Atualização e Carga de *Firmware* em Sistemas Embarcados de Forma Segura e Confiável

RESUMO EXPANDIDO - Disciplina de TCC29009

Paulo Fylippe Sell
Estudante do Curso de Engenharia de Telecomunicações

Roberto de Matos
Professor orientador
Emerson Ribeiro de Mello
Professor coorientador

Semestre 2019-2

Resumo- *Sistemas embarcados estão presentes em grande parte do dia-a-dia das pessoas. Equipamentos que monitoram a saúde, controlam casas e carros inteligentes ou ainda cidades inteiras permitem uma melhor qualidade de vida dos cidadãos. Desta forma, estes dispositivos precisam manter seus firmwares atualizados e seguros, evitando que ataques maliciosos que comprometam seu funcionamento sejam realizados. Tecnologias como Ambientes de Execução Confiável (do inglês Trusted Execution Environments, TEE), que executam processos sensíveis em um microcontrolador de forma reservada e isolada ao restante do sistema, ou Elementos Seguros (do inglês Secure Element, SE), combinados com algorítimos e premissas de criptografia, proveem condições para o desenvolvimento de soluções que garantem a confiabilidade e segurança de dados sensíveis em sistemas embarcados. Este trabalho tem o objetivo de estudar e propor uma maneira de fazer a atualização e carga de firmwares em sistemas embarcados.*

Palavras-chave: Atualização de *firmware.Trusted Execution Environments. Secure Element.*

1 Introdução

Dispositivos que possuem algum tipo de sistema embarcado compartilham um grande espaço na vida das pessoas. De cafeteiras à *smartphones*, estes equipamentos facilitam o cotidiano e proporcionam uma maior comodidade para as pessoas.

Uma vez implantados, estes dispositivos podem apresentar problemas não previstos durante o desenvolvimento, e uma alteração no seu *software* embarcado - também chamado de *firmware* - pode corrigir a falha. Desta forma, os aparelhos devem possuir algum

mecanismo de atualização de *firmware*, de preferência de forma remota, que previna ou corrija o mal funcionamento.

A possibilidade de atualizar o *firmware* de um dispositivo de forma remota trás consigo alguns benefícios. Primeiramente, evita que o usuário tenha que retornar o equipamento ao seu fabricante, poupando tempo e dinheiro do consumidor. Além disso, atualizar o *firmware* remotamente provê uma maior agilidade na melhoria dos equipamentos, uma vez que os dispositivos podem ser atualizados assim que o fabricante dos equipamentos disponibilizar a nova versão do *firmware* (Jain; Mali; Kulkarni, 2016).

A atualização remota de *firmware* pode ainda se dar de duas formas. Na primeira, o dispositivo detecta que existe uma atualização e aplica a melhoria de forma automática, como é o caso de alguns *smartphones*. Existem também casos onde o usuário do dispositivo deve verificar se existe um novo *firmware* disponibilizado pelo fabricante do equipamento e ele mesmo realizar o procedimento para o *upgrade*, como ocorre com alguns roteadores domésticos (Jain; Mali; Kulkarni, 2016). Neste trabalho, o foco será em atualização de *firmware* em dispositivos com sistemas embarcados em microcontroladores, como os *tokens* utilizados em certificados digitais.

Além de mecanismos lógicos de segurança, métodos à níveis de *hardware* podem ser aplicados para aumentar a segurança no momento da atualização de *firmware*. Ambientes de execução confiável (do inglês *Trusted Execution Environments*, TEE) permitem que aplicações confiáveis sejam processadas de forma isolada ao restante do sistema operacional, bem como permite que o *firmware* do dispositivo seja alocado em um ambiente seguro (GLOBALPLATFORM, 2018). Além de utilizar um TEE, também é possível guardar as chaves criptográficas utilizadas na segurança lógica dos dispositivos em plataformas invioláveis, como os elementos seguros (do inglês *Secure Element*, SE). Assim como o ambiente de execução confiável, o elemento seguro também garante que dados sensíveis sejam guardados em lugares protegidos e apenas aplicações autorizadas têm acesso a estes dados (SHEPHERD et al., 2016).

Diante do que foi posto, este trabalho tem como objetivo desenvolver uma solução que permita uma atualização remota de *firmware* de forma segura e confiável em dispositivos microcontrolados.

2 Metodologia

Para cumprimento do objetivo proposto, este trabalho será dividido em três etapas: estudo de premissas básicas de segurança e atualização de *firmware*, bem como estudo de ambientes de execução confiável e elementos seguros, com o intuito de avaliar qual melhor tecnologia se adéqua à solução; levantamento de regras negócio e casos de uso para entender o cenário de atuação da solução e, por fim, a implementação da solução, a fim de validar o desenvolvimento proposto.

2.1 Estudo de premissas de segurança e atualização de *firmware*

Nesta etapa, serão estudados os principais conceitos de segurança e criptografia, bem como estudar soluções de atualização de *firmware* já propostas, com o intuito de entender quais soluções podem ser adotadas para o desenvolvimento do trabalho.

2.2 Levantamento de regras de negócio e casos de uso

Regras de negócio e casos de uso devem ser propostos, a fim de ordenar o funcionamento da solução de atualização de *firmware*. Esta etapa deverá definir a sequência em que cada ator atuará no sistema.

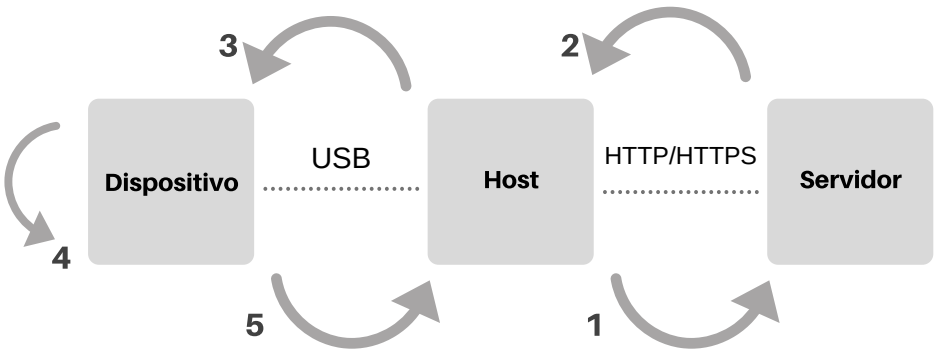
2.3 Desenvolvimento

A partir da base teórica obtida na primeira etapa e das definições de projeto feitas na segunda etapa, será realizado a implementação da solução, a fim de obter no final to trabalho uma prova de conceito realizada.

3 Considerações Parciais/Finais

Um cenário inicial de atualização de *firmware* para o trabalho já foi pensado em conjunto com os professores orientadores. Os componentes desta proposta, bem como a interação entre eles, são apresentados na figura 1. Nela, um dispositivo é conectado a um hospedeiro (computador), que fará o papel de verificar se existe uma nova versão de *firmware* em um servidor (passo 1). Após a resposta do servidor (passo 2), o hospedeiro informa ao dispositivo sobre a atualização e solicita a validação do *firmware* (passos 3 e 4). Caso o dispositivo valide o novo código (passo 5), o computador pode então transferir o novo *firmware* ao dispositivo conectado.

Figura 1 - Primeira proposta de cenário para o trabalho



Com a aplicação dos estudos teóricos realizados e definição da arquitetura de sistema da solução de atualização de *firmware*, espera-se obter um produto mínimo viável após a etapa de desenvolvimento deste trabalho, que deverá propiciar a atualização e carga de *firmware* em sistemas embarcados de forma segura e confiável.

Referências

GLOBALPLATFORM. Tratados e organizações ambientais em matéria de meio ambiente. In: _____. *Introduction to trusted execution environments*. 2018. Disponível em: <https://globalplatform.org/wp-content/uploads/2018/05/Introduction-to-Trusted-Execution-Environment-15May2018.pdf>. Acesso em: 31 ago. 2019.

Jain, N.; Mali, S. G.; Kulkarni, S. Infield firmware update: Challenges and solutions. In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.: s.n.], 2016. p. 1232–1236.

SHEPHERD, C. et al. Secure and trusted execution: Past, present and future – a critical review in the context of the internet of things and cyber-physical systems. In: . [S.l.: s.n.], 2016.