Caracterização de Cenários e Garantias de Segurança em Carregamentos de Software Aeronáutico

Johnny C. Marques[1], Sarasuaty Y. M. Hayashi[1], Lillian M. da Silva Barros[1]

[1]Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brasil

johnny@ita.br, sara.mhy@gmail.com, lilian_michele@yahoo.com.br

Abstract. *On-site software loading occurs when a new software version is needed to correct previously identified errors or include new features or improvements. Field-Loadable Software (FLS) provides external data transmitted through the loading of data in the aircraft from external connections that support maintenance. When there is an addition or modification of connectivity and interfaces, new opportunities for corruption and tampering of systems in the aircraft can be added. The objective of this work is to characterize the software loading scenarios in aircraft and for possible threats that involve information security in this process.*

Resumo. *O carregamento de software em campo acontece quando uma nova versão de software é necessária para corrigir erros previamente identificados ou incluir novas funcionalidades ou melhorias. O software carregado em campo (Field-Loadable Software - FLS) fornece dados externos transmitidos através do carregamento de dados na aeronave a partir das conexões externas que suportam a manutenção. Quando há uma adição ou modificação de conectividade e interfaces, novas oportunidades para corrupção e adulteração de sistemas na aeronave podem ser acrescentadas. O objetivo deste trabalho é caracterizar os cenários de carregamento de software em aeronaves e tratativas para possíveis ameaças que envolvam segurança da informação neste processo.*

1. Introdução

Na aviação civil, principalmente nos grandes jatos, o uso de tecnologias avançadas incorporadas aos sistemas, traduz bem a necessidade de softwares que atendam a um alto grau de tecnologia, combinado com a preocupação em segurança (Marques, 2018).

Agências reguladoras em setores críticos de segurança costumam exigir que os produtos, incluindo o software embarcado, atendam rigorosos requisitos de certificação, como a DO-178C (RTCA, 2011) na aviação. A segurança de voo é a prioridade no software embarcado de uso civil. Podendo todo um projeto de sistema computacional embarcado, cujo software faz parte, ser descartado se não cumprir normas de segurança previstas no processo de certificação (Marques, 2013).

De acordo com Lemes et. al (2003), a certificação de aeronaves é o processo pelo qual um solicitante requer a aprovação de uma autoridade reguladora, como Agência Nacional de Aviação Civil (ANAC), no Brasil, e o *Federal Aviation Administration* (FAA), nos Estados Unidos, em seu projeto aeronáutico. Os processos de certificação de aeronaves usam padrões, orientações, testes, métodos e procedimentos recomendados

para estabelecer a aprovação da certificação. Sendo assim, a segurança de aeronavegabilidade é assunto de interesse no processo de aprovação de sistemas computacionais embarcados da aeronave.

O objetivo deste trabalho é caracterizar os cenários de carregamento de software em aeronaves e tratativas para possíveis ameaças que envolvam segurança da informação neste processo. Este trabalho aprofunda, ilustra e detalha o item 7, "o potencial mal carregamento de atualizações de software nos sistemas da aeronave", previsto na listagem dos potenciais usos indevidos previstos na DO-326A (RTCA, 2014a), ainda inexistente na literatura e com prática não harmonizada entre fabricantes de aeronaves, oficinas de manutenção aeronáutica e companhias aéreas.

2. Contextualização

O desenvolvimento de software já se encontra padronizado na aviação através do uso da RTCA DO-178C (RTCA, 2011) e seus suplementos. Diversos trabalhos dos últimos anos, discutiram avanços e novas metodologias de desenvolvimento de software, em especial, nas seguintes áreas:

- Impactos na transição da DO-178B (RTCA, 1992) para a DO-178C (RTCA, 2011), como explorado nos trabalhos de Marcil (2012) e Youn et. al (2015);
- Desenvolvimento Baseado em Modelos, como explorado nos trabalhos de Sarkis & Dias (2014), Paz & Bousaidi (2016), Eisemann (2016) e Marques & Cunha (2018);
- Uso de Métodos Ágeis no Desenvolvimento de Software, como explorado nos trabalhos de Vanderleest & Buster (2009), Marques et. al (2013) e Marsden et. al (2018); e
- Verificação Formal, como percebido nos trabalhos de Moy et. al (2013) e Marques & Cunha (2017).

Assim, a literatura disponível ainda não endereçou cientificamente os possíveis cenários de carregamento de software aeronáutico. Apesar da DO-178B ter apresentado o conceito de FLS, a indústria ainda não possui um levantamento organizado de cenários e garantias escrito, o que motiva a existência deste trabalho.

O perímetro de segurança cataloga as partes da aeronave ou sistemas que contatam outros sistemas externos. Estas são as partes que suportam as interfaces e processos pelos quais um sistema pode ser afetado ou interagido (RTCA, 2014b).

Segurança de aeronavegabilidade (*airworthness security*) é a proteção necessária que uma aeronave deve prover para mitigar ameaças de segurança da informação. Essas ameaças aparecem como efeitos adversos na segurança devido a ação humana (intencional ou não intencional) usando acesso, uso, divulgação, negação, interrupção, modificação ou destruição de dados e/ou interfaces de dados. Isso inclui as consequências do dano realizado, dos dados falsificados e possíveis acessos de outros sistemas externos aos sistemas da aeronave.

De acordo com a DO-326A (RTCA, 2014a), a aeronavegabilidade de futuras aeronaves será impactada, devido ao uso intencional ou não intencional de sistemas de informação disponíveis nas aeronaves. Exemplos de uso indevido incluem:

1. O potencial de um *malware* infectar um sistema de aeronave;

2. O potencial de um invasor usar o acesso sem fio integrado para acessar as interfaces do sistema da aeronave;
3. O potencial de negação de serviço de interfaces sem fio;
4. O potencial de negação de serviço de sistemas críticos de segurança;
5. O potencial para uso indevido de dispositivos pessoais que acessam sistemas de aeronaves;
6. O potencial para uso indevido de conexões de rede off-board para acessar as interfaces dos sistemas da aeronave; e
7. O potencial mal carregamento de atualizações de software nos sistemas da aeronave.

De acordo com a DO-178C (RTCA, 2011), o software que pode ser FLS é software ou banco de dados que podem ser carregadas sem remover o sistema ou equipamento de sua instalação. Portanto, a instalação refere-se ao avião, ou seja, o software é carregado diretamente na aeronave sem remoção do hardware que o hospeda. O FLS permite que a companhia aérea e suas oficinas de manutenção credenciadas pelo fabricante da aeronave realizem a tarefa de carregamento.

O carregamento de software em campo acontece quando uma nova versão de software é necessária para corrigir problemas previamente identificados, incluir novas funcionalidades ou melhorias. O FLS fornece dados externos transmitidos através do carregamento de dados na aeronave a partir das conexões externas que suportam a manutenção. Quando há uma adição ou modificação de conectividade e interfaces que podem adicionar novas oportunidades para corrupção e adulteração através de sistemas na aeronave, as interfaces para gerenciamento de FLS devem ser incluídas no perímetro de segurança e na identificação de ameaças (RTCA, 2014b).

A Figura 1 apresenta uma enxuta ilustração deste processo. Basicamente um Hardware existentes na aeronave, hospeda um Software A e a companhia aérea recebe um Boletim de Serviço (BS) com instruções para a instalação e um Software B a ser instalado.

3. Cenários de Análise de Segurança

O carregamento de software em campo envolve uma tarefa que fica fora do domínio do fabricante da aeronave, por conta disso, alguns cenários foram identificados pelos autores deste trabalho.

Os fabricantes de aeronaves e de seus sistemas precisam garantir que as operações de carregamento aconteçam de forma segura, correta e completa. Na atividade de carregamento em campo é necessário assegurar que a configuração de software instalada seja a correta para determinada aeronave e sua operação.

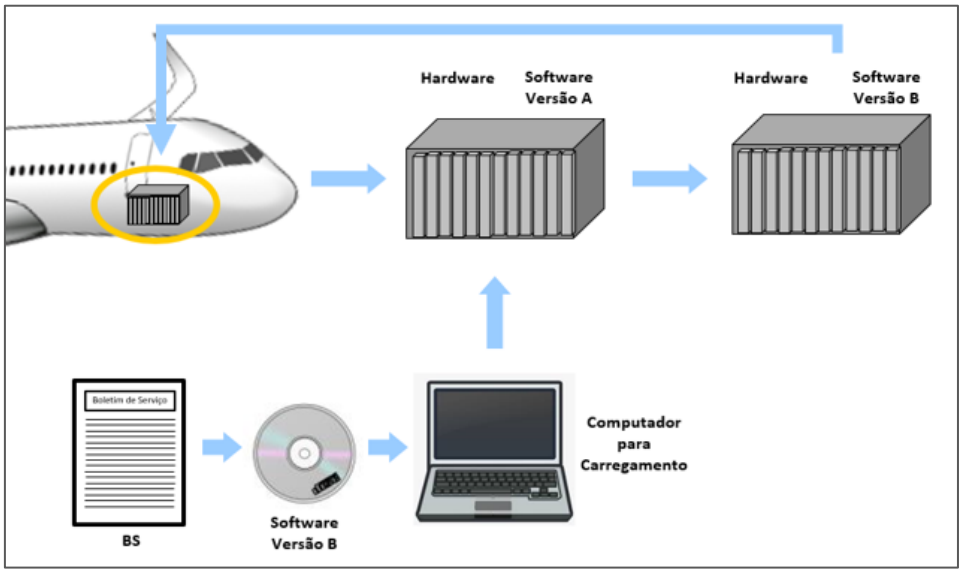

Figura 1. Ilustração do fluxo de carregamento de software em campo

3.1. Cenário 1: Carregamento do Software Impróprio

O Cenário 1 envolve a ameaça de um carregamento de software impróprio, ou seja, a oficina de manutenção faz o carregamento de um software que não é o software esperado a ser carregado.

Considerando que algumas empresas aéreas possuem uma frota diversificada, e em algumas bem numerosa, a gestão dos softwares recebidos pelos fabricantes e suas atualizações em frota é um trabalho intenso no dia-a-dia da companhia aérea. Em alguns casos, até aeronaves de modelos iguais exigem softwares distintos por conta de sua natureza de operação. Um exemplo disso é o Boeing 737-800 SFP (Short Field Performance) que opera em aeroportos com pistas curtas. Muitos de seus softwares podem ser diferentes dos que existem nos Boeing 737-800 padrão.

A Figura 2 apresenta um exemplo hipotético do Cenário 1, envolvendo o carregamento de um software impróprio. Um Boletim de Serviço é emitido pelo fabricante da aeronave XYZ, prevendo a instalação de um novo software para o modelo XYZ-100, no entanto, a companhia aérea pode se confundir e erroneamente realizar a instalação do software previsto para o XYZ-100 numa aeronave de modelo XYZ-200, que possui um software diferente para o sistema em questão.

3.2. Cenário 2: Carregamento de Software Incompleto

Um segundo cenário que envolve a ameaça ao carregamento de software é o chamado carregamento incompleto. Este Cenário 2 envolve o carregamento da versão correta de software na aeronave. No entanto, o carregamento não foi adequadamente completado pela oficina de manutenção responsável pela tarefa de carregamento via Boletim de Serviço (BS). Neste cenário, o software ter sido parcialmente carregado é uma ameaça para a segurança de voo. A Figura 3 apresenta um exemplo hipotético do Cenário 2, envolvendo o carregamento incompleto.

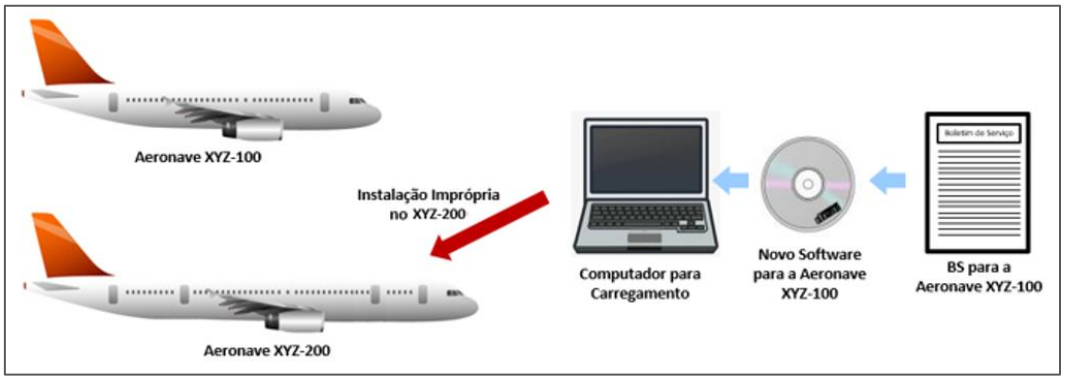

Figura 2. Ilustração do Cenário 1

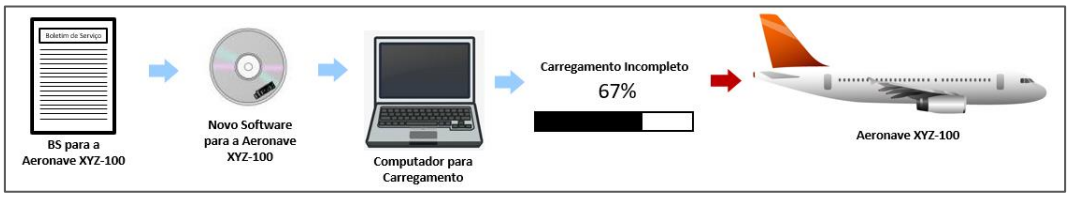

Figura 3. Ilustração do Cenário 2

3.3. Cenário 3: Não Carregamento de Software em Todas as Redundâncias

Em muitos sistemas críticos de segurança, como sistemas *fly-by-wire* e hidráulicos em aeronaves, algumas partes do sistema de controle podem ser triplicadas, o que é formalmente denominado Redundância Modular Tripla (RMT). Um erro em um componente pode ser superado pelos outros dois. Em um sistema triplamente redundante, o sistema tem três subcomponentes, todos os três devem falhar antes que o sistema falhe. Como cada um raramente falha, e espera-se que os subcomponentes falhem independentemente, a probabilidade de todos os três falharem é calculada como extraordinariamente pequena; frequentemente superados por outros fatores de risco, como erro humano (Skalaroff, 1976).

Durante a etapa de carregamento de software, se o sistema possui redundâncias, é necessário garantir que todas as redundâncias recebam a mesma atualização de software. A Figura 4 apresenta um exemplo hipotético do Cenário 3, onde o Software B foi carregado nos Hardwares 1 e 2, mas não foi carregado no Hardware 3.

4. Métodos para Garantia de Segurança no Carregamento

4.1. Garantias para Evitar o Carregamento de Software Impróprio

Uma forma de analisar os impactos do carregamento impróprio, é justamente testar um carregamento desta natureza antes de realizar a entrega de uma nova versão. Existem duas formas de evitar o carregamento de software impróprio: a) usando um Procedimento Manual de Checagem (PMC) a ser colocado no Boletim de Serviço; ou b) introduzindo um sistema computacional na própria aeronave que gerencie os carregamentos de software realizados.

O item a é o mais usual na indústria, já que nasceu naturalmente quando a capacidade de FLS foi introduzida em 1992 na DO-178B (RTCA, 1992). A Figura 5 apresentar uma visão hipotética do PMC.

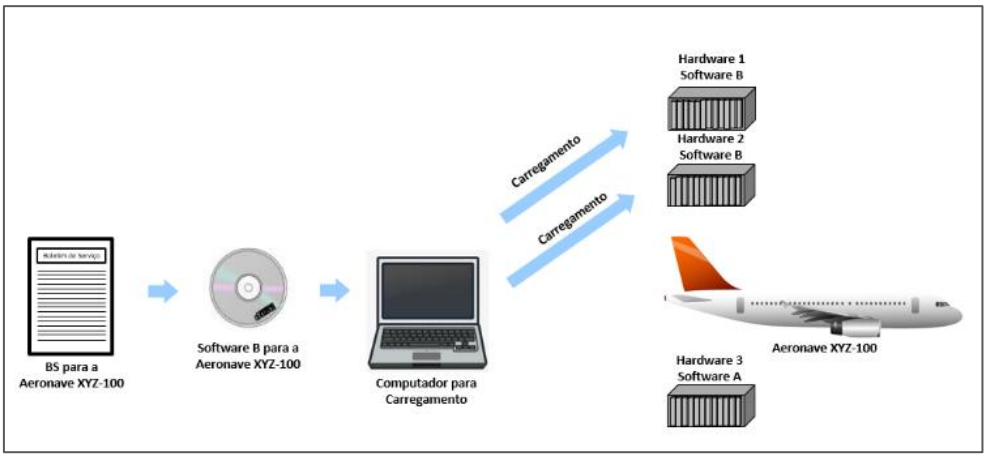

Figura 4. Ilustração do Cenário 3

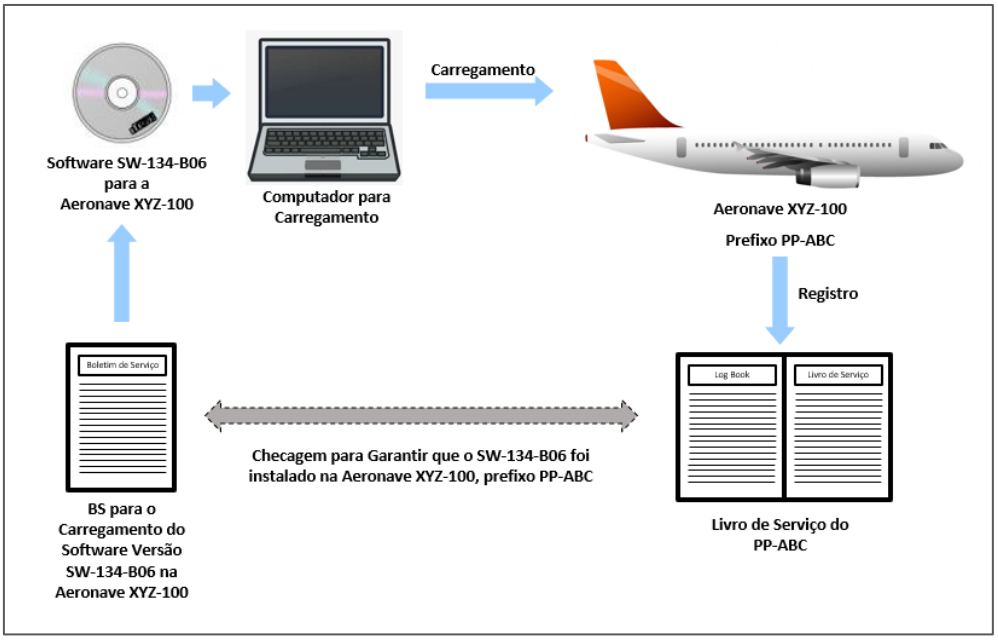

Figura 5. Carregamento com Procedimento Manual de Checagem (PMC)

O item 2 é mais recente, e surgiu no início dos anos 2000 com o uso intensivo de software nos projetos de aeronaves mais atuais. Basicamente, existe a necessidade de utilização de um Sistema de Controle de Configuração da Aeronave (SCCA) que gerencie todos os softwares instalados. A Figura 6 apresenta um carregamento hipotético de um software versão SW-134-B06 na Aeronave XYZ-100. Após o carregamento, o SCCA identifica a existência de uma nova versão e verifica se a mesma é uma configuração válida para aquele modelo de aeronave. Se a for válida, o SCCA atualiza a configuração existente e no caso contrário, uma mensagem é enviada ao cockpit, o que impedirá a operação da aeronave.

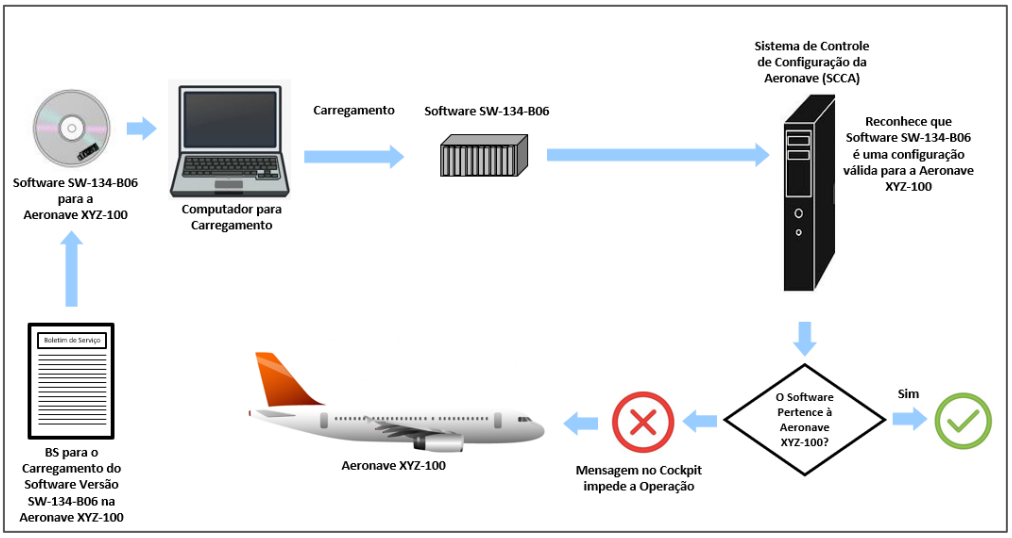

Figura 6. Carregamento com Sistema de Controle de Configuração da Aeronave (SCCA)

4.2. Garantias para Evitar o Carregamento de Software Incompleto

Para evitar o carregamento de software incompleto, é comum analisar a integridade do software carregado e verificar se este encontra-se completo. Os sistemas computacionais *safety-critical*, tipicamente fazem checagem de consistência em sua inicialização, até para assegurar que não houve corrupção no software instalado. Esta prática também é facilmente adaptável para assegurar o carregamento completo de um software recém-instalado.

A Figura 7 apresenta uma ilustração do carregamento com checagem de *Cyclic Redundancy Check* (CRC). Neste exemplo, encontra-se necessária a utilização de um software residente no hardware que receberá o novo carregamento de software. Esse software residente não é carregável e já vem instalado de fábrica no próprio hardware. Após realizado o carregamento, o software residente calcula o CRC do software instalado e compara com o CRC esperado e disponibilizado na ocasião da instalação. O carregamento só é considerado completo se o CRC Esperado e CRC Calculado forem idênticos. O trabalho de Rogers (2008) apresenta os diversos métodos disponíveis para o cálculo de CRC para FLS.

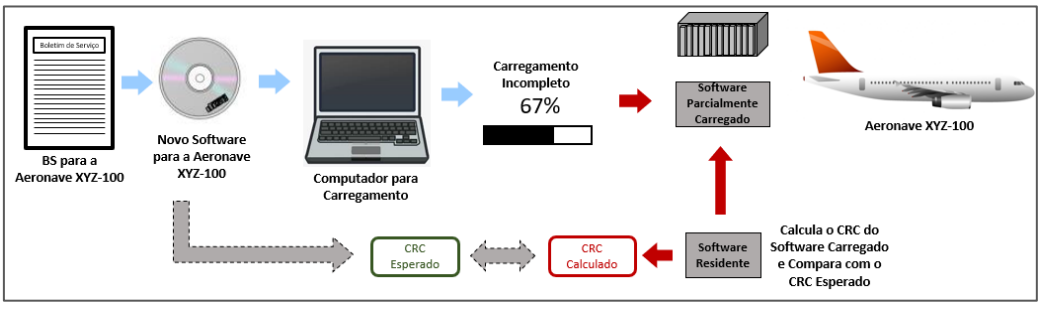

Figura 7. Carregamento com Checagem de CRC

4.3. Garantias para Conformidade no Carregamento de Software Redundante

De acordo com o relatório do *Center for Analysis for Risk and Regulation* (CAAR, 2009), a redundância é indispensável para um mundo em que os riscos tecnológicos devem ser

estreitamente regulados e onde a confiabilidade é interpretada como uma variável que pode ser definida e projetada. O trabalho de Dolega et al. (2016) apresenta a possibilidade de utilização de redundância de software em sistemas de controle de voo, que podem ser usadas em aeronaves não tripuladas (*Unmanned Aircraft Vehicles* – UAV) e na aviação geral.

A Figura 8 apresenta um exemplo hipotético de garantia para conformidade que poderia ser aplicado ao cenário 3, onde um Boletim de Serviço descreve a instalação de uma versão de software SW-134-BS06. O equipamento que hospeda este software tem redundância tripla, porém neste caso, apenas duas das três unidade foram atualizadas, cabendo ao SCCA indicar que uma das unidades encontra-se com software diferente das demais. Em sistemas de aeronaves mais antigas, portanto sem o SCCA, usualmente o próprio Livro de Serviço registraria o carregamento em cada uma das unidades, e a oficina de manutenção deveria verificar manualmente contra o Boletim de Serviço se todas as unidades foram atualizadas com a versão correta.

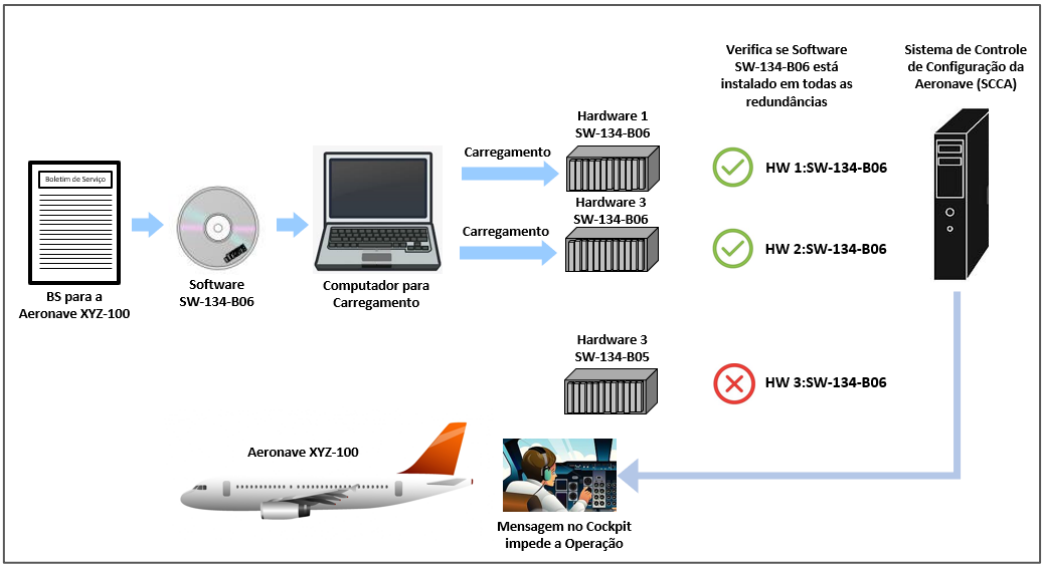

Figura 8. Carregamento com garantia para conformidade de versões de software entre as possíveis redundâncias existentes

5. Conclusão

Conforme apresentado na Seção 1, o objetivo deste trabalho foi caracterizar os cenários de carregamento de software em aeronaves e tratativas para possíveis ameaças que envolvam segurança da informação neste processo. Este trabalho aprofunda, ilustra e detalha o item 7 previsto na listagem dos potenciais usos indevidos previstos na DO-326A (RTCA, 2014a), ainda inexistente na literatura e com prática não harmonizada entre fabricantes de aeronaves, oficinas de manutenção aeronáutica e companhias aéreas.

A seção 3 caracterizou os cenários de carregamento de software. Além dos cenários elencados, também foi possível apresentar uma breve descrição de possíveis garantias. Para cada um dos cenários foram associadas as garantias para evitar possíveis ameaças de segurança da informação (*security*) identificadas nos cenários, que podem acarretar impactos críticos na operação do avião, ligados à segurança dos ocupantes e tripulação (*safety*). A Tabela 1 apresenta a correlação dos Cenários, Ameaças e Garantias.

Tabela 1. Correlação entre ameaças, cenários e garantias

Cenário	Ameaças Identificadas	Garantias Associadas
1	A aeronave opera com um software não válido para a sua configuração.	• Carregamento com Procedimento Manual de Checagem (PMC); e • Carregamento com Sistema de Controle de Configuração da Aeronave (SCCA).
2	A aeronave opera com um software não funcional ou com funcionalidades restritas.	• Carregamento com Checagem de CRC.
3	A aeronave opera com configurações diferentes de software coexistindo no mesmo sistema.	• Carregamento com garantia de conformidade de versões de software entre as possíveis redundâncias existentes.

A formalização destes cenários é a principal contribuição deste trabalho, já que apesar dos mesmos serem praticados, as normas e literatura não os descrevem. Estes cenários são um conjunto completo das utilizações recentes da indústria aeronáutica, mas não se exclui a possibilidade de extensão deles no futuro.

Como trabalho futuro, os autores pretendem estudar uma especialização de cenários, considerando por exemplo, aeronaves mais novas ou mais antigas, que podem ou não contar com o SCCA. Esta continuidade pode definir alguns critérios para apoiar novas hipóteses e definir um conjunto de requisitos e ajudar a estabelecer ensaios mais bem definidos, auditáveis e repetíveis. Outro ponto a explorar é o uso de um esquema de assinatura com certificado digital e como poderiam ser métodos de verificação nos três cenários.

Referências

CAAR (2009). "*When failure is an option: Redundancy, reliability and regulation in complex technical systems*". Em: Relatório do *Center for Analysis for Risk and Regulation.*

Dolega, B., Kopecki, G. and Tomczyk, A. (2016). *Possibilities of using software redundancy in low cost aeronautical control systems.* In. 2016 IEEE Metrology for Aerospace (MetroAeroSpace), páginas 33-37.

Eisemann, U. (2016). *Applying Model-Based Techniques for Aerospace Projects in Accordance with DO178C, DO-331, and DO-333*. Em: 8th European Congress on Embedded Real Time Software and Systems

Lemes, M. J. R., Altoé, F. O., Domiciano, A. J. and Carbonari, A. J. (2003). *Software certification in airborne systems: process and challenges.* Em: 2003 Latin American on Dependable Computing (LADC).

Marcil, L. (2012). Realizing DO-178C's value by using new technology: OOT, MBDV, TQC & FM. Em: 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)

Marques, J. C., Yelisetty, S. M. H., Cunha, A. M., Dias, L. A. V. (2013). *CARD-RM: A Reference Model for Airborne Software*. Em: 2013 10th International Conference on Information Technology: New Generations.

Marques, J. C., Cunha, A. M. (2017). *Verification scenarios of onboard databases under the RTCA DO-178C and the RTCA DO-200B*. Em: 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)

Marques, J. C., Cunha, A. M. (2018). *Tailoring Traditional Software Life Cycles to Ensure Compliance of RTCA DO-178C and DO-331 with Model-Driven Design*. Em: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC).

Marsden, J., Windisch, A., Mayo, R., Grossi, J., Villermin, J., Fabre, L., Aventini, C. (2018). *ED-12C/DO-178C vs. Agile Manifesto – A Solution to Agile Development of Certifiable Avionics Systems*. Em: 9th European Congress of Embedded RealTime Software and Systems.

Moy, Y., Ledinot, E., Delseny, H., Wiels, V., Monate, B. (2013). *Testing or Formal Verification: DO-178C Alternatives and Industrial Experience.* Em: IEEE Software (Volume:30, Issue:3), páginas 50-57.

Paz, A., Bousaidi, G. (2016). *On the Exploration of Model-Based Support for DO-178C-Compliant Avionics Software Development and Certification*. Em: 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW).

Rogers, C. (2008). *Choosing a CRC & specifying its requirements for field-loadable software.* Em. 2008 IEEE/AIAA 27th Digital Avionics Systems Conference.

RTCA (1992). "*DO-178B - Software Considerations in Airborne Systems and Equipment Certification*", Washington, Estados Unidos.

RTCA (2011). "*DO-178C - Software Considerations in Airborne Systems and Equipment Certification*", Washington, Estados Unidos.

RTCA (2014a). "*DO-326A - Airworthiness Security Process Specification*", Washington, Estados Unidos.

RTCA (2014b). "*DO-356 - Airworthiness Security Methods and Consideration*", Washington, Estados Unidos.

Sarkis, A., Dias, L. A. V. (2014). *A Set of Rules for Production of Design Models Compliant with Standards DO-178C and DO-331*. Em: 2014 11th International Conference on Information Technology: New Generations.

Sklaroff, J. R. (1976). *Redundancy Management Technique for Space Shuttle Computers*. Em. IBM Journal of Research and Development (Volume: 20, Issue: 10).

VanderLeest, S. H., Andrew Buter, A. (2009). *Escape the waterfall: Agile for aerospace*. Em: 2009 IEEE/AIAA 28th Digital Avionics Systems Conference.

Youn, W. K., S Hong, S. B., Oh K. R., Sung Ahn O. S. (2015). *Software certification of safety-critical avionic systems: DO-178C and its impacts.* Em: IEEE Aerospace and Electronic Systems Magazine (Volume: 30, Issue: 4), páginas 4-13.