Sumário

Histórico do Documento					
Versão	Elaborado por	Data	Revisado por	Data	Principais Alterações

Objetivo

Prover aos membros da equipe de infraestrutura um framework para avaliar o nível de requerimentos de segurança que deve ser utilizado e guiar a equipe de Segurança Operacional na seleção destes requisitos para sua implementação, garantindo assim uma maior segurança.

Introdução

A implementação de uma metodologia que possa auxiliar no desenvolvimento seguro é um dos principais pontos dentro de um processo de segurança. Os requisitos apresentados neste documento foram definidos com base nas especificações propostas pelo Open Web Application Security Project® (OWASP) e Center For Internet Security (CIS).

Níveis de Implementação

O documento apresenta três níveis de verificação que devem ser observados durante o processo de configuração do serviço. Cada um dos níveis aqui apresentados tem sua particularidade e profundidade.

Nível 1 - Segurança Mínima Necessária	Nível 2 - Padrão para web	Nível 3 - Alto Valor, Garantia, Segurança e Resiliência
Para níveis baixos de garantia de segurança. Apresenta o nível mínimo de segurança esperado para qualquer aplicação web.	Para aplicações que contêm dados sensíveis. É o nível recomendado para a maioria das aplicações.	Para as aplicações mais críticas, que requerem o mais alto nível de confiança.

Nível 1 - Segurança mínima

Para se adequar a este nível, a aplicação não deve possuir vulnerabilidades de fácil descoberta e exploração, bem como as listadas no TOP 10 da OWASP ou mesmo em outras listas de vulnerabilidades similares classificadas como críticos e altos.

O nível 1 é considerado o mínimo de segurança para qualquer aplicação, no entanto, se os dados que devem ser protegidos por sua aplicação são críticos, raramente somente o Nível 1 trará o nível de proteção recomendado de segurança.

Nível 2 – Segurança Recomendada

Um serviço atinge o nível 2 se tiver resiliência suficiente contra a maioria dos riscos associados a ataques através de recursos externos. Portanto, deve ser utilizado como o nível padrão de segurança. Este nível, que é apropriado para aplicações que tratam dados sensíveis, garante que os controles de segurança existem, funcionam e são utilizados dentro da aplicação.

Ameaças a aplicações neste nível serão tipicamente feitas por atacantes habilidosos e motivados, focando em alvos específicos, utilizando ferramentas e técnicas efetivas para o descobrimento e exploração de fraquezas dentro de aplicações.

Características da Aplicação
Armazena, transmite ou processa dados pessoais, confidenciais ou sensíveis de funcionários, parceiros de negócio, terceiros ou prestadores de serviço
Aplicações cuja integridade é crítica.

Nível 3 – Segurança Alto valor, garantia e resiliência

Este nível é o mais alto e é normalmente reservado para serviços que requerem níveis significativos de verificações de segurança, como em situações de infraestrutura crítica, quando realizam processos críticos para o negócio ou funções sensíveis, quando é necessária a garantia da sua integridade para o funcionamento do negócio, ou seja, nível de segurança 3 é exigido em cenários onde uma falha pode provocar um impacto significativo nas operações da organização.

Características da Aplicação
Aplicações que executam funções críticas, onde a falha pode afetar significativamente as operações. Ex: uma aplicação interna que seja utilizada por várias outras, Sefaz Identity. Aplicações que manipulam informações sigilosas ou que realizam processos críticos para o negócio ou funções sensíveis. Ex.: manipulação de informações sujeitas ao sigilo fiscal.

Seleção dos Requisitos de Segurança

Para o processo de seleção dos requisitos de segurança ser eficiente, deve-se aplicar uma avaliação mista, levando em consideração os ativos que devem ser protegidos e os riscos mapeados.

O anexo 1 apresenta uma lista de requisitos de segurança propostos, divididos nos três níveis de segurança. Além disso, os requisitos propostos nos anexos, não deverão ser tratados como únicos. Caso seja notada a necessidade de adição de um requisito, o mesmo pode e deve ser adicionado à sua respectiva tabela dentro do respectivo anexo.

A partir destes anexos, o PO irá gerar a Lista de Requisitos de Segurança, que será passada para a fase de sprint de desenvolvimento. Para gerar esta Lista de Requisitos de Segurança do Projeto, considere utilizar ferramentas, Kanban boards, Scrum ou tabelas que facilitarão na seleção e acompanhamento da implementação dos requisitos.

Anexo 1 – Requisitos de Segurança

Este anexo apresenta os controles que devem ser avaliados, e caso atendam às necessidades de segurança da solução, devem ser aplicados. Esta lista de requisitos foi baseada primordialmente no CIS (Center For Internet Security) que apresenta uma lista das técnicas de boas práticas que deveriam ser incluídas em toda configuração inicial de recursos com finalidade de desenvolvimento de software.

1. Requisitos de Configuração Docker

1.1. Host Configuration

Esta seção cobre as recomendações de segurança para preparar a máquina host que planeja executar cargas de trabalho em contêiner. Proteger o host Docker e seguir as práticas recomendadas de segurança de sua infraestrutura criaria uma base sólida e segura para a execução de cargas de trabalho em contêineres.

#	Descrição	1	2	3	CWE
1.1.1.	Recomenda-se que a versão do Docker seja a mais atualizada. Ao se manter as atualizações do Docker, as vulnerabilidades do software podem ser atenuadas. Um invasor experiente pode ser capaz de explorar vulnerabilidades conhecidas que resultam nelas ser capaz de obter acesso inadequado ou elevar privilégios. Docker_Page: 16	✓			
1.1.2.	Recomenda-se a implementação de um controle de permissão sobre o serviço Docker Daemon. O Daemon Docker atualmente requer acesso ao soquete Docker que é, por padrão, pertencente ao usuário root e ao docker de grupo. O Docker permite que seja compartilhado um diretório entre o host Docker e um contêiner de convidado sem limitar os direitos de acesso do contêiner. Isso significa que pode iniciar um container e mapear o diretório “/” em seu host para o container. O contêiner então pode ser capaz de modificar seu sistema de arquivos host sem quaisquer restrições. Isso significa a probabilidade de ganhos de privilégios elevados simplesmente por ser um membro do grupo docker e, posteriormente, iniciar um contêiner que mapeia o diretório root ”/” no host. Docker_Page: 20				
1.1.3.	Recomenda-se que a auditoria esteja habilitada e configurada para o Docker e arquivos do diretório /var/lib/docker. Além de auditar o sistema de arquivos normal do Linux e chamadas de sistema, deve-se auditar o Daemon do Docker. O daemon Docker é executado com privilégios de root e seu o comportamento depende de alguns arquivos e diretórios principais /var/lib/docker é um desses diretório. Como ele contém todas as informações sobre os containers, ele deve ser auditado. Docker_Page: 22 e 24	✓			

1.1.4.	Recomenda-se que a auditoria esteja habilitada e configurada para o Docker e arquivos do diretório /etc/docker. Este diretório contém vários certificados e chaves usadas para comunicação TLS entre daemon Docker e cliente Docker. Docker_Page: 26	✓			
1.1.5.	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios docker.service. O arquivo docker.service pode estar presente se os parâmetros do daemon tiverem sido alterado por um administrador. Em caso afirmativo, ele contém vários parâmetros para o daemon Docker e deve ser auditado. Docker_Page: 28	✓			
1.1.6.	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios docker.socket. O arquivo docker.socket contém informações de parâmetros para o docker Daemon e deve ser auditado. Docker_Page: 30	✓			
1.1.7.	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios /etc/default/docker. Este diretório contém informações de parâmetros para o docker Daemon e deve ser auditado. Docker_Page: 32	✓			
1.1.8.	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios /etc/sysconfig/docker. Este diretório contém vários parâmetros relacionados ao Daemon Docker quando executado em distribuições baseadas em CentOS e RHEL. Se estiver presente, é importante que seja auditado. Docker_Page: 34	✓			
1.1.9.	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios /etc/docker/daemon.json. Este diretório contém informações de parâmetros para o docker Daemon e deve ser auditado. Docker_Page: 36	✓			

1.1.10	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios /usr/bin/containerd. Docker_Page: 38	✓			
1.1.11	Recomenda-se que a auditoria esteja habilitada e configurada para arquivos e diretórios /usr/sbin/runc Docker_Page: 40	✓			

1.2. Docker Daemon Configuration

Esta seção lista as recomendações que alteram e protegem o comportamento do Docker daemon. As configurações nesta seção afetam TODAS as instâncias de contêiner. As opções de daemon do Docker também podem ser controladas usando arquivos como /etc/sysconfig/docker, /etc/default/docker, o arquivo da unidade systemd ou /etc/docker/daemon.json.

1.2.1.	Recomenda-se que o tráfego de rede seja restrito entre os contêineres. Por padrão, o tráfego de rede irrestrito é habilitado entre todos os contêineres no mesmo host. Assim, cada contêiner tem o potencial de ler todos os pacotes através da rede de contêiner no mesmo host. Isso pode levar a um não intencional erro de divulgação indesejada de informações a outros contêineres. Docker_Page: 43	✓			
1.2.2.	Recomenda-se que esteja ativada a autorização para comandos do cliente Docker. Deve-se utilizar plug-ins de autorização nativos do Docker ou uma autorização de terceiros com o daemon Docker para gerenciar o acesso aos comandos do cliente Docker. O modelo de autorização out-of-the-box do Docker é atualmente "tudo ou nada". Isso significa que qualquer usuário com permissão para acessar o daemon Docker pode executar qualquer comando cliente Docker. O mesmo ocorre para usuários remotos que acessam a API do Docker para entrar em contato com o daemon. Docker_Page: 62				
1.2.3.	Recomenda-se que o nível de registro esteja definido como “info”. Não executar o daemon do Docker na depuração nível de registro Docker_Page: 44	✓			

1.2.4.	Recomenda-se que os registros não seguros não sejam utilizados. Um registro seguro usa TLS. Uma cópia do certificado de CA do registro é colocada no host Docker em /etc/docker/certs.d/<registry-name>/diretório. Um registro inseguro é aquele que não tem um certificado de registro válido ou um que não usa TLS. Registros inseguros não devem ser usados, pois apresentam risco de interceptação e modificação do tráfego. Docker_Page: 48	✓			
1.2.5.	Recomenda-se que a autenticação TLS para Docker Daemon esteja configurada. Por padrão, o daemon Docker se liga a um soquete Unix não conectado à rede e é executado com privilégios root. Se alterar a ligação padrão do daemon do Docker para uma porta TCP ou qualquer outro Soquete Unix, qualquer pessoa com acesso a essa porta ou soquete pode ter acesso total ao Docker daemon e, portanto, por sua vez, para o sistema host. Caso tenha que expor o daemon Docker por meio de um soquete de rede, deve-se configurar o TLS autenticação para o daemon e para quaisquer APIs Docker Swarm (se estiverem em uso). Este tipo de configuração restringe as conexões com o daemon do Docker pela rede para um número limitado de clientes que têm acesso às credenciais do cliente TLS. Docker_Page: 52	✓			
1.2.6.	Recomenda-se que o registro centralizado e remoto esteja configurado. Um método preferível para armazenar logs é um que suporta gerenciamento centralizado e remoto. O registro centralizado e remoto garante que todos os registros de registro importantes estejam seguros, mesmo no evento de um grande problema de disponibilidade de dados. Docker suporta vários métodos de registro. Docker_Page: 64	✓			
1.2.7.	Recomenda-se que o “Live Restore” esteja habilitada apara atender a tríade de disponibilidade. Configurando o sinalizador --live-restore em o daemon do Docker garante que a execução do contêiner não seja interrompida quando não é acessível. Isso também torna mais fácil atualizar e corrigir o daemon do Docker sem tempo de inatividade do aplicativo. Docker_Page: 66	✓			

1.2.8.	Recomenda-se que os containers sejam impedidos de adquirem novos privilégios. Por padrão, deve-se restringir os contêineres de adquirir privilégios adicionais via suid ou sgid. O bit no_new_priv garante que o processo não ganhe quaisquer privilégios adicionais por meio de bits suid ou sgid. Isso reduz os riscos de segurança associado a muitas operações perigosas porque há uma capacidade muito reduzida de subverter binários privilegiados. Definir isso no nível daemon garante que, por padrão, todos os novos contêineres sejam restritos de adquirir novos privilégios. Docker_Page: 74	✓			

1.3. Docker Daemon Configuration Files

Esta seção cobre os arquivos relacionados ao Docker e as permissões e propriedade do diretório. Guardando os arquivos e diretórios, que podem conter parâmetros confidenciais, a segurança é importante para funcionamento correto e seguro do daemon do Docker.

1.3.1.	Recomenda-se que a propriedade do docker.service esteja definida como root:root. O arquivo docker.service contém parâmetros confidenciais que podem alterar o comportamento do Daemon do Docker. Deve, portanto, pertencer ao grupo pelo usuário root e garantir que não seja modificado ou corrompido por um usuário menos privilegiado. Docker_Page: 76 e 78	✓			
1.3.2.	Recomenda-se que a propriedade do docker.socket esteja definida como root:root. O arquivo docker.socket contém parâmetros confidenciais que podem alterar o comportamento do Daemon do Docker. Deve, portanto, pertencer ao grupo pelo usuário root e garantir que não seja modificado ou corrompido por um usuário menos privilegiado. Docker_Page: 80	✓			

1.3.3.	Recomenda-se que as permissões do docker.socket sejam definidas para 644 ou superior. O arquivo docker.socket contém parâmetros confidenciais que podem alterar o comportamento da API remota do Docker. Deve, portanto, ser gravável apenas pelo root, a fim de garantir que não é modificado por usuários menos privilegiados. Docker_Page: 82				
1.3.4.	Recomenda-se que a propriedade do diretório /etc/docker esteja definida com permissões root:root. O diretório contém certificados e chaves, além de vários outros arquivos sensíveis. Docker_Page: 84	✓			
1.3.5.	Recomenda-se que a propriedade do diretório /etc/docker esteja definida com permissões 755 ou superior. O diretório contém certificados e chaves, além de vários outros arquivos sensíveis. Garantindo que nenhum usuário menos privilegiado consiga realizar alterações nesse diretório. Docker_Page: 86				
1.3.6.	Recomenda-se que a propriedade do arquivo de certificado de registro esteja definida como root:root. O diretório /etc/docker/certs.d/ <registry-name> contém o registro do Docker certificados. Esses arquivos de certificado devem ser de propriedade individual e grupo de propriedade root para garantir que usuários menos privilegiados não consigam modificar o conteúdo do diretório. Docker_Page: 88	✓			
1.3.7.	Recomenda-se que as permissões do arquivo de certificado de registro sejam definidas para 444. O diretório /etc/docker/certs.d/ <registry-name> contém o registro do Docker certificados. Esses arquivos de certificado devem ter permissões de 444 ou mais restritivas permissões para garantir que usuários sem privilégios não tenham acesso total a eles. Docker_Page: 90				

1.3.8.	Recomenda-se que a propriedade do arquivo de certificado CA TLS esteja definida como root:root. O arquivo de certificado TLS CA deve ser protegido de qualquer violação. É usado para autenticar o servidor Docker com base em um determinado certificado CA. Deve ser, portanto, propriedade individual e grupo de propriedade do root para garantir que não possa ser modificado por menos usuários privilegiados. Docker_Page: 92	✓			
1.3.9.	Recomenda-se que a permissões do arquivo de certificado TLS CA sejam definidas para 444 ou mais restritiva. O arquivo de certificado TLS CA deve ser protegido de qualquer violação. É usado para autenticar o servidor Docker com base em um determinado certificado CA. Deve ser, portanto, propriedade individual e grupo de propriedade do root para garantir que não possa ser modificado por menos usuários privilegiados. Docker_Page: 94				
1.3.10.	Recomenda-se que a propriedade do arquivo de certificado do Docker Server seja definida como root:root e deve ser atribuído permissão 444 ao arquivo. O arquivo de certificado do servidor Docker deve ser protegido de qualquer violação devido ser usado para autenticar o servidor Docker com base no certificado de servidor fornecido. Docker_Page: 96 e 98.	✓			
1.3.11.	Recomenda-se que a propriedade do arquivo da chave do certificado do servidor Docker esteja definida como root: root e que o arquivo esteja com permissão 400. O arquivo da chave do certificado do servidor Docker deve ser protegido de qualquer violação ou uso desnecessário. Ele contém a chave privada para o certificado do servidor Docker. Deve, portanto, ter permissões de 400 para garantir que o arquivo de chave do certificado não seja modificado. Docker_Page: 100 e 102				

1.3.12.	Recomenda-se que a propriedade do arquivo de soquete Docker esteja definida como root:docker e dever ser atribuído permissão 660. O daemon do Docker é executado como root. O soquete Unix padrão, portanto, deve pertencer a root. Se qualquer outro usuário ou processo possuir este soquete, pode ser possível que esse usuário ou processo não privilegiado interaja com o daemon do Docker. A associação do grupo docker é rigidamente controlada pelo administrador do sistema. No entanto, se qualquer outro grupo possuir este soquete, então pode ser possível para os membros desse grupo interagirem com o daemon do Docker. Tal grupo pode não ser tão controlado quanto o grupo docker. Novamente, isso não está de acordo com a boa prática de segurança. Por esse motivo, o arquivo de soquete Docker Unix padrão deve pertencer ao root e ao grupo de propriedade da docker para manter a integridade do socket. Docker_Page: 104 e 106	✓			
1.3.13.	Recomenda-se que a propriedade do arquivo daemon.json esteja definida como root:root e que o arquivo esteja com permissão 644. . O arquivo daemon.json contém parâmetros sensíveis que podem alterar o comportamento do daemon do docker. Deve, portanto, ser propriedade e propriedade do grupo pelo root para garantir que possa não ser modificado por usuários menos privilegiados. Docker_Page: 108 e 110	✓			
1.3.14.	Recomenda-se que a propriedade do arquivo /etc/default/docker está definida como root: root. O arquivo /etc/default/docker contém parâmetros confidenciais que podem alterar o comportamento de o daemon Docker. Deve, portanto, ser propriedade individual e pertencer ao grupo pelo root para que não possa ser modificado por usuários menos privilegiados. Docker_Page: 112	✓			

1.3.15.	Recomenda-se que a propriedade do arquivo /etc/sysconfig/docker esteja definida como root: root e com permissão 644. O arquivo /etc/sysconfig/docker contém parâmetros confidenciais que podem alterar o comportamento do daemon Docker. Deve, portanto, ser propriedade individual e pertencer ao grupo pelo root para garantir que não seja modificado por usuários menos privilegiados. Docker_Page: 114 e 116.				

1.4. Container Images

Imagens de contêiner e arquivos de compilação gerenciam os fundamentos de como uma instância de contêiner de uma determinada imagem se comportaria. Garantir que esteja usando imagens de base adequadas e arquivos de compilação apropriados podem ser muito importantes para a construção de sua infraestrutura.

1.4.1.	Recomenda-se que seja criado um usuário além do root. É uma boa prática executar contêineres com usuário não root. Isso pode ser feito seja por meio da diretiva USER no Dockerfile, quando usado como parte das diretivas CMD ou Entrypoint. Docker_Page: 120	✓			
1.4.2.	Recomenda-se que os containers utilizem imagens de bases confiáveis. Os repositórios oficiais contêm imagens do Docker selecionadas e otimizadas pelo Docker comunidade ou por seu fornecedor. Não há garantia de que essas imagens sejam seguras e não contêm vulnerabilidades de segurança ou código malicioso. Deve-se ter cuidado, portanto, ao obter imagens de contêiner do Docker e de terceiros e executar essas imagens deve ser revisado de acordo com a política de segurança organizacional. Docker_Page: 122				
1.4.3.	Não utilizar pacotes desnecessários no container. Softwares desnecessários não devem ser instalados em contêineres, pois isso aumenta sua superfície de ataque. Docker_Page: 124				

1.4.4.	Recomenda-se que as imagens sejam digitalizadas e reconstruídas para incluir patches de segurança. Imagens devem ser verificadas com frequência em busca de vulnerabilidades. Deve-se reconstruir todas as imagens para incluir esses patches e, em seguida, instanciar novos contêineres a partir deles. Docker_Page: 126				
1.4.5.	Recomenda-se que a confiança para o docker esteja habilitada. A confiança de conteúdo é desabilitada por padrão e deve ser habilitada de acordo com política de segurança. A confiança de conteúdo fornece a capacidade de usar assinaturas digitais para dados enviados e recebidos de registros remotos do Docker. Essas assinaturas permitem a verificação do lado do cliente do identidade e o editor de tags de imagem específicas e garante a procedência do contêiner imagens. Docker_Page: 128	✓			
1.4.6.	Recomenda-se que as permissões setuid e setgid sejam removidas. Remover as permissões setuid e setgid nas imagens pode evitar o aumento de privilégios ataques dentro de contêineres. As permissões setuid e setgid podem ser usadas para escalonamento de privilégios. Enquanto estas permissões podem ser legitimamente necessárias às vezes, deve considerar removê-las de pacotes que não precisam deles. Docker_Page: 134	✓			
1.4.7.	Recomenda-se que não seja adicionado informações confidenciais no DockerFile. As informações e comandos ficam salvos no dockerfile e sendo assim é recomendado que não seja incluído nenhuma informação sensível. Docker_Page: 138	✓			

1.5. Container Runtime Configuration

Existem muitas implicações de segurança associadas à maneira como os contêineres são iniciados. Alguns parâmetros de tempo de execução podem ser fornecidos com consequências de segurança que podem comprometer o host e os contêineres em execução nele. Portanto, é muito importante verificar a forma como os contêineres são iniciados e quais parâmetros estão associados eles.

1.5.1.	Recomenda-se que, se aplicável, um perfil do AppArmor esteja habilitado. O AppArmor protege o sistema operacional Linux e os aplicativos de várias ameaças, impondo uma política de segurança que também é conhecida como perfil do AppArmor. Pode-se criar o próprio perfil do AppArmor para contêineres ou usar o perfil padrão do Docker. Docker_Page: 142				
1.5.2.	Recomenda-se que os recursos do kernel sejam restritos aos contêineres. O Docker oferece suporte à adição e remoção de recursos. Deve-se remover todos os recursos não necessários para o funcionamento correto do contêiner. Especificamente, no conjunto de recursos padrão fornecido pelo Docker, o recurso NET_RAW deve ser removido se não for explicitamente necessário, pois pode dar a um invasor acesso a um contêiner a capacidade de criar tráfego de rede falsificado. Docker_Page: 146				
1.5.3.	Recomenda-se que diretórios confidências do sistema host não estejam montados. Se os diretórios sensíveis forem montados no modo de leitura e gravação, pode ser possível fazer mudanças nos arquivos. Isso tem implicações de segurança óbvias e deve ser evitado. Docker_Page: 151				
1.5.4.	Recomenda-se que o sshd não seja executado dentro de containers. Executar SSH dentro do contêiner aumenta a complexidade do gerenciamento de segurança: Docker_Page: 153				
1.5.5.	Recomenda-se que apenas as portas necessárias estejam abertas no container. Um contêiner pode ser executado apenas com as portas definidas no Dockerfile para sua imagem ou pode como alternativa, receber parâmetros de tempo de execução arbitrariamente para abrir uma lista de portas. A abertura de portas desnecessárias aumenta a superfície de ataque do contêiner e o aplicativo em contêiner associado. Docker_Page: 157				

1.5.6.	Recomenda-se que o namespace de rede do host não seja compartilhado. Quando compartilhado permite que o processo do contêiner abra portas reservadas com numeração baixa da mesma forma que qualquer outro processo root pode fazer. Também permite o contêiner para acessar serviços de rede, como D-bus no host Docker. Um processo de contêiner pode realizar ações indesejadas, como desligar o host Docker. Esta opção não deve ser usada a menos que haja um motivo muito específico para ativá-la. Docker_Page: 159	✓			
1.5.7.	Recomenda-se que o sistema de arquivos root do contêiner esteja montado como somente leitura. Ativar esta opção força os contêineres em tempo de execução a definir explicitamente a gravação de seus dados, estratégia para persistir ou não seus dados. Isso também reduz os vetores de ataque de segurança, pois o sistema de arquivos da instância do contêiner não pode ser adulterado ou escrito, a menos que tenha permissões explícitas de leitura e gravação em sua pasta e diretório do sistema de arquivos. Docker_Page: 165	✓			
1.5.8.	Recomenda-se que o tráfego de contêiner de entrada está vinculado a um host específico interface. Por padrão, os contêineres do Docker podem fazer conexões com o mundo exterior, mas o exterior não pode se conectar a contêineres, cada conexão de saída vai originar um dos próprios endereços IP da máquina host. Caso exista várias interfaces de rede no host, o contêiner pode aceitar conexões em portas expostas em qualquer interface de rede. Isso pode não ser desejável e pode não ser protegido. Em muitos casos, uma interface específica desejada é exposta externamente e serviços como detecção de intrusão, prevenção de intrusão, firewall, balanceamento de carga são todos executados com a intenção de filtrar o tráfego público de entrada. Não deve, portanto, aceitar conexões de entrada. Docker_Page: 168	✓			

1.5.9.	Recomenda-se que a política de reinicialização do contêiner 'em caso de falha' esteja definida como '5'. Usando o sinalizador --restart no comando docker run, pode-se especificar uma política de reinicialização para como um contêiner deve ou não ser reiniciado na saída, escolher a política de reinicialização em caso de falha e limitar as tentativas de reinicialização a 5. Caso continue tentando indefinidamente iniciar o contêiner, isso pode levar a uma negação de serviço no host. Docker_Page: 170	✓			
1.5.10.	Recomenda-se que o namespace UTS do host não seja compartilhado. Compartilhar o namespace UTS com o host fornece permissão total para que cada contêiner mude o nome do host. Docker_Page: 170	✓			
1.5.11.	Recomenda-se que os comandos docker exec não sejam usados com a opção privilegiada nem com usuário root. Docker_Page: 186 e 188.				
1.5.12.	Recomenda-se que o contêiner seja impedido de adquirir privilégios. Docker_Page: 192				
1.5.13.	Recomenda-se que a integridade do contêiner seja verificada no tempo de execução. Se a imagem do contêiner que você está usando não tem uma instrução HEALTHCHECK predefinida, use o parâmetro --health-cmd para verificar a integridade do contêiner em tempo de execução. O parâmetro --health-cmd no tempo de execução do contêiner para verificar a integridade do contêiner. Docker_Page: 194	✓			
1.5.14.	Recomenda-se que o Docker sempre use as últimas versões da imagem. Docker_Page: 196				

1.5.15.	Recomenda-se que o limite cgroup PIDs está definido. Usar o parâmetro cgroup PIDs --pids-limit evitaria excessos de solicitações, restringindo o número que poderia ocorrer dentro de um contêiner dentro de um período de tempo especificado. Docker_Page: 198				

1.6. Docker Security Operations

Esta seção cobre alguns dos problemas de segurança operacional associados ao Docker. Essas são as práticas recomendadas que devem ser seguidas sempre que possível.

1.6.1.	Recomenda-se que não exista múltiplos containers no mesmo host. A flexibilidade dos contêineres facilita a execução de várias instâncias de aplicativos e portanto, leva indiretamente a imagem do Docker que podem existir em vários níveis de patch de segurança. Isto também significa que você está consumindo recursos do host que de outra forma poderiam ter sido usados executando contêineres 'úteis'. Docker_Page: 209				

1.7. Docker Enterprise Configuration

Esta seção contém recomendações para proteger os componentes do Docker Enterprise.

1.7.1.	Recomenda-se a configuração do serviço LDAP O sistema integrado de autenticação de usuário gerenciado do UCP oferece suporte apenas à criação e exclusão de usuários e deficiência. Ao usar um ponto de extremidade LDAP externo, pode ter mais controle sobre os usuários, grupos e outras organizações hierárquicas que podem acessar e manipular recursos via UCP. Docker_Page: 231				
1.7.2.	Recomenda-se o uso de certificados externos. Quando é instalado o Universal Control Plane sem fornecer seus próprios certificados TLS, por padrão, configurar certificados autoassinados. Em vez disso, deve-se usar certificados assinados por uma autoridade certificada externa e confiável, Docker_Page: 233				

1.7.3.	Recomenda-se a execução de análise de vulnerabilidade. É importante garantir que imagens do Docker estejam livres de vulnerabilidades. O Docker Trusted Registry (DTR) inclui varredura de vulnerabilidade de imagem que pode verificar qualquer pacote incluído na imagem em bancos de dados de vulnerabilidades conhecidas. Docker_Page: 246				