OWASP Top 10
Riscos Segurança Aplicacional – 2017

A1:2017-Injeção	Falhas de injeção, tais como injeções de SQL, OS e LDAP ocorrem quando dados não-confiáveis são enviados para um interpretador como parte de um comando ou consulta legítima. Os dados hostis do atacante podem enganar o interpretador levando-o a executar comandos não pretendidos ou a aceder a dados sem a devida autorização.
A2:2017-Quebra de Autenticação	As funções da aplicação que estão relacionadas com a autenticação e gestão de sessões são muitas vezes implementadas incorretamente, permitindo que um atacante possa comprometer passwords, chaves, *tokens* de sessão, ou abusar doutras falhas da implementação que lhe permitam assumir a identidade de outros utilizadores (temporária ou permanentemente).
A3:2017-Exposição de Dados Sensíveis	Muitas aplicações web e APIs não protegem de forma adequada dados sensíveis, tais como dados financeiros, de saúde ou dados de identificação pessoal (PII). Os atacantes podem roubar ou modificar estes dados mal protegidos para realizar fraudes com cartões de crédito, roubo de identidade, ou outros crimes. Os dados sensíveis necessitam de proteções de segurança extra como encriptação quando armazenados ou em trânsito, tal como precauções especiais quando trocadas com o navegador web.
A4:2017-Entidades Externas de XML (XXE)	Muitos processadores de XML mais antigos ou mal configurados avaliam referências a entidades externas dentro dos documentos XML. Estas entidades externas podem ser usadas para revelar ficheiros internos usando o processador de URI de ficheiros, partilhas internas de ficheiros, pesquisa de portas de comunicação internas, execução de código remoto e ataques de negação de serviço, tal como o ataque *Billion Laughs*.
A5:2017-Quebra de Controlo de Acessos	As restrições sobre o que os utilizadores autenticados estão autorizados a fazer nem sempre são corretamente verificadas. Os atacantes podem abusar destas falhas para aceder a funcionalidades ou dados para os quais não têm autorização, tais como dados de outras contas de utilizador, visualizar ficheiros sensíveis, modificar os dados de outros utilizadores, alterar as permissões de acesso, entre outros.
A6:2017-Configurações de Segurança Incorretas	As más configurações de segurança são o aspeto mais observado nos dados recolhidos. Normalmente isto é consequência de configurações padrão inseguras, incompletas ou *ad hoc*, armazenamento na nuvem sem qualquer restrição de acesso, cabeçalhos HTTP mal configurados ou mensagens de erro com informações sensíveis. Não só todos os sistemas operativos, *frameworks*, bibliotecas de código e aplicações devem ser configurados de forma segura, como também devem ser atualizados e alvo de correções de segurança atempadamente.
A7:2017-Cross-Site Scripting (XSS)	As falhas de XSS ocorrem sempre que uma aplicação inclui dados não-confiáveis numa nova página web sem a validação ou filtragem apropriadas, ou quando atualiza uma página web existente com dados enviados por um utilizador através de uma API do browser que possa criar JavaScript. O XSS permite que atacantes possam executar scripts no browser da vítima, os quais podem raptar sessões do utilizador, descaraterizar sites web ou redirecionar o utilizador para sites maliciosos.
A8:2017-Desserialização Insegura	Desserialização insegura normalmente leva à execução remota de código. Mesmo que isto não aconteça, pode ser usada para realizar ataques, incluindo ataques por repetição, injeção e elevação de privilégios.
A9:2017-Utilização de Componentes Vulneráveis	Componentes tais como, bibliotecas, *frameworks* e outros módulos de software, são executados com os mesmos privilégios que a aplicação. O abuso dum componente vulnerável pode conduzir a uma perda séria de dados ou controlo completo de um servidor. Aplicações e APIs que usem componentes com vulnerabilidades conhecidas podem enfraquecer as defesas da aplicação possibilitando ataques e impactos diversos.
A10:2017-Registo e Monitorização Insuficiente	O registo e monitorização insuficientes, em conjunto com uma resposta a incidentes inexistente ou insuficiente permite que os atacantes possam abusar do sistema de forma persistente, que o possam usar como entrada para atacar outros sistemas, e que possam alterar, extrair ou destruir dados. Alguns dos estudos demonstram que o tempo necessário para detetar uma violação de dados é de mais de 200 dias e é tipicamente detetada por entidades externas ao invés de processos internos ou monitorização.