opção que recomendo utilizar;

Ao final do procedimento de validação as empresas irão lhe fornecer um arquivo zip contendo os Certificados SSL a serem instalados em seu servidor. Se o certificado selecionado for de Validação de Domínio no tipo

Domínio Simples elas devem em geral fornecer até 3 arquivos, o Bundle, o Chained e o CA-Key.

GoDaddy

Por sua popularidade acredito ser interessante fazer no mínimo um comentário sobre como proceder com a GoDaddy, e se você tiver outra empresa para recomendar mande sua observação que terei grande prazer em adicionar sua referência na próxima edição.

A GoDaddy irá disponibilizar dois arquivos, um onde o nome contém a palavra bundle e outro apenas com números, ao renomear tais arquivos manter o bundle no nome do arquivo e adicionar no outro a descrição "chained".

NameCheap

NameCheap é uma empresa que objetiva levar aos clientes baixo custo na aquisição de certificados. Nos últimos anos vinha comprando meus certificados todos com ela, por isto acabei gerando um pequeno tutorial para agilizar meu fluxo de trabalho, tutorial este que compartilho com os amigos neste livro.

Após a aquisição do pacote na NameCheap e a geração do CSR você segue com a sequência abaixo:

- Vá em namecheap.com e após efetuar login no namecheap vá em "Account -> Product List -> SSL Certificates" e clique no botão "Active" relacionado ao certificado que está atendendo;

- Copie e cole o conteúdo do arquivo [ano]-dominio.csr aguarde ele carregar o nome do domínio e clique em "Next"
- Selecione "Any other server" e clique em "Next"
- Selecione a opção "HTTP-base" e clique em "Next", o sistema irá solicitar um email para encaminhar o arquivo, informe o email do sysadmin na WB4B e clique em "Next" na tela seguinte clique em "Send";
- Acessar na área administrativa do namecheap o item "domain list" no menu principal (lateral esquerda), trocar o filtro de seleção no canto direito superior de "domains" para "all products", clicar na seta para baixo do domínio relacionado e selecionar o botão "manage" do certificado de segurança o qual está sendo gerado, clicar em "Edit methods" e selecionar a opção "Download file".
- Na janela que irá abrir será apresentado o nome do arquivo em "File to Download" (representado neste tutorial como "NomeArquivo.txt") e o botão "Download File" com o qual efetuamos o download do arquivo.

O Namecheap nem sempre irá encaminhar o email solicitado no passo anterior, quando encaminhar nós linkamos o chamado recebido com o chamado de atendimento de atualização do arquivo.

Resumindo o fluxo de geração de certificados

Faz já algum tempo que não instalo certificados diferentes de um Validador de Domínio do tipo Domínio Simples (o mais básico dos certificados), mas acredito que o fluxo continue sendo basicamente o mesmo em qualquer Autoridade Certificadora, independente do certificado, por isto vou detalhar abaixo sete passos que recomendo ao instalar o certificado e falar um pouco sobre minha experiência, tentando dar dicas básicas de como proceder em cada um dos momentos.

Seleção

É o momento de selecionar sua Autoridade Certificadora, seu parceiro que irá lhe autenticar perante a sociedade!

Neste caso é difícil errar quando falamos de certificados para domínios, mas é sempre bom lembrar. A primeira dica na parte de seleção é que estamos tratando de tecnologia, por isso é importante ficar esperto quanto ao objetivo do certificado, em algumas poucas situações uma Autoridade Certificadora pode não oferecer o certificado que você precisa. P. ex., nem todas oferecem o certificado com menos de 2048 bits e por algum motivo você pode vir a precisar de algum destes elementos, ou seja, fique esperto com as suas necessidades. Como disse, no caso de instalação de um Certificado de SSL para Domínio Simples, como ocorre em 99% dos casos da TI basta seguir a orientação do fornecedor, mas quando não estiver lidando com servidores como Apache, nginx, ISS e outros, vale observar as características e limitações deste servidor para evitar adquirir um certificado não adequado a sua demanda.

Aquisição

Começa imediatamente após a seleção com a confecção do pedido do certificado para a Autoridade Certificadora.

A primeira dica é um pouco menos comum, mas o conhecimento pode lhe ser útil um dia, como me foi no passado. Sempre surgem novos produtos, a dica que posso dar neste momento é de avaliar se na lista de produtos do sua Autoridade Certificadora tem o produto que deseja, as vezes ocorre de um novo produto surgir de forma que o produto que você utilizou no ano anterior acaba sendo uma opção mais cara.

Você também não precisa ficar em um único fornecedor, pode adquirir um certificado no primeiro ano com um primeiro fornecedor e outro no ano seguinte com outro fornecedor. Geralmente a primeira compra do cadastro tem bons descontos, um certificado que geralmente custa US$50 sai por US$10 na primeira compra. A venda de certificados é um comércio como todos os outros e descontos são concedidos como todos os outros comércios eletrônicos. As vezes você também ganha um desconto em uma conversa com atendente de algum chat, aqui a frase “quem não chora não mama” é a dica!

Uma outra dica é que você pode adquirir vários certificados de uma única vez e criar como se fosse um estoque de certificados, então se você prever que precisará instalar vários no decorrer de vários meses pode comprar tudo em uma única vez, pode inclusive pleitear algum desconto com o vendedor, claro que esta dica é mais valiosa para quem atende várias empresas, mas é comum empresas grandes lançarem vários sites no decorrer do ano, então a dica também vale para grandes empresas.

Geração da chave

Para a geração de Chaves já dei as dicas no capítulo “Boas Práticas”, mas importante então fica a lembrança: 1) “Quanto maior melhor”, e 2) “Altere Chaves periodicamente”

Geração do certificado

As vezes vendo a documentação nos perdemos pensando que, p. ex., o processo de validação é demorado ou complicado. A dica aqui mais uma vez já foi repassada em momento anterior, mas entenda que essas empresas são altamente automatizadas, os processos são todos simples e rápidos. Se você está considerando que algum email pode demorar 1 dia para chegar busque entender o que aconteceu pois provavelmente o email já foi encaminhado e por algum motivo você e seu servidor é que não o recebeu.

Confie mais na qualidade do atendimento da Autoridade Certificadora.

Instalação do Certificado

A instalação é bem simples, basta acessar algum arquivo de configuração e apontar seu servidor para visualizar os certificados. Irei detalhar abaixo como proceder com os servidores mais populares, o Apache HTTP e o nginx. Veja os detalhes na seção “Como instalar o arquivo no servidor’ logo a seguir, conforme sua demanda.

Uma dica que imagino ser util para iniciantes e válida para certificados

mais simples é que o certificado que estamos apresentando aqui pode ser instalado em um servidor de desenvolvimento, como em um servidor local e a partir disto você pode validar a instalação e garantir que o certificado esteja ok, basta alterar no seu arquivo hosts para que o domínio aponte para o IP do servidor onde você instalou o certificado, se for, p. ex., uma máquina virtual, basta apontar para o IP do servidor virtual.

Validação do Certificado

Assim como a validação do procedimento de instalação esta validação do certificado também pode ser feita em uma máquina virtual ou servidor alternativo. Quando eu estava começando a trabalhar com certificados uma de minhas dúvidas era se o certificado era relacionado ao IP, tinha receio de que após a instalação do certificado em uma máquina eu tivesse que adquirir outro certificado para instalar em outra máquina, mas não, nenhum certificado que eu tenha conhecimento irá restringir a instalação ou uso por IP do servidor, apenas o domínio, e eles não restringem o uso, apenas são vinculados ao domínio, ou seja, se forem utilizados em outra máquina eles irão permitir a encriptação dos dados, mas não a confiabilidade destes (por isto, tecnicamente, um não deve ser utilizado sem o outro para atingir o total da funcionalidade de segurança), então se você configurar o servidor para responder conforme o certificado você poderá validar a instalação e o certificado em outra máquina, basta configurar o hosts em sua máquina cliente.

Caso você queira validar se o certificado está instalado conforme padrão de mercado você também pode utilizar várias ferramentas disponíveis na internet, basta buscar por frases com “validar certificado ssl instalado em servidor” e logo virão vários resultados para você validar seu serviço de instalação. Porém se quiser pode utilizar algum dos links abaixo que utilizo a alguns anos para validar o meu serviço

- Quality SSL Labs - SSL Server Test: https://www.ssllabs.com/ssltest/
- SSLShopper - SSL Checker (Uma das mais completas que conheço): https://www.sslshopper.com/ssl-checker.html

- RapidSSLonline Digicert: https://www.rapidsslonline.com/ssl-tools/ssl-checker.php

Uma dica para o uso destas ferramentas online é sempre utilizar mais de uma para validar, assim garante que não foi cache ou algo similar.

Como instalar o arquivo no servidor

Como pode ver a disponibilidade dos arquivos a serem criados é similar, independente do fornecedor e basicamente a instalação é a configuração por meio de edição de arquivos texto-simples em UTF-8.

Basta configurar e reiniciar o servidor, seja lá qual for seu ambiente e/ou aplicativo de servidor de páginas.

Esta dica vale para iniciantes: Para treinar você também pode utilizar certificados gratuitos!

Servidores Apache

No caso do Apache você indica nos parâmetros SSLCertificateFile, SSLCertificateKeyFile e SSLCertificateChainFile em uma definição de um Virtual Host respondendo a porta 443 nos arquivos de configuração (Ex: /etc/httpd/conf/httpd.conf) e indica que o uso do certificado pode ser ativo pela variável “SSLEngine” marcando ela como “on”. Veja abaixo um exemplo de virtual host com o SSL configurado.

```
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/ca.crt
    SSLCertificateChainFile /etc/pki/tls/certs/gd_bundle.crt
    SSLCertificateKeyFile /etc/pki/tls/private/ca.key
    DocumentRoot "/home/dominiocom/public_html"
    ServerName www.dominio.com
    ServerAlias dominio.com
    <Directory /home/dominiocom/public_html/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
```

```
        Order allow,deny
        allow from all
    </Directory>
    ErrorLog "/var/log/httpd/https-dominiocom-error_log"
    CustomLog "/var/log/httpd/https-dominiocom-access_log" common
    ServerSignature On
    LogLevel warn
</VirtualHost>
```

Servidores nginx

O nginx em minha opinião é mais simples, ao menos a sintaxe me parece mais legível, mas basicamente é a mesma coisa, acertar o caminho para os arquivos de Senha.

Note que aqui eu configurei apenas um arquivo de Certificado (no parâmetro ssl_certificate), isto pois tanto o certificado quanto o bundle estavam juntos no mesmo arquivo. O mesmo pode ser feito no Apache.

Note também que coloquei o ano em que o domínio foi gerado no nome do arquivo, com isso você tem a ideia de quando ele irá vencer simplesmente olhando para o arquivo.

```
server {
    listen 443;
    error_log /var/log/nginx/dominiocom-ssl-error.log;
    access_log /var/log/nginx/dominiocom-ssl-access.log;
    ssl on;
    ssl_certificate ssl/2015-dominiocom.crt;
    ssl_certificate_key ssl/dominiocom.key;
    ssl_session_cache shared:SSL:10m;
    ssl_prefer_server_ciphers on;
    ssl_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-
AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-
AES128-SHA256:ECDHE-RSA-RC4-SHA:ECDHE-RSA-AES256-
```

```
SHA:ECDHE-RSA-AES128-SHA:RC4-SHA:!aNULL:!eNULL:!LOW:!3DES:!
MD5:!EXP:!PSK:!SRP:!DSS:!RC4;
    include sites-available/dominiocom.inc;
}
```

Como checar se seu certificado tem validade

Basicamente você abre seu navegador e visualiza a informação, cada navegador tem um jeito. Fazem alguns anos alguns navegadores alteraram a forma de se visualizar, entretanto basta uma busca rápida na internet que você encontrará o caminho para visualizar em seu navegador. Você também pode pegar a informação diretamente na linha de comando, mas é um pouco mais chato. Em linux é bem tranquilo. Para facilitar eu simplesmente criei um script de linha de comando, nele adiciono todos os meus servidores que preciso monitorar e quando executa ele me apresenta um relatório indicando qual a data de validade dos certificados dos servidores que tenho responsabilidade.

Não poderia deixar de compartilhar o script neste livro. Para utilizar-lo, basta colocar os servidores entre parênteses na declaração da variável “$servers” que o script irá realizar a validação de todos os domínios um a um. Você pode inclusive implementar melhorias para avaliar quais irão vencer no decorrer do mês atual e encaminhar um relatório mensal para você no email.

```
#!/bin/bash

declare -a servers=("dominio1.com.br" "dominio2.com.br");

echo 'Data de validade dos servidores:'
for server in "${servers[@]}"; do
  echo -n "$server:";
  (echo | \
          openssl s_client -showcerts -servername $server
                  -connect $server:443 2>/dev/null | \
          openssl x509 -inform pem -noout -text) | \
          grep 'Not After' | \
          awk -F':' '{printf("%s:%s:%s\n", $2,$3,$4);}';
```

```
done
```

Possíveis problemas

O único problema que realmente conheço com Certificados SSL é quando a pessoa responsável por instalar não compreende a arquitetura de certificados e não instala o certificado da Autoridade Certificadora, se fizer um teste com uma das ferramentas que passei anteriormente verá que o certificado fica incompleto, mas alguns navegadores não irão apresentar erros, isto faz com que nós seguimos com o trabalho e só identifiquemos o erro um tempo depois, as vezes meses depois! Isto sim é um problema. No mais são questões de suporte comum com o cliente que até vou comentar, mas não são problemas com o certificado e sim com a falta de conhecimento das pessoas em relação a tecnologia de uma forma mais generalisada.

Gostaria de compartilhar dois cenários que vez por outra, como desenvolvedores e tecnólogos, temos em relação a certificados SSL. O primeiro e o mais comum é a validade do certificado. É muito mais que comum esquecermos, o problema não é nós como desenvolvedores esquecermos, o problema é quando o cliente ou até o administrador de sistemas esquece e o cliente acha que a culpa é nossa! Entenda que por mais que a culpa não seja sua o cliente vai entender que sua falta de interesse com o problema é um problema ainda maior do que o fato de outros terem esquecido a data de validade do bendito arquivo! Por isso, dependendo de seu envolvimento com o projeto, recomendo você a utilizar técnicas como o script que compartilhei acima para evitar maus bocados com seus clientes.

Nota do autor: História verídica para ilustrar a situação acima! Éramos gestores de loja virtual e tínhamos um parceiro de hospedagem para gerenciar o sistema operacional, o SGDB e etc., nós devíamos apenas cuidar do código e da loja, cadastrar promoções, produtos e realizar parte do atendimento ao cliente, mas vez por outra o email de notificação de validade do certificado era “ignorado” pelo responsável e ao invés de se organizar para não ter

> esse tipo de problema nos anos seguintes ele achou melhor botar a culpa em nossa equipe e fazer com que nós avisássemos ele no ano seguinte. Não me entenda mal, o cliente tem sempre razão, mas era o tipo de stress que vez por outra tínhamos com o cliente.

Outro problema que acontece é quando o cliente não consegue acessar o site corretamente. Tive um caso especial em que a máquina do cliente estava em outra rede, ele conseguia visualizar o site mas não o site correto que eu havia implementado de forma que ele achava que estava validando meu trabalho, mas não estava. Perdi uns 3 dias e alguns pontos no relacionamento com o cliente. Neste cenário temos que entender o que está acontecendo com o cliente e auxiliá-lo a entender o que ele está fazendo de errado.

Links para evolução do estudo

Blog ProgramaBrasil.org

É meu blog pessoal, uso ele mais como um bloco de notas aberto ao público, tem muitos tutoriais desatualizados e alguns que a maioria do público não entende, mas com o lançamento desta série de livros me comprometi a postar conteúdo mais orientado com o que estava escrevendo por isto se você busca atualização ou quer compartilhar suas observações sobre o livro vá até a página http://www.blogspot.com.br/ssh_private_key_files e faça um comentário no post.

Videos de Fábio Akita

Como disse você deve utilizar áudio e vídeo para complementar seus estudos, por isto gostaria de recomendar um vídeo para que você evolua este assunto. O vídeo que recomendarei não fala apenas de Chaves de Segurança, mas fala sobre Segurança Digital como um todo, são dois vídeos do Fábio Akita que é um empresário do setor de tecnologia, autor de um dos primeiros livros brasileiros sobre Ruby on Rails, um dos organizadores do RubyConf Brazil, palestrante e atualmente também youtuber.

Entendendo Conceitos Básicos de CRIPTOGRAFIA | Parte 1/2 https://youtu.be/CcU5Kc_FN_4

Entendendo Conceitos Básicos de CRIPTOGRAFIA | Parte 2/2

https://youtu.be/HCHqtpipwu4

Livro ITU-T X.509

Este livro é gratuíto e basicamente é o que descreve Certificados SSL

como conhecemos.

O título original é “SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY“

Link para documento: https://www.itu.int/itu-t/recommendations/rec.aspx?rec=13031

Livro CISSP CBK

O nome do livro é “Official (ISC)2 Guide to the CISSP CBK”. ISBN: 978-1-4822-6276-6. Editado por Adam Gordon.

O livro se auto intitula “O compêndio mais completo de conhecimento da indústria, elaborado pelos principais especialistas em segurança global. Um item obrigatório para quem busca obter a credencial “Profissional Certificado de Segurança de Sistemas de Informação“.

É uma obra completa sobre vários assuntos de segurança, dentre eles destaco no capítulo “Domain 4 - Comunications & Network Security” o item “Cryptography used to maintain communications security”.

Wikipedia

http://en.wikipedia.org/wiki/Ssh-keygen

https://en.wikipedia.org/wiki/Public_key_certificate

https://en.wikipedia.org/wiki/X.509

Uso de SSH com keys

http://sshkeychain.sourceforge.net/mirrors/SSH-with-Keys-HOWTO/SSH-with-Keys-HOWTO.html#toc4

RFCs (Request for Comments)

Para os que não conhecem os RFCs são as normas que determinam (ao menos deveriam determinar) a funcionalidade da internet.

Lista das RFCs de meu conhecimento importantes para o entendimento das tecnologias discutidas neste livro

- **4716**: The Secure Shell (SSH) Public Key File Format
- **4251**: The Secure Shell (SSH) Protocol Architecture
- **4252**: The Secure Shell (SSH) Authentication Protocol
- **4253**: The Secure Shell (SSH) Transport Layer Protocol
- **4254**: The Secure Shell (SSH) Connection Protocol
- **5280**: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile;
- **8017**: PKCS #1: RSA Cryptography Specifications Version 2.2