Esquemas de autenticação sem senha baseados na tecnologia FIDO

Palavras-Chave: FIDO, autenticador, autenticação, chaves

Autores:
Henrique Morais Filho – FEEC, UNICAMP
Prof. Dr. Marco Amaral Henriques – FEEC, UNICAMP

INTRODUÇÃO:

Com o avanço de formas de burlar a autenticação, os métodos de autenticação usuais, por login e senha, são frequentemente associados a sua falta de efetividade, demonstrando não serem tão seguros quanto se espera. Isso, somado à grande quantidade de serviços usados pelos usuários e, por consequência, a uma grande quantidade de senhas a serem geridas,ressalta a falta de eficiência desses métodos, o que leva a péssimas práticas, como a reutilização de senhas. A reutilização de senhas, assim como o uso de senhas fracas, pode causar sérios danos às pessoas e empresas, pois facilitam um ataque. É apontado que 81% dos roubos de dados e falsas autenticações estão relacionados ao uso de senhas fracas descobertas, ou de senhas fortes que foram roubadas [5].

Visando uma maior segurança e eficiência na autenticação, foi desenvolvido um novo sistema de autenticação que promete ser mais seguro: o FIDO (Fast Identity Online). Para isso, o protocolo FIDO realiza a autenticação com um par de chaves criptográficas pública/privada. Somente o usuário tem acesso à chave privada por meio de Client-to-Authenticator Protocols (CTAP) que são responsáveis por conectar de forma segura os dispositivos que contêm as chaves privadas (autenticadores) e o servidor. O servidor pede acesso ao autenticador para que seja provado que este possui as chaves privadas, prova essa feita ao assinar digitalmente um desafio gerado pelo servidor que requisitou a autenticação do usuário. O desafio é recebido e, após a identificação do usuário, a chave privada é utilizada para assinar o mesmo. A assinatura será enviada de volta para o servidor, onde será validada com a chave pública correspondente do usuário.

Nesse contexto, o objetivo deste estudo é analisar e avaliar as vantagens e desvantagens desse sistema de autenticação sem senha baseado no protocolo FIDO em diferentes contextos e para diferentes tipos de usuários, entendendo assim as dificuldades e os principais pontos de atenção para sua implementação.

METODOLOGIA:

Para a realização deste estudo, foram pesquisadas, estudadas e utilizadas diferentes formas de implementação da tecnologia FIDO tanto como um segundo fator de autenticação, quanto como único fator de autenticação. Essa tecnologia possui três padrões principais que podem ser implementados: UAF, U2F e FIDO2.

UAF: O padrão Universal Authentication Framework foi criado para substituir a autenticação por senhas em serviços online, como aplicativos e navegadores. Nele, o usuário se registra em um serviço online e escolhe um autenticador local, como a leitura de impressão digital. Após o registro, o usuário pode se autenticar utilizando o autenticador escolhido. Para implementar esse padrão, os serviços precisam instalar e utilizar um conjunto de aplicações e especificações que compõem o UAF [3], Nesse caso, o navegador ou aplicativo tem que se comunicar com o cliente, uma aplicação específica do FIDO como plugins e SDKs, e interagir com os autenticadores utilizando o Authenticador Specifc Modlue (ASM), como é mostrado na Figura 1.

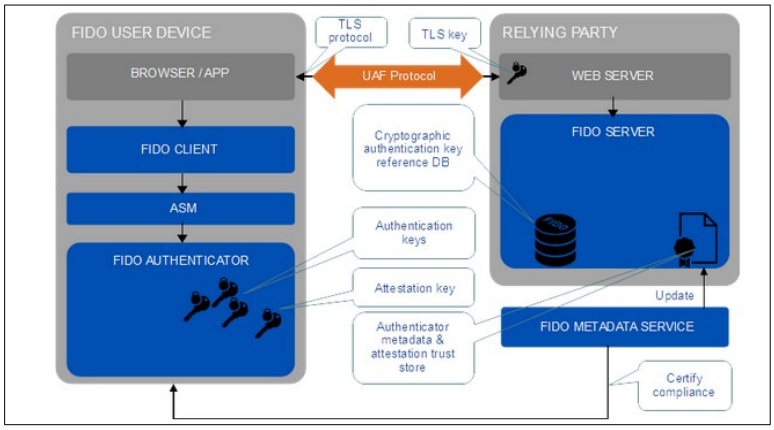

Figura 1: Ilustração do protocolo FIDO UAF em alto nível [3]

U2F: O Universal 2nd Factor foi criado para servir como um segundo fator de autenticação, no qual o usuário continua usando login e senha e seleciona um autenticador para ser o segundo fator de autenticação que possa se comunicar através de USB, NFC (Near Field Communication) ou BlueTooth LE (Low Energy). Agora U2F é nomeado CTAP1 devido ao lançamento do FIDO2.

FIDO2: É o padrão de autenticação mais atual, permitindo seu uso como segundo fator, autenticação multifator ou em substituição completa às senhas. Seu principal diferencial é a integração nativa com navegadores, através da API JavaScript WebAuthn da W3C, como mostrado na Figura 2, que gerencia a comunicação entre os serviços online, o navegador e o autenticador, simplificando sua adoção [2]. Este padrão utiliza o protocolo CTAP2 para a comunicação entre o autenticador e a plataforma [4], ao mesmo tempo que mantém retrocompatibilidade com autenticadores legados CTAP1. Ademais, possui suporte para autenticadores com biometria, a fim de confirmar que o usuário que estiver utilizando o autenticador seja, de fato, o dono dele [1]. Para utilizar o FIDO2, é necessário um autenticador compatível, como as chaves de segurança YubiKey ou smartphones, sendo preferíveis aqueles com sensores biométricos. Durante o registro em um serviço, a plataforma solicita a geração de chaves criptográficas, enviando dados do usuário, do serviço e um desafio.

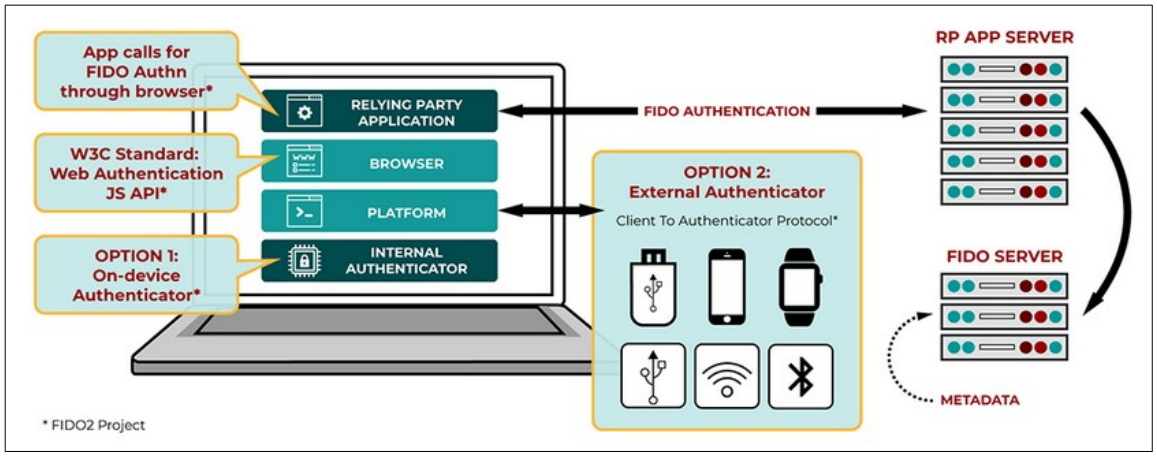

Figura 2: Ilustração do protocolo FIDO2 em alto nível [4]

O funcionamento do FIDO2 baseia-se nessa API WebAuthn, pela qual o navegador intermedeia a comunicação entre o serviço e o autenticador. Durante o registro, o autenticador gera um par de chaves, assina o desafio enviado pelo serviço e retorna a chave pública encapsulada em um *attestationObject*, que é armazenada no servidor. Na autenticação, o serviço envia um novo desafio, assinado pelo autenticador mediante autorização do usuário com a chave privada. O navegador retorna a assinatura e os metadados de segurança (*authenticatorData*), permitindo que o serviço valide a conformidade do dispositivo e confirme a identidade do usuário.

Na etapa de autenticação, o serviço gera um desafio que é enviado ao navegador e repassado ao autenticador. Com a autorização do usuário, a chave privada assina o desafio, e o resultado, juntamente com o objeto *authenticatorData*, retorna ao navegador. Esses dados são então encaminhados ao serviço, que valida o autenticador segundo seus parâmetros de segurança e confirma a correspondência entre a chave privada utilizada e a chave pública associada ao usuário.

Após a avaliação dos diferentes tipos de padrões dessa tecnologia, foram testadas diferentes implementações desses padrões para autenticação com servidores web (HTTPS) e SSH (Secure Shell), sendo elas: o WebAuthn.io, desenvolvido pelo Duo Labs para testar os autenticadores e a API WebAuthn para comunicações HTTPS; o Akamai Krypton MFA, que possui a finalidade de transformar o celular em um autenticador FIDO2 para autenticações via SSH, através de um aplicativo desenvolvido pela Akamai; e uma versão de FIDO virtual (de demonstração) criada pelo perfil bulwarkid do GitHub, chamada “Virtual-fido”, que visa utilizar um USB virtual na própria máquina do usuário para funcionar como um autenticador FIDO2.

O Akamai Krypton MFA, desenvolvido pela Akamai, propõe utilizar um aparelho celular para realizar autenticações via SSH por meio do aplicativo Akamai MFA, em conjunto com a ferramenta de linha de comando akr, que trabalha exclusivamente com esse aplicativo. Ao emparelhar o dispositivo móvel com o terminal por meio de um código QR, é possível gerar um par de chaves criptográficas. A chave privada permanece armazenada no dispositivo móvel e a chave pública é exibida no terminal para ser registrada no servidor desejado. Durante a autenticação via SSH, quando a chave pública

correspondente estiver presente no servidor, o aplicativo recebe uma notificação solicitando a confirmação do usuário para autorizar o uso da chave privada, permitindo, assim, a autenticação.

Já o Virtual-fido é uma versão muito inicial da utilização de um “USB virtual” que emula os protocolos CTAP do FIDO2 no computador local. Essa API cria um servidor IP/USB na rede local via TCP. As solicitações para a criação de credenciais são feitas pelo terminal que a API esteja rodando.

RESULTADOS E DISCUSSÃO:

Comparativo entre os diferentes padrões da tecnologia FIDO: o FIDO UAF não possui suporte nativo em navegadores e sistemas operacionais, exigindo configuração adicional por parte das aplicações cliente específicas do UAF para sua utilização. Isso dificulta sua ampla adoção por diferentes serviços online, tornando-o mais restrito a nichos. Já o U2F, agora CTAP1, integra-se ao FIDO2 e, por isso, passou a ser mais utilizado em conjunto com esse padrão, que se tornou mais popular em relação aos outros dois. Tendo isso em vista, as implementações da tecnologia FIDO testadas nesta pesquisa foram baseadas no padrão FIDO2.

Observa-se que nem todos os navegadores e sistemas operacionais dispõem de implementação nativa do FIDO. Navegadores como o Firefox e distribuições Linux como o Ubuntu, por exemplo, podem não oferecer suporte nativo ao protocolo. Em contrapartida, o Google integrou essa tecnologia ao Google Play Services, o que possibilita que dispositivos móveis que utilizem tais serviços atuem como autenticadores. A empresa também fez uma implementação nativa em navegadores vinculados ao ecossistema do Google. O sistema operacional Windows também oferece suporte ao FIDO2. Consequentemente, quando um navegador não implementa o FIDO de forma nativa, ele passa a depender do suporte do sistema operacional para viabilizar a autenticação; inversamente, em sistemas operacionais sem suporte nativo, a funcionalidade pode ficar a cargo do navegador.

Em relação às autenticações via SSH, tokens USB autenticadores que utilizam o FIDO2 são utilizados em conjunto com a coleção de protocolos de software livre OpenSSH para gerar o par de chaves por meio do terminal. É importante ressaltar que não há uma tecnologia consolidada para a utilização de outros autenticadores, como dispositivos móveis, para realizar essa operação via SSH, evidenciando uma limitação atual dessa tecnologia para autenticações com servidores SSH.

Implementações dessa tecnologia: Utilizando o WebAuthn.io, é possível testar de maneira simples a autenticação FIDO2 com diferentes autenticadores, como tokens USB e dispositivos móveis, em um servidor web. Para os testes, foi utilizado um aparelho celular Android como autenticador, por meio do qual foi possível criar e armazenar as chaves criptográficas utilizando o Google Play Services. Dessa forma, para autenticações em servidores HTTPS, o FIDO2 já se apresenta como uma tecnologia mais verstil e acessível aos usuários na maioria dos sistemas operacionais e navegadores modernos. Entretanto, o mesmo não se aplica a servidores SSH, uma vez que, atualmente, não há um método oficial para realizar a autenticação sem o uso de tokens USB.

Com o objetivo de avaliar alternativas ao uso exclusivo de tokens em autenticações via SSH, foram analisados o Akamai Krypton MFA e o Virtual-Fido. O Krypton, atualmente em versão beta,

demonstrou-se promissor como solução para autenticação nesse contexto, uma vez que apresenta configuração simplificada e operação semelhante ao uso de dispositivos móveis como autenticadores em servidores HTTPS. A principal diferença é que, nesse caso, as notificações de autorização de comandos são realizadas por meio do aplicativo da Akamai. Para validação prática, foi estabelecida uma conexão SSH com um repositório remoto no GitHub, a qual ocorreu de forma bem-sucedida.

No que se refere ao Virtual-Fido, por emular o prótocolo CTAP do FIDO2, é possível utiliza-lo em autenticações via HTTPS e SSH. Apesar de apresentar resultados positivos em testes preliminares, como no site da Yubico, sugerido pelos próprios desenvolvedores da aplicação, a solução ainda se mostra limitada, uma vez que não demonstrou operacionalidade em outros ambientes de validação, como o WebAuthn.io. Além disso, durante o processo de geração de chaves, o sistema expõe no terminal informações sensíveis que, em um cenário real, não deveriam ser reveladas, o que compromete sua viabilidade prática e segurança.

CONCLUSÕES:

A tecnologia FIDO, por meio do padrão FIDO2, mostra-se promissora para reduzir as fragilidades da (e os ataques dirigidos à) autenticação baseada em login e senha. Entretanto, o acesso a essa tecnologia ainda é um problema devido ao custo dos tokens USB, a ausência de alternativas consolidadas que permitam o uso de outros tipos de autenticadores (por exemplo, dispositivos móveis) em autenticações via SSH e a cobertura limitada de suporte em alguns sistemas operacionais e navegadores, o que restringe os usuários aos ambientes que suportam o FIDO.

BIBLIOGRAFIA

[1] FIDO Alliance. Client to authenticator protocol (ctap), 2019. https://fidoalliance.org/specs/fido-v2.0-ps-20190130/fido-client-to-authenticator-protocol-v2.0-ps-20190130.htmlauthenticatorMakeCredential.

[2] FIDO Alliance. Web authentication: An api for accessing public key credentials, 2019. https://www.w3.org/TR/webauthn/sctn-biometric-privacy.

[3] FIDO Alliance. Fido uaf architectural overview, 2020. https://fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-overview-v1.2-ps-20201020.htmlfido-uaf-goals.

[4] FIDO Alliance. User authentication specifications overview, 2020. https://fidoalliance.org/specifications/.

[5] C. Ward and N. Pritam, Cyberespionage and ransomware attacks are on the increase warns the Verizon 2017 Data Breach Investigations Report. Verizon News Archives, April 2017.

[6] bulwarkid. Virtual-FIDO. https://github.com/bulwarkid/virtual-fido/tree/master?tab=readme-ov-file.

[7] Akamai. Krypton akr FIDO2 SSH Agent. https://techdocs.akamai.com/mfa/docs/akr-fido2-ssh-agent.