ISCTE IUL
Instituto Universitário de Lisboa
Lisbon University Institute

Departamento de Ciências e Tecnologias da Informação

Auditoria de Segurança em Aplicações na *World Wide Web* Portuguesa

Nuno Filipe Martins da Silveira Teodoro

Dissertação submetida como requisito parcial para a obtenção do Grau de

Mestre em Engenharia Informática
Especialidade em Sistemas de Informação e Gestão do Conhecimento

Orientador: Prof. Doutor Carlos José Corredoura Serrão

Junho, 2011

ISCTE IUL
Instituto Universitário de Lisboa
Lisbon University Institute

Departamento de Ciências e Tecnologias da Informação

Auditoria de Segurança em Aplicações na *World Wide Web* Portuguesa

Nuno Filipe Martins da Silveira Teodoro

Dissertação submetida como requisito parcial para a obtenção do Grau de

Mestre em Engenharia Informática
Especialidade em Sistemas de Informação e Gestão do Conhecimento

Orientador: Prof. Doutor Carlos José Corredoura Serrão

Junho, 2011

“Nothing great was ever achieved without enthusiasm.”

Por Ralph Waldo Emerson

Escritor e poeta (1803 - 1882)

Resumo

Numa época em que grande parte dos Sistemas de Informação são desenvolvidos com o objectivo de serem utilizados sobre plataformas *Web*, e em que grande parte dos utilizadores recorre diariamente a aplicações que são executadas remotamente, é fundamental garantir que essas aplicações são seguras, protegendo as informações confidenciais dos utilizadores e organizações.

O aumento da problemática anteriormente frisada é melhor percepcionado dado que, num esforço do próprio governo Português, existe um acréscimo significativo do acesso dos cidadãos a este tipo de aplicações, realizando operações sensíveis, onde a partilha de informação pessoal e confidencial é obrigatória para a execução da maioria dos serviços estatais.

Assim, este trabalho tem como objectivo a identificação de grupos críticos, e das aplicações *Web* mais relevantes dentro do próprio grupo, culminando numa análise quantitativa e qualitativa do grau de segurança associado, numa primeira fase, à aplicação *Web per si*, e posteriormente ao grupo que a contém. Esta análise procura recolher indicadores que permitam a extrapolação dos resultados obtidos para uma aproximação ao panorama geral em Portugal.

A metodologia desenvolvida para a realização desta auditoria não só é fiável e concordante com entidades internacionalmente reconhecidas na área da segurança *Web*, como flexível e de utilização eficiente em contextos organizacionais variados, levando a uma maior consciencialização por parte das entidades que a implementem, e à consequente melhoria do estado da segurança *Web* portuguesa.

Palavras-chave: *Web*, segurança, vulnerabilidades, auditoria, estudo de vulnerabilidades em aplicações *Web*

Abstract

Nowadays, most information systems are developed towards Web platforms' usage. These users perform, on a daily basis, operation on these remote applications, making it crucial to ensure their security, protecting users and organizations' confidential data. This work represents an extremely relevant analysis regarding the study of the security of Portuguese Web applications.

The previous problem takes an even bigger relevance on the Portuguese panorama as the Portuguese government is enforcing a huge growth of citizen's access to public administration Web applications. These services represent the execution of sensitive operations, where sharing personal and confidential data is mandatory.

Based on the previous facts, this project has the purpose of identifying critical groups and the most relevant Web applications contained inside them. This will allow to perform a quantitative and qualitative security analysis of the Web applications, the services group they represent and to extrapolate those results towards a loyal approximation of the whole Portuguese panorama.

The methodology created for this project it's not only reliable and in sync with internationally known web application's security entities, but also flexible and efficient for the usage on several organizational contexts, increasing the awareness on the security levels for the organizations implementing it, leading to an improvement of the National Web security state.

Keywords: *Web*, security, vulnerabilities, audit, web applications vulnerability study

Agradecimentos

Ao professor Carlos Serrão, por todo o trabalho de orientação, pelos debates produtivos e pelo esforço dispendido na realização desta dissertação. Pelo entusiasmo com que ensina todos os dias e pela oportunidade que me deu no envolvimento com a OWASP, o qual muito contribuiu para a escolha da minha carreira profissional. Pela inspiração e amizade, qualidades que me garantem que nesta viagem, ganhei igualmente um amigo.

Às várias entidades que participaram no presente estudo. Obrigado pelo tempo dispendido e pela vontade em contribuir para o avanço na melhoria da segurança *Web* em Portugal. Sem o seu apoio esta dissertação não seria possível.

À minha família que sempre me apoiou na minha busca pessoal pela excelência e auto-realização, tanto académica como profissional.

A ti, Ana Isabel, pela paciência, pela boa disposição, carinho e compreensão nestes anos de grande azáfama onde pouco tempo tive para dedicar a quem mais amo. Sem ti não teria conseguido.

Glossário

OWASP: *Open Web Application Security Project*. Organização mundial não lucrativa constituída por peritos, profissionais da indústria e organizações, que produz material *open source* e internacionalmente reconhecido para as melhores práticas de segurança aplicacional.

WASC: *Web Application Security Consortium*. Organização mundial não lucrativa constituída por peritos, profissionais da indústria e organizações, que produz material *open source* e internacionalmente reconhecido para as melhores práticas de segurança para a *World Wide Web*.

URL: *Uniform Resource Locator*. Formato de uma designação universal para referenciar um recurso na Internet ou Intranet.

P&S: *Point and Shot*. Modo de execução dos *web scanners* que pressupõe que nenhuma configuração é realizada antes da sua utilização, excepto da introdução do URL da aplicação a auditar.

GUI: *Graphical User Interface*. – Interface gráfica que representa as acções possíveis de realizar num dispositivo e/ou programa, facilitando a sua interacção com o utilizador.

WWW: *World Wide Web*. Serviço que funciona sobre a Internet e que interliga um conjunto de recursos digitais baseado em protocolos *standards* de comunicação como HTTP e TCP/IP.

SaaS: *Software as a Service*. É um modelo de utilização de *software* em que este e os dados que lhes estão associados então tipicamente centrados na *Internet* e são acedidos por clientes através de um *browser*.

Internet: Sistema global interligado de redes digitais de computadores que utiliza protocolos *standard* de comunicação para servir todos os seus utilizadores.

Intranet: Rede de computadores privada que utilizada os mesmos princípios da Internet, num ambiente controlado.

Indíce

Lista de Tabelas

Lista de Figuras

Capítulo 1 Introdução

1.1 Assunto e Motivação

A *World Wide Web* (WWW*)* é um dos serviços baseados na Internet que maior crescimento tem tido nos últimos tempos. Uma das suas principais características é o facto de permitir aceder a qualquer aplicação através de um único meio, um *browser Web*, criando uma total independência do local e do sistema cliente em utilização (ubiquidade). Este último ponto vem precisamente escalar o número de utilizadores que cada vez menos inércia sente na utilização destes sistemas, havendo ainda lugar, por parte da oferta tecnológica actual, a um forte incentivo a uma cultura de total mobilidade e permanente comunicação com a Internet.

Assente na consciência das actuais expectativas e necessidades dos utilizadores, existe inevitavelmente uma total conversão do modelo tecnológico para uma base sustentada em aplicações *Web*, onde o actual contexto do desenvolvimento de aplicações é cada vez mais direccionado à WWW.

Seguindo esta tendência tecnológica internacional, Portugal não é excepção na criação de medidas que permitam o acompanhamento desta evolução. Desta forma, não só as empresas sedidadas em território nacional procuram uma maior interacção com a WWW, como o próprio Governo, procura reforçar a ideia de inovação e avanço tecnológico nesta área, facto este que pode ser comprovado com medidas como é o caso do SIMPLEX (SIMPLEX, 2006).

De um ponto de vista lógico, colocar uma aplicação a funcionar na WWW resulta na criação de uma linha de comunicação entre virtualmente qualquer entidade no Mundo e os mecanismos de processamento dessa aplicação, os quais estão permanentemente à espera de qualquer pedido que lhes seja transmitido. Este processo revela ambas, a principal funcionalidade e fraqueza deste tipo de aplicações. Esta inerente facilidade de acesso e total disponibilidade vem fazer com que as aplicações *Web* sejam verdadeiramente apetecíveis do ponto de vista da exploração maliciosa.

Acrescendo aos factores disponibilidade e disseminação, o tipo de operações realizadas hoje em dia a partir das aplicações *Web,* é também por si só um dos maiores catalizadores de atractividade para agentes maliciosos. Não existe, nem deverá existir, uma restrição do tipo de operações realizadas neste tipo de aplicações. Assim, é fácil analisar a diferença da antiguidade para a actualidade, onde existe uma total migração do conceito "aplicação *Web*", não servindo esta apenas como portal de exibição de informação inócua e inerte, como outrora, mas passando para o principal meio de criação de valor no núcleo organizacional e pessoal, criando não apenas interligação interna, como externa, para com clientes, parceiros de negócio e prestadores de serviços.

Assim, existindo um aumento do número de aplicações *Web*, dos seus utilizadores e da importância das operações realizadas, é fácil perceber que existe um acréscimo do problema de segurança relacionado com este tipo de aplicações (Holz, T., Marechal S. e Raynal, F., 2009).

Infelizmente, quando falamos em segurança, os problemas são mais complexos e não se reduzem exclusivamente à segurança das redes e restantes infra-estruturas de comunicação. Apesar da segurança sempre ter sido uma das principais preocupações relativamente a sistemas críticos, a nova conjectura tecnológica relacionada com as aplicações *Web* obriga a uma nova forma de pensar a segurança, não só a nível tecnológico, mas também a um nível mais lógico. Existe hoje em dia uma migração da perspectiva tecnológica, na área da segurança, para a camada aplicacional do modelo OSI (Day & Zimmermann, 1983), relegando para patamares menos importantes as camadas inferiores, como é o caso da camada de rede e transporte. A consciencialização desta nova perspectiva tecnológica, na maioria dos casos, está ainda longe do desejável.

O facto anterior é facilmente confirmado pelo número de ataques documentados por algumas entidades. O Instituto Nacional de *Standards* e Tecnologia (NIST), uma agência do departamento de comércio dos Estados Unidos, possui uma base de dados de vulnerabilidades nacionais (NIST, National Vulnerability Database), a qual contém mais de 45000 vulnerabilidades identificadas na camada aplicacional desde 1997 (NIST, Software Vulnerabilities).

Estes factos são corroborados por inúmeros estudos realizados por entidades internacionalmente reconhecidas e acreditadas na área das Tecnologias de Informação (TI), e em especial na área da segurança aplicacional. Num estudo realizado pelo

Gartner Group, é estimado que actualmente cerca de 74% dos ataques organizacionais derivam da camada aplicacional (Gartner, 2011).

Adicionalmente a CISCO, emite anualmente um relatório de segurança não só com os principais acontecimentos no ano transacto, como com as principais tendências para o ano seguinte. Sem grandes surpresas, no relatório anual de 2010 foi identificado como principal tema de 2010 e de crescimento para 2011, a exploração de vulnerabilidades em ambientes *Web* (CISCO, 2010), como pode ser observado na Figura 1.

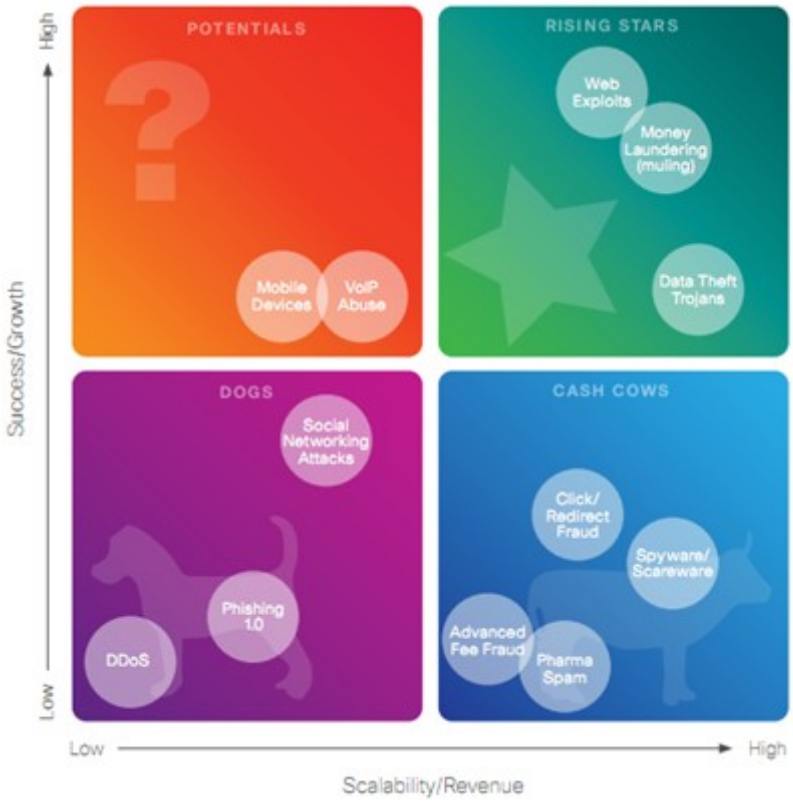


Figura 1 - *Cisco Cybercrime Return on Investment Matrix*

Por sua vez, o WASC (WASC, Web Application Security Consortium) elaborou um estudo com mais de 12000 aplicações *Web,* onde o resultado obtido derivou em mais de 97000 vulnerabilidades, resultando em que cerca de 98% das aplicações testadas tenham sido consideradas inseguras (WASC, 2008).

De igual forma, entidadades reconhecidas no domínio da segurança elaboram estudos onde os resultados são elucidativos da problemática da segurança *Web.* Alguns dos mais relevantes são a WhiteHat (WhiteHat, 2011) e a IBM (IBM, 2010). De um modo geral todos os estudos apontam para uma falta de maturidade neste particular, afectando

virtualmente todas as pessoas com uma ligação à Internet que realizem operações críticas neste domínio.

Como consequência do aumento dos problemas derivados de vulnerabilidades em aplicações *Web*, é cada vez mais influente a presença de entidades como a OWASP (OWASP, Open Web Application Security Project), WASC e o CERT.PT (CERT, Computer Emergency Response Team).

De um modo geral, o que se verifica, não só a nível internacional, como nacional, é que existe uma persistência das medidas reactivas, em vez das proactivas. A reactividade, neste domínio específico, significa que o problema já ocorreu, e as consequências já tiveram lugar no cerne da organização. Obviamente este não será o procedimento desejado e a proacividade deverá estar presente em todo o processo de desenvolvimento e maturação das aplicações *Web*, terminando apenas com o fim da sua vida útil.

1.2 Descrição do Problema

Tal como anteriormente descrito, a segurança *Web* deve ser hoje em dia um dos principais focos de interesse organizacional, especialmente se existir uma ligação directa entre o negócio e estas aplicações.

A falta de formação e/ou preocupação com a segurança aplicacional é um facto que persiste ainda nos dias de hoje, onde, apesar de serem realizadas operações sensíveis como pagamento de impostos, segurança social, despesas de saúde, operações bancárias, entre outras, não existe ainda um esforço, a nível nacional, na criação de bases sólidas para a preparação de profissionais capazes de lidar com esta temática.

Como influência adicional, é notório que o mercado de desenvolvimento de software não é propício ao desenvolvimento de aplicações de software muito centradas na segurança aplicacional. O simples facto de existir um acréscimo temporal na entrada em produção de uma aplicação, devido à inclusão de metodologias e processos de segurança no ciclo de vida de desenvolvimento de software, praticamente inviabiliza esta abordagem, especialmente quando considerando factores monetários e de conhecimentos/capacidades técnicas para o efeito. Adicionalmente, a segurança aplicacional não é um factor vendável no contexto do mercado actual. O detrimento da

segurança em favor da inclusão de características visuais apelativas e novas funcionalidades é uma máxima indubitavelmente aceite.

Inevitavelmente, a componente de segurança das aplicações *Web* é deixada para uma fase posterior à entrada em produção, recaindo maioritariamente numa análise de *black-box* (Beizer, 1990). A anterior afirmação delega neste trabalho, juntamente com as condições temporais e de acessibilidade restritivas do mesmo, a limitação da realização de auditorias de vulnerabilidades em domínios e condições muito específicas. Com base nestes factos, é entendido como abordagem ao problema, o seguinte:

- Definição de uma metodologia rápida e eficaz, baseada nas melhores práticas de entidades internacionalmente reconhecidas;
- Identificação das principais vulnerabilidades em aplicações *Web* na actualidade;
- Identificação de grupos críticos em Portugal;
- Identificação das principais vulnerabilidades em cada aplicação *Web;*
- Realização de uma análise de risco adequada a cada grupo; e
- Identificação dos grupos mais problemáticos.

1.3 Objectivos

Realizar uma auditoria ao nível da segurança das aplicações *Web* em Portugal, que permita estabelecer o nível de segurança destas aplicações através da comparação de resultados da análise de sectores críticos no panorama nacional. Este tipo de auditoria permitirá não só derivar o nível de maturidade da segurança aplicacional das organizações mais representativas no nosso país como identificar pontos de falha que devam ser endereçados com maior preocupação.

O principal objectivo deste trabalho é o de realizar auditorias de segurança num grupo de aplicações *Web* bem definidas, dentro do contexto da WWW em Portugal, estabelecendo uma visão geral do nível da segurança aplicacional nacional. Esta análise irá fornecer um importante indicador às entidades oficiais responsáveis por assegurar esta temática em Portugal.

Adicionalmente, dada a estreita relação entre o tema retratado na tese, os elementos que nela participam e a OWASP, o culminar desta tese de mestrado tem também como objectivo escalar a interligação da OWASP com a temática da segurança aplicacional

em Portugal, fornecendo um passo importante numa cooperação fundamental para o progresso da segurança aplicacional nacional, interferindo não só no domínio tecnológico como também no académico.

Este trabalho tem como objectivo a definição de uma metodologia para a análise da segurança de aplicações *Web* de modo a determinar o seu nível de segurança, apoiando-se nas vulnerabilidades mais perigosas e vulgarmente identificadas como tendo maior probabilidade de ocorrência e impacto nos sistemas de informação das organizações.

Adicionalmente, a metodologia desenvolvida tem como objectivo, ter a característica de ser flexível e adaptável o suficiente para aplicação directa num ambiente organizacional onde não exista o conhecimento suficiente nem a maturidade para a implementação de políticas de segurança aplicacional. Deste modo, o facilitismo em determinar vulnerabilidades nas aplicações *Web* é aumentado, diminuindo a probabilidade de ocorrência de acções maliciosas, levando a uma melhoria gradual do panorama nacional da segurança *Web*.

Uma descrição dos objectivos propostos para este trabalho encontra-se abaixo descrita:

- Realizar uma análise das diferentes abordagens para conduzir auditorias de segurança em aplicações *Web*;
- Identificar quais são as ameaças mais comuns na segurança em aplicações *Web*, e identificar ferramentas para realizar os testes a essas vulnerabilidades;
- Seleccionar um conjunto de aplicações *Web* relevantes para a realização dos testes;
- Definir uma metodologia para realizar a auditoria de segurança das aplicações *Web* seleccionadas;
- Aplicar a metodologia de teste às aplicações *Web* identificadas (inclui o uso de ferramentas de teste automáticas, realização de testes de intrusão, entre outros); e
- Reunir os resultados dos testes (incluindo como a vulnerabilidade pode ser explorada, quais são os riscos dessa exploração e qual é o impacto dessa vulnerabilidade na aplicação *Web*), tratar os dados e produzir conclusões, incluindo recomendações em como solucionar as vulnerabilidades encontradas.

1.4 Contribuições

As contribuições deste trabalho estão directamente relacionadas com os seus principais objectivos. Se por um lado, não existem factores de inovação tecnológicos associados ao culminar desta dissertação, por outro, existe não só a criação de um estudo imparcial sobre o estado da segurança de aplicações *Web* em nichos críticos em Portugal, como a promoção da necessidade de consiencialização das entidades, para um problema cada vez mais presente.

Será também criada uma metodologia rápida e eficiente, baseada nas melhores práticas, concordantes com entidades internacionalmente reconhecidas no domínio da segurança *Web*, e em ferramentas automáticas eficazes, robustas e *open-source*, que permitam uma análise coerente das vulnerabilidades existentes.

É também tido como contribuição evidente, devido à intensa utilização de ferramentas automáticas *open-source* de análise de vulnerabilidades em aplicações *Web* (*web scanners*), a análise sobre a qualidade destas ferramentas, e a sua aplicabilidade em contextos reais e de elevada importância. Adicionalmente, será criada uma metodologia para comparação das características destas ferramentas, baseada em *standards* criados por entidades internacionalmente reconhecidas, fornecendo uma importante contribuição para a elaboração de um modelo simples e eficaz na análise e escolha de *web scanners.*

Dado apenas serem utilizados *web scanners open-source* (cuja decisão se encontra devidamente fundamentada no Capítulo 3, secção 3.4), é também entendido como contribuição a análise dos resultados que é possível obter, por parte de qualquer utilizador (inclusivé naqueles que sejam mal intencionados), a partir da utilização destas ferramentas. Este tópico é extremamente relevante pois permite que as entidades que participaram neste estudo tenham a percepção da possível fragilidade das suas aplicações *Web* a um conjunto de ferramentas livremente acessíveis por qualquer pessoa.

A criação de valor para as organizações enquadradas e que aceitaram participar neste estudo é outra das contribuições principais que resulta deste trabalho. Através da análise dos relatório enviados, não só as organizações poderão beneficiar directamente de um

aumento da segurança das suas aplicaões *Web*, como também os seus parceiros de negócio e os seus clientes finais.

Por fim, é demais evidente que a análise do estado da segurança aplicacional *Web* em Portugal é a maior contribuição deste trabalho. A análise de grupos de entidades críticas no domínio português permitirá recolher indicadores passíveis de extrapolar para todo o panorama nacional, com os devidos cuidados, fornecendo uma visão geral de uma das maiores preocupações tecnológicas da actualidade, a segurança *Web*.

1.5 Organização da Dissertação

Para além deste capítulo introdutório, a presente dissertação está organizada em mais quatro capítulos.

O Capítulo 2 descreve a revisão da literatura relacionada com o trabalho desenvolvido ao longo da dissertação. Neste capítulo não só irá ser analisada a temática das principais vulnerabilidades presentes em aplicações *Web,* como os métodos mais reconhecidos e internacionalmente aceites para a realização de auditorias de vulnerabilidades em aplicações *Web*. Finalmente, será realizada uma revisão da literatura relacionada com os *web scanners*. Apesar destas ferramentas não serem o objectivo de análise principal neste estudo, estão directamente relacionadas com a metodologia desenvolvida, assim como com a qualidade e quantidade de dados obtidos para análise, facto pelo qual ganham especial relevo ao longo de todo o processo de auditoria às aplicações *Web*.

O Capítulo 3 tem como objectivo a descrição da solução que é proposta para a resolução do problema identificado. Numa primeira instância irá ser analisado o contexto do enquadramento legal em Portugal para a realização destas auditorias. Seguidamente irão ser identificados os grupos escolhidos para a realização da auditoria, partindo de uma análise de contexto e de relevância para o trabalho. Como última etapa deste capítulo será definida a metodologia a utilizar na realização dos testes de auditoria.

No Capítulo 4 irá ser realizada uma descrição detalhada da análise dos resultados obtidos. Não só irão ser analisadas as vulnerabilidades identificadas de um modo geral, como irá ser estudado o estado de segurança das aplicações *Web* analisadas. Seguidamente irá ser realizada uma análise de risco, tanto num panorama restrito a cada grupo em análise como na generalidade, fornecendo o panorama geral, concordante com

todas as vulnerabilidades encontradas. Finalmente, serão apresentados os resultados relativos à fiabilidade das detecções efectuadas pelos *web scanners* utilizados.

Por fim, no Capítulo 5 são apresentadas as principais conclusões sobre todo o trabalho elaborado e apontadas directrizes para um trabalho futuro.

Capítulo 2 Estado da Arte

Neste capítulo serão detalhados os principais tópicos referentes ao estado da arte de auditoria de vulnerabilidades em aplicações *Web*. Este tema, de acordo com os pressupostos identificados para a realização da presente dissertação, compõe não só uma análise do estado da arte das principais vulnerabilidades em aplicações *Web*, como das metodologias mais fiáveis e robustas para a sua identificação.

Adicionalmente, a importante e extensa utilização de ferramentas automáticas de análise de vulnerabilidades em aplicações *Web* (*web scanners*), torna também imprescindível a análise do estado da arte destas ferramentas.

2.1 Principais Vulnerabilidades em Aplicações Web

Desde que a temática da segurança aplicacional tomou relevo na indústria de desenvolvimento de software que várias entidades começaram a abordar este assunto de uma forma mais séria.

Numa primeira instância, e fruto das antigas tendências do mercado de software ainda afastadas de uma centralização em aplicações *Web*, começaram a surgir boas práticas de verificação de vulnerabilidades em software cliente (SANS, MITRE, etc).

Contudo, com o evoluir das tendências de desenvolvimento de *software*, houve uma necessidade de criação de entidades e documentação específica para abordar os problemas de segurança que incidem sobre as aplicações *Web*. A identificação destes problemas tem sido baseada em duas grandes entidades internacionais:

- WASC; e
- OWASP.

Estas duas entidades centram-se puramente no domínio das aplicações *Web*, fornecendo directrizes específicas para esta área, criando uma fonte de conhecimento altamente reconhecido e espacializado na comnidade de segurança *Web*.

O documento elaborado pela WASC – " *WASC Threat Classification*" – foi exibido ao público na sua primeira versão em 2004 (WASC, Web Application Security

Consortium: Threat Classification v1), tendo sido substituído pela segunda versão em Janeiro de 2010 (WASC, Web Application Security Consortium: Threat Classification v2).

O WASC *Threat Classification* deriva de um esforço colaborativo de toda a comunidade WASC para enumerar e classificar os ataques e fraquezas que podem comprometer uma aplicação *Web*, os seus dados ou utilizadores. Estes ataques e fraquezas, na sua versão mais recente, estão descritos na Tabela 1.

Ataques	Fraquezas
Abuse of Functionality	*Application Misconfiguration*
Brute Force	*Directory Indexing*
Buffer Overflow	*Improper Filesystem Permissions*
Content Spoofing	*Improper Input Handling*
Credential/Session Prediction	*Improper Output Handling*
Cross-Site Scripting	*Information Leakage*
Cross-Site Request Forgery	*Insecure Indexing*
Denial of Service	*Insufficient Anti-automation*
Fingerprinting	*Insufficient Authentication*
Format String	*Insufficient Authorization*
HTTP Response Smuggling	*Insufficient Password Recovery*
HTTP Response Splitting	*Insufficient Process Validation*
HTTP Request Smuggling	*Insufficient Session Expiration*
HTTP Request Splitting	*Insufficient Transport Layer Protection*
Integer Overflows	*Server Misconfiguration*
LDAP Injection	
Mail Command Injection	
Null Byte Injection	
OS Commanding	
Path Traversal	
Predictable Resource Location	
Remote File Inclusion (RFI)	
Routing Detour	
Session Fixation	
SOAP Array Abuse	
SSI Injection	
SQL Injection	
URL Redirector Abuse	
XPath Injection	
XML Attribute Blowup	
XML External Entities	
XML Entity Expansion	
XML Injection	
XQuery Injection	

Tabela 1 - WASC *Threat Classification Attacks and Weaknesses*

Por sua vez, o OWASP produziu o "OWASP *Top* 10" pela primeira vez em 2004 (OWASP, OWASP Top 10 2004), documento este que foi revisto em 2007 (OWASP, OWASP Top 10 2007), tendo sido produzida a última versão em 2010 (OWASP, OWASP Top 10 2010).

O OWASP *Top* 10 é um documento que tem como principal objectivo educar os programadores, arquitectos de software, gestores e organizações acerca das consequências das fraquezas de segurança *Web* mais relevantes. Adicionalmente, este documento fornece técnicas básicas de protecção contra estes problemas.

Sendo um documento em constante evolução e análise por parte da comunidade da OWASP é natural que as suas alterações reflictam os problemas actuais das aplicações *Web*. Um exemplo destas modificações resulta de uma comparação directa entre a versão de 2007 e a de 2010 (as duas versões mais relevantes do *OWASP Top* 10), a qual pode ser vista na Tabela 2.

OWASP Top 10 2007	OWASP Top 10 2010
A1: Cross Site Scripting (XSS)	*A1: Injection*
A2: Injection Flaws	*A2: Cross-Site Scripting (XSS)*
A3: Malicious File Execution	*A3: Broken Authentication and Session Management*
A4: Insecure Direct Object Reference	*A4: Insecure Direct Object References*
A5: Cross Site Request Forgery (CSRF)	*A5: Cross-Site Request Forgery (CSRF)*
A6: Information Leakage and Improper Error Handling	*A6: Security Misconfiguration*
A7: Broken Authentication and Session Management	*A7: Insecure Cryptographic Storage*
A8: Insecure Cryptographic Storage	*A8: Failure to Restrict URL Access*
A9: Insecure Communications	*A9: Insufficient Transport Layer Protection*
A10: Failure to Restrict URL Access	*A10: Unvalidated Redirects and Forwards*

Tabela 2 – OWASP *Top* 10 2007 *vs* OWASP *Top* 10 2010

O OWASP *Top 10*, quando comparado com o WASC *Threat Classification*, apresenta-se mais completo para a análise da segurança em vários cenários organizacionais. O OWASP *Top 10* não só identifica as principais vulnerabilidades como se apoia no restante material da OWASP, não só a nível de categorização de vulnerabilidades, como métodos de mitigação, análises de risco, entre outras, tornando-se no *standard* na área da segurança.

2.2 Auditoria de Vulnerabilidades em Aplicações Web

A auditoria de vulnerabilidades em aplicações *Web* é um processo de elevada complexidade que é composto por duas áreas principais: o tipo de auditorias a realizar e quais as metodologias utilizadas para realizar todo o processo de análise de vulnerabilidades.

2.3 Tipo de Auditorias

Uma auditoria de segurança a uma aplicação *Web* envolve uma série de análises técnicas e documentais. Numa perspectiva geral, embora pouco exequível no contexto real do mercado de trabalho, uma análise completa envolveria análise à documentação (manuais, documentos de arquitectura, fluxos, entre outros), assim como a toda a estrutura de rede onde a aplicação *Web* está integrada e à própria aplicação em si.

Assim sendo, uma auditoria de segurança pode ser baseada em várias técnicas, entre as quais se destacam as seguintes:

- Revisão e inspecção documental;
- Modelação de ameaças;
- Revisão de código; e
- Testes de intrusão.

A revisão e inspecção documental tem como principal objectivo o de assegurar que existe de facto uma preocupação na criação de documentação que apoie o desenvolvimento de software, e por outro lado, que essa documentação é feita correctamente. Para além de questões relacionadas com o desenvolvimento, arquitectura e usabilidade da aplicação, a documentação deverá correctamente identificar medidas explicitamente tomadas no âmbito da segurança aplicacional, assim como vulnerabilidades identificadas mesmo que sem resolução.

A modelação de ameaças (Ebenezer Oladimeji, 2006) é considerada umas das melhores técnicas de previsão de possíveis vulnerabilidades para a camada aplicacional. Novamente, a OWASP possui recursos extremamente valiosos neste domínio (OWASP, Application Threat Modeling), os quais permitem seguir uma abordagem estruturada na

análise da segurança de uma aplicação, identificando, quantificando e endereçando os riscos de segurança que lhe estão associados.

Outra das principais actividades relacionada com auditorias de segurança centra-se na revisão de código. Sendo uma aplicação não mais do que um conjunto de linhas de código, criadas em torno de uma arquitectura tecnológica e envolto num modelo de negócio, faz todo o sentido que essas linhas de código sejam analisadas por padrões que correspondam a possíveis falhas de segurança. Normalmente podem ser seguidas abordagens de revisão de código manuais ou automáticas, sendo que cada uma apresenta os seus prós e contras (IEEE, Static Analysis for Security, 2004).

Finalmente, e talvez um dos mais conhecidos e importantes vectores na auditoria de segurança em aplicações *Web*, os testes de intrusão (IEEE, About Penetration Testing, 2007). De um modo geral os testes de intrusão, ou *pen tests,* são bastante atractivos pois podem ser facilmente aplicados no fim do ciclo de vida de desenvolvimento da aplicação. Deste modo, são efectuados testes que asseguram a segurança das aplicações, não forçando a alteração dos processos de desenvolvimento de software instalados na organização.

2.4 Metodologias de Auditoria de Vulnerabilidades

A auditoria de vulnerabilidades é uma tarefa fastidiosa e por isso mesmo sujeita a falhas humanas que podem comprometer a seriedade e fiabilidade dos resultados obtidos. O método mais fiável de conseguir realizar auditorias de um modo credível centra-se na utilização de metodologias reconhecidas.

Todas as metodologias internacionalmente reconhecidas são desenvolvidas por um grupo de profissionais com larga experiência na área e com um forte suporte por parte de um grupo ou entidade também por si só bastante acreditado na sua área de actuação. Quando se fala em metodologias de auditoria internacionalmente reconhecidas no contexto de aplicações *Web*, são imediatamente apontadas como referência as seguintes:

- *Information Systems Security Assessment Framework* (ISSAF);
- *Open Source Security Testing Methodology Manual* (OSSTMM); e
- OWASP *Testing Guide*.

Apesar destas serem das metodologias de auditoria mais reconhecidas a nível internacional, existe uma diferença significativa entre a ISSAF, OSSTMM e o OWASP *Testing Guide*. Enquanto que a ISSAF e o OSSTMM são metodologias de auditoria de sistemas de informação em toda a sua extensão, incluíndo aplicações *Web*, o OWASP *Testing Guide* centra-se puramente em auditoria de aplicações *Web*.

Ainda que o facto anterior não retire nenhuma credibilidade à ISSAF e OSSTMM no contexto de auditoria de aplicações *Web*, o contexto que enquadra o OWASP *Testing Guide* torna-o numa referência ainda mais forte neste domínio em particular.

A ISSAF é um projecto *open source* pertencente à *Open Information Systems Security Group* (ISSAF, Information Systems Security Assessment Framework) que teve a sua primeira publicação em Dezembro de 2004. Mais recentemente, em Março de 2006, foi publicada a versão 0.2, a última da ISSAF. Sendo 2006 a data da última publicação, por mais força e notoriedade que tenha o grupo que suporta esta metodologia, esta é colocada num patamar bastante inferior em termos de acompanhamento das mais recentes necessidade de segurança da informação, catalisando o seu absentismo neste mercado.

O OSSTMM é, assim como a ISSAF, um projecto *open source*. Este projecto é mantido pelo *Institute for Security and Open Source Methodologies* (OSSTMM, Open Source Security Testing Methodology Manual), tendo tido a sua primeira publicação em Dezembro de 2000. Ao contrário da ISSAF, o OSSTMM tem actualizações regulares e conta com uma versão bastante actualizada, tendo sido publicada em Dezembro de 2010. Contudo, o OSSTMM tem uma particularidade que o distingue das restantes metodologias. Apesar de ser uma metodologia gratuita, o documento só é fornecido ao público na sua ante-penúltima versão. Deste modo as duas versões mais actuais estão salvaguardadas para membros pagantes da ISECOM. No entanto, considera-se irrelevante este ponto dado que a quantidade de actualizações ao documento faz com que o fluxo de publicações seja grande o suficiente para não ser desconsiderado na utilização no mercade de segurança de informação.

O OWASP *Testing Guide* é um documento grátis criado pela OWASP. A primeira versão foi publicada em Julho de 2004, sendo que em Dezembro de 2008 foi publicada a terceira e última versão deste documento. O OWASP *Testing Guide* (actualmente na sua terceira versão) é um documento internacionalmente reconhecido como

metodologia de auditoria de aplicações *Web*. Por ser totalmente dedicado à *Web*, e por ser suportado por uma entidade como a OWASP, tornou-se a metodologia *standard* para efectuar auditorias deste tipo de aplicações.

A vantagem do OWASP *Testing Guide* é não só o documento em si, como todo um conjunto de material que enquadra o seu desenvolvimento. Este documento não só é grátis em todas as suas versões, como possui uma estrutura interna que descreve uma metodologia de auditoria bastante detalhada e completa, tendo também como vantagem a inclusão de sugestões de ferramentas e resultados esperados para cada processo especificado.

Adicionalmente, o OWASP *Testing Guide* traz enormes vantagens para as auditorias de aplicações *Web*, não só no ponto de vista do auditor, como da organização alvo da auditoria. Estas vantagens são traduzidas na extrema intimidade de relação e mapeamento directo com os seguintes documentos da OWASP:

- OWASP *Top* 10;
- OWASP *Common Vulnerability List*; e
- OWASP *Risk Rating Methodology*.

Esta comunicação entre variado e importante material da OWASP faz com que seja muito apetecível a utilização do *Testing Guide*.

Para 2011 está já proposta a publicação de uma nova versão do OWASP *Testing Guide* (Matteo Meucci, Girogio Fedon e Pavol Luptak, 2011), a qual trará inúmeras melhorias não só à versão anterior como na interligação como o restante material da OWASP.

2.5 *Web Scanners*

De um modo muito sucinto, os *web scanners* podem ser definidos como ferramentas automáticas que realizam testes de intrusão em aplicações *Web* (Fong & Okun, 2007). Nesta secção é proposto não só definir o que é um *web scanner* e quais as suas principais funcionalidades, como descrever quais são usualmente os melhores métodos de análise e comparação destas ferramentas.

2.6 Definição e funcionalidades

Os *web scanners* analisam uma aplicação através de métodos de *crawling*, percorrendo toda a aplicação e analisando os seus conteúdos através da aplicação de funções internas que inspeccionam o código e o seu comportamento de acordo com uma lista de vulnerabilidades conhecidas. Esta inspecção pretende imitar ataques realizados por agentes maliciosos que geram *inputs* específicos, enquanto que as respostas e comportamentos gerados são analisados. Estes *inputs* maliciosos são não apenas criados a partir de uma lista conhecida que pretende simular ataques que forcem o aparecimento de determinadas vulnerabilidades, mas também utilizando técnicas de *fuzzing* (Oehlert, 2005), as quais são cada vez mais usuais nos testes aplicacionais.

Hoje em dia os *web scanners* estão disponíveis no mercados sob a forma de *software* comercial e *open source*, assim como através da oferta de *SaaS*. Esta abrangência de vectores de oferta mostra bem o crescimento que estas ferramentas têm tido, especialmente em domínios de integração de segurança no Ciclo de Vida de Desenvolvimento de Software (CVDS). Na maioria dos casos esta relação é bastante conhecida, sendo que este tipo de ferramentas são maioritariamente utilizadas no fim do CVDS, altura onde se realizam testes de pré-produção.

De um modo geral, é possível definir três etapas principais no funcionamento dos *web scanners*:

- Configuração;
- *Crawling;* e
- Análise.

Estas três etapas são, não só essenciais, como permitem que os *web scanners* sejam cada vez mais, considerados ferramentas de extrema relevância em auditorias, conciliando as necessidades específicas de cada utilizador, com a efectividade e a facilidade de utilização.

A fase da configuração inclui a definição de um URL ou IP da aplicação a ser testada, assim como a definição de diferentes parâmetros relacionados com as funcionalidades dos processos de *crawling* e análise. Esta é uma característica importante dos *web scanners* pois permite que sejam definidas profundidades e excepções de *crawling*,

assim como parameterizações das análises de vulnerabilidades e ser efectuadas. Estas capacidades afectam as questões de granularidade de *crawling* e análise, criando variações no tempo de execução do w*eb scanner*, factor que pode ser importante em determinados cenários.

As configurações dos parâmetros de análise de vulnerabilidades e a capacidade de parar e continuar com o processo de análise são de extrema relevância já que fornece a flexibilidade que muitos dos testes requerem. Estas configurações usualmente incluem a definição de perfis de utilização onde as costumizações são agrupadas, criando um nível adicional na facilidade e rapidez de utilização. Para além das características enunciadas destaca-se ainda a importância da funcionalidade de calendarização. Esta funcionalidade permite definir um processo automático para testes previamente definidos e configurados, iniciando e terminando o processo de análise em espaços temporais e intervalos de tempo controlados.

A capacidade de realizar manualmente ataques e pedidos específicos, dentro do ambiente gráfico destas ferramentas, é algo muito apreciado pelos profissionais de segurança, utilizadores deste tipo de ferramentas. Isto deve-se ao facto de que estas ferramentas não conseguem detectar vulnerabilidades para as quais não foram programadas para detectar, catalisando a necessidade de inclusão de funcionalidades de testes manuais, fornecendo um nível de conciliação entre a automatização e especificidade de análise de determinadas vulnerabilidades, fornecendo uma melhor base para atingir altos níveis de confiança na sua utilização.

A fase de *crawling* (Gautam Pant, Padmini Srinivasan e Filippo Menczer , 2003) é uma das mais importantes pois é nesta fase que o *web scanner* se "apercebe" da aplicação a testar. Como resultado, o *web scanner* irá produzir um mapeamento da estrutura da aplicação, contruindo uma árvore com essa estrutura, onde serão mapeados todos os conteúdos como páginas HTML, pastas, ficheiros, *scripts*, etc. Este mapeamento é crucial. A falha na identificação de recursos da aplicação irá resultar num resultado global deficiente, já que existem conteúdos que não foram testados. De um modo geral, o processo de *cawling* pode ser visto como exemplificado na Figura 2.

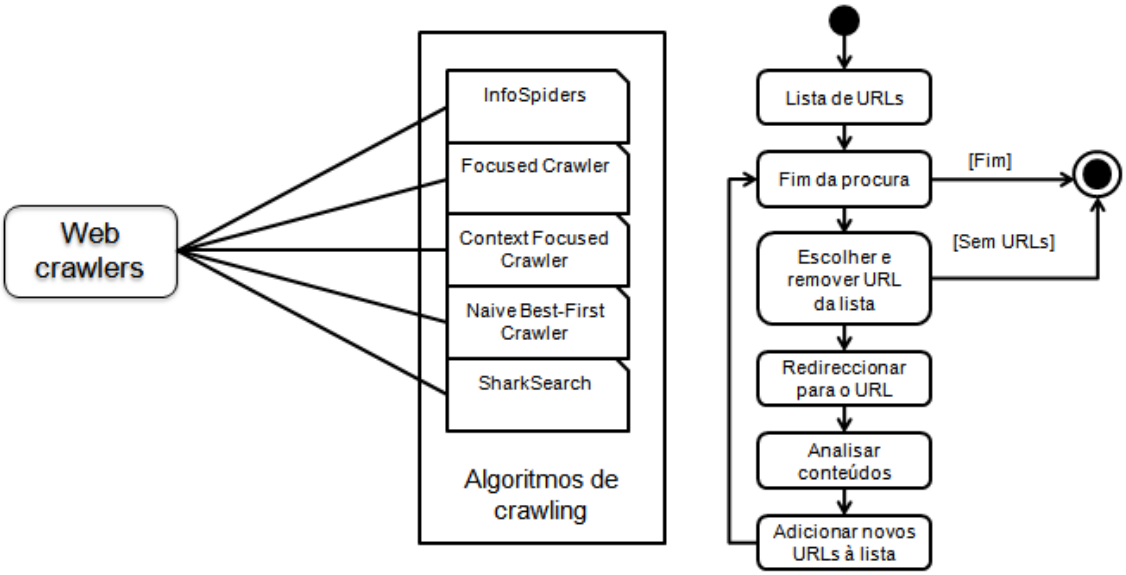


Figura 2 - Processo de *crawling*

Tal como demonstrado pela figura, o *crawler* parte de um URL inicial, a partir do qual indexa todos os *links* que vai encontrando, baseando-se em vários algoritmos, examinando o conteúdo que vai percorrendo.

Por último, a terceira fase, onde se efectuam as análises de vulnerabilidades, é de facto a razão da utilização destas ferramentas. Nesta fase o *web scanner* irá realizar testes de intrusão automatizados à aplicação *Web* em causa. Estes testes envolvem a simulação de ataques à aplicação, sendo que numa abordagem *black box*, seria exactamente o que um atacante faria através da criação de *inputs* específicos, analisando posteriormente os *outputs* obtidos. Nesta fase será efectuado um elevado número de testes, usualmente configurados e pré-definidos durante a fase de configuração. O resultado destes testes será tratado e apresentado ao utilizador numa forma estruturada e perceptível.

Adicionalmente, a capacidade de ver resultados em tempo real e de permitir a sua manipulação, fornece um alto nível de interactividade para profissionais que não querem depender apenas de soluções *Point and Shot* (P&S). A maioria dos *web scanners* actuais migrou de um ambiente de execução em *shell* para a utilização de uma *Graphical User Interface* (GUI) que permite uma exibição de dados ordenada e agradável. Esta característica é extremamente útil pois reduz a inércia de utilização deste tipo de ferramentas devido à redução de conhecimento técnico que é necessário para as utilizar.

A geração de relatórios é uma das características mais procuradas nos *web scanners*. A importância dada a esta funcionalidade deriva da própria capacidade do *web scanner* fornecer relatórios com as vulnerabilidades encontradas, os vectores de ataque, formas de mitigação, mapeamento das falhas de segurança com regras internacionais de conformidade, entre outros. Este factor permite diminuir o esforço na elaboração de relatório fastidiosos, ao mesmo tempo em que fornece um entregável perceptível a pessoas de topo na organização, onde os resultados técnicos pouco importam, mas sim o resultado final, numa linguagem reconhecida por eles.

2.7 Análise e comparação

Os *web scanners* existem em grande número e com variadas funcionalidades. Esta enorme oferta torna extremamente complicado e moroso perceber qual é que "a melhor" ferramenta a utilizar.

Embora seja uma tendência perguntar qual é o melhor *web scanner*, a verdade é que, apesar desta pergunta não ter uma resposta única e de fácil explicação, existem métodos que podem fornecer uma visão aproximada das necessidades de cada utilizador. O factor anteriormente descrito é facilmente comprovado se pensarmos na especificidade de cada utilizador no que toca às suas necessidades, tempo para realização dos testes, conhecimento técnico, plataforma alvo, entre outros.

Como reposta diferentes metodologias para análise e avaliação de *web scanners* foram desenvolvidas. As metodologias aqui apresentadas possuem uma grande aceitação por parte dos profissionais de segurança que utilizam estas ferramentas pois fornecem mecanismos de medição quantitativa e qualitatva das características dos *web scanners*.

2.8 *Web Application Security Scanner Functional Specification*

O Instituto Nacional de Standards e Tecnologia (NIST, National Institute of Standards and Technology) possui um projecto denominado *Software Assurance Metrics and Tool Evaluation* (NIST, SAMATE) o qual é apoiado pelo *Department of Homeland Security National Cyber Security Division* dos Estados Unidos da América (DHS, National Cyber Security Division).

Parte do projecto SAMATE consiste na identificação e avaliação da segurança de *software*, onde os *web scanners* se incluem. Mais concretamente, no âmbito dos *webs scanners*, este projecto denomina-se *Web Application Security Scanner Functional Specification* (NIST, WASSFS). O principal objectivo do WASSFS consiste em definir requisitos mínimo para características obrigatórias e opcionais de *web scanners*. De um modo geral, é definido que os *web scanners* devem ser capazes de:

- Identificar tipos especificos de vulnerabilidades em aplicações *Web*;
- Para cada vulnerabilidade identificada, gerar um relatório que indique um ataque efectuado; e
- Identificar vulnerabilidades com uma taxa de falsos positivos baixa o suficiente para ser aceitável.

Opcionalmente, um *web scanner* deve reunir as seguintes condições:

- Produzir um relatório compatível com outras ferramentas;
- Permitir que determinadas vulnerabilidades sejam suprimidas pelo utilizador; e
- Utilizar nomes padrão para classificação de vulnerabilidades.

Apesar de ter sido um projecto relativamente importante, desde 1 de Janeiro de 2010 deixou de ser continuado, passando a suportar a *framework* de análise e avaliação de *web scanner* que o viria a substituir, o WASSEC.

2.9 *Web Application Security Scanner Evaluation Criteria*

O *Web Application Security Scanner Evaluation Criteria* (WASC, WASSEC) é um extenso documento, produzido pela WASC, que pretende cobrir a maioria dos aspectos relacionados com a avaliação dos *web scanners*. De uma forma bastante sucinta, os tópicos definidos no WASSEC para avaliação de *web scanners*, são:

- Suporte de protocolos;
- Autenticação;
- Gestão de sessões;
- *Crawling*;
- *Parsing*;
- Testes;

- Comandos e controlos; e
- Relatórios.

2.10 Modelos de avaliação costumizados

Para além das *frameworks* de avaliação anteriormente descritas, existem modelos bastante mais flexíveis e que se apoiam num consenso geral em vez de uma listagem de pontos definidos num documento.

Na sua generalidade, os modelos de avaliação costumizados focam-se unicamente nas capacidade de detecção de vulnerabilidades dos *web scanners*.

O princípio por detrás destes modelos consiste na escolha de um conjunto de vulnerabilidades a testar e, para cada uma, implementar múltiplas instâncias de teste, desde o facilmente explorável e detectável, até ao limite de impossível de explorar e detectar. Para que tal seja possível deverão ser criadas camadas sobre cada vulnerabilidade, consistindo essas camadas em mecanismos de protecção. De um modo sucinto este processo pode ser visto na seguinte sequência:

1. Seleccionar vulnerabilidade a testar;
2. Criar níveis de defesa baseados na informação disponível em como prevenir a exploração da vulnerabilidade; e
3. Analisar o comportamento do *web scanner* para cada nível.

A sequência de passos acima definidos, juntamente com a metodologia WASSEC, quando utilizados em conjunto, definem muito certamente uma das formas mais fiáveis de análise e avaliação de *web scanners*, combinando resultados práticos com funcionalidades teóricas das ferramentas.

Muitos fabricantes optam por desenvolver aplicações *Web* públicas, com um conjunto de vulnerabilidades bem identificadas, precisamente para facilitar o teste dos *web scanners*. Usualmente este tipo de testes e os resultados não são considerados fiáveis dado que estas aplicações são desenvolvidas de modo a que as vulnerabilidades existentes serão sempre identificadas pelas ferramentas, dando uma falsa sensação de fiabilidade. Apesar do facto anteriormente referido, no mínimo estes testes fornecerão

alguma informação superficial considerada relevante pelo que, em última alternativa, deverão ser considerados. De entre as aplicações para teste destacam-se as seguintes:

- Cenzic;
- Watchfire;
- WebMaven/ Buggy Bank;
- HackmeBank;
- Stanford SecuriBench; e
- OWASP Site Generator.

Capítulo 3 Proposta de Solução

3.1 Análise do Enquadramento Legal das Auditorias de Segurança

Antes de se poder realizar qualquer tipo de avaliação da solução proposta para a realização deste trabalho, é necessário perceber e avaliar o enquadramento legal da realização de auditorias de segurança *Web* em Portugal.

Em 23 de Julho de 2009, foi aprovada a Lei do Cibercrime – Lei nº 109/2009, de 15 de Setembro (PGDL, 2009), tendo sido posteriormente promulgada a 29 de Agosto de 2009.

A análise desta lei é fundamental para os processos de pré-auditoria já que, no artigo 6º, alínea 1 do Capítulo 1, é apontado o seguinte:

"Quem, sem permissão legal ou sem para tanto estar autorizado pelo proprietário, por outro titular do direito do sistema ou de parte dele, de qualquer modo aceder a um sistema informático, é punido com pena de prisão até 1 ano ou com pena de multa até 120 dias."

Adicionalmente, no artigo 7º, alínea 1 do Capítulo 1, é descrito o seguinte:

"Quem, sem permissão legal ou sem para tanto estar autorizado pelo proprietário, por outro titular do direito do sistema ou de parte dele, e através de meios técnicos, interceptar transmissões de dados informáticos que se processam no interior de um sistema informático, a ele destinadas ou dele provenientes, é punido com pena de prisão até 3 anos ou com pena de multa."

Sendo que, qualquer acção realizada para o processo de auditoria necessita inevitavelmente de um acesso ao sistema informático e da intercepção das suas respostas, torna-se obrigatória a notificação e obtenção do consentimento legal por parte das entidades que farão parte do processo de auditoria.

3.2 Identificação de Grupos para Auditoria

A identificação dos grupos para a realização dos testes de auditoria foi uma das principais actividades para determinar o sucesso do cumprimento dos objectivos propostos para a presente dissertação.

Se por um lado, existe a necessidade de escolher grupos representativos e que realizem operações críticas em Portugal, por outro, existe também a consciência da dificuldade de aceitação, por parte das entidades pertencentes a esses grupos, para participação nos testes de auditoria que enquadram este trabalho.

São então identificados dois pré-requisitos para a selecção de entidades constituíntes dos grupos para as auditorias:

- Realização de operações críticas e manipulação de dados sensíveis no contexto português; e
- Abrangência de um elevado número de pessoas em Portugal.

Analisando os mais recentes e abrangentes eventos de migração de serviços para ambientes *Web,* em Portugal, torna-se imperativa a inclusão de entidades governamentais neste estudo. Esta migração deriva da tentativa do governo português de diminuir o fosso entre cidadãos e serviços do estado, tendo sido criado o programa Simplex (SIMPLEX, 2006), derivando numa maior e mais rápida interacção entre ambos.

Um grupo de entidades que é sempre meritório de atenção no que respeita a segurança *Web* e realização de operações sensíveis, são as entidades financeiras. Este grupo de entidades não só preenche os requisitos identificados previamente, como são um vector importante para a análise do estado de segurança *Web* em Portugal, já que são dos principais grupos que investe em segurança a nível mundial.

As Forças Armadas sempre representaram o ícone da segurança Nacional em qualper país, e em Portugal isso não é excepção. Conscientes desse facto, foram identificadas entidades representativas deste grupo para inclusão nos testes de auditoria.

O facto deste trabalho derivar de uma vertente académica, e de um dos seus objectivos ser precisamente a criação de uma maior proximidade entre a área académica e a

OWASP, foi igualmente decidido incluir como grupo de entidades para testes, instituições do Ensino Superior em Portugal.

Finalmente, foi decidido criar um último grupo, constituído por entidades prestadoras de serviços, onde seriam incluídas quaisquer entidades que, embora mantendo a conformidade com os pré-requisitos identificados, mas não se enquadravam nos grupos anteriormente definidos.

O enquadramento das entidades anteriormente descritas pode ser sumarizado nos grupos abaixo descritos na Figura 3.

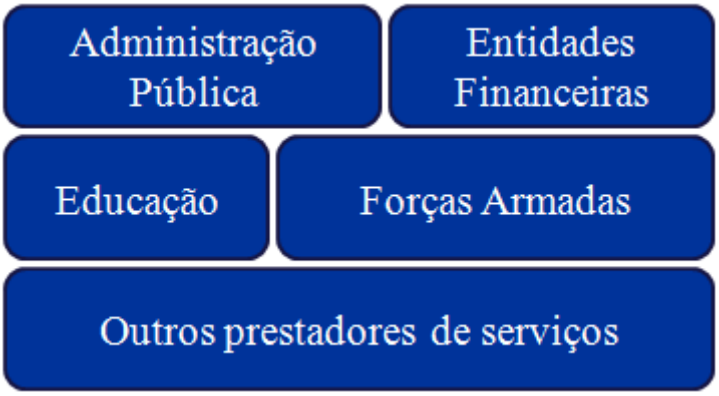


Figura 3 - Grupos de entidades para testes de auditoria

Posteriormente à identificação dos grupos, surge a necessidade da prioritização e análise de risco de rejeição de cada um, processo este que foi feito criando uma escala de 1 a 5 valores (sendo 1 o valor mínimo e 5 o máximo), numa matriz de intersecção, onde foram atribuídas cores identificativas.

Risco / **Prioridade**	1	2	3	4	5
1					
2					
3					
4					
5					

Tabela 3 – Matriz de Prioridade/Risco

Seguidamente a matriz previamente definida foi aplicada aos grupos identificados, identificando os grupos mais críticos e com maior risco de rejeição de participação nos testes de auditoria.

Risco / **Prioridade**	1	2	3	4	5
1					
2	Outros prestadores de serviços				
3	Educação				
4					Forças Armadas
5				Administração Pública	Entidades Financeiras

Tabela 4 – Matriz de Prioridade/Risco aplicada aos grupos

Como é possível verificar, o grupo das Forças Armadas, Bancos e Governo, são os que apresentam maior risco. Este risco pode estar não só associado à recusa na participação nos testes de auditoria, como à aceitação de participação mas sob circunstâncias muito restritas, tanto operacional como legalmente.

3.3 Definição da Metodologia

A definição da metodologia é uma passo extremamente importante no decorrer do restante trabalho, tendo influência directa na capacidade de atingir os objectivos e nos resultados esperados. De um modo geral, na definição da metodologia têm que ser cumpridos dois objectivos principais:

- Fiabilidade; e
- Rapidez.

A fiabilidade é obviamente uma condição inerente a qualquer trabalho sério a ser realizado. A rapidez, neste caso, é importante pois permite a inclusão de mais entidades nos testes, elevando a amostra, criando um resultado total mais apoximado da realidade do panorama Nacional.

3.4 Definição de pressupostos

Sendo uma auditoria de segurança a aplicações *Web* um conjunto de testes de *software*, é possível fazer uma análise dos prévia dos recursos disponibilizados pelas organizações para a realização deste trabalho, comparando com os métodos de testes de software existentes, criando uma base sólida para a criação de uma metodologia.

De modo a diminuir a relutância das organizações em aceitar participar neste trabalho, a aumentar a velocidade de acesso aos dados, e a manter intactas questões de propriedade intelectual (código fonte e conhecimento sobre infra-estrutura), foi decidido optar por realizar auditorias de segurança numa abordagem *black box* como é possível ver na figura 5.

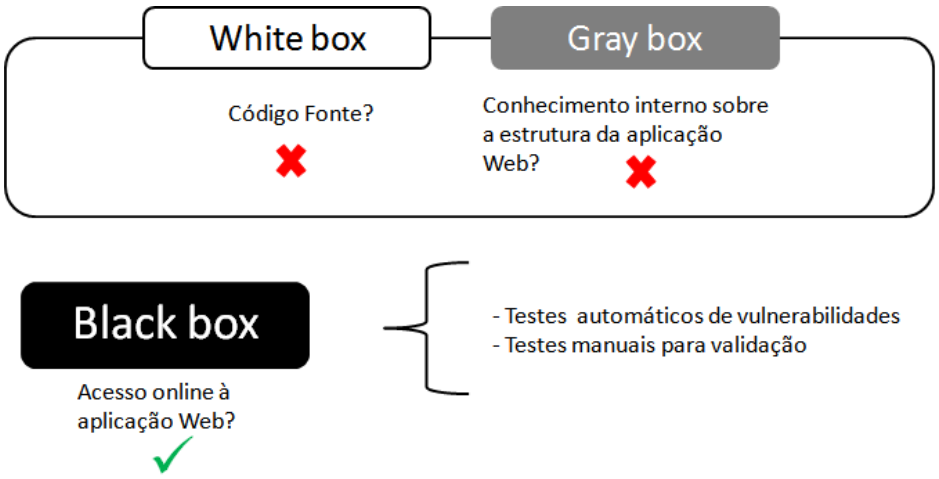


Figura 4 - Abordagem de testes de software para a auditoria

A adopção de uma abordagem *black box* é também concordante com a utilização de ferramentas anteriormente descritas, os *web scanners*.

3.5 Metodologia pré-testes de auditoria

Antes da realização dos testes de auditoria, devido às restrições legais descritas na secção 3.1 do Capítulo 3, é necessário desenvolver uma metodologia de contacto com as entidades que serão envolvidas nos testes (Figura 5).

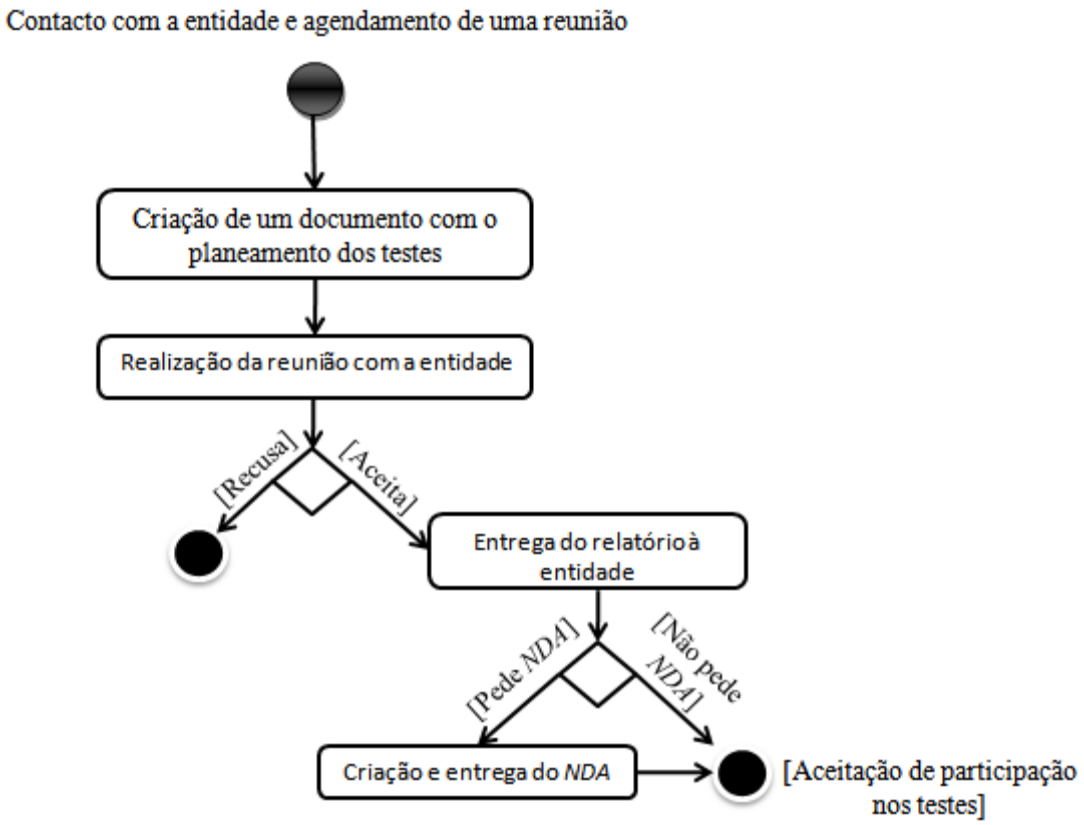


Figura 5 - Metodologia pré-testes de auditoria

Após o contacto com a entidade alvo, será criado um documento com a descrição dos testes e dos termos em que estes serão realizados (Anexo A), o qual será enviado para discussão e análise durante a reunião agendada.

Adicionalmente, e caso seja requerido, será enviado um *Non-Disclosure Agreement* (Anexo B), procurando salvaguardar a entidade que será submetida aos testes de auditoria, através de um contrato legal.

Excepcionalmente, e caso a entidade alvo dos testes necessite de salvaguardar mais questões legais do que as que foram definidas no *Non-Disclosure Agreement* (NDA), esta poderá criar um documento próprio, contendo as informações que no seu entender são relevantes. Posteriormente, este documento será analisado quanto à sua exequibilidade no contexto do trabalho em questão.

3.6 Metodologia dos testes de auditoria

Terminado o processo de pré-auditoria é necessário criar uma metodologia para a realização dos testes, que obedeça aos critérios anteriormente definidos.

Baseado no pressupostos anteriormente descritos, esta metodologia tem que corresponder aos processos identificados na Figura 6.

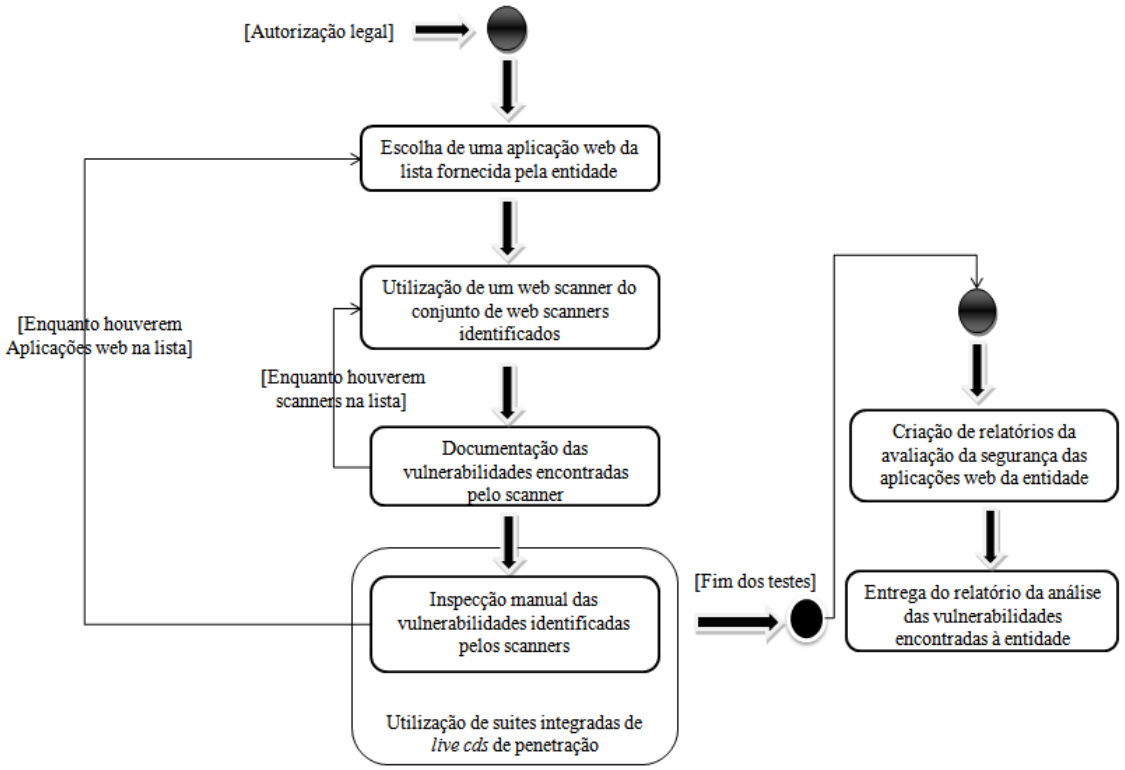


Figura 6 - Metodologia dos testes de auditoria

Como se pode observar, todo o processo tem início após obtenção de autorização legal para a realização dos testes. Seguidamente tem início um processo iterativo onde, por cada aplicação *Web* a ser testada, serão utilizados os *web scanners* e documentados os seus resultados. Finalmente, será realizada uma análise manual a todos os resultados obtidos da análise automática dos *web scanners*, validando os resultados através da eliminação de falsos positivos e da recolha de evidência das vulnerabilidades. Todo o processo culmina na criação e entrega de um relatório à entidade em causa.

A metodologia aqui descrita segue uma abordagem de testes de intrusão que consiste nos seguintes pontos:

1. Reconhecimento – Nesta fase irá ser explorada e examinada a aplicação web e a infra-estrutura que a engloba de um modo superficial. O objectivo desta fase é o de mapear algum contexto que engloba a aplicação *Web*, obtendo alguma informação sobre a mesma;
2. Enumeração – Esta fase é onde irão ser obtidas as informações sobre os serviços e recursos específicos que estão associados à aplicação *Web* que está a ser testada. A obtenção desta informação permite saber quais são as vulnerabilidades que poderão estar associadas a esses serviços, permitindo explorá-las e saber como mitigar a existência das mesmas; e
3. Testes aplicacionais – Nesta etapa é onde os testes aplicacionais realmente irão ser efectuados. Estes testes irão ser levados a cabo consoante as informações recolhidas nas fases anteriores. Deste modo o sucesso de descoberta de possíveis vulnerabilidades irá ser muito maior, levando a uma auditoria da segurança aplicacional muito mais rápida e eficiente.

Caso a meio do processo de auditoria seja confirmada uma vulnerabilidade crítica e de fácil exploração, que represente perigo imediato para a entidade, esta será imediatamente avisada através dos meios de comunicação identificados no documento de proposta de auditoria, previamente aprovado nas actividas de pré-auditoria.

Como é possível observar na metodologia (Figura 6), terminado o processo de auditoria automática realizado pelos *web scanners*, é necessário proceder à inspecção manual das vulnerabilidades identificadas.

Esta inspecção manual é um dos pontos mais importantes da metodologia de auditoria de vulnerabilidades já que a veracidade dos resultados apresentados no relatório derivam precisamente da confirmação das detecções dos *web scanners*. A inspecção manual é também a altura mais morosa das auditorias de vulnerabilidades devido aos complexos processos que são necessários, cujo fluxo qual está descrito na Figura 7.

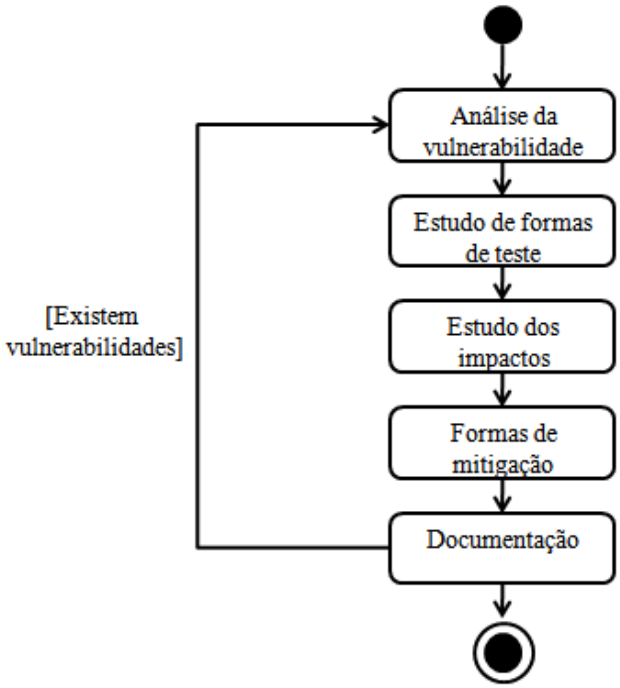


Figura 7 - Metodologia de inspecção manual

O processo de inspecção manual tem início com a análise da vulnerabilidade identificada. Esta análise tem como objectivo perceber com que tipo de vulnerabilidade se está a lidar e quais as suas principais características. Esta primeira abordagem permite que seja recolhida informação preciosa para servir de *input* para as próximas fases.

Seguidamente são estudadas formas de testar a existência dessa vulnerabilidade. Estes testes recorrem a técnicas e ferramentas específicas, sendo que um dos principais utilitários é uma distribuição Linux para testes de intrusão – Backtrack (Linux, Backtrack).

Na próxima fase são identificados quais os impactos que a vulnerabilidade tem na aplicação *Web,* assim como as formas de mitigação.

Finalmente todo este processo e evidências serão documentados, fazendo parte integrante do relatório final.

3.7 Análise e escolha de *web scanners*

Como última etapa da proposta de solução, é necessário analisar e identificar as ferramentas que terão um papel fundamental nos processos da metodologia anteriormente descrita, os *web scanners*.

A escolha dos *web scanners* consistiu, numa primeira instância, na análise entre dois grandes grupos:

- Comerciais; e
- *Open source*.

A identificação de *web scanners* que pertencessem a estes grupos foi feita com recurso à lista fornecida pela WASC, a qual divide estas ferramentas em 3 grupos: comerciais, *software-as-a-service* e open source (WASC, Web Application Security Scanner List).

Como um dos objectivos passa pela criação de uma *framework* de auditoria de vulnerabilidades barata, rápida e eficiente, que possa ser aplicada a contextos organizacionais que possam não ter capacidade de investimento em metodologias e/ou ferramentas de segurança *Web,* ou que o seu grau de maturidade em segurança de informação não o exiga, foi decidido que seriam adoptadas apenas ferramentas *open source*.

Adicionalmente, dada a natureza académica do presente estudo, não foi possível obter fundos para realizar um investimento em ferramentas comerciais. Foram, no entanto, solicitadas cópias funcionais destas ferramentas, sem qualquer tipo de limitações ao nível das funcionalidades que permitissem o normal decorrer dos processos de auditoria enquadrados num estudo desta natureza, solicitações essas que foram rejeitadas pelos respectivos fabricantes.

Mesmo com foco apenas na comunidade *open source*, o número de ferramentas existentes, adicionando à quantidade de análises que seriam necessárias para avaliar todos os resultados obtidos, torna impossível a utilização de todas estas ferramentas. Apesar de não ser desejável possuir demasiadas ferramentas, comprometendo o tempo de execução de testes e obrigando à diminuição do número de aplicações a testar, é também indesejável a utilização de apenas uma ferramenta.

Deste modo, a utilização de apenas uma ferramenta potencia a obtenção de resultados demasiados parciais aos métodos de detecção presentes nessa ferramenta, diminuindo o grau de fiabilidade dos resultados. Devido ao facto anterior, foi decidido que deveriam ser escolhidas 3 ferramentas, criando um equilibro entre o cruzamento de dados provenientes de fontes diferentes, cobrindo potenciais falhas individuais, e ainda assim mantendo a rapidez necessária aos testes.

É então necessário proceder à avaliação destas ferramentas, avaliação essa que deverá seguir os seguintes passos:

- Identificação dos *web scanners* existentes na comunidade *open source*;
- Avaliação da qualidade dos *web scanners* baseada no *feedback* da comunidade; e
- Aplicação do WASSEC.

Os *web scanners* identificados na comunidade *open source* para avaliação estão representados na Figura 8.

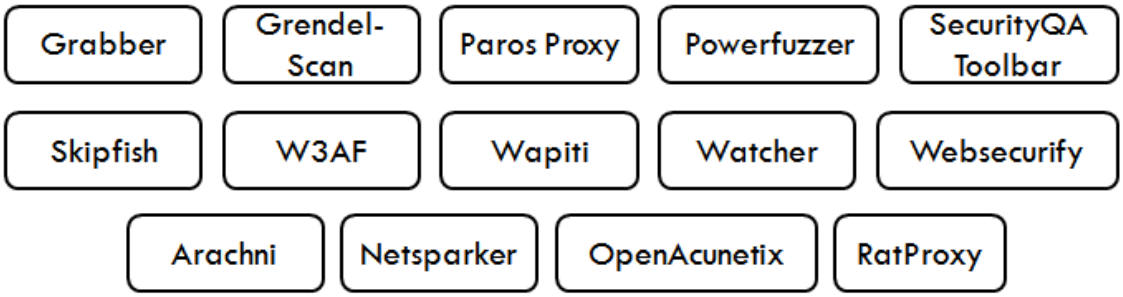


Figura 8 - *Web scanners open source*

Após a identificação dos principais *web scanners* na comunidade *open source*, foi necessário proceder à análise do *feedback* dos profissionais de segurança *web* que utilizam estas ferramentas regularmente.

Após a análise de algumas das características dos *web scanners*, das opiniões de membros da comunidade open source (Open Source Web Application Scanner Poll Results) e de testes realizados a estas ferramentas (Adam Doupé, Marco Cova, e Giovanni Vigna, 2010), foi decidido retirar os *web scanners* indicados seguidamente (Figura 9).

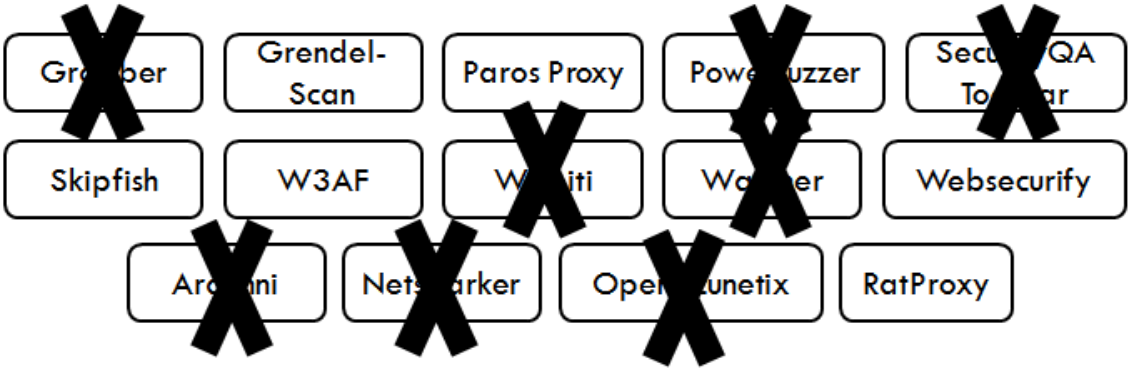


Figura 9 - Primeira filtragem de *web scanners open-source*

Terminada a primeira fase de triagem torna-se necessário proceder à sua avaliação através do WASSEC, obtendo uma lista final, de onde serão escolhidas as 3 com maior pontuação.

No entanto, não será feita uma aplicação directa do WASSEC. De modo a cobrir todas as necessidades individuais, o documento de avaliação WASSEC (WASC, WASSEC Evaluation Spreadsheet) já prevê a inclusão de características costumizadas, criando uma flexibilidade de avaliação bastante importante. Deste modo, foi decidido incluir no WASSEC os seguintes critérios para avaliação:

- Detecção das vulnerabilidades presentes no OWASP *Top* 10 2010;
- Actividade e actualizações recentes;
- Suporte a novas tecnologias;
- Facilidade de interacção com os programadores/criadores para resolução de problemas; e
- Apoio e suporte do *web scanner* por parte de uma organização.

Os três primeiros pontos são auto-explicativos quanto à sua necessidade de inclusão nos pontos de avaliação costumizados no documento de avaliação do WASSEC.

A facilidade de interacção com os programadores/criadores do *web scanners* tem de facto uma importância acentuada. Essa importância deve-se ao facto de que qualquer problema que surja no decorrer dos testes poderá ser mais facilmente mitigado com uma comunidade que tenha capacidade de comunicação e resolução de problemas, limitando o risco de paragem dos testes e consequente atraso na entrega do relatório de auditoria.

Por sua vez, o apoio do *web scanner* por parte de uma organização traz enormes vantagens. A associação da imagem da organização ao *web scanner* faz com que esta liberte recursos que trazem vantagens não só ao nível da capacidade de resposta da comunidade a problemas apresentados, como ao próprio desenvolvimento da ferramentas relativamente às suas funcionalidades e fiabilidade.

Finalmente, após a aplicação do documento de avaliação WASSEC, com as devidas costumizações, aos 6 *web scanners* identificados para esta última fase (Anexo C), é possível observar que os 3 *web scanners* a utilizar para realizar as auditorias de vulnerabilidades serão:

- **W3af** – Bonsai-sec & Rapid7;
- **Websecurify** – Gnucitizen; e
- **Skipfish** – Google.

Os 3 *web scanners* identificados cumprem os requisitos previamente identificados e possuem capacidades de análise e detecção de vulnerabilidades completas e importantes no contexto das aplicações *Web* tal como é possível ver na descrição presente na secção de *Listagem e Descrição das Ferramentas* no Anexo A.

3.8 Classificação de evidências

A classificação de evidências é um factor importante para a identificação e agrupamento de vulnerabilidades no relatório entregue às entidades auditadas. Esta classificação permite não só saber com detalhe quais as falhas de segurança identificadas, como também analisar o risco presente, numa primeira fase em cada aplicação auditada, e posteriormente, no panorama global da organização.

3.9 Vulnerabilidades

Apesar da existência de inúmeras classificações de vulnerabilidades (CVSS, CWE, CCE, entre outros), foi decidido que, de acordo com a metodologia definida, seria mais coerente manter uma classificação concordante com as classificações indicadas pelos *web scanners*.

Adicionalmente, qualquer vulnerabilidade mal classificada, ou sem uma classificação definida por parte do *Web scanner* será classificada, sempre que possível, segundo a OWASP *Common Vulnerability List* (OWASP, Common Vulnerability List).

O agrupamento de vulnerabilidades com características semelhantes é também uma classificação importante, na medida que fornece uma perspectiva de alto nível de estado de segurança das aplicações *Web* auditadas. Deste modo, é mais fácil transmitir às entidades auditadas o nível global de criticidade associado às vulnerabilidades descobertas. Foram então definidos três graus de criticidade, so quais estão abaixo descritos e devidamente identificados com as respectivas cores:

Grau 1 – Vulnerabilidades consideradas muito graves e que podem comprometer gravemente o sistema. Serão também incluídas neste grupo vulnerabilidades que, devido à sua fácil exploração, produzam uma elevada probabilidade de ataques;

Grau 2 – Vulnerabilidades consideradas graves e que podem comprometer o sistema até determinado ponto, sem a profundidade do Grau 1. Possuem um grau de exploração relativamente complicado, derivando uma probabilidade de ataques média; e

Grau 3 – Vulnerabilidades consideradas pouco graves ou informações que poderão levar ao aparecimento de vulnerabilidades ou ataques. Geralmente neste grau são enquadrados dados informacionais identificados pelas ferramentas, os quais não acarretam riscos de negócio.

3.10 Análise de Risco

A descoberta de vulnerabilidades é algo importante mas que só tem um significado concreto quando associadas ao risco que podem significar para o negócio da organização. Assim, a classificação do risco deriva directamente do tipo de vulnerabilidades confirmadas pela auditoria. De modo a promover uma melhor interacção entre a análise de risco, a metodologia de teste, e a classificação das vulnerabilidades, foi decidido optar pela adopção da análise de risco definida pela OWASP (OWASP Risk Rating Methodology), que também é parte constituínte do OWASP *Testing Guide*.

De acordo com a metodologia de análise de risco escolhida, são identificados 4 grandes grupos de análise, cada um com os respectivos sub-tópicos:

- **Factores dos agentes de ameaça**
  - Nível de perícia
  - Motivo
  - Oportunidade
  - Abrangência
- **Factores da vulnerabilidade**
  - Facilidade de descoberta
  - Facilidade de exploração
  - Nível de conhecimento
  - Capacidade de detecção da intrusão
- **Impacto técnico**
  - Perda de confidencialidade
  - Perda de integridade
  - Perda de disponibilidade
  - Capacidade de responsabilização
- **Impacto no negócio**
  - Danos financeiros
  - Danos de reputação
  - Não concordância com normas internacionais
  - Violação da privacidade

No entando, existem alguns pontos chave que tornam necessária a costumização da metodologia de análise de risco anteriormente definida, facto também suportado pela própria *OWASP*, onde é veemente a necessidade de produzir um modelo de análise que se adapte às necessidades e condições envolventes ao próprio domínio da análise. Estes pontos são os seguintes:

- Fraco conhecimento das peculiaridades do negócio de cada entidade – todo o enquadramento do processo de auditoria baseia-se numa abordagem puramente técnica, facto pelo qual não existe conhecimento do negócio com o nível de profundidade requerido para correctamente encarar a metodologia de análise de

risco anteriormente descrita na sua totalidade, assim como outras metodologias de análise de risco mais profundas;

- Necessidade de agilização da análise de risco – dado que o foco principal está centrado na temática tecnológica da segurança aplicacional, torna-se necessário elaborar uma análise de risco que seja ágil o suficiente para poder fornecer dados importantes à entidade auditada, e ainda assim não comprometer os processos de auditoria de vulnerabilidades pelo tempo dispendido na sua elaboração;

Foi então decidido eliminar os seguintes sub-tópicos:

- **Factores dos agentes de ameaça**
    - Abrangência

A abrangência dos factores de ameaça, pretende identificar o tamanho do grupo dos agentes de ameaça (por exemplo, programadores, utilizadores de Intranet, administradores de base de dados, parceiros de negócio, entre outros). Este tipo de análise não é possível dado o desconhecimento da constituição e fluxo de processos internos das organizações auditadas.

- **Factores da vulnerabilidade**
    - Capacidade de detecção de intrusão

A capacidade de detecção de intrusão pretende identificar os mecanismos (caso existam) postos em prática na organização para detectar e registar a tentativa de ataques. Não existe, baseado nos pressupostos para os processos de auditoria e na relação com as organizações envolvidas, o conhecimento do sistema de infra-estrutura ao nível do registo das acções realizadas por agentes externos.

- **Impacto técnico**
    - Capacidade de responsabilização

A capacidade de responsabilização pretende avaliar a capacidade de identificar a origem individual do agente de ameaça. A remoção deste factor do impacto técnico prende-se com o facto de que o agente de ameaça é uma constante, estando sempre enquadrado com as ferramentas automáticas de análise de vulnerabilidades, não acrescentado uma mais valia a sua inclusão na análise técnica.

O modelo costumizado de análise de risco irá então compreender os seguintes grupos e respectivos sub-tópicos:

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade

Tabela 5 – Costumização do grupo Factores dos agentes de ameaça

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento

Tabela 6 – Costumização do grupo Factores da vulnerabilidade

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade

Tabela 7 – Costumização do grupo Factores dos agentes de ameaça

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade

Tabela 8 – Costumização do grupo Impacto no negócio

Todos os sub-tópicos serão avaliados numa escala de 0 a 9, sendo que os valores numérios, enquadrados num intervalo definido, correspondem a uma descrição qualitativa do risco, a qual se encontra descrita na tabela abaixo representada.

Probabilidade e níveis de impacto	
[0,3[	Baixo
[3,6[	Médio
[6,9]	Alto

Tabela 9 – Mapeamento da probabilidade e níveis de impacto

As condições de realização dos testes, enquadradas nos pressupostos da análise de vulnerabilidades definida, deriva em valores constantes para alguns sub-tópicos, valores esses definidos na própria metodologia da OWASP. Nos restantes sub-tópicos, será feita uma comparação entre todas as vulnerabilidades encontradas, classificando-as quantitativamente, tendo em conta a natureza da vulnerabilidade e o impacto que esta geralmente apresenta para o sub-tópico em análise.

Dado o interesse da análise de risco ser baseada numa perspectiva de análise de sectores de actividade, a aplicação do modelo de análise de risco costumizado, previamente definido, irá ser feito sempre aplicado ao nível do grupo e não das aplicações *Web* individualmente ou das entidades.
Deste modo, serão analisadas todas as vulnerabilidades na totalidade do grupo e será feito o mapeamento com os valores numéricos definidos na Tabela 8 e com as métricas de análise de risco anteriormente descritas.

Como última etapa, será utilizada a matriz de mapeamento de probabilidade e impacto abaixo representada (Tabela 10).

Valor de Risco Geral				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 10 – Cálculo do Valor de Risco Geral

3.11 Protecção dos dados da auditoria

Dada a natureza dos dados recolhidos na auditoria, torna-se imperativo assegurar a protecção e confidencialidade dos dados de todo o processo.

O relatório, resultados das análises das ferramentas, resultados de análises manuais e qualquer outro tipo de informação referente à organização será armazenado num dispositivo com encriptação em *hardware* de AES de 256 *bits*, onde qualquer tentativa de força bruta levará à eliminação permanente dos dados, protegendo-os em qualquer situação.

Adicionalmente serão realizadas cópias de segurança num volume TrueCrypt encriptado com AES de 256 *bits* e algoritmo *hash* SHA de 512 *bits* ao longo da auditoria, assegurando que nenhuma informação é perdida antes da entrega do relatório final.

Todos os dados originais e cópias serão eliminadas após o fim da auditoria e com a entrega do relatório final à respectiva entidade, assegurando que a única fonte dos

resultados se mantém com a entidade auditada, não deixando lugar para qualquer tipo de fuga de informação por parte da equipa de auditoria.

Capítulo 4 Análise de Resultados

Ao longo do trabalho realizado nesta dissertação foram auditadas um total de 42 aplicações *Web* num conjunto de 7 entidades, resultando em mais de 3500 páginas de relatórios. Por motivos de confidencialidade, apesar da apresentação de resultados enquadrada em cada grupo, nunca serão indicadas quais as aplicações *Web* específicas que foram alvo da auditoria nem o nome das entidades que as suportam.

4.1 Vulnerabilidades Identificadas

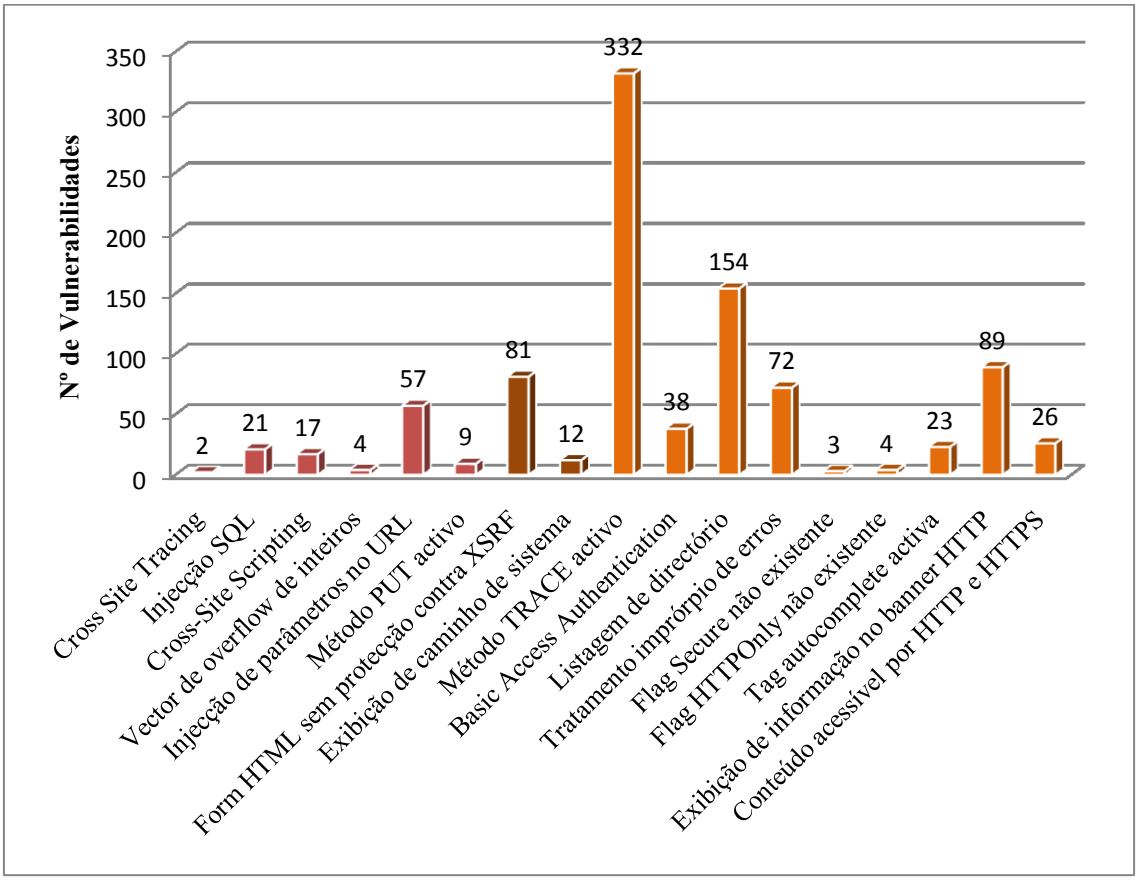


Figura 10 - Total de vulnerabilidades do estudo

Como resultados finais, foram identificadas e confirmadas um total de 944 vulnerabilidades, as quais estão identificadas no gráfico abaixo representado.

Como se pode observar, foram confirmadas um total de 944 vulnerabilidades em que 110 vulnerabilidades são de Grau 1, 93 vulnerabilidades de Grau 2 e 741 de Grau 3. Esta distribuição pode ser vista na Figura 11.

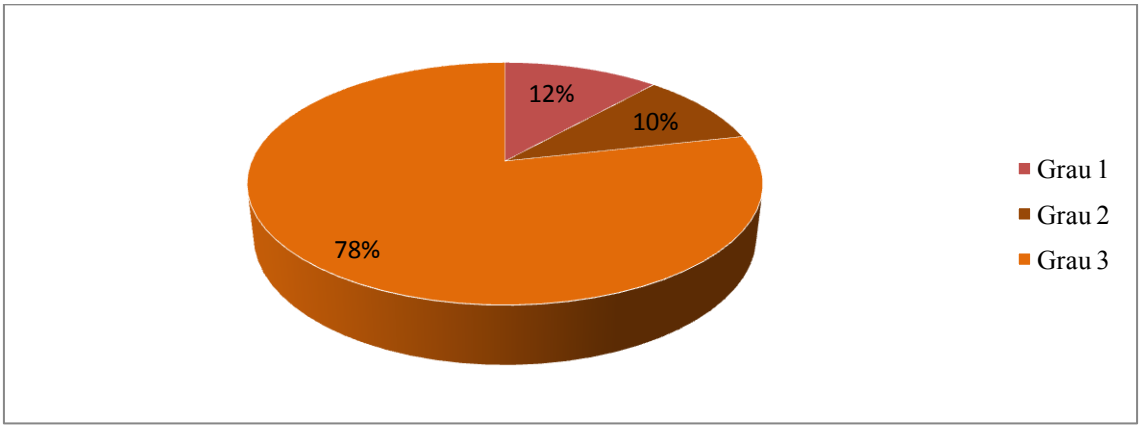


Figura 11 - Distribuição da totalidade das vulnerabilidades

Adicionalmente, no gráfico abaixo pode ser verificada a distribuição na classificação do OWASP Top 10.

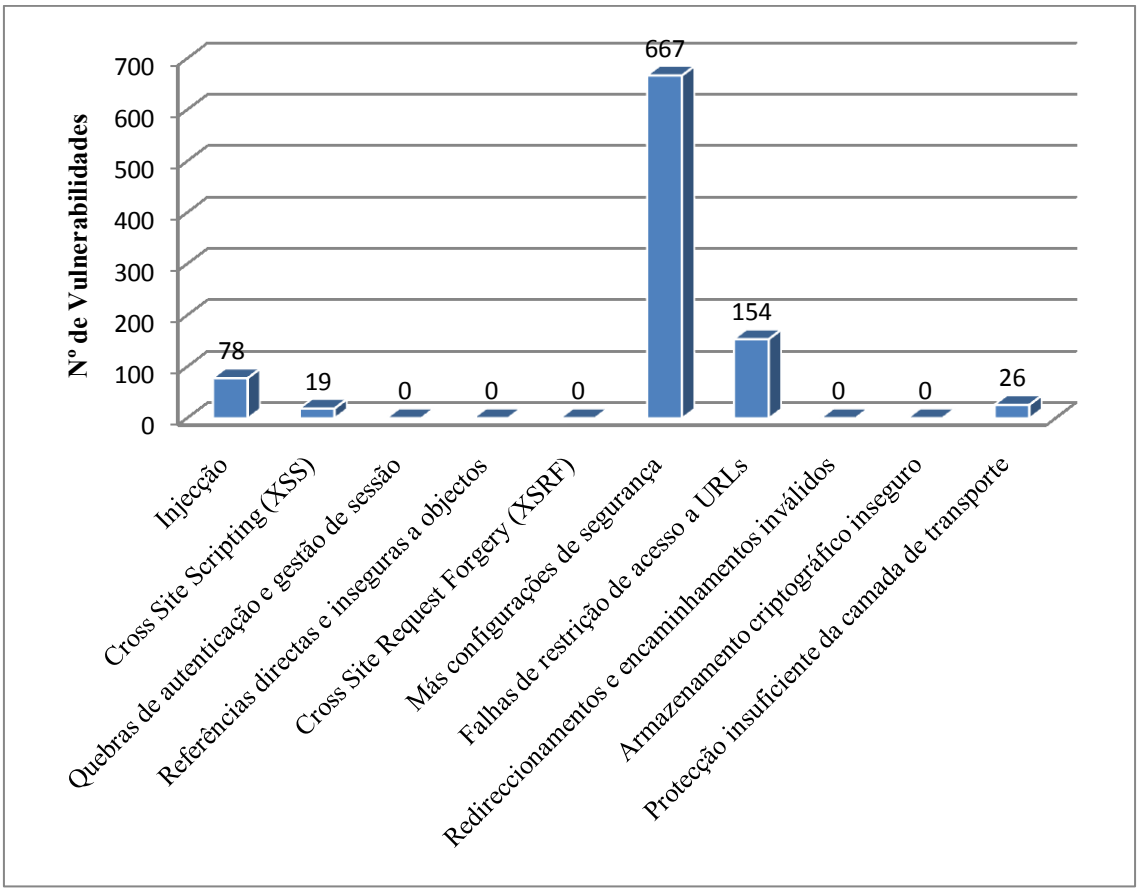


Figura 12 - Resultados totais no OWASP Top 10

4.1.1 Análise de Risco

A análise de risco será feita, nesta fase tendo em conta a totalidade das vulnerabilidades identificadas em todos os grupos de estudo previamente definidos. Esta análise tem como objectivo perceber o grau de maturidade geral em segurança em aplicações *Web*, permitindo extrapolar os resultados para um âmbito mais alargado, dentro do espectro português.

De acordo com a metodologia de análise de risco definida, torna-se necessário realizar um conjunto de atribuições de métricas que permitam obter resultados consistentes e fiáveis. Esta atribuição de métricas servirá como base para todas as análises de risco realizadas no presente estudo, sendo aqui definidos os valores e ponderações.

No grupo dos factores dos agentes de ameaça os valores, excepto o motivo, os valores atribuídos aos restantes sub-tópicos são estáticos, sendo definidos pela própria metodologia, os quais, enquadrados no contexto do presente estudo, relacionam-se da seguinte maneira:

- Nível de perícia – perícia concordante com profissionais de segurança e testes de intrusão, sendo atribuído o valor 9.
- Oportunidade – São necessário alguns recursos em termos de interacção com as entidades e de software para a realização dos testes, sendo atribuído o valor 7.

O valor associado ao motivo estará associado à prioridade previamente definida na Tabela 3, sendo neste caso 9.

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade
9	9	7

O valor associado aos factores de ameaça é então definido por:

(9+9+7) / 3 = 8,333

No grupo dos factores da vulnerabilidade, todos os valores são estáticos, sendo igualmente definidos pela própria metodologia, os quais, enquadrados no contexto do presente estudo, relacionam-se com os respetivos sub-tópicos da seguinte maneira:

- Facilidade de descoberta – facilidade de descoberta baseada puramente em ferramentas automáticas, sendo atribuído o valor 9.
- Facilidade de exploração – Dado que a presente medotologia de estudo pressupõe que nunca existirá lugar a exploração, esta será sempre tida como teórica, pelo que será atribuído o valor 1.
- Facilidade de exploração – Dada a descoberta de todas as vulnerabilidades ser realizada através de ferramentas de análise automática, as vulnerabilidades têm obrigatoriamente que ser do conhecimento público, facto pelo qual é atribuído o valor 9.

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento
9	1	9

O valor associado aos factores da vulnerabilidade é então definido por:

(9+1+9) / 3 = 6.333

Para que seja possível calcular o valor associado ao grupo do impacto técnico é necessário definir os valores associados aos sub-tópicos de confidencialidade, integridade e disponibilidade. A definição deste valores foi definida tendo em conta dois factores:

- Percentagem de tipo de vulnerabilidade, na totalidade das vulnerabilidades confirmadas; e
- Peso ponderado entre 0 e 9 associado ao vector de confidencialidade, integridade ou disponibilidade que essa vulnerabilidade representa de acordo com os impactos mais comuns derivado dessa mesma vulnerabilidade.

Vulnerabilidade	**Confidencialidade**
Injecção SQL	9
Cross-Site Scripting	9
Exibição de caminho de sistema	9
Listagem de directório	9
Conteúdo acessível por HTTP e HTTPS	9
Flag Secure não existente	8
Flag HTTPOnly não existente	8
Tratamento impróprio de erros	8
Exibição de informação no *banner* HTTP	8
Form HTML sem protecção contra XSRF	8

Cross Site Tracing	7
Basic Access Authentication	5
Vector de *overflow* de inteiros	5
Método TRACE activo	5
Tag autocomplete activa	4
Injecção de parâmetros no URL	1
Método PUT activo	1

Vulnerabilidade	**Integridade**
Injecção SQL	9
Cross-Site Scripting	8
Cross Site Tracing	8
Injecção de parâmetros no URL	8
Basic Access Authentication	7
Vector de *overflow* de inteiros	5
Método PUT activo	5
Conteúdo acessível por HTTP e HTTPS	5
Tratamento impróprio de erros	5
Flag HTTPOnly não existente	4
Exibição de caminho de sistema	4
Flag Secure não existente	4
Método TRACE activo	3
Listagem de directório	1
Form HTML sem protecção contra XSRF	1
Exibição de informação no *banner* HTTP	1
Tag autocomplete activa	1

Vulnerabilidade	**Disponibilidade**
Cross-Site Scripting	9
Injecção SQL	8
Basic Access Authentication	8
Vector de *overflow* de inteiros	7
Método PUT activo	6
Tratamento impróprio de erros	5
Flag HTTPOnly não existente	5
Exibição de caminho de sistema	5
Flag Secure não existente	5
Cross Site Tracing	2
Form HTML sem protecção contra XSRF	2
Tag *autocomplete* activa	2
Método TRACE activo	1
Injecção de parâmetros no URL	1
Listagem de directório	1
Conteúdo acessível por HTTP e HTTPS	1
Exibição de informação no *banner* HTTP	1

Utilizando as tabelas de mapeamento das vulnerabilidades e pesos previamente definidas, serão criados os valores totais correspondentes aos vectores de confidencialidade, integridade e disponibilidade, como se pode ver nas tabelas abaixo definidas.

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Confidencialidade (%/100)*Peso**
Cross-Site Scripting	17	1,801	0,018	9	0,162
Injecção SQL	21	2,225	0,022	9	0,198
Basic Access Authentication	38	4,025	0,040	9	0,36
Vector de *overflow* de inteiros	5	0,530	0,005	9	0,045
Método PUT activo	9	0,953	0,010	9	0,09
Tratamento impróprio de erros	72	7,627	0,076	8	0,608
Flag HTTPOnly não existente	4	0,424	0,004	8	0,032
Exibição de caminho de sistema	12	1,271	0,013	8	0,104
Flag Secure não existente	3	0,318	0,003	8	0,024
Cross Site Tracing	2	0,212	0,002	8	0,016
Form HTML sem protecção contra XSRF	81	8,581	0,086	7	0,602
Tag autocomplete activa	23	2,436	0,024	5	0,12
Método TRACE activo	332	35,169	0,352	5	1,76
Injecção de parâmetros no URL	57	6,038	0,060	5	0,3
Listagem de directório	154	16,314	0,163	4	0,652
Conteúdo acessível por HTTP e HTTPS	26	2,754	0,028	1	0,028
Exibição de informação no *banner* HTTP	89	9,428	0,094	1	0,094
Total	944	100	1		5

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Integridade (%/100)*Peso**
Cross-Site Scripting	17	1,801	0,018	9	0,162
Injecção SQL	21	2,225	0,022	8	0,176
Basic Access Authentication	38	4,025	0,040	8	0,32
Vector de *overflow* de inteiros	5	0,530	0,005	8	0,04
Método PUT activo	9	0,953	0,010	7	0,07
Tratamento impróprio de erros	72	7,627	0,076	5	0,38
Flag HTTPOnly não existente	4	0,424	0,004	5	0,02
Exibição de caminho de sistema	12	1,271	0,013	5	0,065
Flag Secure não existente	3	0,318	0,003	5	0,015
Cross Site Tracing	2	0,212	0,002	4	0,008

Form HTML sem protecção contra XSRF	81	8,581	0,086	4	0,344
Tag autocomplete activa	23	2,436	0,024	4	0,096
Método TRACE activo	332	35,169	0,352	3	1,056
Injecção de parâmetros no URL	57	6,038	0,060	1	0,06
Listagem de directório	154	16,314	0,163	1	0,163
Conteúdo acessível por HTTP e HTTPS	26	2,754	0,028	1	0,028
Exibição de informação no *banner* HTTP	89	9,428	0,094	1	0,094
Total	944	100	1		3

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Disponibilidade (%/100)*Peso**
Cross-Site Scripting	17	1,801	0,018	9	0,162
Injecção SQL	21	2,225	0,022	8	0,176
Basic Access Authentication	38	4,025	0,040	8	0,32
Vector de *overflow* de inteiros	5	0,530	0,005	7	0,035
Método PUT activo	9	0,953	0,010	6	0,06
Tratamento impróprio de erros	72	7,627	0,076	5	0,38
Flag HTTPOnly não existente	4	0,424	0,004	5	0,02
Exibição de caminho de sistema	12	1,271	0,013	5	0,065
Flag Secure não existente	3	0,318	0,003	5	0,015
Cross Site Tracing	2	0,212	0,002	2	0,004
Form HTML sem protecção contra XSRF	81	8,581	0,086	2	0,172
Tag autocomplete activa	23	2,436	0,024	2	0,048
Método TRACE activo	332	35,169	0,352	1	0,352
Injecção de parâmetros no URL	57	6,038	0,060	1	0,06
Listagem de directório	154	16,314	0,163	1	0,163
Conteúdo acessível por HTTP e HTTPS	26	2,754	0,028	1	0,028
Exibição de informação no *banner* HTTP	89	9,428	0,094	1	0,094
Total	944	100	1		2

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade
5	3	2

O valor associado ao impacto técnico é:

(5+3+2) / 3 = 3,333

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade
9	9	7	9

O valor associado ao impacto no negócio é:

(9+9+7+9) / 4 = 8,5

O cálculo do valor de probabilidade total é Alto, dado que:

(8.333 + 6.333) /2 = 7

O cálculo do valor de impacto total é Alto, dado que:

(3.333 + 8.5) /2 = 6

Utilizando a matriz de mapeamendo dos vectores de probabilidade e impacto obtém-se o grau geral de risco de Crítico, como se pode verificar na tabela abaixo.

Grau de Risco				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 11 – Grau de risco total do estudo

A classificação do grau de risco como Crítico espelha bem que, nos sectores em que as entidades auditadas se enquadram, o número e tipo de vulnerabilidades encontradas não está concordante com o grau de maturidade de segurança de informação que deveria existir.

Este resultado demonstra que o impacto, técnico e no negócio, derivado das vulnerabilidades encontradas nas entidades auditadas, associado à probabilidade de ocorrência de exploração destas, pode ter resultados penosos para os nichos de negócio

envolvidos, assim como para o cenário nacional, assumindo que a extrapolação de resultados, embora abusiva dado o tamanho da amostra, retrata a realidade nacional.

4.2 Estado da Segurança das Aplicações Web Analisadas

Para que possam ser mais detalhadamente analisados os resultados individuais que compõem o conjunto de vulnerabilidades previamente explanados, serão aqui expostos os resultados da auditoria a cada aplicação *Web*.

Adicionalmente, será aplicado aos grupos em estudo, a análise de risco previamente definida, procurando fornecer uma perspectiva de mais alto nível, dos problemas associados aos diversos sectores, derivados da sua natureza e das vulnerabilidades neles detectadas.

4.2.1 Administração Pública

Foi avaliada uma entidade no grupo "Administração Pública", sendo que foram auditadas um total de dez aplicações *Web*.

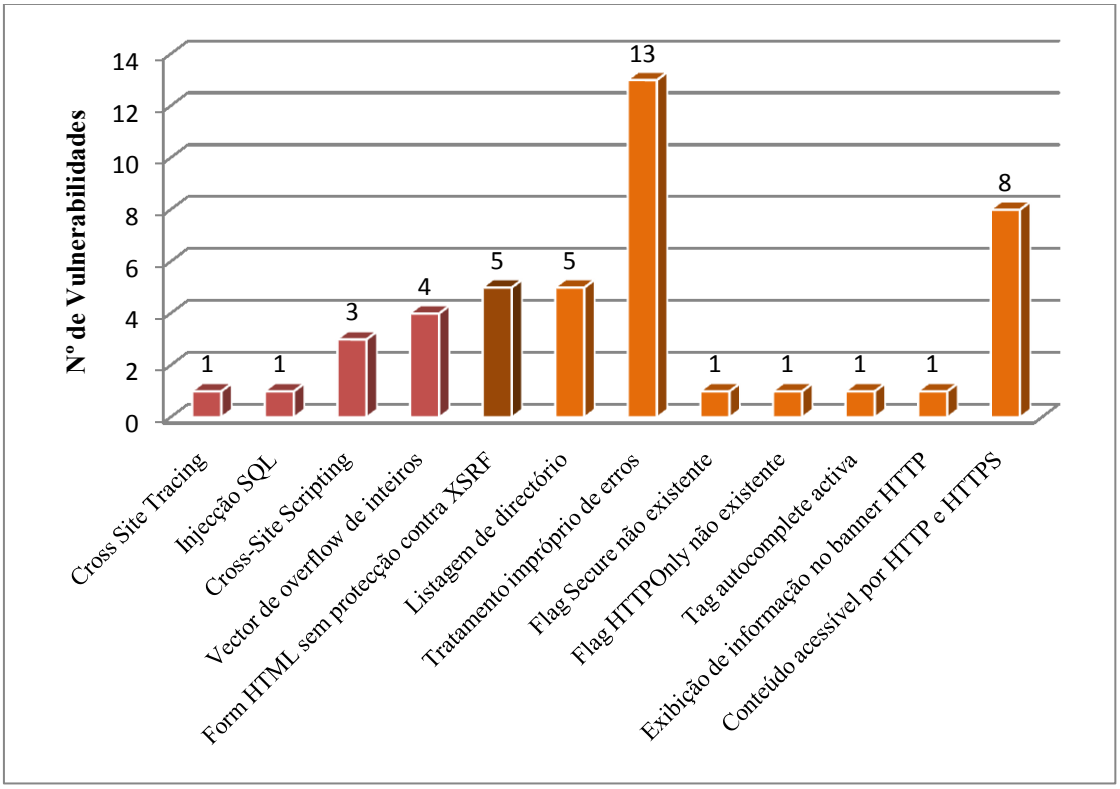


Figura 13 – Aplicação *Web* 1 da Administração Pública

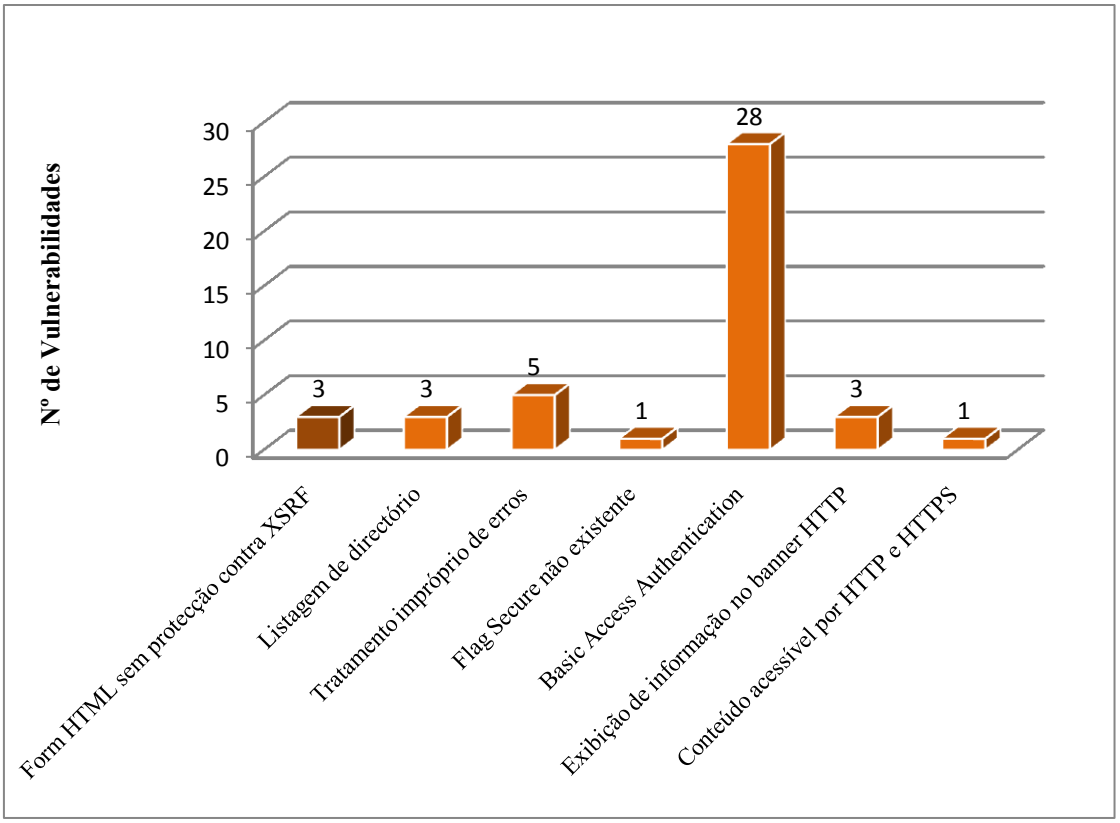


Figura 14 – Aplicação *Web* 2 da Administração Pública

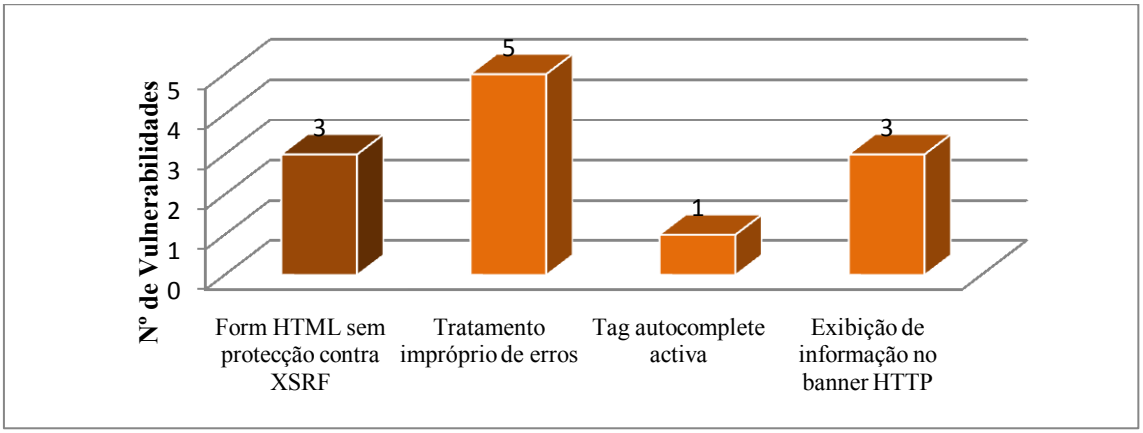


Figura 15 – Aplicação *Web* 3 da Administração Pública

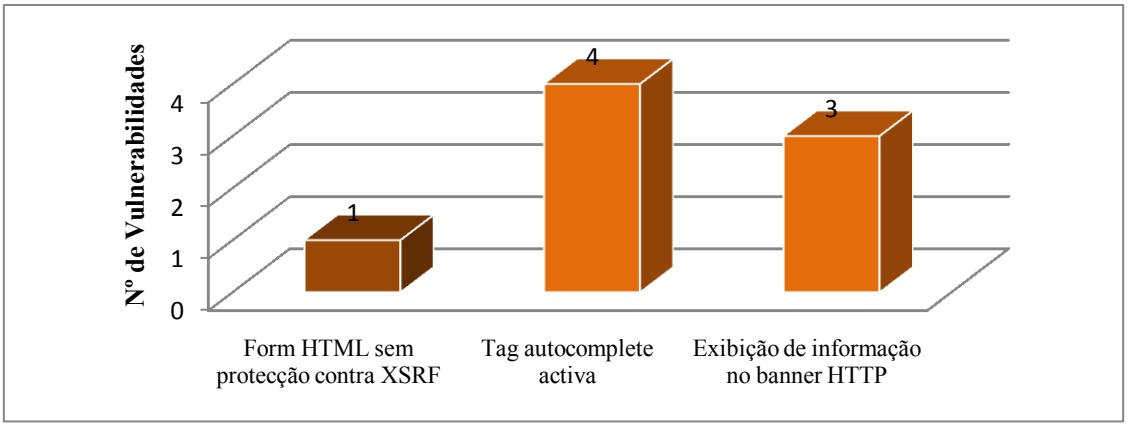


Figura 16 – Aplicação *Web* 4 da Administração Pública

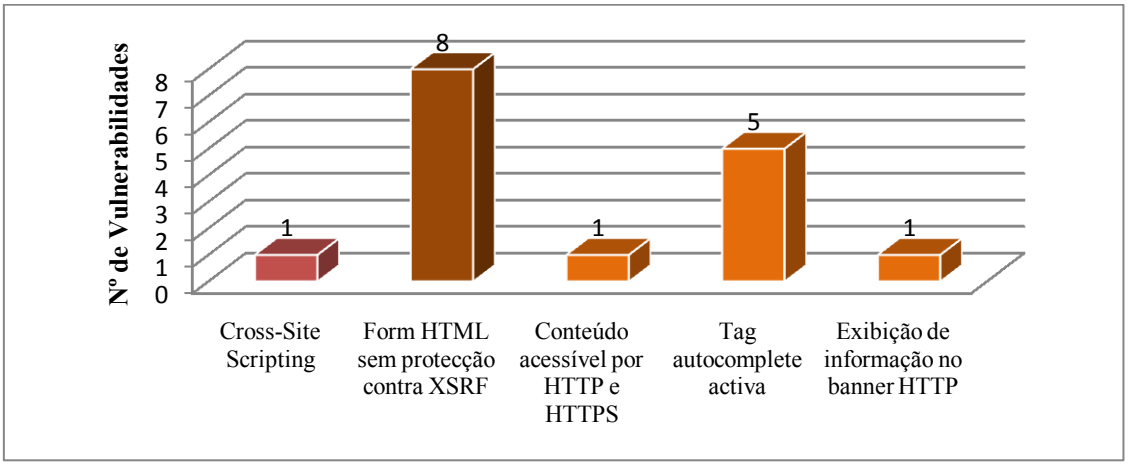


Figura 17 – Aplicação *Web* 5 da Administração Pública

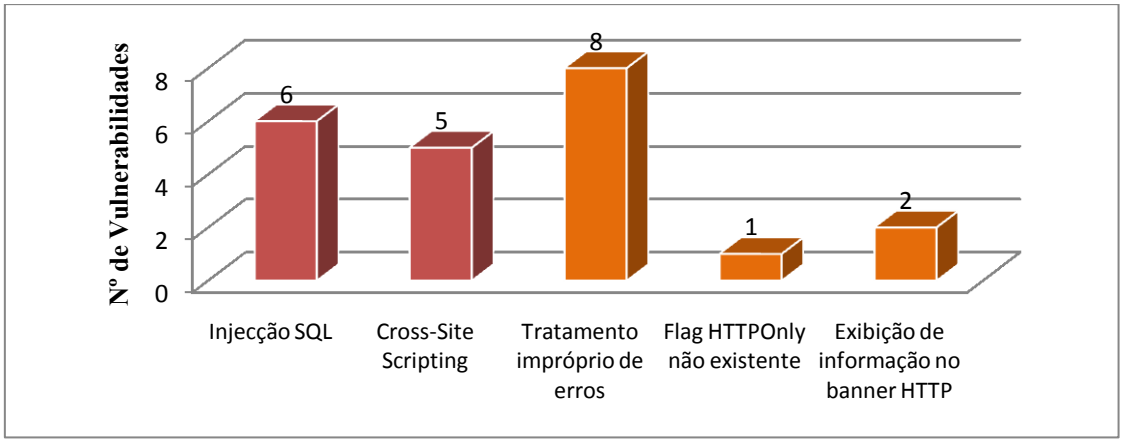


Figura 18 – Aplicação *Web* 6 da Administração Pública

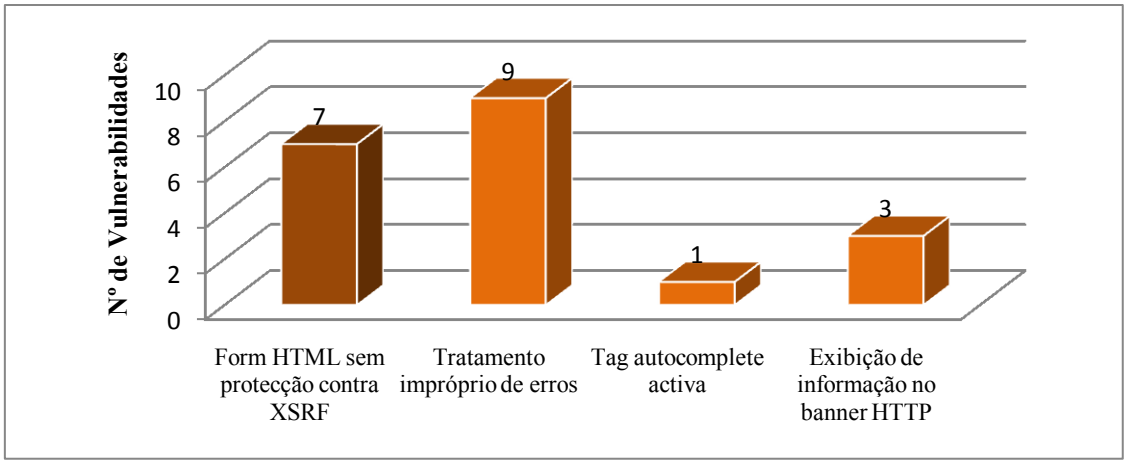


Figura 19 – Aplicação *Web* 7 da Administração Pública

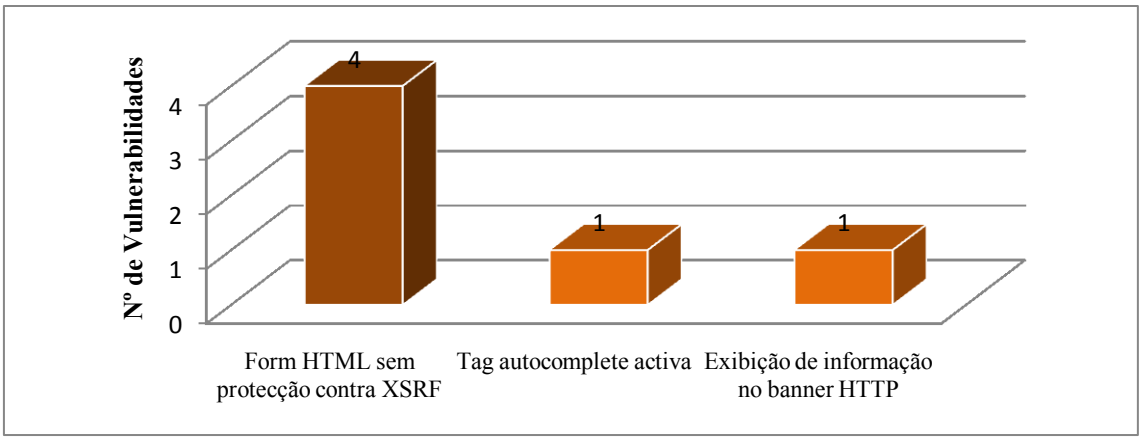


Figura 20 – Aplicação *Web* 8 da Administração Pública

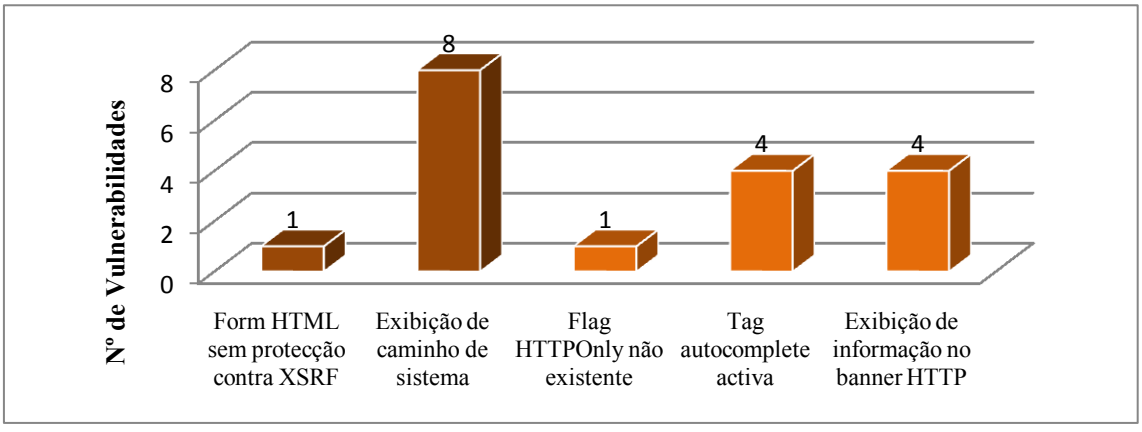


Figura 21 – Aplicação *Web* 9 da Administração Pública

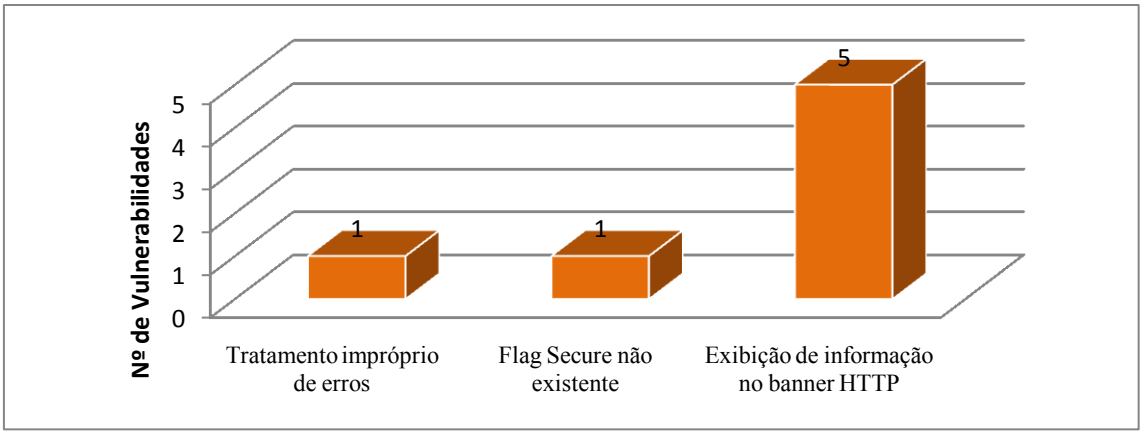


Figura 22 – Aplicação *Web* 10 da Administração Pública

Como se pode observar pelos resultados anteriormente descritos, foram confirmadas um total de 197 vulnerabilidades de entre as quais 21 estão classificadas como Grau 1, 40 em Grau 2 e 136 como Grau 3. Esta distribuição pode ser vista no gráfico abaixo representado.

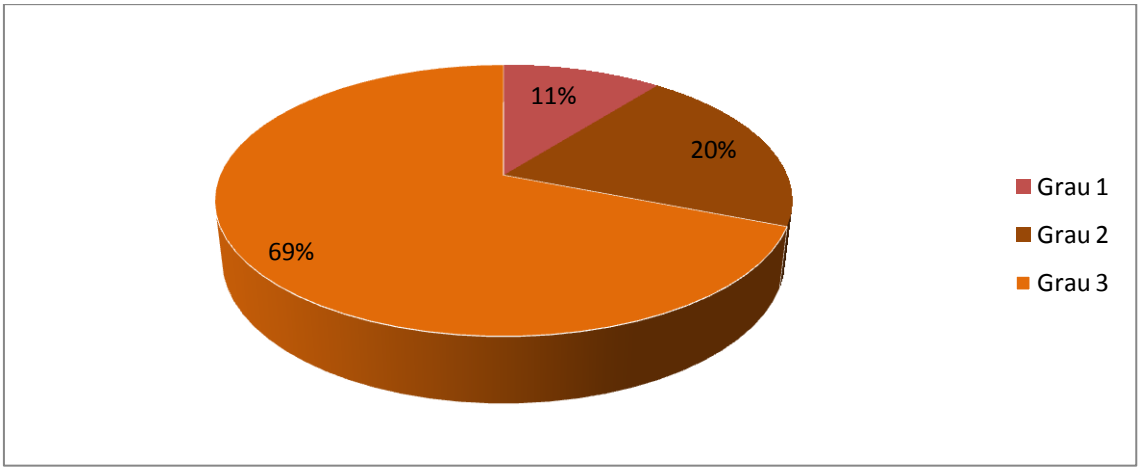


Figura 23 – Distribuição da totalidade das vulnerabilidades da Administração Pública

No gráfico abaixo pode ser verificado que a distribuição na classificação do OWASP Top 10 se centra sobretudo nas más configurações de segurança.

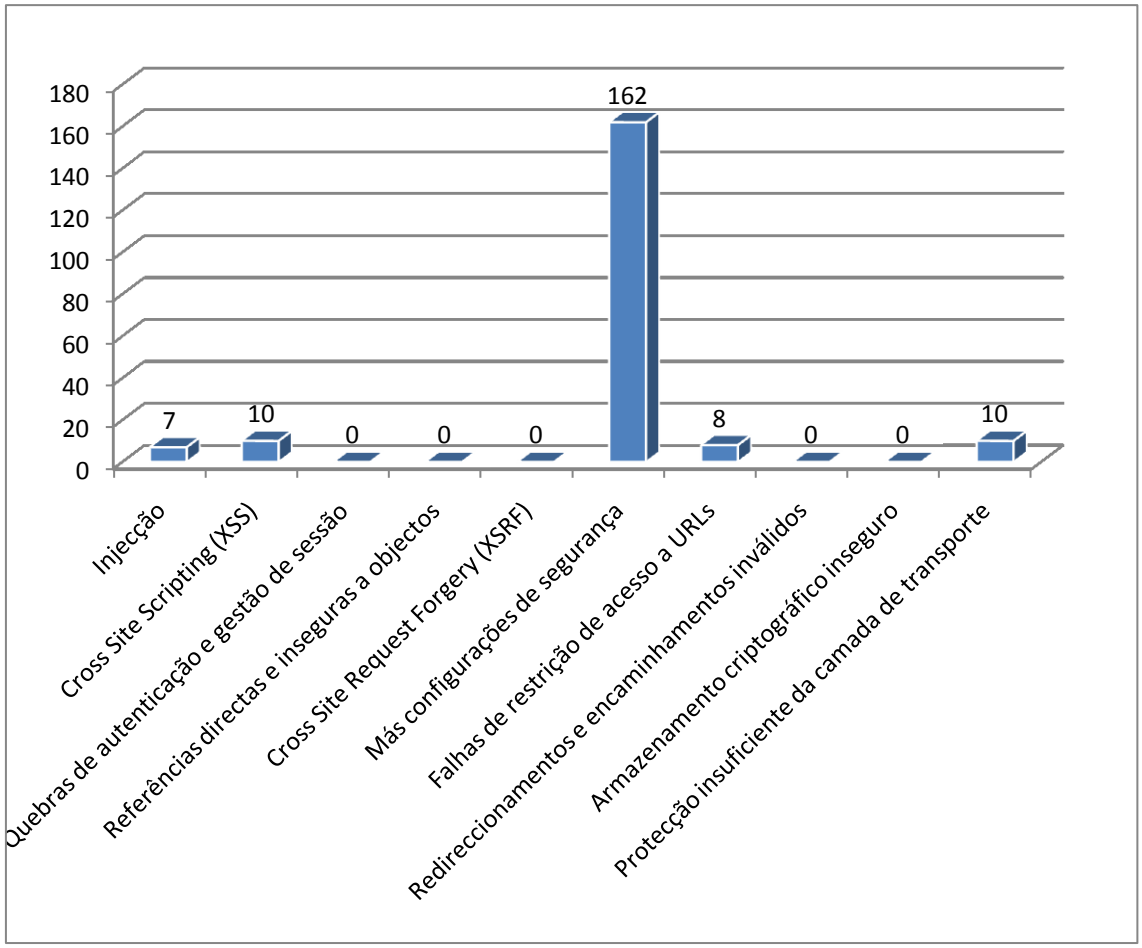


Figura 24 – Distribuição no OWASP Top 10 das vulnerabilidades da Administração Pública

4.2.1.1 Análise de Risco

Com os resultados obtidos na fase de auditoria de vulnerabilidades, foi possível obter uma base sólida para a elaboração de uma análise de risco baseada nas vulnerabilidades identificadas, a qual se encontra abaixo representada.

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade
9	9	7

O valor associado aos factores de ameaça é:

(9+9+7) / 3 = 8,333

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento
9	1	9

O valor associado aos factores da vulnerabilidade é:

(9+1+9) / 3 = 6.333

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Confidencialidade (%/100)*Peso**
Cross Site Tracing	1	0,508	0,005	7	0,036
Injecção SQL	7	3,553	0,036	9	0,320
Cross-Site Scripting	9	4,569	0,046	9	0,411
Vector de *overflow* de inteiros	4	2,030	0,020	5	0,102
Form HTML sem protecção contra XSRF	32	16,244	0,162	8	1,299
Listagem de directório	8	4,061	0,041	9	0,365
Tratamento impróprio de erros	41	20,812	0,208	8	1,665
Flag Secure não existente	3	1,523	0,015	8	0,122
Flag HTTPOnly não existente	3	1,523	0,015	8	0,122
Tag autocomplete activa	17	8,629	0,086	4	0,345
Exibição de informação no *banner* HTTP	26	13,198	0,132	8	1,056
Conteúdo acessível por HTTP e HTTPS	10	5,076	0,051	9	0,457
Basic Access Authentication	28	14,213	0,142	5	0,711
Exibição de caminho de sistema	8	4,061	0,041	9	0,365
Total	197	100	1		7

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Integridade (%/100)*Peso**
Cross Site Tracing	1	0,508	0,005	8	0,041
Injecção SQL	7	3,553	0,036	9	0,320
Cross-Site Scripting	9	4,569	0,046	8	0,365
Vector de *overflow* de inteiros	4	2,030	0,020	3	0,061
Form HTML sem protecção contra XSRF	32	16,244	0,162	1	0,162
Listagem de directório	8	4,061	0,041	1	0,041
Tratamento impróprio de erros	41	20,812	0,208	5	1,041
Flag Secure não existente	3	1,523	0,015	4	0,061
Flag HTTPOnly não existente	3	1,523	0,015	4	0,061
Tag autocomplete activa	17	8,629	0,086	1	0,086
Exibição de informação no *banner* HTTP	26	13,198	0,132	1	0,132

Conteúdo acessível por HTTP e HTTPS	10	5,076	0,051	5	0,254
Basic Access Authentication	28	14,213	0,142	7	0,995
Exibição de caminho de sistema	8	4,061	0,041	4	0,162
Total	197	100	1		4

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Disponibilidade (%/100)*Peso**
Cross Site Tracing	1	0,508	0,005	2	0,010
Injecção SQL	7	3,553	0,036	8	0,284
Cross-Site Scripting	9	4,569	0,046	9	0,411
Vector de *overflow* de inteiros	4	2,030	0,020	7	0,142
Form HTML sem protecção contra XSRF	32	16,244	0,162	2	0,325
Listagem de directório	8	4,061	0,041	1	0,041
Tratamento impróprio de erros	41	20,812	0,208	5	1,041
Flag Secure não existente	3	1,523	0,015	5	0,076
Flag HTTPOnly não existente	3	1,523	0,015	5	0,076
Tag autocomplete activa	17	8,629	0,086	2	0,172
Exibição de informação no *banner* HTTP	26	13,198	0,132	1	0,132
Conteúdo acessível por HTTP e HTTPS	10	5,076	0,051	1	0,051
Basic Access Authentication	28	14,213	0,142	8	1,137
Exibição de caminho de sistema	8	4,061	0,041	5	0,203
Total	197	100	1		4

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade
7	4	4

O valor associado ao impacto técnico é:

(7+4+4) / 3 = 5

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade
9	9	7	9

O valor associado ao impacto no negócio é:

(9+9+7+9) / 4 = 8.5

O cálculo do valor de probabilidade total é Alto, dado que:

(8.333 + 6.333) /2 = 7

O cálculo do valor de impacto total é Alto, dado que:

(5 + 9) /2 = 7

Utilizando a matriz de mapeamendo dos vectores de probabilidade e impacto obtém-se o grau geral de risco de Crítico, como se pode verificar na tabela abaixo.

Grau de Risco				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 12 – Grau de risco associado à Administração Pública

4.2.2 Entidades Financeiras

Foram contactadas um total de 3 entidades de destaque do sector bancário em Portugal, as quais todas responderam afirmativamente à proposta de trabalho apresentada. Foram realizadas reuniões com essas entidades, e foram entregues os respectivos documentos de planeamento de testes de auditoria para sua análise.

Embora o interesse demonstrado tenha sido notório, com cerca de 1 ano de espera pela resposta destas entidades, foi decidido que não seria exequível continuar com a mesma e aguardar o desenrolar do restante processo, motivo pelo qual não foi possível enquadrá-las nos resultados obtidos.

Foi de facto um revês sério nas aspirações deste estudo, pois teria sido muito interessante analisar a maturidade da segurança das aplicações *Web* que utilizamos diariamente, e que têm a seu cargo o nosso património financeiro.

4.2.3 Forças Armadas

No grupo das Forças Armadas foram avaliadas duas entidades, às quais correspondem um total de 16 aplicações *Web* auditadas, cujos resultados se podem observar nos gráficos abaixo representados.

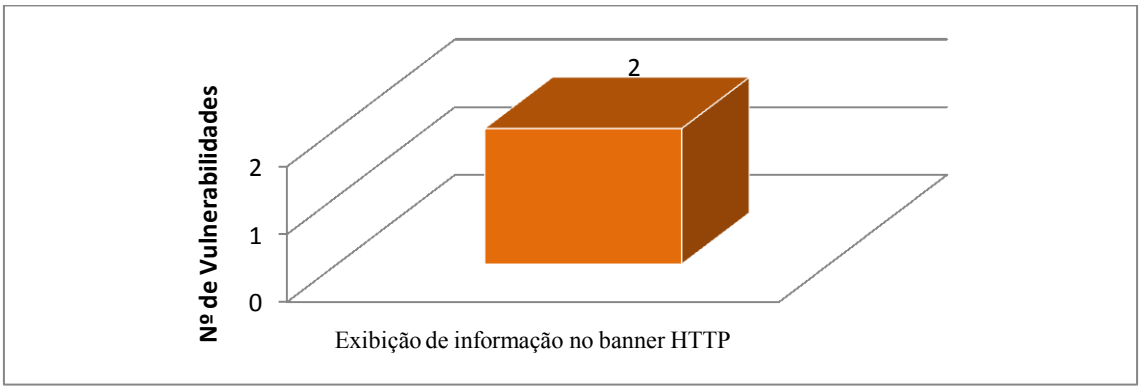


Figura 25 – Aplicação *Web* 1 das Forças Armadas

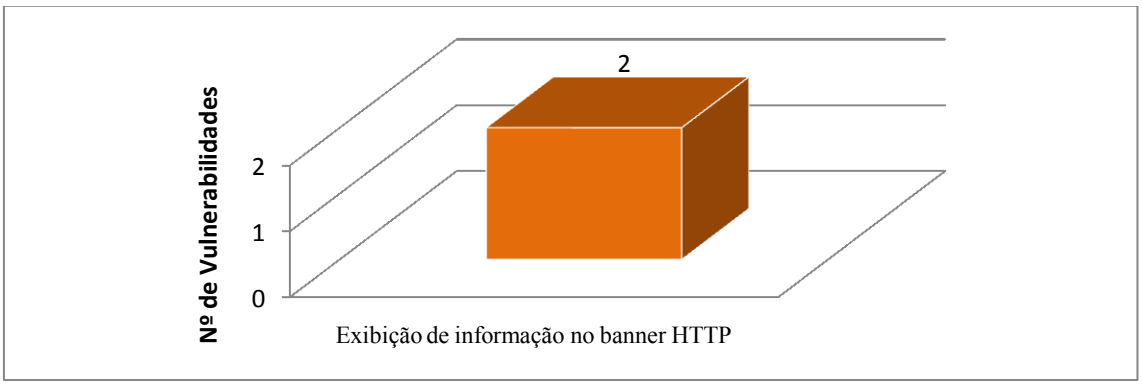


Figura 26 – Aplicação *Web* 2 das Forças Armadas

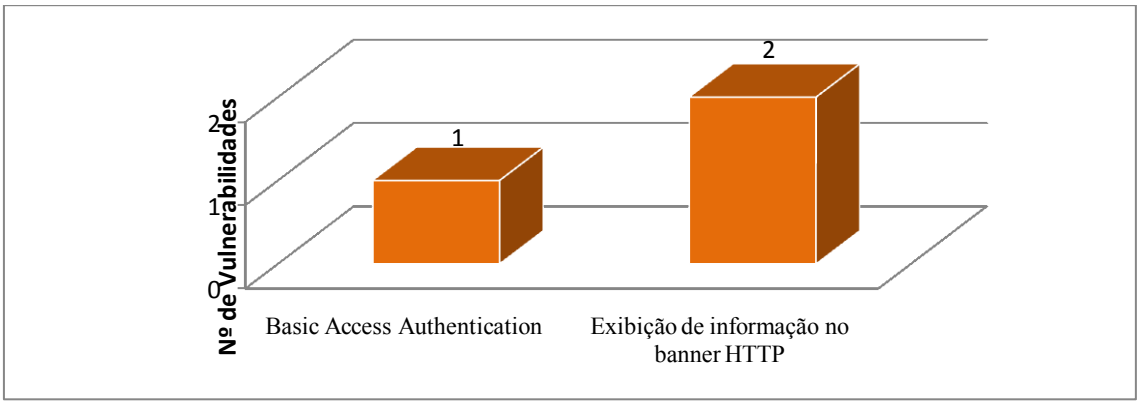


Figura 27 – Aplicação *Web* 3 das Forças Armadas

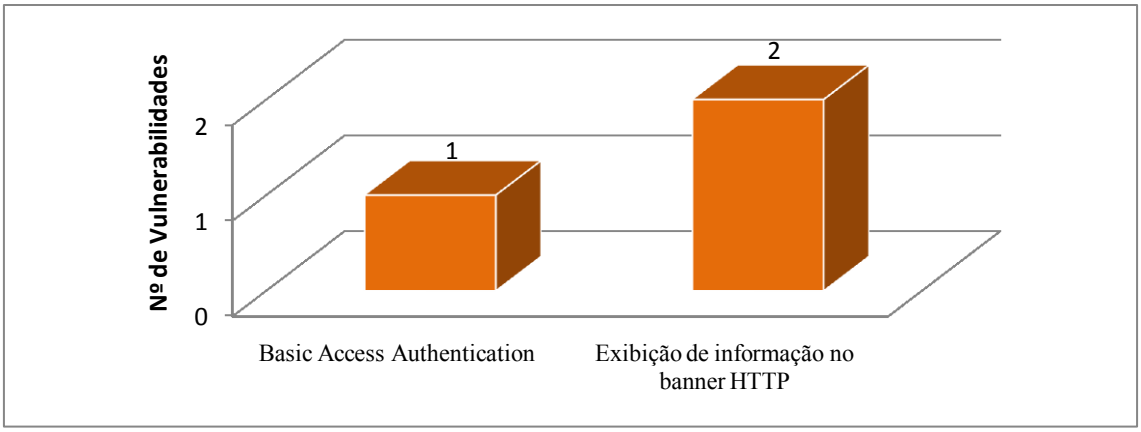


Figura 28 – Aplicação *Web* 4 das Forças Armadas

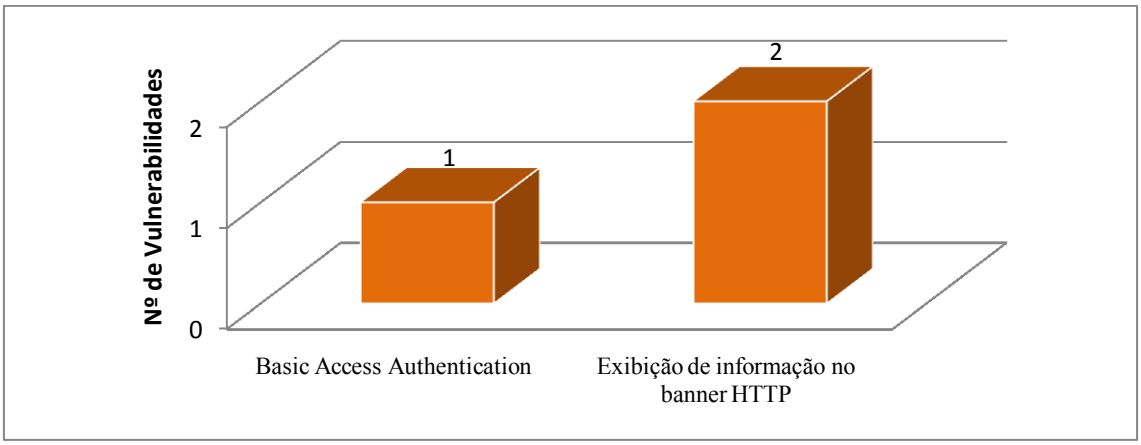


Figura 29 – Aplicação *Web* 5 das Forças Armadas

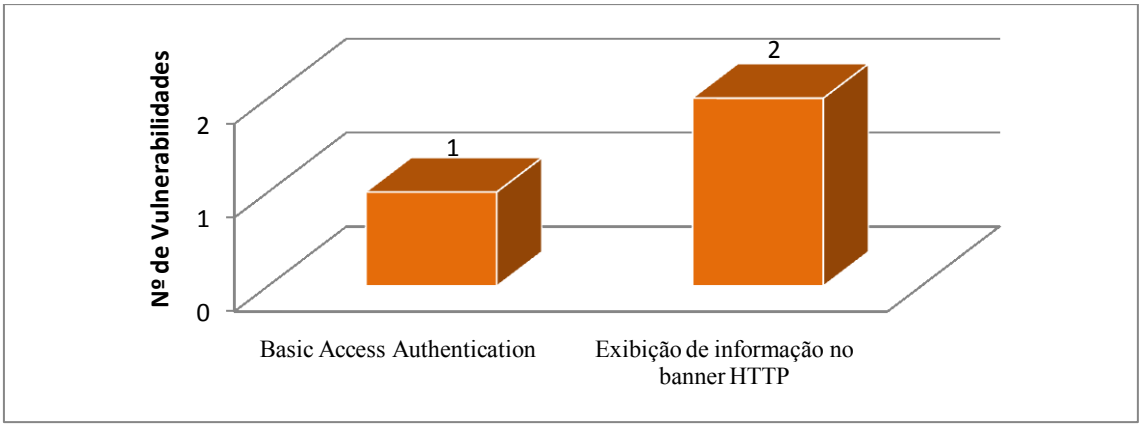


Figura 30 – Aplicação *Web* 6 das Forças Armadas

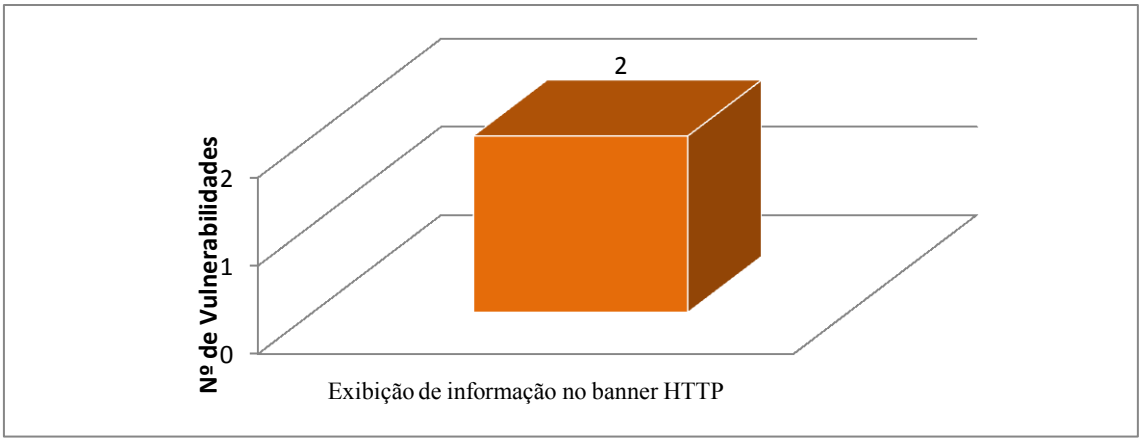


Figura 31 – Aplicação *Web* 7 das Forças Armadas

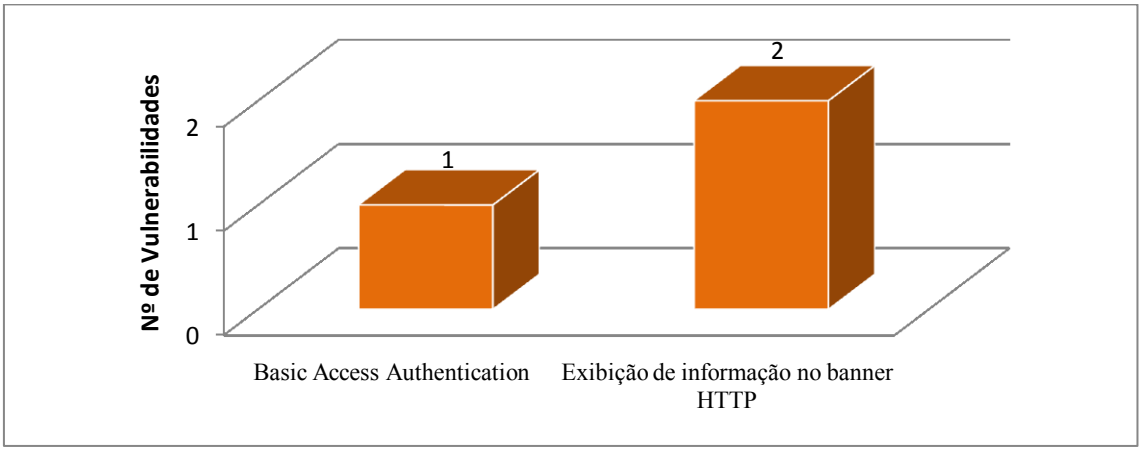


Figura 32 – Aplicação *Web* 8 das Forças Armadas

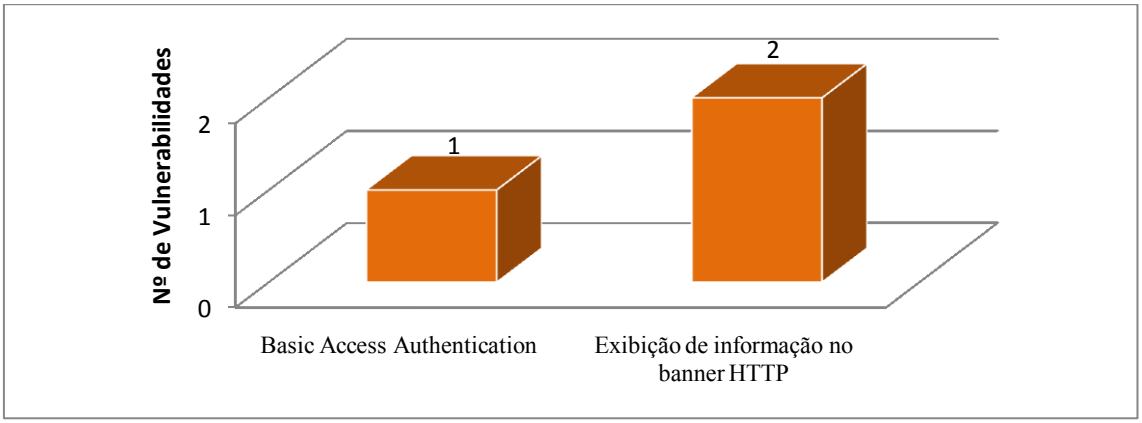


Figura 33 – Aplicação *Web* 9 das Forças Armadas

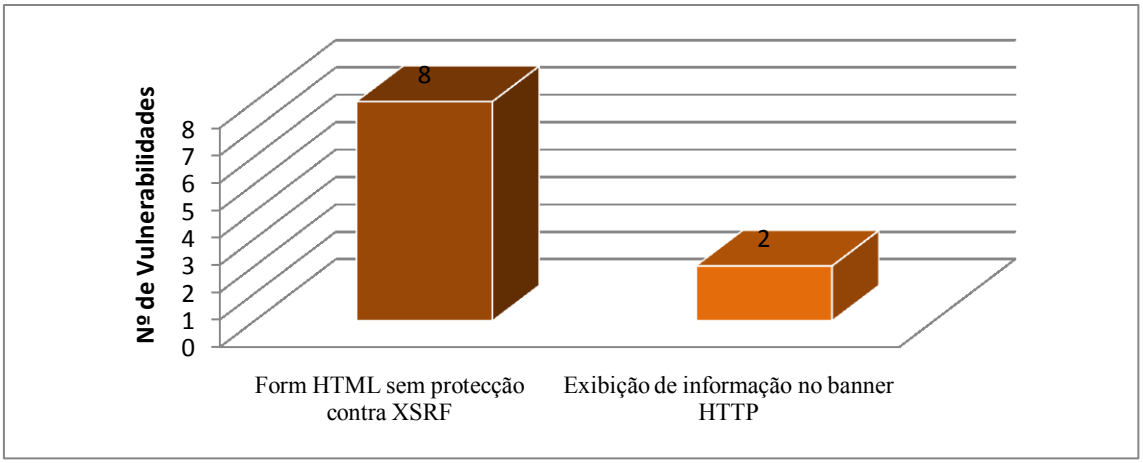


Figura 34 – Aplicação *Web* 10 das Forças Armadas

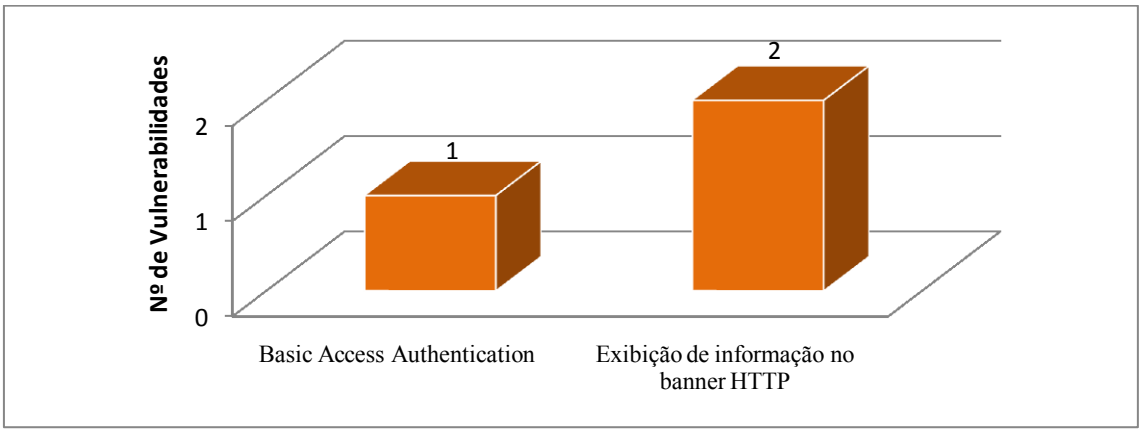


Figura 35 – Aplicação *Web* 11 das Forças Armadas

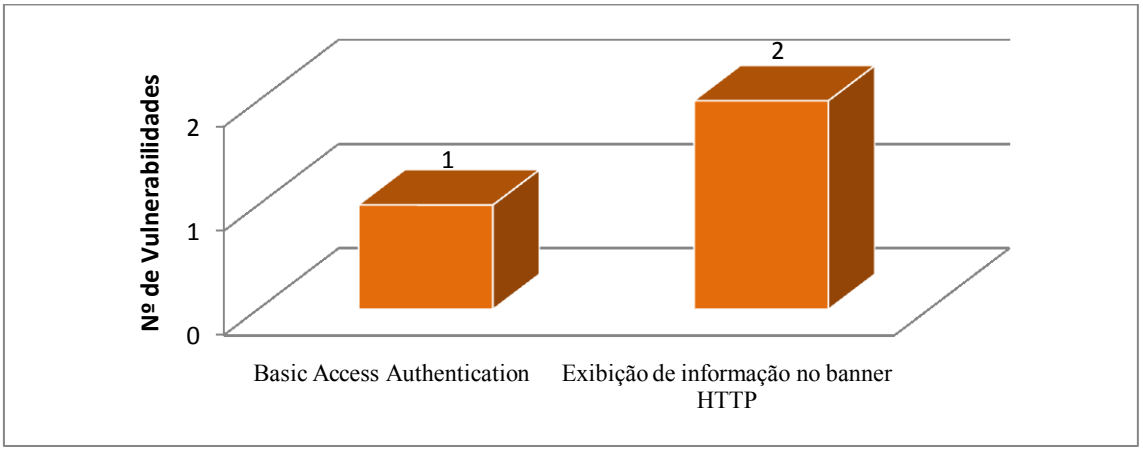


Figura 36 – Aplicação *Web* 12 das Forças Armadas

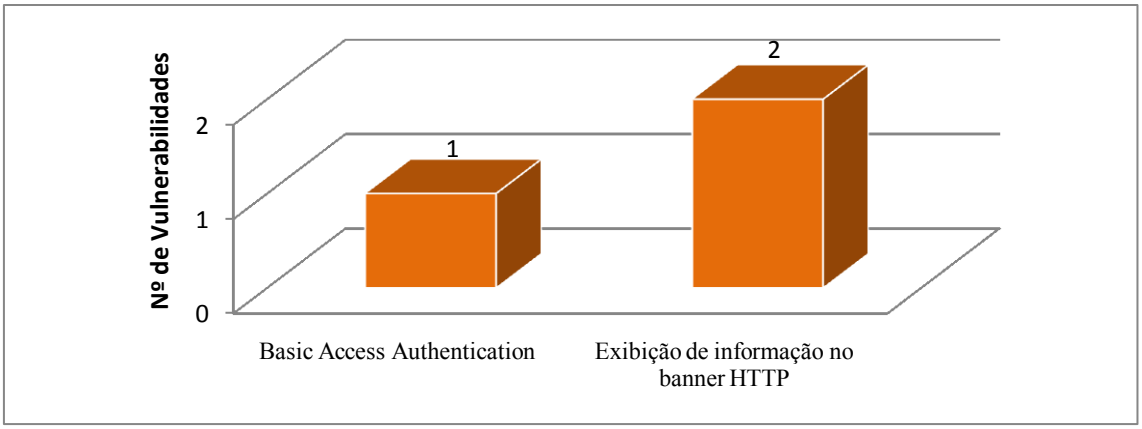


Figura 37 – Aplicação *Web* 13 das Forças Armadas

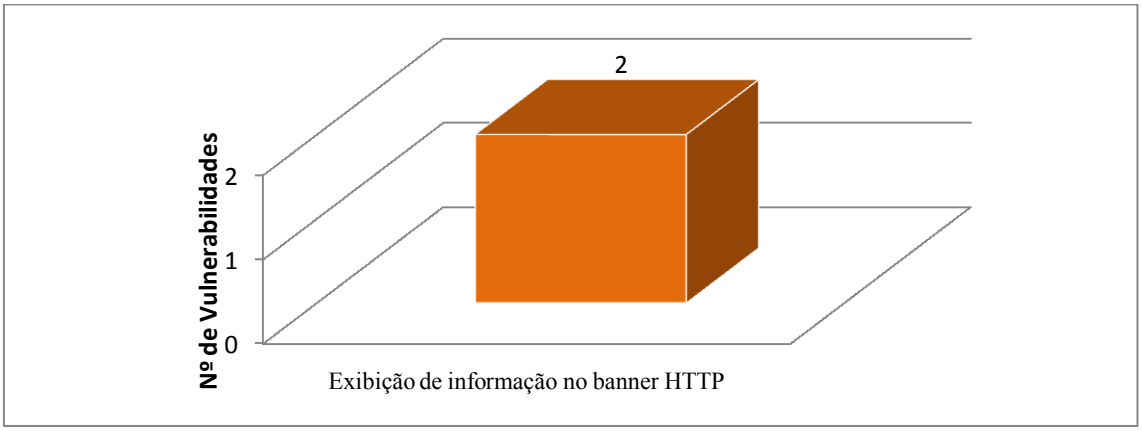


Figura 38 – Aplicação *Web* 14 das Forças Armadas

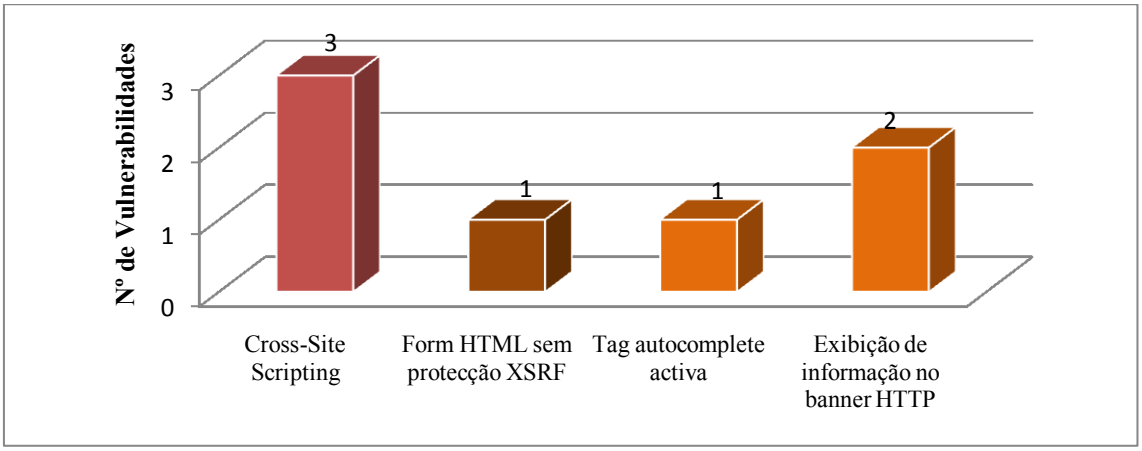


Figura 39 – Aplicação *Web* 15 das Forças Armadas

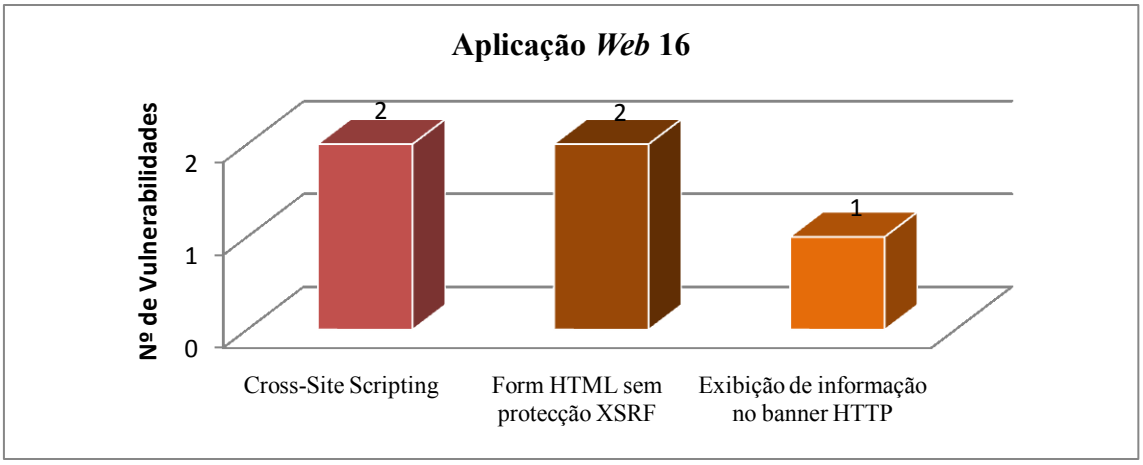


Figura 40 – Aplicação *Web* 16 das Forças Armadas

Como se pode observar pelos resultados anteriormente descritos, foram confirmadas um total de 57 vulnerabilidades, estando 5 classificadas como Grau 1, 11 classificadas como Grau 2 e 41 como Grau 3, distribuição esta que pode ser vista no gráfico abaixo representado.

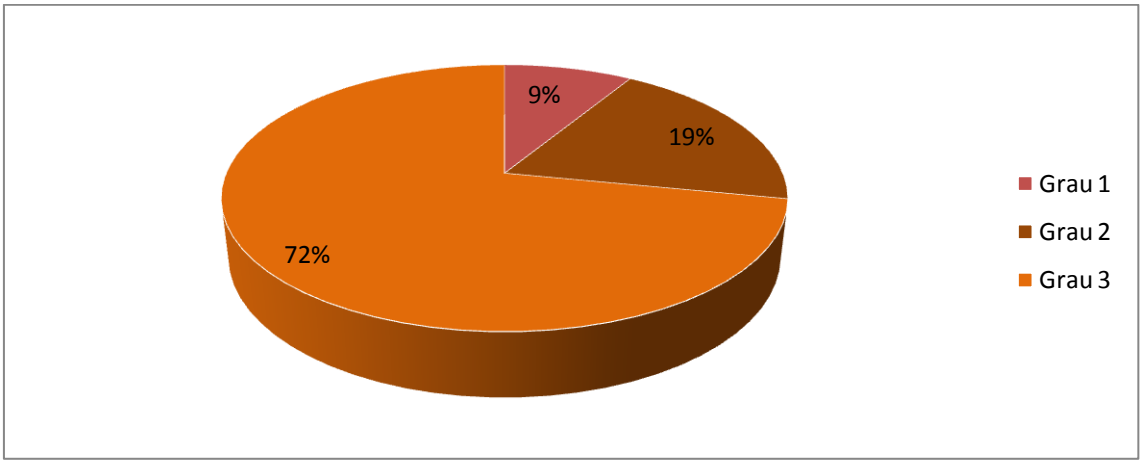


Figura 41 – Distribuição da totalidade das vulnerabilidades das Forças Armadas

Adicionalmente, no gráfico abaixo pode ser verificado que a distribuição na classificação do OWASP Top 10 se encontra totalmente no grupo das más configurações de segurança.

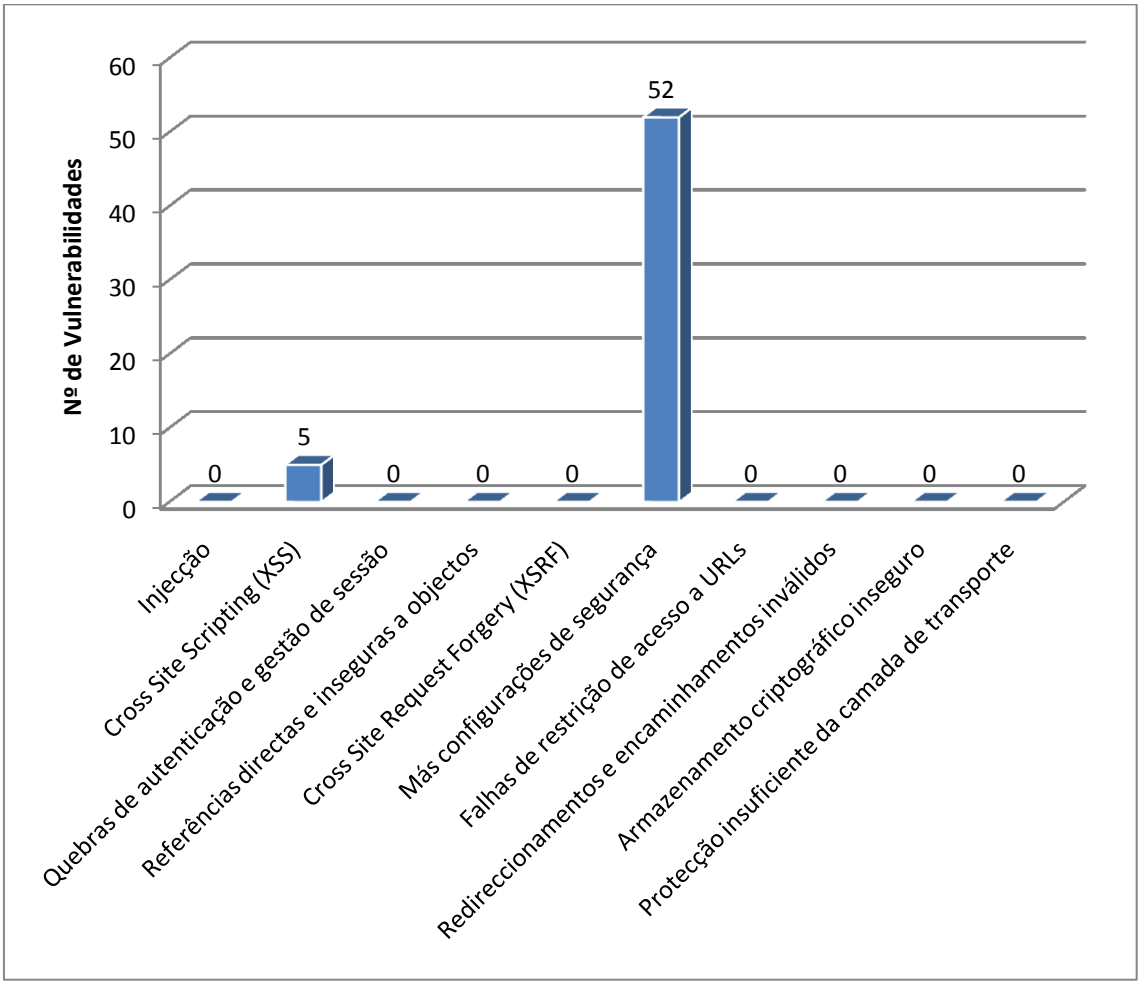


Figura 42 – Distribuição no OWASP Top 10 das vulnerabilidades das Forças Armadas

4.2.3.1 Análise de Risco

Com os resultados obtidos na fase de auditoria de vulnerabilidades, foi possível obter uma base sólida para a elaboração de uma análise de risco baseada nas vulnerabilidades identificadas, a qual se encontra abaixo representada.

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade
9	8	7

O valor associado aos factores de ameaça é:

(9+8+7) / 3 = 8

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento
9	1	9

O valor associado aos factores da vulnerabilidade é:

(9+1+9) / 3 = 6.333

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Confidencialidade (%/100)*Peso**
Cross-Site Scripting	5	8,772	0,088	9	0,792
Form HTML sem protecção contra XSRF	11	19,298	0,193	8	1,544
Exibição de informação no *banner* HTTP	31	54,386	0,544	8	4,352
Basic Access Authentication	9	15,789	0,158	5	0,79
Tag autocomplete activa	1	1,754	0,018	4	0,072
Total	57	100	1		8

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Integridade (%/100)*Peso**
Cross-Site Scripting	5	8,772	0,088	8	0,704
Form HTML sem protecção contra XSRF	11	19,298	0,193	1	0,193
Exibição de informação no *banner* HTTP	31	54,386	0,544	1	0,544
Basic Access Authentication	9	15,789	0,158	7	1,106
Tag autocomplete activa	1	1,754	0,018	1	0,018
Total	57	100	1		3

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Disponibilidade (%/100)*Peso**
Cross-Site Scripting	5	8,772	0,088	9	0,792
Form HTML sem protecção contra XSRF	11	19,298	0,193	2	0,386
Exibição de informação no *banner* HTTP	31	54,386	0,544	1	0,544
Basic Access Authentication	9	15,789	0,158	8	1,264
Tag autocomplete activa	1	1,754	0,018	2	0,036
Total	57	100	1		3

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade
8	3	3

O valor associado ao impacto técnico é:

(8+3+3) / 3 = 4,667

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade
1	9	5	7

O valor associado ao impacto no negócio é:

(1+9+5+7) / 4 = 5.5

O cálculo do valor de probabilidade total é Alto, dado que:

(8 + 6.333) /2 = 7

O cálculo do valor de impacto total é Alto, dado que:

(5 + 5.5) /2 = 5

Utilizando a matriz de mapeamendo dos vectores de probabilidade e impacto obtém-se o grau geral de risco de Crítico, como se pode verificar na tabela abaixo.

Grau de Risco				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 13 – Grau de risco associado às Forças Armadas

4.2.4 Educação

No grupo da Educação foram avaliadas um total de duas entidades, às quais correspondem um total de 14 aplicações *Web,* como se pode observar nos gráficos abaixo representados.

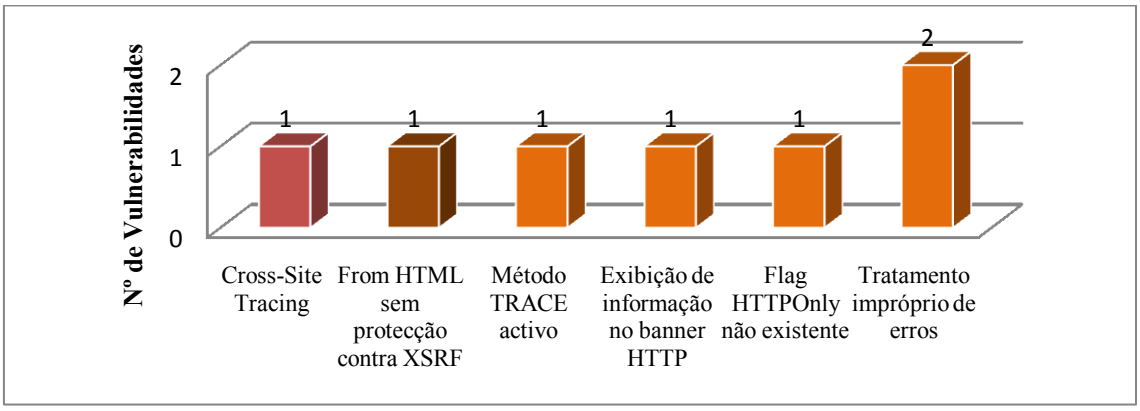


Figura 43 – Aplicação *Web* 1 da Educação

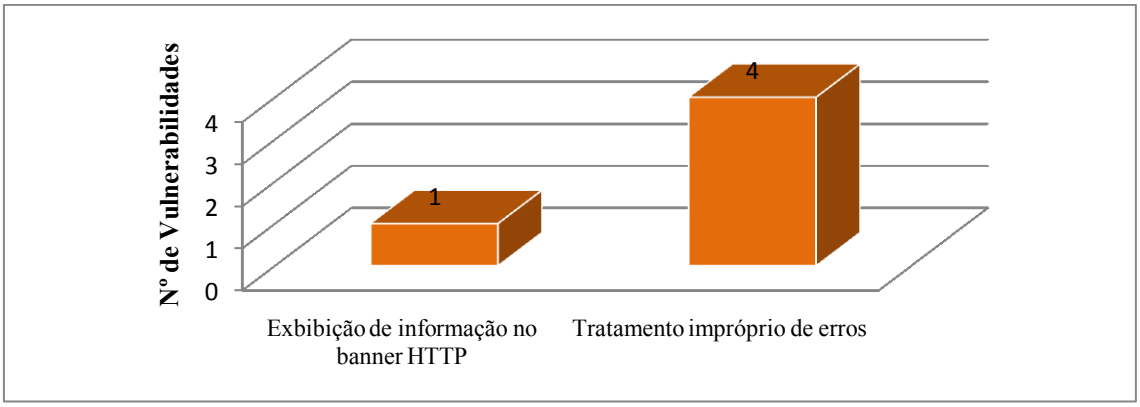


Figura 44 – Aplicação *Web* 2 da Educação

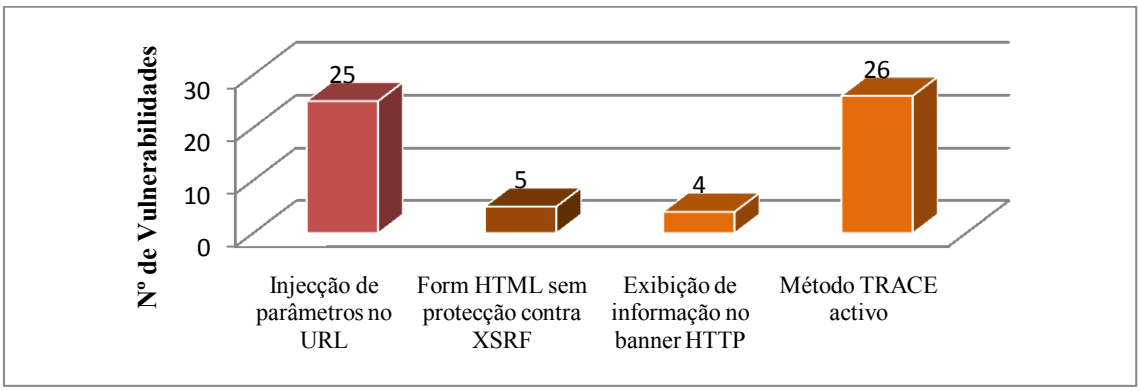


Figura 45 – Aplicação *Web* 3 da Educação

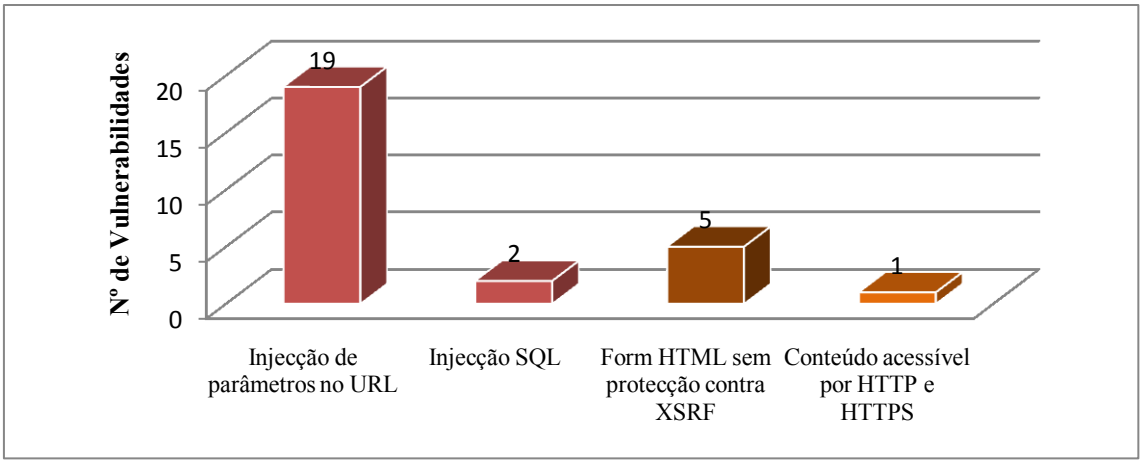


Figura 46 – Aplicação *Web* 4 da Educação

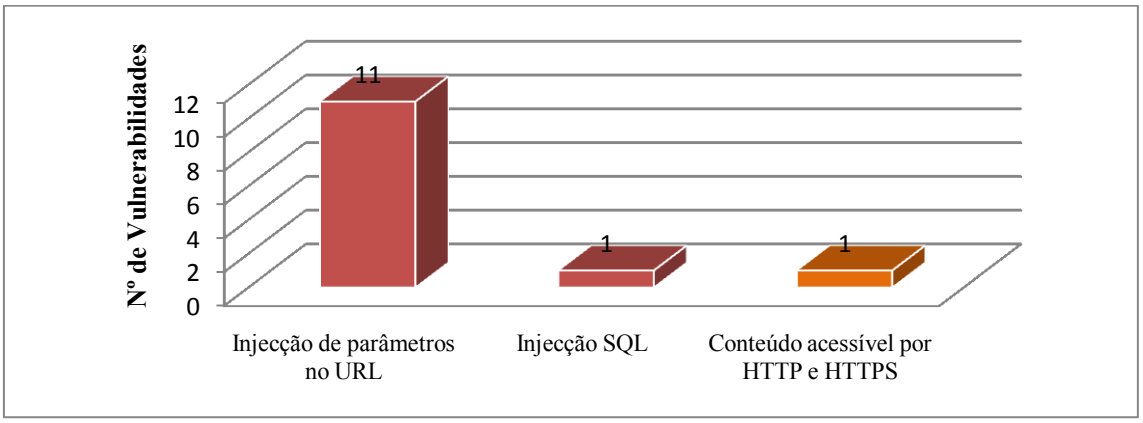


Figura 47 – Aplicação *Web* 5 da Educação

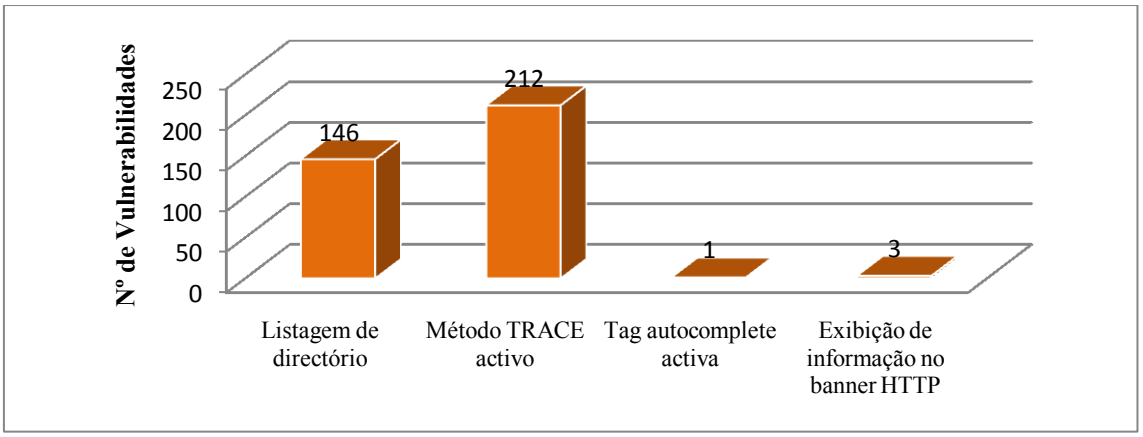


Figura 48 – Aplicação *Web* 6 da Educação

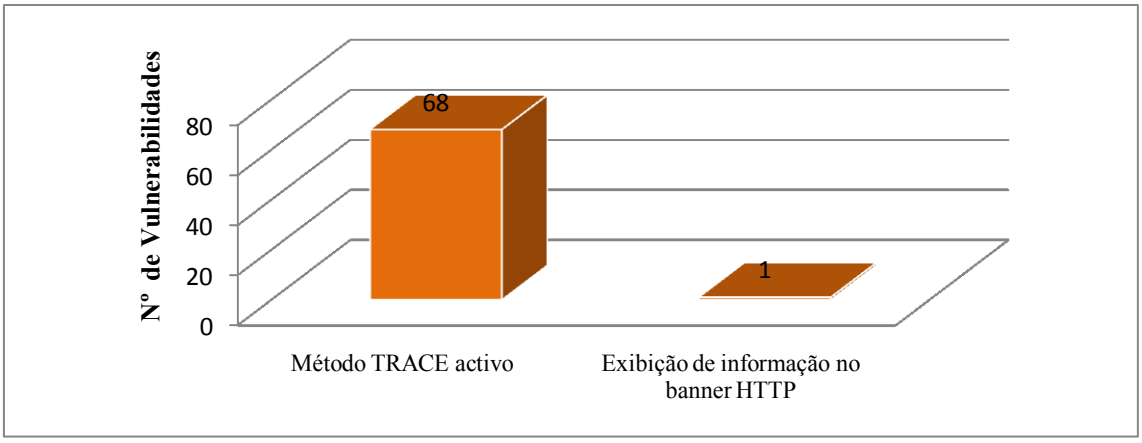


Figura 49 – Aplicação *Web* 7 da Educação

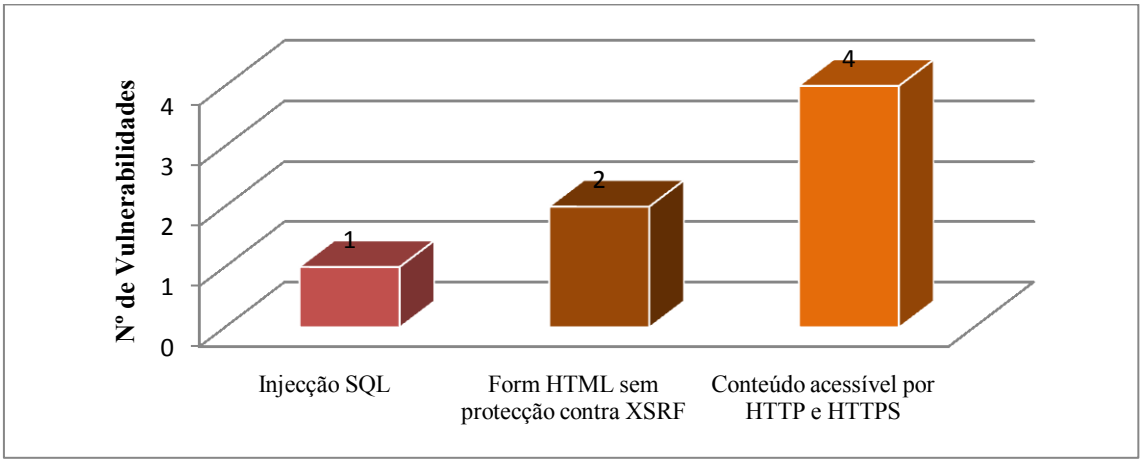


Figura 50 – Aplicação *Web* 8 da Educação

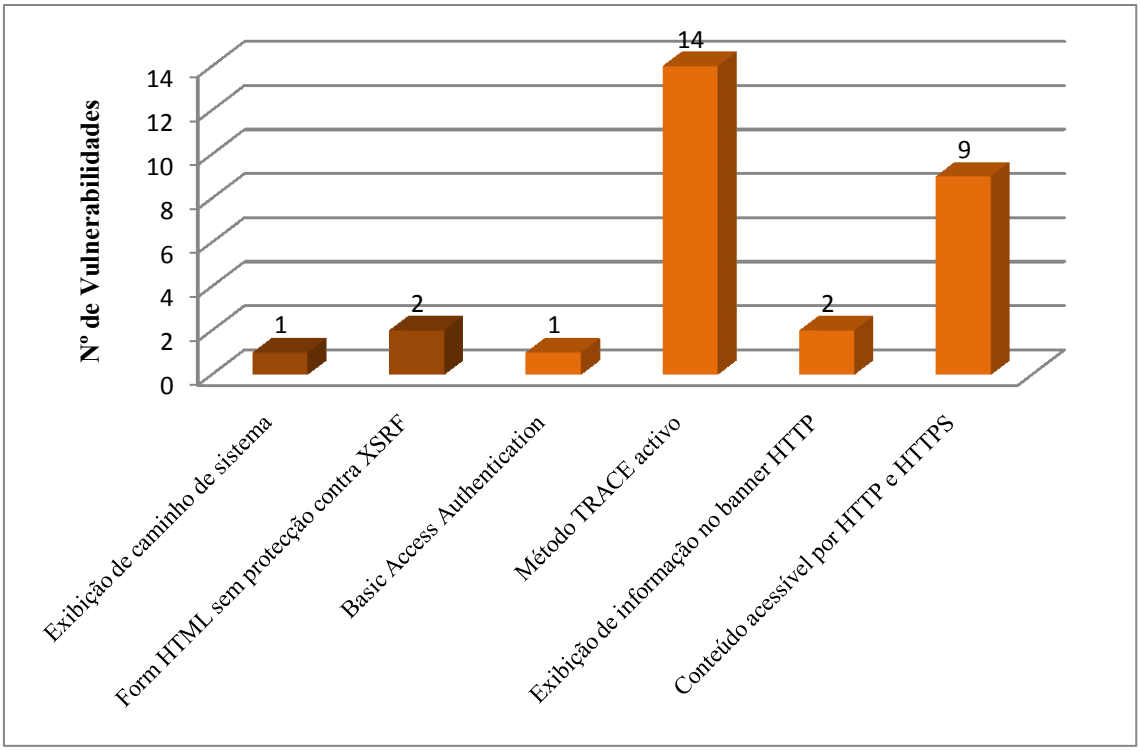


Figura 51 – Aplicação *Web* 9 da Educação

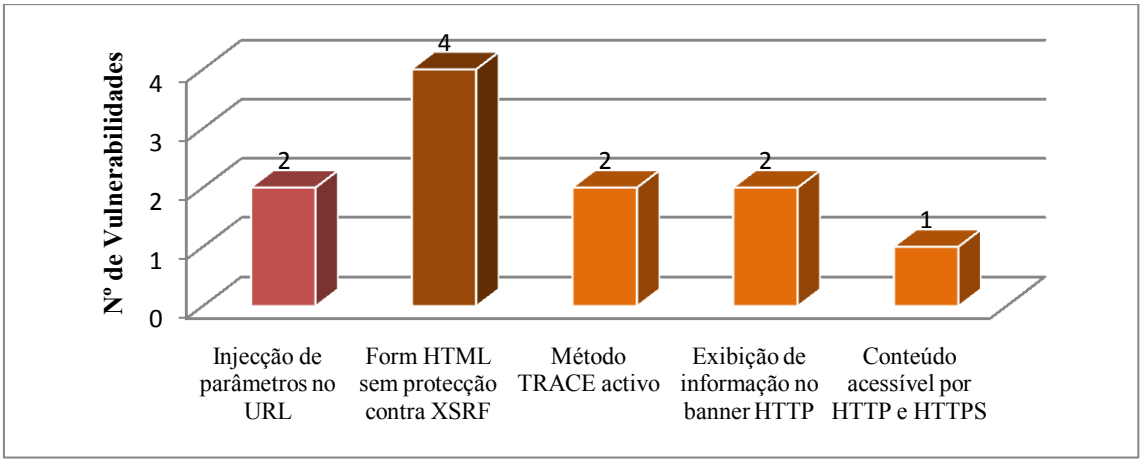


Figura 52 – Aplicação *Web* 10 da Educação

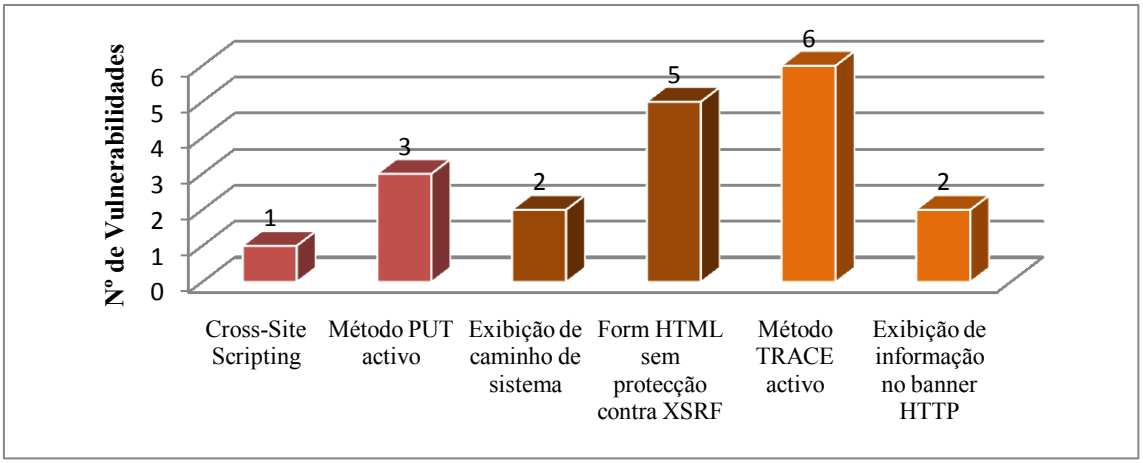


Figura 53 – Aplicação *Web* 11 da Educação

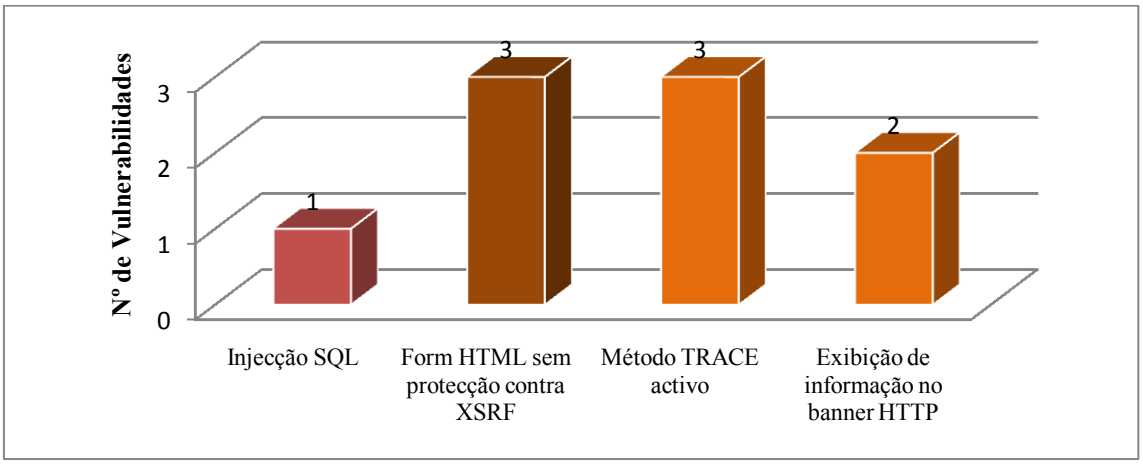


Figura 54 – Aplicação *Web* 12 da Educação

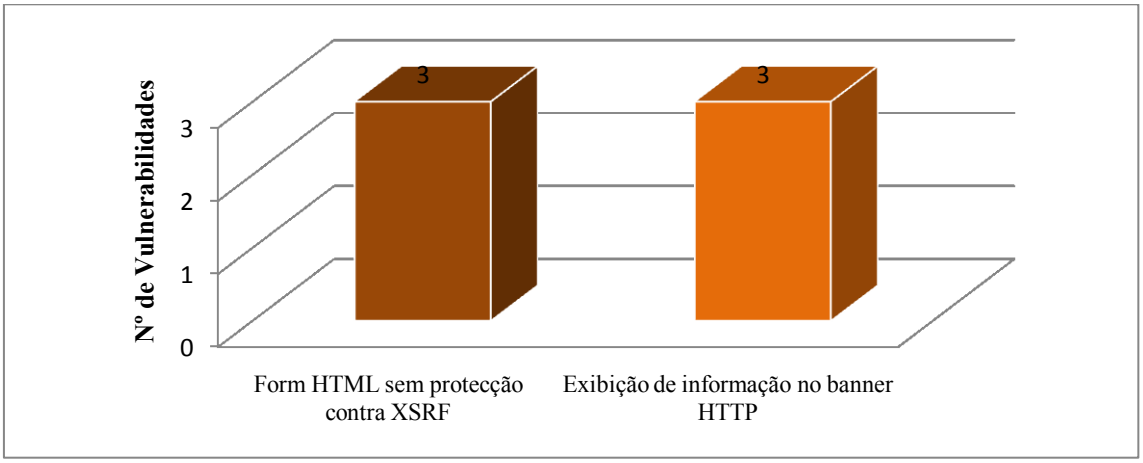


Figura 55 – Aplicação *Web* 13 da Educação

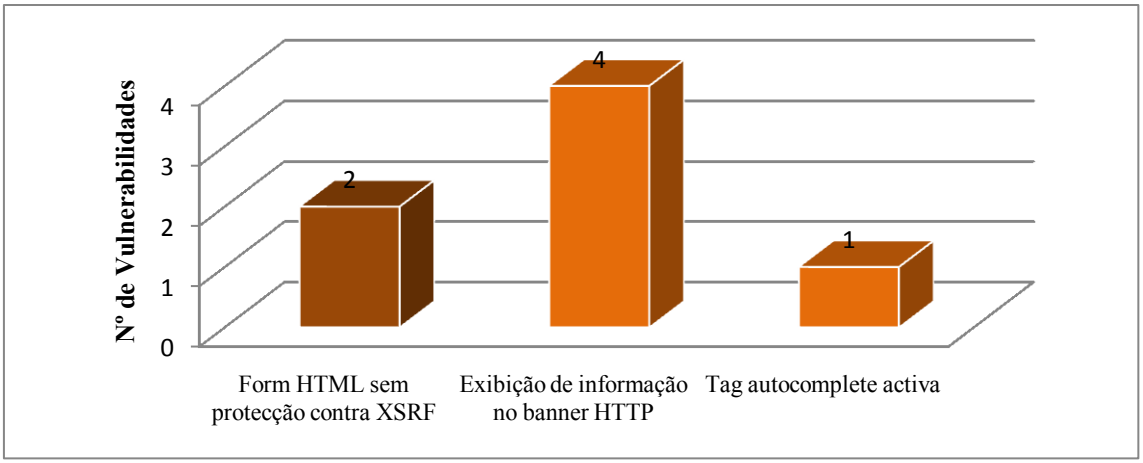


Figura 56 – Aplicação *Web* 14 da Educação

Como se pode observar pelos resultados anteriormente descritos, foram confirmadas um total de 631 vulnerabilidades de entre as quais 67 estão classificadas como Grau 1, 33em Grau 2 e 531 como Grau 3, distribuição esta que pode ser vista no gráfico abaixo representado.

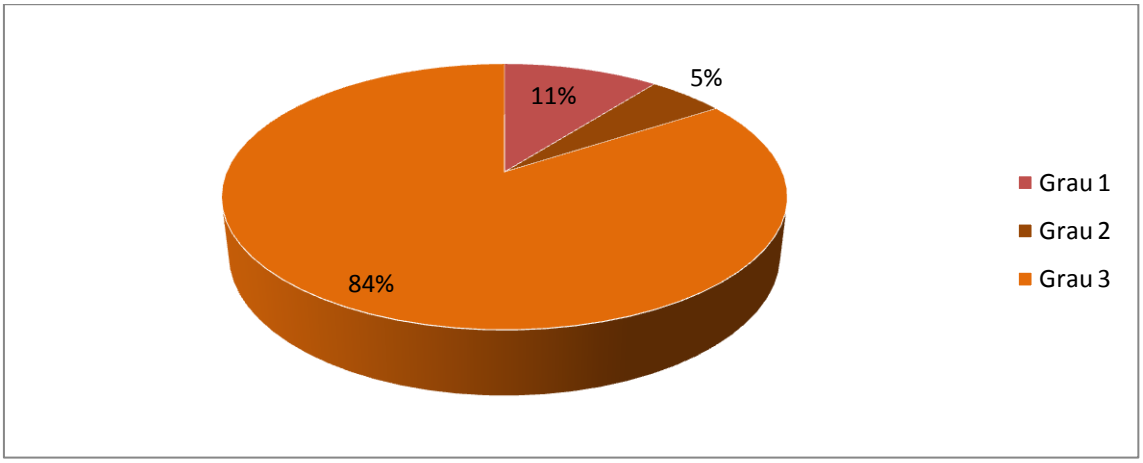


Figura 57 – Distribuição da totalidade das vulnerabilidades da Educação

Adicionalmente, no gráfico abaixo pode ser verificado que a distribuição na classificação do OWASP Top 10 se centra sobretudo nas más configurações de segurança apesar da existência de várias vulnerabilidades críticas de injecção e XSS.

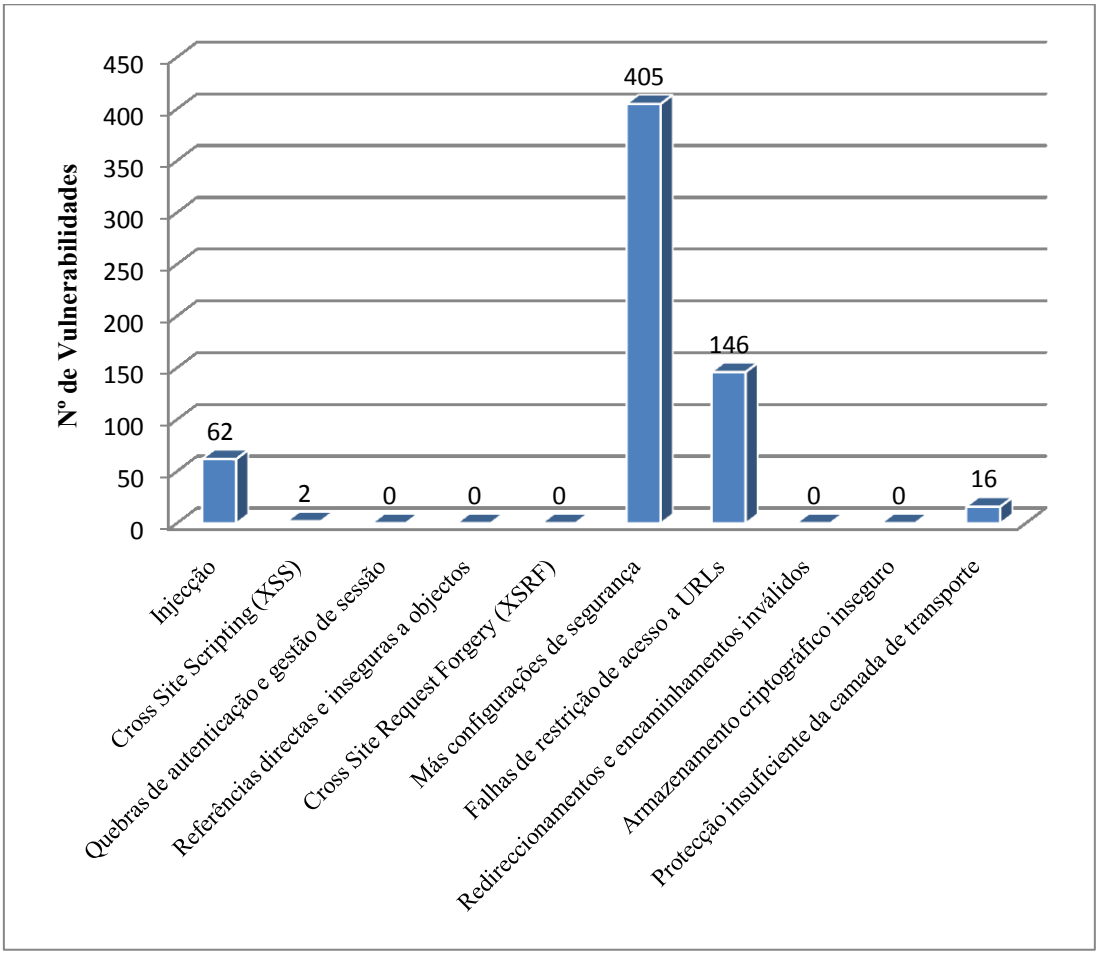


Figura 58 – Distribuição no OWASP Top 10 das vulnerabilidades da Educação

4.2.4.1 Análise de Risco

Com os resultados obtidos na fase de auditoria de vulnerabilidades, foi possível obter uma base sólida para a elaboração de uma análise de risco baseada nas vulnerabilidades identificadas.

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade
9	1	7

O valor associado aos factores de ameaça é:

(9+1+7) / 3 = 5,666

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento
9	1	9

O valor associado aos factores da vulnerabilidade é:

(9+1+9) / 3 = 6.333

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Confidencialidade (%/100)*Peso**
Cross Site Tracing	1	0,158	0,002	7	0,011
Injecção SQL	5	0,792	0,008	9	0,071
Cross-Site Scripting	1	0,158	0,002	9	0,014
Injecção de parâmetros no URL	57	9,033	0,090	1	0,090
Form HTML sem protecção contra XSRF	32	5,071	0,051	8	0,406
Listagem de directório	146	23,138	0,231	9	2,082
Tratamento impróprio de erros	6	0,951	0,010	8	0,076
Basic Access Authentication	1	0,158	0,002	5	0,008
Flag HTTPOnly não existente	1	0,158	0,002	8	0,013
Tag autocomplete activa	2	0,317	0,003	4	0,013
Exibição de informação no *banner* HTTP	25	3,962	0,040	8	0,317
Conteúdo acessível por HTTP e HTTPS	16	2,536	0,025	9	0,228
Método TRACE activo	332	52,615	0,526	5	2,631
Exibição de caminho de sistema	3	0,475	0,005	9	0,043
Método PUT activo	3	0,475	0,005	1	0,005
Total (arredondado às unidades)	631	100	1		6

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Integridade (%/100)*Peso**
Cross Site Tracing	1	0,158	0,002	8	0,013
Injecção SQL	5	0,792	0,008	9	0,071
Cross-Site Scripting	1	0,158	0,002	8	0,013
Injecção de parâmetros no URL	57	9,033	0,090	8	0,723
Form HTML sem protecção contra XSRF	32	5,071	0,051	1	0,051
Listagem de directório	146	23,138	0,231	1	0,231
Tratamento impróprio de erros	6	0,951	0,010	5	0,048
Basic Access Authentication	1	0,158	0,002	7	0,011

Flag HTTPOnly não existente	1	0,158	0,002	4	0,006
Tag autocomplete activa	2	0,317	0,003	1	0,003
Exibição de informação no *banner* HTTP	25	3,962	0,040	1	0,040
Conteúdo acessível por HTTP e HTTPS	16	2,536	0,025	5	0,127
Método TRACE activo	332	52,615	0,526	5	3,157
Exibição de caminho de sistema	3	0,475	0,005	4	0,019
Método PUT activo	3	0,475	0,005	5	0,024
Total (arredondado às unidades)	631	100	1		5

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Disponibilidade (%/100)*Peso**
Cross Site Tracing	1	0,158	0,002	2	0,003
Injecção SQL	5	0,792	0,008	8	0,063
Cross-Site Scripting	1	0,158	0,002	9	0,014
Injecção de parâmetros no URL	57	9,033	0,090	1	0,090
Form HTML sem protecção contra XSRF	32	5,071	0,051	2	0,101
Listagem de directório	146	23,138	0,231	1	0,231
Tratamento impróprio de erros	6	0,951	0,010	5	0,048
Basic Access Authentication	1	0,158	0,002	8	0,013
Flag HTTPOnly não existente	1	0,158	0,002	5	0,008
Tag autocomplete activa	2	0,317	0,003	2	0,006
Exibição de informação no *banner* HTTP	25	3,962	0,040	1	0,040
Conteúdo acessível por HTTP e HTTPS	16	2,536	0,025	1	0,025
Método TRACE activo	332	52,615	0,526	1	0,526
Exibição de caminho de sistema	3	0,475	0,005	5	0,024
Método PUT activo	3	0,475	0,005	6	0,029
Total (arredondado às unidades)	631	100	1		1

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade
6	5	1

O valor associado ao impacto técnico é:

(6+5+1) / 3 = 4

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade
1	5	7	7

O valor associado ao impacto no negócio é:

(1+5+7+7) / 4 = 5

O cálculo do valor de probabilidade total é Alto, dado que:

(5.666 + 6.333) /2 = 6

O cálculo do valor de impacto total é Alto, dado que:

(4 +5) /2 = 5

Utilizando a matriz de mapeamendo dos vectores de probabilidade e impacto obtém-se o grau geral de risco de Alto, como se pode verificar na tabela abaixo.

Grau de Risco				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 14 – Grau de risco associado à Educação

4.2.5 Outros Prestadores de Serviços

Foram avaliadas um total de duas entidades no grupo de "Outros prestadores de serviços", tendo sido auditada uma aplicação *Web* por cada entidade.

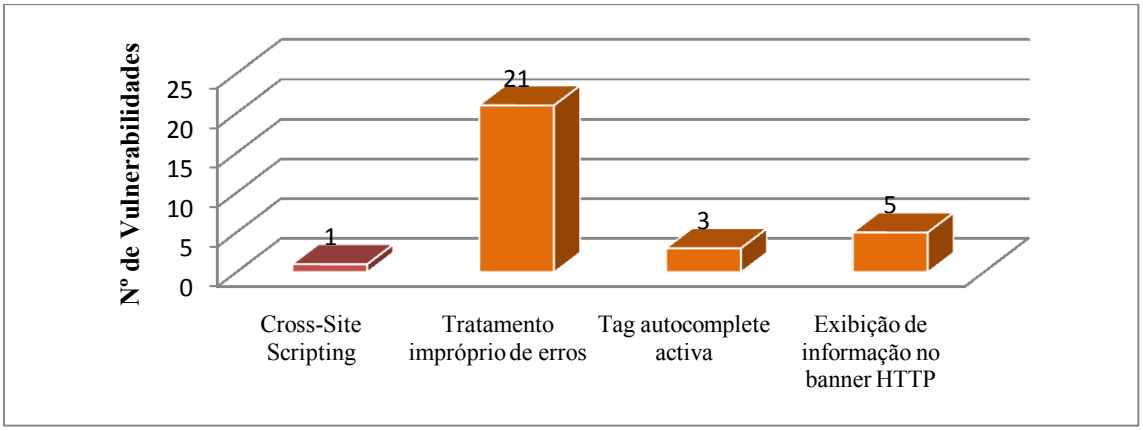


Figura 59 – Aplicação *Web* 1 dos Outros prestadores de serviços

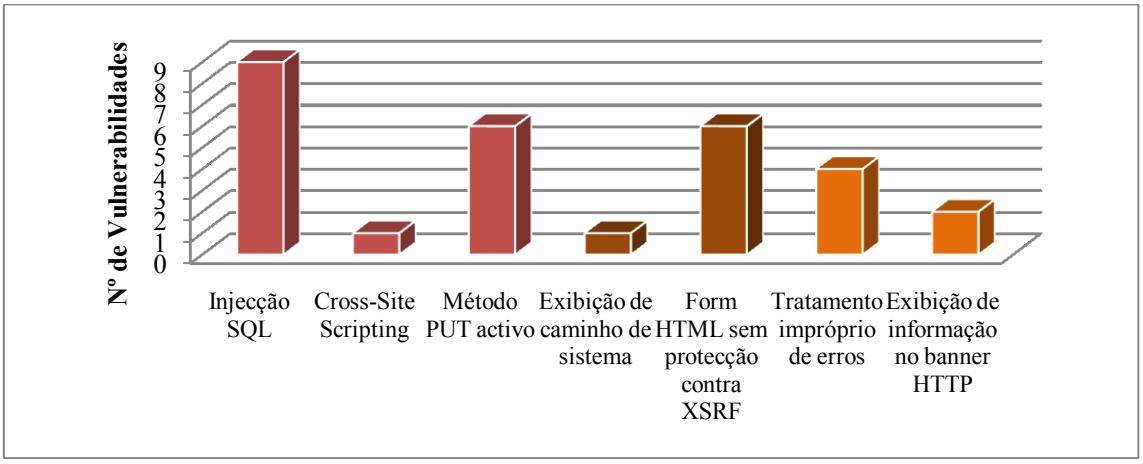


Figura 60 – Aplicação *Web* 2 dos Outros prestadores de serviços

Como se pode observar pelos resultados anteriormente descritos, foram confirmadas um total de 59 vulnerabilidades de entre as quais 17 estão classificadas como Grau 1, 7 em Grau 2 e 35 como Grau 3, distribuição esta que pode ser vista no gráfico abaixo representado.

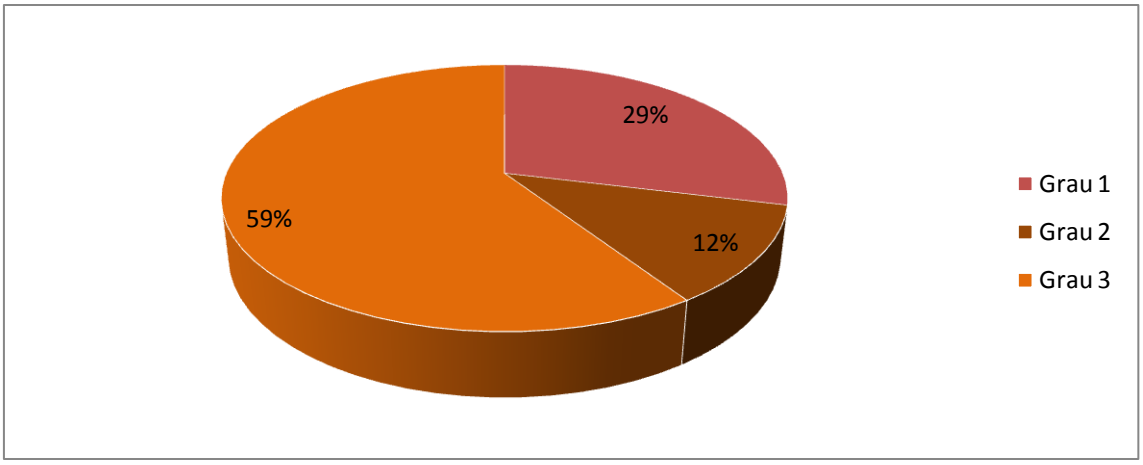


Figura 61 – Distribuição da totalidade das vulnerabilidades dos Outros prestadores de serviços

Adicionalmente, no gráfico abaixo pode ser verificado que a distribuição na classificação do OWASP Top 10 se centra sobretudo nas más configurações de segurança apesar da existência de várias vulnerabilidades críticas de injecção e XSS.

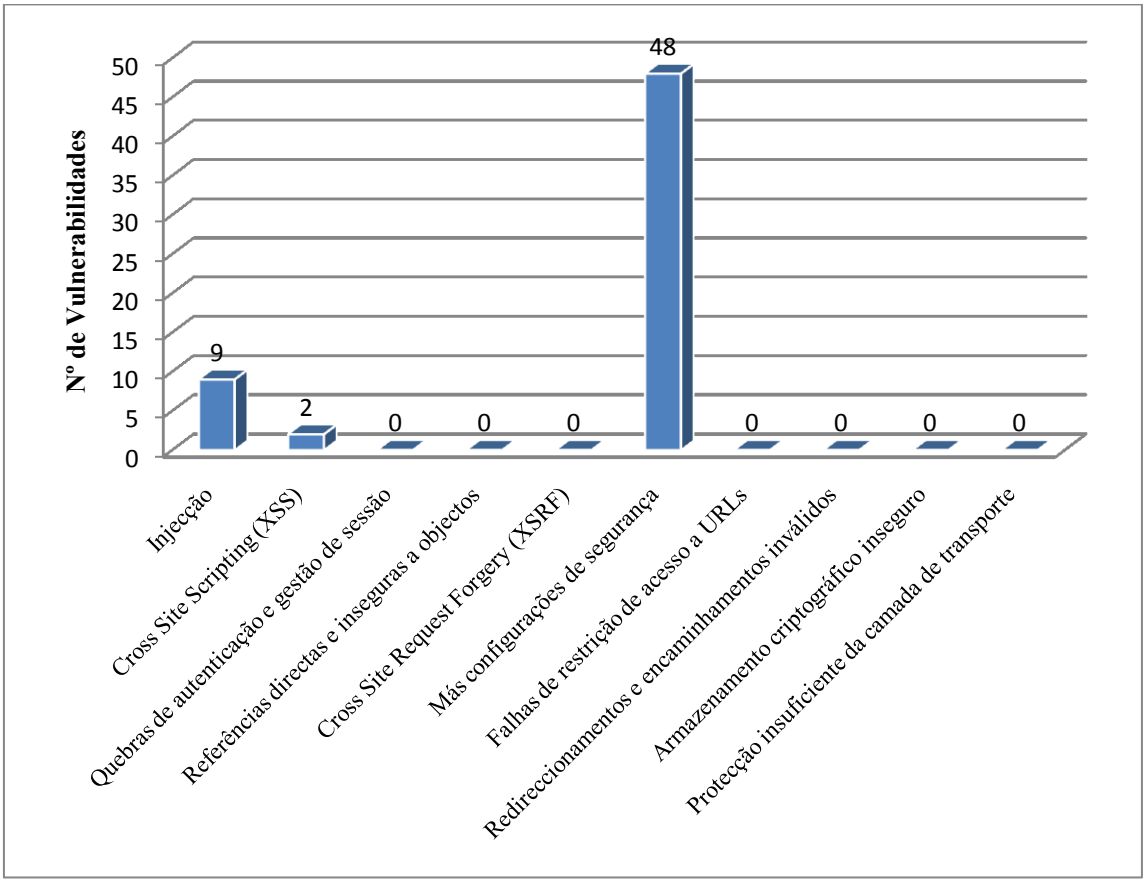


Figura 62 – Distribuição no OWASP Top 10 das vulnerabilidades dos Outros prestadores de serviços

4.2.5.1 Análise de Risco

Com os resultados obtidos na fase de auditoria de vulnerabilidades, foi possível obter uma base sólida para a elaboração de uma análise de risco baseada nas vulnerabilidades identificadas.

Factores dos agentes de ameaça		
Nível de perícia	Motivo	Oportunidade
9	4	7

O valor associado aos factores de ameaça é:

(9+4+7) / 3 = 6,666

Factores da vulnerabilidade		
Facilidade de descoberta	Facilidade de exploração	Nível de conhecimento
9	1	9

O valor associado aos factores da vulnerabilidade é:

(9+1+9) / 3 = 6.333

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Confidencialidade (%/100)*Peso**
Injecção SQL	9	15,254	0,153	9	1,373
Cross-Site Scripting	2	3,390	0,034	9	0,305
Método PUT activo	6	10,169	0,102	1	0,102
Form HTML sem protecção contra XSRF	6	10,169	0,102	8	0,814
Tratamento impróprio de erros	25	42,373	0,424	8	3,390
Tag autocomplete activa	3	5,085	0,051	4	0,203
Exibição de informação no *banner* HTTP	7	11,864	0,119	8	0,949
Exibição de caminho de sistema	1	1,695	0,017	9	0,153
Total (arredondado às unidades)	59	100	1		7

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Integridade (%/100)*Peso**
Injecção SQL	9	15,254	0,153	9	1,373
Cross-Site Scripting	2	3,390	0,034	8	0,271
Método PUT activo	6	10,169	0,102	9	0,915
Form HTML sem protecção contra XSRF	6	10,169	0,102	1	0,102

Tratamento impróprio de erros	25	42,373	0,424	5	2,119
Tag autocomplete activa	3	5,085	0,051	1	0,051
Exibição de informação no *banner* HTTP	7	11,864	0,119	1	0,119
Exibição de caminho de sistema	1	1,695	0,017	4	0,068
Total (arredondado às unidades)	59	100	1		5

Descrição da vulnerabilidade	**Qt.**	**%**	**% / 100**	**Peso (0-9)**	**Disponibilidade (%/100)*Peso**
Injecção SQL	9	15,254	0,153	8	1,220
Cross-Site Scripting	2	3,390	0,034	9	0,305
Método PUT activo	6	10,169	0,102	8	0,814
Form HTML sem protecção contra XSRF	6	10,169	0,102	2	0,203
Tratamento impróprio de erros	25	42,373	0,424	5	2,119
Tag autocomplete activa	3	5,085	0,051	2	0,102
Exibição de informação no *banner* HTTP	7	11,864	0,119	1	0,119
Exibição de caminho de sistema	1	1,695	0,017	5	0,085
Total (arredondado às unidades)	59	100	1		5

Impacto técnico		
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade
7	5	5

O valor associado ao impacto técnico é:

(7+5+5) / 3 = 5.666

Impacto no negócio			
Danos financeiros	Danos de reputação	Não concordância com standards	Violação da privacidade
3	1	5	5

O valor associado ao impacto no negócio é:

(3+1+5+5) / 4 = 3.5

O cálculo do valor de probabilidade total é Alto, dado que:

(6.666 + 6.333) /2 = 7

O cálculo do valor de impacto total é Médio, dado que:

(5.666 + 3.5) /2 = 5

Utilizando a matriz de mapeamendo dos vectores de probabilidade e impacto obtém-se o grau geral de risco de Alto, como se pode verificar na tabela abaixo.

Grau de Risco				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Residual	Baixo	Médio
		Baixo	Médio	Alto
	Probabilidade			

Tabela 15 – Grau de risco associado aos Outros prestadores de serviços

4.3 Fiabilidade dos Web Scanners Utilizados

A análise de 42 aplicações *Web* permitiu estabelecer uma base de confiança para a análise da fiabilidade dos *Web scanner* utilizados. Ao longo do estudo realizado foram analisadas diversas tecnologias desde PHP (15 aplicações *Web*), Java (1 aplicação *Web*) e .Net/ASPX (26 aplicações *Web*). O anterior facto é importante pois permite uma melhor percepção da capacidade de análise por parte dos *web scanners* num conjunto de tecnologias para a *Web,* dominantes no mercado, ao invés da centralização em apenas uma (para a qual o *scanner* até possa estar optimizado, viciando uma análise de fiabilidade).

Os resultados individuais de cada *web scanner*, na contagem da totalidade de vulnerabilidades detectadas e dos falsos positivos associados, pode ser visto na figura abaixo representada (Figura 64).

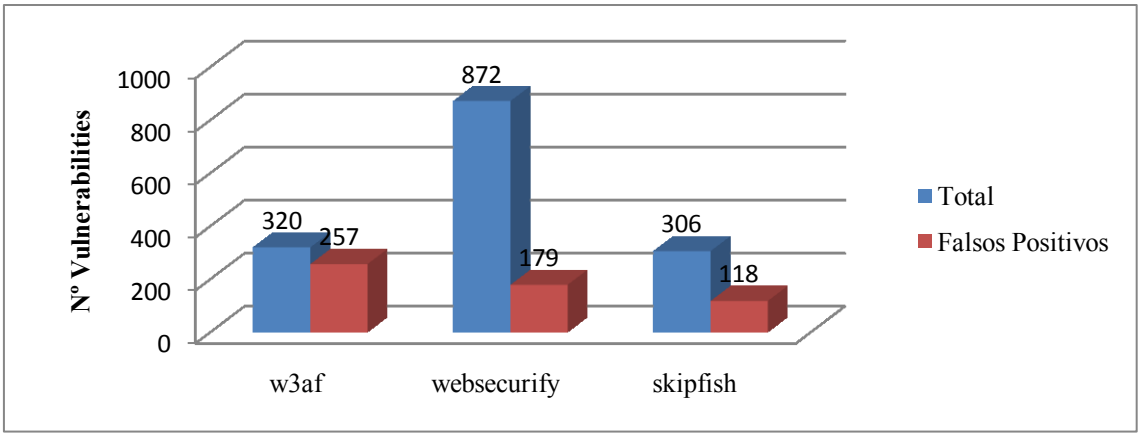


Figura 63 – Análise total das vulnerabilidades detectadas pelos *web scanners*

Individualmente, o rácio de total de descoberta de vulnerabilidades face aos falsos positivos identificados pode ser observado nos gráficos abaixo identificados.

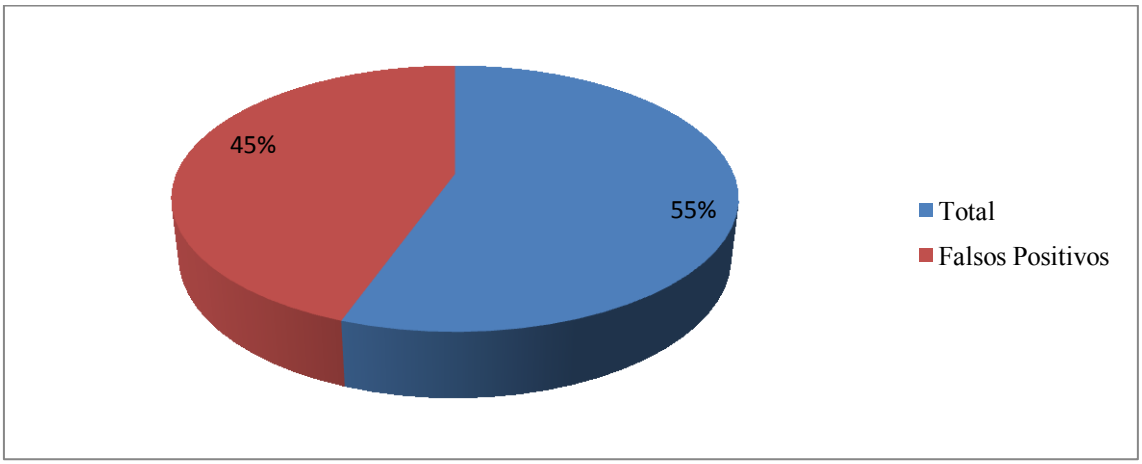


Figura 64 – Percentagem de falsos positivos do w3af

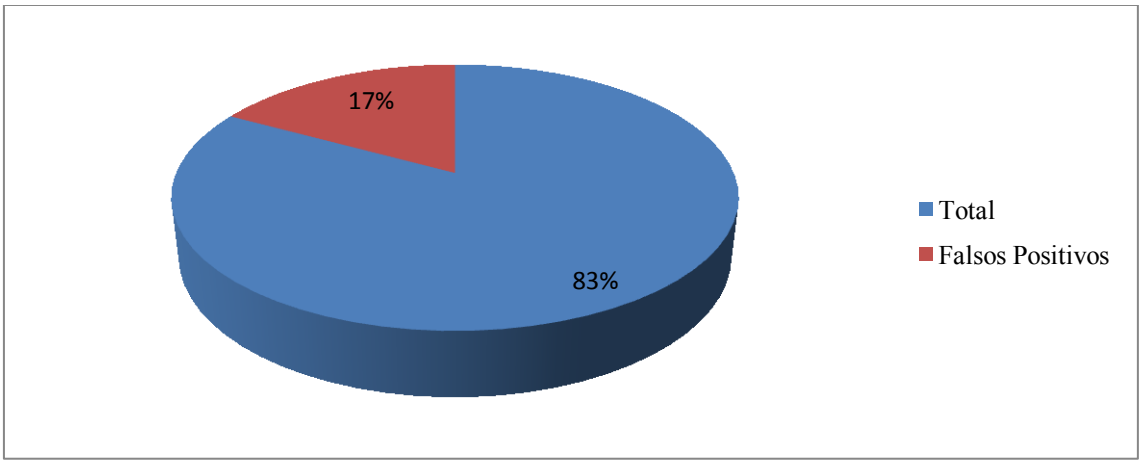


Figura 65 – Percentagem de falsos positivos do Websecurify

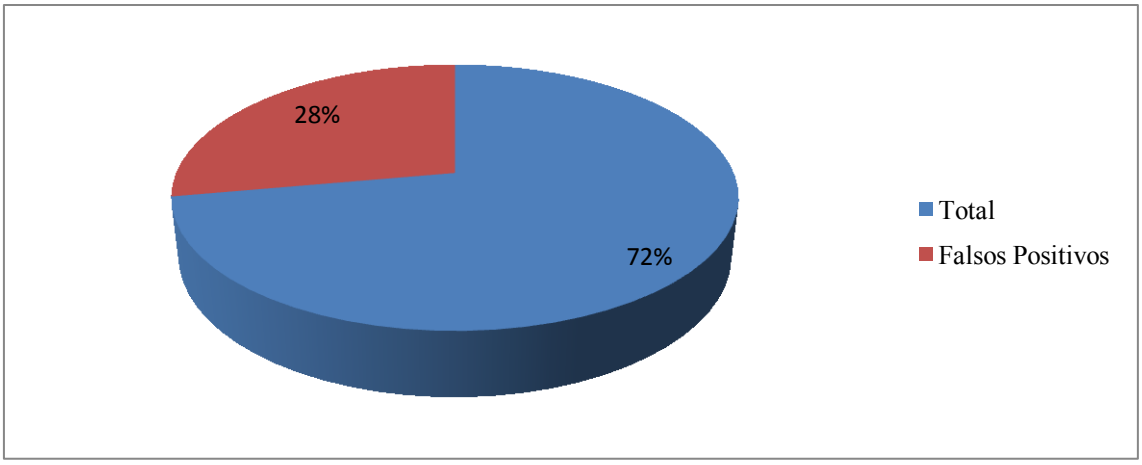


Figura 66 – Percentagem de falsos positivos do skipfish

Como é possível observar, o *web scanner* que apresenta maior percentagem de falsos positivos é o **w3af** com 45%, seguindo-se o **skipfish** com 28%. Por último, e com menor percentagem de falsos positivos encontra-se o **Websecurify** com apenas 17%. Como visão agregada da totalidade dos resultados, temos que apenas 27% das vulnerabilidades identificadas constituem falsos positivos como é possível observar no gráfico abaixo.

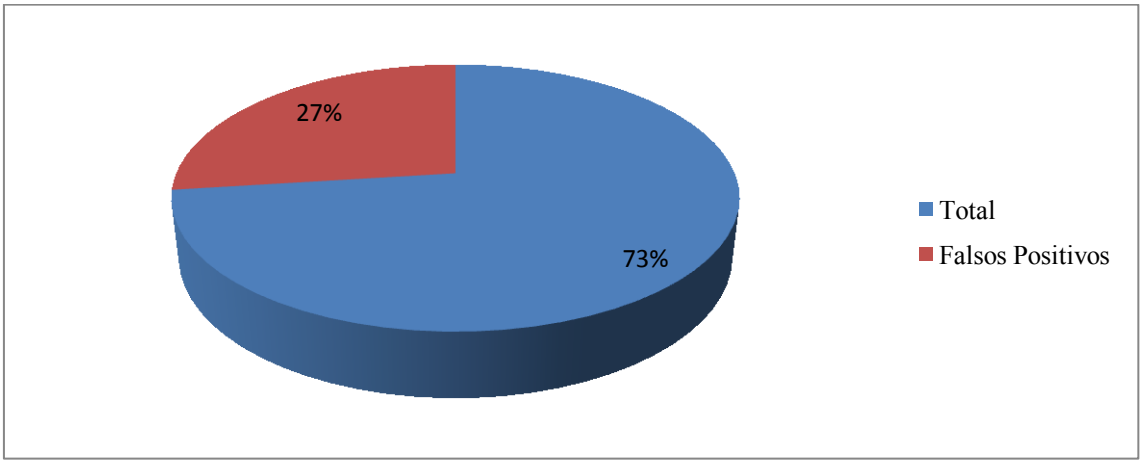


Figura 67 – Total de falsos positivos de todos os *web scanners*

Estes resultados mostram que a utilização de *web scanners* pode de facto fornecer bons resultados e indicadores do estado de segurança de aplicações Web, embora o número de falsos positivos exiga que estes sejam utilizados com algumas reservas. No entanto, num total de 1498 vulnerabilidades, apenas 554 serem falsos positivos, faz com que estas ferramentas mereçam não ser descartadas pelo menos numa fase inicial da auditoria.

Capítulo 5 Conclusões

O trabalho realizado nesta dissertação, além de ter ajudado a criar uma *framework* para avaliar o estado de segurança de aplicações *Web* e o risco associado a estas vulnerabilidades, baseado em documentação e ferramentas gratuitas, permitiu reunir uma série de indicadores para avaliar o estado geral da segurança *Web* em determinados nichos em Portugal com o intuito de extrapolar os resultados (embora com algumas reservas, visto a amostra ser diminuta).

Na secção 5.1 são revistos todos os objectivos propostos e atingidos, resumindo o trabalho desenvolvido na presente dissertação. A secção 5.2 aponta que as principais linhas de desenvolvimento futuro da presente *framework* de análise de vulnerabilidades, passam principalmente pelo incremento e variação das aplicações *Web* assim como das ferramentas utilizadas e da metodologia de análise de risco.

5.1 Objectivos

Tal como foi referido ao longo da dissertação, os principais objectivos da mesma relacionam-se com o desenvolvimento de uma *framework* baseada em ferramentas e documentação aberta e gratuita que permita avaliar o estado de segurança de aplicações *Web* assim como utilizar esses resultados para a realização de uma análise de risco associada ao negócio.

Numa visão geral do estado de segurança *Web*, foram confirmadas um total de 944 vulnerabilidades em 42 aplicações *Web* (dando uma média de 22 vulnerabilidades por aplicação), sendo que 110 vulnerabilidades são de Grau 1, 93 vulnerabilidades de Grau 2 e 741 de Grau 3. Com uma percentagem de 12% de vulnerabilidades de alta criticidade num domínio de estudo extremamente relevante (apesar de diminuto) no panorama português, e onde as medidas e investimento em segurança da informação e protecção dos dados abundam na sua maioria, pode-se extrapolar que em domínios menos relevantes, o cenário seja ainda mais preocupante. A preocupação com a temática da segurança *Web* é perfeitamente corroborada com os resultados derivados do estudo realizado nesta dissertação. É possível verificar que, segundo a metodologia de análise

de risco utilizada, o risco associado à totalidade dos grupos se situava num nível crítico, espelhando bem que este é um tema longe da maturidade desejável.

A avaliação dos grupos de entidades que fizeram parte do estudos nesta dissertação originou igualmente alguns resultados preocupantes.

No grupo da Administração Pública foram confirmadas um total de 197 vulnerabilidades de entre as quais 21 estão classificadas como Grau 1, 40 em Grau 2 e 136 como Grau 3. Este grupo conta com 11% das vulnerabilidades classificadas com o maior nível de criticidade, tendo sido obtida uma classificação final de análise de risco de nível crítico.

Nas Forças Armadas foram confirmadas um total de de 57 vulnerabilidades, estando 5 classificadas como Grau 1, 11 classificadas como Grau 2 e 41 como Grau 3, derivando em 9% das vulnerabilidades classificadas com o maior grau de criticidade, tendo sido obtida uma classificação final de análise de risco de nível alto.

O grupo da Educação conta com um total de 631 vulnerabilidades de entre as quais 67 estão classificadas como Grau 1, 33 em Grau 2 e 531 como Grau 3 derivando em 11% das vulnerabilidades classificadas com grau máximo de criticidade e numa classificação da análise de risco de nível alto.

O enquadramento dos outros prestadores de serviços conta com um total de 59 vulnerabilidades confirmadas, sendo 17 classificadas como grau 1, 7 em grau 2 e 35 como grau 3, derivando num total de 29% das vulnerabilidades enquadradas no nível máximo de criticidade e numa classificação da análise de risco de nível alto.

É demais evidente a necessidade de melhoria no domínio da segurança aplicacional. Os resultados acima descritos mostram uma maturidade na segurança aplicacional muito longe do necessário, especialmente tendo em conta o tipo de entidades e aplicações envolvidas no estudo.

Destes resultados poderão ser retiradas algumas conclusões, as quais são:

1. Falta de capacidade dos programadores na temática da segurança aplicacional;
2. Impossibilidade de inclusão de medidas de segurança aplicacional; e
3. Falta de compreensão da gravidade originada pela insegurança aplicacional.

O primeiro ponto relaciona-se com a falta de preparação dos programadores no domínio específico da segurança aplicacional. Apesar dos programadores não terem que ser especialistas no domínio da segurança, muitas vezes, as bases que são necessárias para que muitas vulnerabilidades sejam eliminadas, não são sequer conhecidas. Este é um problema que, a nível nacional não tem resposta ao nível da formação académica da maioria dos recém-licenciados que iniciam a carreira de programadores, e acabam por ser incluídos em equipas que desenvolvem aplicações com elevada importância no panorama nacional.

Neste particular, a OWASP pode ter um papel importante. A OWASP *Academies* (OWASP, Academies) é um projecto que pretende, a nível internacional, proporcionar um modelo de interacção entre alunos e professores, com foco na temática da segurança aplicacional, diminuindo o fosso de necessidade/conhecimento que até então predomina no mercado.

O segundo ponto relaciona-se com o facto de que na maioria dos casos a segurança é deixada para um patamar de importância inferior ao dos prazos de entrega e da beleza estética das funcionalidades implementadas nas aplicações. Este facto liga-se directamente ao terceiro ponto dado que, na minha opinião pessoal, esta troca de prioridades deve-se precisamente à falta de compreensão dos riscos que a que o negócio e os seus intervenientes estão sujeitos na eventualidade de algumas vulnerabilidades mais críticas serem exploradas.

É também possível concluir que, numa opinião pessoal, as entidades governamentais ou reguladoras das tecnologias em Portugal, deveriam procurar mais apoio e interligação com entidades internacionais como a OWASP, especialmente dada a sua forte presença e motivação no tema da segurança aplicacional em Portugal. Este tipo de relações poderão ser muito frutíferas, especialmente dados os resultados preocupantes obtidos neste estudo.

No decorrer deste estudo foram elaborados e aceites artigos em 5 conferências, uma nacional e 4 internacionais, as quais se encontram abaixo descritas:

- *Ibero-American Web Application Security Conference* 2009 (IBWAS'09) – Esta conferência teve lugar em Madrid, onde foi apresentado o artigo "*Web*

Applications Security Assessment in the Portuguese World Wide Web Panorama" (Nuno Teodoro e Carlos Serrão, 2009);

- Conferência da Associação Portuguesa de Sistemas de Informação 2010 (CAPSI 2010) – Esta conferência teve lugar em Viana do Castelo, onde foi apresentado o artigo "Desenvolvimento de Aplicações e Sistemas de Informação para a *World Wide Web* – uma actividade de elevado risco" (Carlos Serrão, Nuno Teodoro e Joaquim Marques, 2010);
- *Ibero-American Web Application Security Conference* 2010 (IBWAS'10) – Esta conferência teve lugar em Lisboa, onde foi apresentado o artigo "*Automating Web Applications Security Assessments through Scanners"* (Nuno Teodoro e Carlos Serrão, 2010);
- *World Congress on Internet Security* (WCIS 2011) – Esta conferência teve lugar em Londres, onde foi apresentado o artigo "*Assessing the Portuguese Web Applications Security*" (Nuno Teodoro e Carlos Serrão, 2011); e
- International Conference on Information Society 2011 (i-Society 2011) – Esta conferência teve lugar em Londres, onde foi apresentado o artigo "*Web Application Security – Improving Critical Web-based Applications Quality through in-depth Security Analysis*" (Nuno Teodoro e Carlos Serrão, 2011).

5.2 Trabalho Futuro

Ainda que todos os objectivos propostos tenham sido alcançados com sucesso existem sempre melhorias que podem ser realizadas.

Numa primeira fase será relevante analisar um maior número, e uma maior diversidade de entidades e de aplicações *Web*, mas que ainda assim possuam os requisitos identificados para a escolha das entidades incluídas neste estudo. Contudo, as possibilidades de trabalho futuro não se esgotam no puro aumento e variação de aplicações *Web*. Seria interessante a avaliação de novos *web scanners* dentro da comunidade *open source*, como é o caso do Arachni, Netsparker, entre muitos outros.

Adicionalmente, realizando uma nova auditoria de vulnerabilidades exactamente às mesmas aplicações *Web* deste estudo, constituiria uma análise interessante, avaliando o comportamento que os mesmos *web scanners*, numa versão mais recente e melhorada.

Igualmente, uma nova auditoria às mesmas entidades e aplicações deveria estar presente no trabalho futuro, indicando se foram implementadas as melhorias apresentadas nos relatórios entregues, assim como se nas novas aplicações desenvolvidas foram contemplados os problemas descobertos ao longo da auditoria.

O trabalho realizado nesta dissertação elevou o nível de conhecimento de segurança *Web* em entidades e grupos críticos em Portugal e permitiu a criação de uma *framework* de avaliação de segurança *Web* e análise de risco sem custos financeiros directamente associados (excepto do tempo das pessoas dispendido nas auditorias), fornecendo uma importante base para as organizações estabelecerem um ponto de referência da sua maturidade na segurança das suas aplicações *Web.*

Referências

Adam Doupé, Marco Cova, e Giovanni Vigna, 2010. Why Johnny Can't Pentest:An Analysis of Black-box Web Vulnerability Scanners.

Administrativa, G. d. (s.d.). *SIMPLEX*. Obtido em 6 de Fevereiro de 2011, de http://www.simplex.pt

Beizer, B. (1990). *Software Testing Techniques.* New York: Van Nostrand Reinhold.

Carlos Serrão, Nuno Teodoro e Joaquim Marques. (2010). Desenvolvimento de Aplicações e Sistemas de Informação para a World Wide Web – uma actividade de elevado risco. *CAPSI.* Viana do Castelo.

CERT, Computer Emergency Response Team. (s.d.). *Serviço de Resposta a Incidentes de Segurança Informática*. Obtido em 12 de Fevereiro de 2011, de http://www.cert.pt

CISCO, 2010. (s.d.). *Security Annual Report 2010.* Obtido em 8 de Março de 2011, de http://www.cisco.com/en/US/prod/collateral/vpndevc/security_annual_report_2010.pdf

Day, J., & Zimmermann, H. (1983). The OSI reference model. *Proceedings of the IEEE* (pp. 1334-1340). IEEE.

DHS, National Cyber Security Division. (s.d.). *Department of Homeland Security (DHS) National Cyber Security Division*. Obtido em 6 de Março de 2011, de http://www.dhs.gov/xabout/structure/editorial_0839.shtm

Ebenezer Oladimeji, S. S. (2006). Security threat Modeling and Analysis: A Goal-Oriented Approach. *The 10th IASTED International Conference on Software Engineering and Applications.* Dallas.

Fong, E., & Okun, V. (2007). Web Application Scanners: Definitions and Functions. *40th Annual Hawaii International Conference on System Sciences* (pp. 280b-280b). IEEE.

Gartner CyberThreat Landscape. (s.d.). Obtido em 8 de Março de 2011, de http://www.dts.ca.gov/pdf/news_events/sec_awareness/Gartner_CyberThreat_landscape_2010.pdf

Gartner, 2011. (s.d.). *CyberThreat Landscape.* Obtido em 8 de Março de 2011, de http://www.dts.ca.gov/pdf/news_events/sec_awareness/Gartner_CyberThreat_landscape_2010.pdf

Gautam Pant, Padmini Srinivasan e Filippo Menczer . (2003). Crawling the Web.

Holz, T., Marechal S. e Raynal, F. (19 de Outubro de 2009). New Threats and Attacks on the World Wide Web. *IEEE Security & Privacy* , pp. 72-75.

IBM, 2010. (s.d.). *Designing a strategy for comprehensive web protection.* Obtido em 8 de Março de 2011, de http://public.dhe.ibm.com/common/ssi/ecm/en/raw14246usen/RAW14246USEN.PDF

IEEE. (12 de Dezembro de 2007). About Penetration Testing. *Security & Privacy* , p. 84.

IEEE. (2004). Static Analysis for Security. *IEEE Security & Privacy* , pp. 32-35.

ISSAF, Information Systems Security Assessment Framework. (s.d.). *Open Information Systems Security Group - ISSAF*. Obtido em 1 de Março de 2011, de http://www.oissg.org/issaf

Linux, Backtrack. (s.d.). *Backtrack*. Obtido em 26 de Fevereiro de 2011, de http://www.backtrack-linux.org/

Matteo Meucci, Girogio Fedon e Pavol Luptak, 2011. (s.d.). *Planning the OWASP Testing Guide v4.* Obtido em 8 de Março de 2011, de http://www.owasp.org/images/b/b2/OWASP_Testing_Guide_-_OWASP_Summit_2011.pdf

NIST. (s.d.). *National Institute of Standards and Technology*. Obtido em 12 de Fevereiro de 2011, de http://www.nist.gov

NIST, National Institute of Standards and Technology. (s.d.). *National Institute of Standards and Technology - NIST*. Obtido em 6 de Março de 2011, de http://www.nist.gov

NIST, National Vulnerability Database. (s.d.). *National Vulnerability Database*. Obtido em 12 de Fevereiro de 2011, de http://nvd.nist.gov/

NIST, SAMATE. (s.d.). *Software Assurance Metrics and Tool Evaluation*. Obtido em 6 de Março de 2011, de http://samate.nist.gov

NIST, Software Vulnerabilities . (s.d.). *Software Vulnerabilities*. Obtido em 8 de Março de 2011, de http://web.nvd.nist.gov/view/vuln/search-results?query=&search_type=all&cves=on

NIST, WASSFS. (s.d.). *Web Application Security Scanner Functional Specification.* Obtido em 6 de Março de 2011, de http://samate.nist.gov/docs/webapp_scanner_spec_sp500-269.pdf

Nuno Teodoro e Carlos Serrão. (2011). Assessing the Portuguese Web Applications Security. Londres: IEEE.

Nuno Teodoro e Carlos Serrão. (2010). Automating Web Applications Security Assessments through Scanners. Lisboa: Springer.

Nuno Teodoro e Carlos Serrão. (2011). Improving Critical Web-based Applications Quality through in-depth Security Analysis. Londres: IEEE.

Nuno Teodoro e Carlos Serrão. (2009). Web Applications Security Assessment in the Portuguese World Wide Web panorama. Madrid: Springer.

Oehlert, P. (2005). Violating assumptions with fuzzing. *Security & Privacy* (pp. 58- 62). IEEE.

Open Source Web Application Scanner Poll Results. (s.d.). Obtido em 27 de Fevereiro de 2011, de http://www.ethicalhack3r.co.uk/security/open-source-web-application-scanner-poll-results/

OSSTMM, Open Source Security Testing Methodology Manual. (s.d.). *Institute for Security and Open Source Methodologies - OSSTMM*. Obtido em 01 de Março de 2011, de http://www.isecom.org/osstmm/

OWASP Risk Rating Methodology. (s.d.). Obtido em 27 de Fevereiro de 2011, de http://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

OWASP, Academies. (s.d.). *OWASP Academies*. Obtido em 29 de Maio de 2011, de https://www.owasp.org/index.php/OWASP_Academies#tab=OWASP_Academies_meeting_-_5th.2C_6th_January

OWASP, Application Threat Modeling. (s.d.). *OWASP Application Threat Modeling*. Obtido em 8 de Março de 2011, de http://www.owasp.org/index.php/Application_Threat_Modeling

OWASP, Common Vulnerability List. (s.d.). *OWASP Common Vulnerability List*. Obtido em 27 de Fevereiro de 2011, de http://www.owasp.org/index.php/OWASP_Common_Vulnerability_List#tab=Vulnerability_List_.28DRAFT.29

OWASP, Open Web Application Security Project. (s.d.). *OWASP*. Obtido em 12 de Fevereiro de 2011, de http://www.owasp.org

OWASP, OWASP Top 10 2004. (s.d.). *OWASP Top 10 2004*. Obtido em 16 de Fevereiro de 2011, de http://kent.dl.sourceforge.net/project/owasp/Top%20Ten/2004/OWASPTopTen2004.pdf

OWASP, OWASP Top 10 2007. (s.d.). *OWASP Top 10 2007.* Obtido em 26http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf de Maio de 2011

OWASP, OWASP Top 10 2010. (s.d.). *OWASP Top 10 2010*. Obtido em 16 de Fevereiro de 2011, de http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf

PGDL, 2009. (s.d.). *Lei do Cibercrime*. Obtido em 22 de Fevereiro de 2011, de http://www.pgdlisboa.pt/pgdl/leis/lei_mostra_articulado.php?nid=1137&tabela=leis&ficha=1&pagina=1

SIMPLEX. (s.d.). Obtido em 6 de Fevereiro de 2011, de http://www.simplex.pt

SIMPLEX, 2006. (s.d.). Obtido em 6 de Fevereiro de 2011, de http://www.simplex.pt

WASC, 2008. (s.d.). *Web Application Security statistics.* Obtido em 8 de Março de 2011, de https://files.pbworks.com/download/PvEkez8z0n/webappsec/13247070/WASS-SS-2008.pdf

WASC, WASSEC Evaluation Spreadsheet. (s.d.). *WASSEC Evaluation Spreadsheet.* Obtido em 27 de Fevereiro de 2011, de http://projects.webappsec.org/f/WASSEC+Evaluation+Spreadsheet.xls

WASC, WASSEC. (s.d.). *Web Application Security Scanner Evaluation Criteria - WASSEC*. Obtido em 6 de Março de 2011, de http://projects.webappsec.org/w/page/13246986/Web-Application-Security-Scanner-Evaluation-Criteria

WASC, Web Application Security Consortium. (s.d.). *WASC*. Obtido em 12 de Fevereiro de 2011, de http://www.webappsec.org

WASC, Web Application Security Consortium: Threat Classification v1. (s.d.). *Web Application Security Consortium: Threat Classification v1*. Obtido em 16 de Fevereiro de 2011, de https://files.pbworks.com/download/J0SZxkW8oP/webappsec/13247053/WASC-TC-v1_0.pdf

WASC, Web Application Security Consortium: Threat Classification v2. (s.d.). *Web Application Security Consortium: Threat Classification v2*. Obtido em 16 de Fevereiro de 2011, de https://files.pbworks.com/download/bZTVJ1dyrt/webappsec/13247059/WASC-TC-v2_0.pdf

WASC, Web Application Security Scanner List. (s.d.). *Web Application Security Scanner List*. Obtido em 26 de Fevereiro de 2011, de http://projects.webappsec.org/w/page/13246988/Web-Application-Security-Scanner-List

WhiteHat, 2011. (s.d.). *WhiteHat Website Security Statistics Report.* Obtido em 8 de Março de 2011, de http://www.whitehatsec.com/home/resource/stats.html

Anexo A – Proposta de Serviços de Auditoria

Introdução

Pretende-se com este documento estabelecer as condições iniciais segundo as quais serão levados a cabo os diversos testes a realizar no âmbito da bateria de testes de segurança aplicacionais Web, no contexto de uma tese de Mestrado em Engenharia Informática (MEI) a realizar no Instituto Superior de Ciências do Trabalho e da Empresa – Instituto Universitário de Lisboa (ISCTE-IUL).

Este trabalho será executado pelo aluno Nuno Filipe Teodoro e supervisionado pelo Professor Doutor Carlos Serrão.

Sendo o Professor Carlos Serrão líder do capítulo português da OWASP e o aluno Nuno Teodoro membro dessa mesma organização, este projecto terá uma grande envolvência por parte dos materiais e conhecimento produzidos por esta organização. Esta organização é mundialmente reconhecida na área da segurança aplicacional, conferindo ainda uma maior relevância e notoriedade ao contexto deste trabalho de mestrado. O culminar desta tese de mestrado tem também como objectivo escalar a interligação da OWASP com a temática da segurança aplicacional em Portugal, fornecendo o primeiro passo de uma cooperação fundamental para o progresso da segurança digital nacional.

Objectivo e Âmbito

O objectivo do trabalho a realizar consiste em realizar um conjunto de testes automáticos e manuais a uma série de aplicações Web previamente identificadas e autorizadas, de forma a procurar identificar potenciais vulnerabilidades nas mesmas.

A identificação das aplicações Web a analisar serve o propósito de identificar quais as aplicações mais críticas para a organização, sendo que são essas que deverão ser o alvo dos testes, conferindo ao projecto levado a cabo no trabalho de mestrado, uma importância acentuada, revelando o panorama geral da segurança das aplicações Web em Portugal em dominios críticos.

Faz parte do âmbito deste trabalho o seguinte:

- A realização de um conjunto de **testes automatizados e manuais**, segundo uma metodologia definida e um conjunto de ferramentas seleccionadas (e que serão identificadas neste documento);
- A **verificação dos resultados** produzidos por essas mesmas ferramentas, e eventuais formas de exploração das vulnerabilidades encontradas (se alguma) nunca explorando a fundo as mesmas;
- A produção de **relatórios** com a listagem das vulnerabilidades encontradas, formas de as explorar, e **recomendações** para as resolver.

A metodologia a usar no trabalho será do tipo white-hat, **não havendo lugar a qualquer tipo de intrusão não autorizada nem escalamento dos privilégios das aplicações testadas**. Os testes serão realizados tendo como premissa que poderiam ser utilizadores legítimos que os poderiam realizar não sendo necessários acessos privilegiados à aplicação, sendo levados a cabo numa metodologia de **black-box**.

Estrutura do Documento

Este documento começa por introduzir os objectivos do trabalho a realizar, detalhando e contextualizando os mesmos no âmbito de um trabalho académico.

Seguidamente será apresentada a metodologia a ser utilizada na realização do trabalho, sendo depois apresentadas as diversas ferramentas que serão utilizadas nos testes automatizados. Posteriormente, será descrito o formato para os resultados a apresentar.

Finalmente será apresentada uma proposta de planeamento para a realização dos múltiplos testes a realizar e a identificação das aplicações web seleccionadas.

Acrónimos e Abreviaturas

Acrónimos/Abreviaturas	Descrição
ADETTI	Adetti – Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática
OWASP	Open Web Application Security Project
WASC	Web Application Security Consortium
MEI	Mestrado em Engenharia Informática
ISCTE-IUL	Instituto Superior de Ciências do Trabalho e da Empresa – Instituto Universitário de Lisboa

Documentos de Referência

[DR.1] OWASP, "Open Web Applications Security Project", http://www.owasp.org

[DR.2] OWASP, "Testing Guide", http://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf

[DR.3] Nuno Teodoro e Carlos Serrão, "Web Applications Security Assessment in the Portuguese World Wide Web Panorama", IBWAS'09, Novembro 2009

[DR.4] Testes de penetração baseados em black-box, http://en.wikipedia.org/wiki/Penetration_test

[DR.5] OWASP Top 10, http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project, 2010

[DR.6] OWASP Testing Guide, http://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf

[DR.7] A. Andreu, Professional Pen Testing for Web Applications (Programmer to Programmer), Wrox, 2006.

[DR.8] J. Faircloth, J. Beale, R. Temmingh, H. Meer, C.v. Walt, and H. Moore, Penetration Tester's Open Source Toolkit, Syngress, 2005.

[DR.9] Professional Penetration Testing, http://www.syngress.com/hacking-and-penetration-testing/Professional-Penetration-Testing/, visitado em Maio de 2010

[DR.10] WASC Threat Classification, http://projects.webappsec.org/Threat-Classification

[DR.11] W3af, Web Application Attack and Audit Framework, http://w3af.sourceforge.net/

[DR.12] Skipfish, http://code.google.com/p/skipfish/

[DR.13] Websecurify, http://www.websecurify.com/

[DR.14] Backtrack, http://www.backtrack-linux.org/

Objectivos do Trabalho

Este trabalho surge no contexto de um trabalho académico iniciado no âmbito do Mestrado em Engenharia Informática do ISCTE - IUL. Neste mestrado propôs-se a realização da avaliação do estado de segurança de algumas das principais aplicações que funcionam sobre a World Wide Web em Portugal.

No seguimento de uma divisão sectorial, o trabalho proposto tem como objectivo a realização de uma bateria de testes para avaliar o estado da segurança aplicacional de uma aplicação *Web* previamente definida pela entidade. Os testes terão como premissa de execução, um ambiente do tipo "*black box*" [DR.4]. Técnicas manuais e ferramentas

automatizadas serão empregues na determinação de vulnerabilidades, bem como metodologias definidas por entidades com créditos firmados na área da segurança aplicacional.

Os testes a realizar terão como objectivo determinar a existência de um conjunto de problemas, em especial aquelas que são designadas em comum por OWASP Top 10 [DR.5] (embora também se possa recorrer a outro tipo de classificação de vulnerabilidades, determinadas por outras entidades). No caso específico do OWASP Top 10, serão avaliadas as seguintes vulnerabilidades:

- Acesso e manipulação de sistemas e bases de dados, através da injecção directa de comandos enviados à aplicação;
- Acesso e manipulação de sessões, pelo envio de comandos dirigidos aos browsers dos utilizadores, em nome da aplicação;
- Subversão dos mecanismos que controlam a autenticação e as sessões dos utilizadores;
- Acesso directo a documentos por falha no controlo de acessos;
- Manipulação dos utilizadores e execução de operações na aplicação, não confirmadas, em seu nome;
- Acesso e manipulação de sistemas, serviços e bases de dados cujos controlos estejam mal parametrizados;
- Acesso directo a documentos por ausência de controlo de acessos;
- Encaminhamento dos utilizadores para páginas maliciosas;
- Acesso a informação sensível que não esteja protegida por cifra (ou protegida por cifra inadequada) nas bases de dados; e
- Acesso a informação sensível que não seja protegida enquanto atravessa a Internet.

Como resultado dos testes existe o objectivo de identificar, documentar e alertar a entidade para as vulnerabilidades encontradas, assim como, sempre que possível, uma possível mitigação para os problemas.

Descrição da Metodologia a Utilizar

A metodologia a usar neste trabalho será numa primeira fase a utilização de ferramentas automáticas para testes de vulnerabilidades aplicacionais, numa segunda fase, realização de testes manuais, e por fim, a elaboração de um relatório com as vulnerabilidade encontradas, se existirem, assim como uma breve descrição das mesmas e possíveis soluções. Os testes com ferramentas automáticas seguem um procedimento bastante simples e directo. Do conjunto de ferramentas seleccionadas, estas serão executadas tendo como alvo a aplicação web previamente definidas e os resultados obtidos, serão

analisados e confirmados. Serão alvo de avaliação e confirmação manual, estes resultados, evitando com maior rigor falsos positivos e falsos negativos que possam surgir. **A elaboração do relatório visa sobretudo fornecer um documento útil para a organização, validando a existência de vulnerabilidades detectadas, e oferecendo soluções que possam ajudar a mitigar as mesmas.**

Os testes manuais serão bastante extensos e serão criados a partir de uma compilação de vários materiais que pretendem auxiliar na criação de uma metodologia única, extensa e precisa para efectuar testes de penetração. Cada metodologia não pode ser considerada individualmente a melhor a ser utilizada, isto porque muitas vezes a conjunção da experiência dos autores de várias metodologias permitem colmatar lacunas e atingir uma abrangência de testes muito mais eficiente, sendo essa a abordagem seguida neste trabalho.

Os testes manuais seguirão várias metodologias, das quais se destacam a metodologia seguida no "OWASP Testing Guide" [DR.6], no livro "Professional Pen Testing for Web Applications" [DR.7], no livro "Penetration Tester's Open Source Toolkit" [DR.8] e em "Professional Penetration Testing: Creating and Operating a Formal Hacking Lab" [DR.9]. Estes testes procurarão também incluir os principais problemas nas aplicações web que são definidos pela WASC mais precisamente no "WASC Threat Classification", sendo que este é um documento também bastante referênciado na análise de vulnerabilidades e ameaças no domínio da segurança aplicacional [DR.10].

Sempre que exista alguma alteração nas ferramentas abordadas neste documento, será pedida uma autorização à entidade testada para utilização das mesmas na fase de testes. Deste modo, poderá ser mais eficazmente mantido um controlo sobre os acessos que serão efectuados às aplicações Web na altura dos testes.

As metodologias aqui descritas seguem uma abordagem de testes de intrusão que consiste nos seguintes pontos:

1. Reconhecimento
   a. Nesta fase irá ser explorada e examinada a aplicação web e a infra-estrutura que a engloba de um modo superficial. O objectivo desta fase é o de mapear algum contexto que engloba a aplicação web, obtendo alguma informação sobre esta.
2. Enumeração
   a. Esta fase é onde irão ser obtidas as informações sobre os serviços e recursos específicos que estão associados à aplicação web que está a ser

testada. A obtenção desta informação permite saber quais são as vulnerabilidades que poderão estar associadas a esses serviços, permitindo explorá-las e saber como mitigar a existência das mesmas.

3. Testes aplicacionais
   a. Nesta etapa é onde os testes aplicacionais realmente irão ser efectuados. Estes testes irão ser levados a cabo consoante as informações recolhidas nas fases anteriores. Deste modo o sucesso de descoberta de possíveis vulnerabilidades irá ser muito maior, levando a uma auditoria da segurança aplicacional muito mais rápida e eficiente.

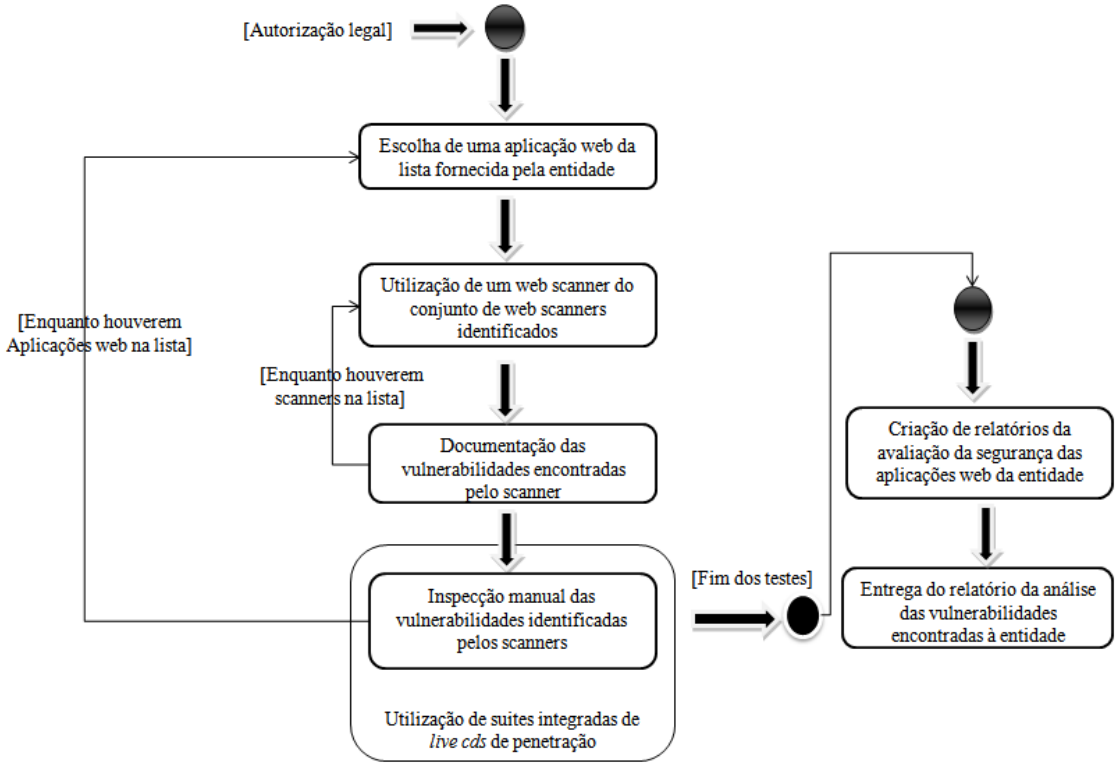


Figura 1. Descrição de alto nível da metodologia usada

Listagem e Descrição das Ferramentas

As ferramentas listadas nesta secção serão as principais ferramentas a utilizar neste trabalho. Esta lista é meramente indicativa do tipo de ferramentas a utilizar e a adição de novas ferramentas será efectuada mediante consulta e aprovação por parte da entidade testada. O facto de não ser possível definir uma lista precisa e imutável deve-se ao elevado número de ferramentas já existentes no mercado e a novas ferramentas que seriam desconhecidas ou inexistentes no mercado até à data e que poderão, a determinada altura, ser consideradas importantes na avaliação das aplicações Web.

A utilização de várias ferramentas automáticas de análise de vulnerabilidades (web scanners) deve-se ao facto de que deste modo será possível cobrir uma maior extensão de análises de vulnerabilidades. Estas ferramentas nem sempre possuem as mesmas capacidades e funcionalidades, sendo que utilizando uma gama de ferramentas suficientemente abrangente, poderão ser obtidos resultados mais precisos nesta análise. As ferramentas de análise de vulnerabilidades em aplicações Web open-source identificadas até ao momento para utilização nos testes, e as suas respectivas principais funcionalidades no que respeita a auditoria em aplicações Web são:

- **W3af** [DR.11]
  - Cross Site Request Forgeries (XSRF)
  - Más configurações do .htaccess
  - SQL injection
  - Auditoria aos parâmetros do certificado SSL
  - Exploração de inseguranças nos forms de upload de ficheiros
  - MX injection
  - Procura genérica de bugs
  - Local file include
  - Verifica se URL's com HTTPS estão acessíveis via HTTP
  - XPATH injection
  - Commandos do SO
  - Remote file inclusion
  - Erros de configuração do WebDAV
  - Server side include (SSI)
  - Injecção da função eval()
  - buffer overflow
  - Cross Site Scripting (XSS)
  - Cross Site Tracing (XST)
  - Blind SQL injection
  - Bugs em formatos de strings
  - preg_replace (PHP)
  - global redirection
  - LDAP injection
  - Phishing vectors
  - Configurações extensões de Frontpage (upload de ficheiros para o servidor)
  - Splitting
- **Skipfish** [DR.12]
  - Server-side SQL injection (including blind vectors, numerical parameters).
  - Explicit SQL-like syntax in GET or POST parameters.
  - Server-side shell command injection (including blind vectors).
  - Server-side XML / XPath injection (including blind vectors).
  - Format string vulnerabilities.
  - Integer overflow vulnerabilities.
  - Locations accepting HTTP PUT.

  - Stored and reflected XSS vectors in document body (minimal JS XSS support present).
  - Stored and reflected XSS vectors via HTTP redirects.
  - Stored and reflected XSS vectors via HTTP header splitting.
  - Directory traversal (including constrained vectors).
  - Assorted file POIs (server-side sources, configs, etc).
  - Attacker-supplied script and CSS inclusion vectors (stored and reflected).
  - External untrusted script and CSS inclusion vectors.
  - Mixed content problems on script and CSS resources (optional).
  - Incorrect or missing MIME types on renderables.
  - Generic MIME types on renderables.
  - Incorrect or missing charsets on renderables.
  - Conflicting MIME / charset info on renderables.
  - Bad caching directives on cookie setting responses.
  - Directory listing bypass vectors.
  - Redirection to attacker-supplied URLs (stored and reflected).
  - Attacker-supplied embedded content (stored and reflected).
  - External untrusted embedded content.
  - Mixed content on non-scriptable subresources (optional).
  - HTTP credentials in URLs.
  - Expired or not-yet-valid SSL certificates.
  - HTML forms with no XSRF protection.
  - Self-signed SSL certificates.
  - SSL certificate host name mismatches.
  - Bad caching directives on less sensitive content
- **Websecurify** [DR.13]
  - Cross Site Scripting (XSS)
  - SQL injection
  - Carriage Return and Line Feed (CRLFI ou HTTP Response Splitting)
  - LFI
  - Directory Listing
  - System Path disclosure vulnerabilities
  - Information leakage

Uma das ferramentas mais utilizada pelos profissionais de segurança é as suites integradas para testes de penetração. Estas suites integradas geralmente são disponibilizadas através um Live CD que contempla uma série de ferramentas bastante utilizadas e acreditadas no domínio da segurança e da análise de vulnerabilidades. Sendo que a descrição de cada uma das ferramentas encontradas nestes Live CD seria demasiado extensa, será aqui enunciada qual a distribuição que será utilizada no âmbito deste trabalho, a qual é:

- Backtrack 4 [DR.14]

Os *web scanners* aqui enunciados são ferramentas automáticas que procuram vulnerabilidades a nível aplicacional, aplicando uma série de ataques pré-programados

após a análise da aplicação. Estes ataques procurando explorar quais as vulnerabilidades existentes nas aplicações Web, sendo possível depois definir medidas que mitiguem os problemas encontrados.

Resultados a Apresentar

O relatório final a ser apresentado deverá ser detalhado em todas as acções que foram efectuadas a fim de descobrir as vulnerabilidades (**caso estas existam**). Deverão também constar no relatório as acções possíveis de ser efectuadas para a mitigação das vulnerabilidades, levando ao aumento da segurança e qualidade das aplicações Web testadas.

Os resultados a apresentar no relatório final deverão incluir as seguintes secções:

- Relatório sumarizado
  Esta secção destina-se a fornecer uma visão de alto nível dos testes efectuados. Deverá possuir informações sem grande nível de detalhe mas que permitam ter uma visão geral sobre o resultado da auditoria ao nível de segurança das aplicações testadas. Deverão fazer parte desta secção os seguintes tópicos:
  - Introdução
    - Descrição de alto nível do objectivo e metodologia dos testes
    - Descrição das aplicações testadas
  - Resultado final
    - Resultados por aplicação web testada
    - Resultados elucidativos do panorama geral
- Relatório detalhado
  Esta secção pretende entrar em profundidade nos testes efectuados e nos resultados obtidos, permitindo a compreensão de todos os elementos envolvidos nos testes. Deverão então ser apresentadas todas as informações relativas aos processos de testes, assim como as medidas de mitigação para as vulnerabilidades detectadas. Esta secção deverá possuir os seguintes tópicos:
  - Lista da totalidade das ferramentas utilizadas
  - Listagem da totalidade das vulnerabilidades encontradas
  - Listagem de acções de mitigação para cada vulnerabilidade (quando aplicável)
    - Para cada ferramenta
      - Vulnerabilidades encontradas
      - Impacto das vulnerabilidades
      - Mitigação das vulnerabilidades
      - Output dos testes da ferramenta
      - Inspecções manuais efectuadas para confirmação da vulnerabilidade (quando aplicável)
        - Ferramentas utilizadas neste nível
        - Resultado da inspecção manual

Planeamento do Trabalho

O trabalho será realizado fora da rede interna da organização, salvo raras excepções, previamente acordadas. Os testes serão executados em janelas temporais previamente definidas e acordadas com a entidade tendo em conta os períodos críticos para esta.

Alguns dos testes são extremamente morosos em termos de execução, devido essencialmente a dois factores importantes: a dimensão do site a testar e a complexidade do mesmo, assim como a complexidade e profundidade dos testes a realizar. Estes dois factores podem contribuir para que o trabalho exaustivo de identificação de vulnerabilidades em aplicações Web possa ser bastante alargado no tempo.

Grande parte deste trabalho, apesar de ser moroso, é automatizado pelas ferramentas de análise, que funcionam quase na sua maioria de forma independente do seu operador. A intervenção do operador é apenas necessária para analisar os resultados recolhidos e para aferir alguns desses mesmos resultados com alguns testes mais localizados.

Identificação das aplicações alvo da auditoria

A equipa de testes compromete-se apenas a realizar testes de segurança aplicacional nas aplicações Web identificadas pela organização.

Protecção dos dados da auditoria

A equipa de auditoria compromete-se a assegurar a protecção e confidencialidade dos dados de todo o processo de auditoria.

O relatório, resultados das análises das ferramentas, resultados de análises manuais e qualquer outro tipo de informação referente à organização será armazenado num dispositivo com encriptação AES em hardware, onde qualquer tentativa de força bruta levará à eliminação permanente dos dados, protegendo-os em qualquer situação.

Dado o enquadramento do projecto numa tese de mestrado, é a natureza pública da mesma, é de salientar que **todos os resultados serão ofuscados, não só no seu conteúdo, como na sua natureza, impedindo ambos a exibição dos mesmo, e o possível mapeamentos dos mesmos com qualquer organização.**

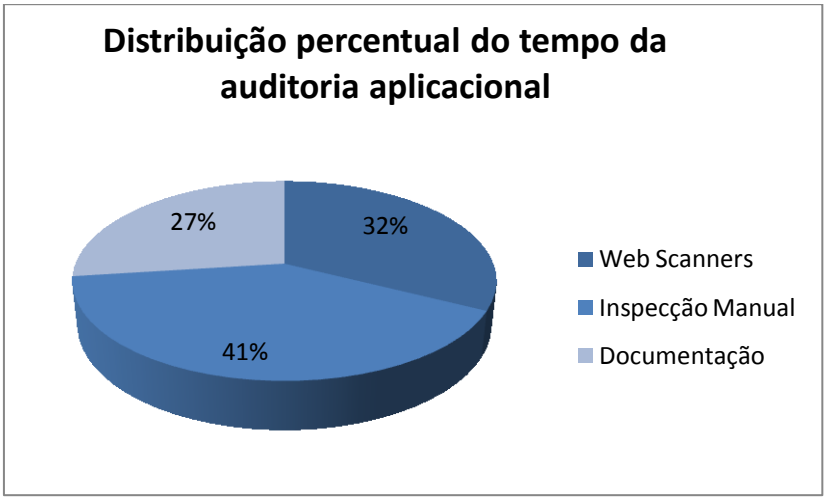
Distribuição Temporal

O planeamento do trabalho descrito nesta secção foi feito tendo em conta o desconhecimento do tempo de *scanning* que determinadas ferramentas poderão ter em cada aplicação. Sendo assim, foi feita uma estimativa do tempo que se acha necessário, sendo que qualquer operação que caia fora das janelas temporais definidas nunca será executada sem que antes se entre em contacto com a entidade, garantindo sempre o conhecimento por parte desta, das operações que estão a ser efectuadas, e quando, nos seus sistemas. Esta divisão temporal permite ainda o despiste de ataques reais que sejam efectuados contra a organização, para que não sejam confundidos com os testes em curso, e a organização possa tomar as devidas precauções.

Em termos de definição de datas teremos então uma distribuição representada na tabela em baixo:

Início dos testes	<data de recepção dos URLs>
Fim dos testes	+6 dias
Início da inspecção manual	<fim dos testes automáticos>
Fim da inspecção manual	+25 dias
Início da documentação de resultados	<fim da inspecção manual>
Fim da documentação de resultados	+5 dias

No que respeita à percentagem de tempo necessário para a auditoria aplicacional, apesar de poderem existir algumas alterações na alocação temporal, a sua distribuição percentual no tempo manter-se-á semelhante à indicada no seguinte gráfico.



A realização dos testes com *web scanners* poderá representar alguma carga nos servidores da entidade. Apesar de esta carga ser considerada mínima, de modo a procurar não causar qualquer tipo de impacto nos servidores, os testes considerados mais profundos serão realizados num horário acordado com a entidade.

Antes do início dos testes, caso seja do interesse e necessidade da organização, poderá ser assinado um *non-disclosure agreement* (NDA). Mesmo no caso de não ser assinado um NDA, a equipa de testes e elementos envolvidos (orientador da tese de mestrado) comprometem-se a não divulgar qualquer informação relativa aos resultados dos testes, ou que possa ser alvo de associação entre os mesmos e a entidade.

Os contactos utilizados entre as duas entidades estão indicados na tabela abaixo.

Equipa de auditoria aplicacional	
Nome	Nuno Filipe Martins da Silveira Teodoro
E-mail	nuno.filipe.teodoro@gmail.com
Telefone	919370496

Orientador da tese de Mestrado	
Nome	Carlos Serrão
E-mail	carlos.serrao@iscte.pt
Telefone	-

Validações a efectuar pela organização

É então esperado que a organização em causa efectue uma validação dos pontos enunciados neste documento, com especial atenção para as datas propostas para a realização dos testes, assim como uma definição do horário para a realização dos mesmos, o qual poderá ser enviado por e-mail para a equipa de auditoria.

Acções de Emergência

Sempre que algum evento inesperado aconteça, a auditoria será de imediato suspensa e serão contactados os responsáveis da organização, onde lhes será fornecida uma explicação da ocorrência. Analogamente, sempre que alguma acção suspeita e fora do âmbito do definido neste projecto seja detectada por parte da organização, esta deverá entrar em contacto de imediato com a equipa de auditoria aplicacional, a fim de perceber se tais eventos foram causados por esta.

Anexo B – Exemplo de *Non-disclosure Agreement*

TERMOS DE RESPONSABILIDADE E CONFIDENCIALIDADE

A EQUIPA DE AUDITORIA, VEM DECLARAR, QUE SE OBRIGA AO CUMPRIMENTO DO DEVER DE SIGILO E CONFIDENCIALIDADE DAS INFORMAÇÕES QUE TENHA CONHECIMENTO NO ÂMBITO DA REALIZAÇÃO DE TESTES DE AUDITORIA DE VULNERABILIDADES DE APLICAÇÕES WEB ACORDADO COM A ENTIDADE, E QUE SE OBRIGA, TAMBÉM, A NÃO TRANSMITIR, POR QUALQUER MEIO, PARA O EXTERIOR, NEM UTILIZAR EM PROVEITO PRÓPRIO OU DE TERCEIROS, ESSA INFORMAÇÃO CONFIDENCIAL E ESSES DADOS.

O SIGNATÁRIO ACEITA TODAS AS RESPONSABILIDADES PELAS CONSEQUÊNCIAS DE QUALQUER FALHA NOS SISTEMAS DA ENTIDADE, CAUSADA PELAS ACTIVIDADES POR SI DESENCADEADAS.

OBRIGA-SE AINDA A ACORDAR PREVIAMENTE E A PLANEAR COM A ESTRUTURA DA ENTIDADE QUALQUER ACTIVIDADE QUE PRETENDA DESENCADEAR COM EVENTUAIS CONSEQUÊNCIAS PARA ESTA ORGANIZAÇÃO, E A OBTER AUTORIZAÇÂO PARA A REALIZAÇÃO DA MESMA, ACAUTELANDO QUE NÃO HAJA DEGRADAÇÕES NO DESEMPENHO E RENDIMENTO DOS SISTEMAS E DOS ROUTERS E FIREWALLS ASSOCIADOS.

NO CASO DE ALGUNS DADOS PODEREM SER TEMPORARIAMENTE ALTERADOS E DOS SISTEMAS FICAREM TEMPORARIAMENTE INDISPONÍVEIS (“HANG” OU “CRASH”) COMO RESULTADO DAS VULNERABILIDADES TESTADAS, O SIGNATÁRIO DISPONIBILIZA-SE PARA COLABORAR COM A ENTIDADE PARA IDENTIFICAR OS MEIOS NECESSÁRIOS PARA A SUA RECUPERAÇÃO.

O SIGNATÁRIO OBRIGA-SE AINDA A NÃO DIVULGAR O RESULTADO DOS TESTES REALIZADOS, SEM O PRÉVIO CONSENTIMENTO POR ESCRITO DA ENTIDADE.

Lisboa, 14 de DEZEMBRO de 2010

Anexo C – Aplicação do *WASSEC* aos *web scanners* identificados

			Grendel-Scan		Paros Proxy		Skipfish		w3af		Websecurify		RatProxy	
Section Number	Section Title	Criticality Level	Support Level	Score - Do Not Edit	Support Level	Score - Do Not Edit	Support Level	Score - Do Not Edit	Support Level	Score - Do Not Edit	Support Level	Score - Do Not Edit	Support Level	Score - Do Not Edit
1	**Protocol Support**													
1,1	Transport Support													
1.1.1	HTTP 1.1	5	5	25	5	25	5	25	5	25	5	25	5	25
1.1.2	HTTP 1.0	5	5	25	5	25	5	25	5	25	5	25	5	25
1.1.3	SSL/TLS	5	5	25	5	25	5	25	5	25	5	25	5	25
1.1.4	HTTP Keep-Alive	5	5	25	5	25	5	25	5	25	5	25	5	0
1.1.5	HTTP compression	0		0		0		0		0		0		0
1.1.6	HTTP user agent configuration	0		0		0		0		0		0		0
Section Score			*100*		*100*		*100*		*100*		*100*		*75*	
1,2	Proxy Support													
1.2.1	HTTP 1.0 proxy	0		0		0		0		0		0		0
1.2.2	HTTP 1.1 proxy	0		0		0		0		0		0		0
1.2.3	Socks 4 proxy	0		0		0		0		0		0		0
1.2.4	Socks 5 proxy	0		0		0		0		0		0		0
1.2.5	PAC file	0		0		0		0		0		0		0
Section Score			*0*		*0*		*0*		*0*		*0*		*0*	
2	**Authentication**													
2,1	Authentication													

	Schemes													
2.1.1	Basic	0		0		0		0		0		0		0
2.1.2	Digest	0		0		0		0		0		0		0
2.1.3	HTTP negotiate	0		0		0		0		0		0		0
2.1.4	HTML Form-based													
2.1.4.1	Automated	0		0		0		0		0		0		0
2.1.4.2	Scripted	0		0		0		0		0		0		0
2.1.4.3	Non-Automated	0		0		0		0		0		0		0
2.1.5	Single sign on	0		0		0		0		0		0		0
2.1.6	Client SSL certificates	0		0		0		0		0		0		0
2.1.7	Custom implementations	0		0		0		0		0		0		0
Section Score			*0*		*0*		*0*		*0*		*0*			
3	**Session Management**													
3,1	Session Management Capabilities													
3.1.1	Start a new session	5	5	25	5	25	5	25	5	25	5	25	5	25
3.1.2	Session token refresh	5	5	25	5	25	5	25	5	25	5	25	5	25
3.1.3	Session expired	0		0		0		0		0		0		0
3.1.4	Reacquire session tokens	0		0		0		0		0		0		0
3,2	Session Management Token Type Support													
3.2.1	HTTP cookies	5	5	25	5	25	5	25	5	25	5	25	5	25
3.2.2	HTTP parameters	5	5	25	5	25	5	25	5	25	5	25	5	25
3.2.3	HTTP URL path	4	5	20	5	20	5	20	5	25	5	25	5	20
3,3	Session Token Detection Configuration													
3.3.1	Automatic session token detection	0		0		0		0		0		0		0
3.3.2	Manual session token configuration	0		0		0		0		0		0		0
3,4	Session Token Refresh													

	Policy													
3.4.1	Fixed session token value	0		0		0		0		0		0		0
3.4.2	Login process provided token value	0		0		0		0		0		0		0
3.4.3	Dynamic token value	0		0		0		0		0		0		0
Section Score		*0*		*0*		*0*		*0*		*0*		*0*		
4	**Crawling**													
4,1	Web Crawler Configuration													
4.1.1	Define a starting URL	5	5	25	5	25	5	25	5	25	5	25	5	25
4.1.2	Define additional hostnames (or IPs)	1	0	0	0	0	5	5	0	0	0	0	0	0
4.1.3	Define exclusions for													
4.1.3.1	Specific hostnames (or IPs)	0		0		0		0		0		0		0
4.1.3.2	Specific URLs or URL patterns	0		0		0		0		0		0		0
4.1.3.3	Specific file extensions	0		0		0		0		0		0		0
4.1.3.4	Specific parameters	0		0		0		0		0		0		0
4.1.4	Limit redundant requests	0		0		0		0		0		0		0
4.1.5	Supporting concurrent sessions	0		0		0		0		0		0		0
4.1.6	Specify request delay	0		0		0		0		0		0		0
4.1.7	Define maximum crawl depth	0		0		0		0		0		0		0
4.1.8	Training the crawler	0		0		0		0		0		0		0
4,2	Web Crawler Functionality													
4.2.1	Identify newly discovered hostnames	5	5	25	5	25	5	25	5	25	5	25	5	25
4.2.2	Support automated form submission	0		0		0		0		0		0		0

4.2.3	Detect error pages/custom 404 responses	0		0		0		0		0		0		0
4.2.4	Redirect Support													
4.2.4.1	Follow HTTP redirects	5	5	25	5	25	5	25	5	25	5	25	5	25
4.2.4.2	Follow meta refresh redirects	0		0		0		0		0		0		0
4.2.4.3	Follow JavaScript redirects	0		0		0		0		0		0		0
4.2.5	Identify and accept cookies	4	5	20	5	20	5	20	5	20	5	20	5	20
4.2.6	Support AJAX applications	4	5	20	4	16	5	20	5	20	5	20	5	20
Section Score			*115*		*111*		*120*		*115*		*115*		*115*	
5	**Parsing**													
5,1	Web Content Types													
5.1.1	HTML	5	5	25	5	25	5	25	5	25	5	25	5	25
5.1.2	JavaScript	5	5	25	5	25	5	25	5	25	5	25	5	25
5.1.3	VBScript	0		0		0		0		0		0		0
5.1.4	XML	0		0		0		0		0		0		0
5.1.5	Plaintext	4	5	20	5	20	5	20	5	20	5	20	5	20
5.1.6	ActiveX Objects	0		0		0		0		0		0		0
5.1.7	Java Applets	0		0		0		0		0		0		0
5.1.8	Flash	0		0		0		0		0		0		0
5.1.9	CSS	0		0		0		0		0		0		0
5,2	Character Encoding Support													
5.2.1	ISO-8859-1	0		0		0		0		0		0		0
5.2.2	UTF-7	0		0		0		0		0		0		0
5.2.3	UTF-8	0		0		0		0		0		0		0
5.2.4	UTF-16	0		0		0		0		0		0		0
5,3	Parser tolerance	0		0		0		0		0		0		0
5,4	Parser customization	0		0		0		0		0		0		0

5,5	Extraction of dynamic content	0		0		0		0		0		0		0
Section Score			*70*		*70*		*70*		*70*		*70*		*70*	
6	**Testing**													
6,1	Testing Configuration													
6.1.1	Host names or IPs	5	5	25	5	25	5	25	5	25	5	25	5	25
6.1.2	URL patterns	0		0		0		0		0		0		0
6.1.3	File extensions	0		0		0		0		0		0		0
6.1.4	Parameters	0		0		0		0		0		0		0
6.1.5	Cookies	0		0		0		0		0		0		0
6.1.6	HTTP headers	0		0		0		0		0		0		0
6,2	Testing Capabilities													
6.2.1	Authentication													
6.2.1.1	Brute Force													
6.2.1.1.1	Lack of account lockout	0		0		0		0		0		0		0
6.2.1.1.2	Different login failure message	0		0		0		0		0		0		0
6.2.1.2	Insufficient authentication	3	1	3	5	15	5	15	5	15	5	15	1	3
6.2.1.3	Weak password recovery	0		0		0		0		0		0		0
6.2.1.4	Lack of SSL on login pages	4	1	4	5	20	3	12	5	20	0	0	1	4
6.2.1.5	Auto-complete enabled on pass parameters	4	1	4	0	0	0	0	0	0	5	20	1	4
6.2.2	Authorization													
6.2.2.1	Credential/Session Prediction													
6.2.2.1.1	Sequential session token	0		0		0		0		0		0		0
6.2.2.1.2	Non-Random session token	0		0		0		0		0		0		0
6.2.2.2	Insufficient													

	Authorization													
6.2.2.2.1	Forcefully browse to "logged-in" URL	0		0		0		0		0		0		0
6.2.2.2.2	Forcefully browse to high-privilege URL	0		0		0		0		0		0		0
6.2.2.2.3	HTTP verb tampering	0		0		0		0		0		0		0
6.2.2.3	Insufficient session expiration	0		0		0		0		0		0		0
6.2.2.4	Session Fixation													
6.2.2.4.1	Failure to generate new session ID	0		0		0		0		0		0		0
6.2.2.4.2	Permissive session management	0		0		0		0		0		0		0
6.2.2.5	Session Weaknesses													
6.2.2.5.1	Session token passed in URL	0		0		0		0		0		0		0
6.2.2.5.2	Session cookie not set with secure attribute	0		0		0		0		0		0		0
6.2.2.5.3	Session cookie not set with HTTPOnly	0		0		0		0		0		0		0
6.2.2.5.4	Session cookie not sufficiently random	0		0		0		0		0		0		0
6.2.2.5.5	Site does not force SSL connection	0		0		0		0		0		0		0
6.2.2.5.6	Site uses SSL but references insecure objects	0		0		0		0		0		0		0
6.2.2.5.7	Site supports weak SSL ciphers	0		0		0		0		0		0		0
6.2.3	Client-side Attacks													
6.2.3.1	Content spoofing	0		0		0		0		0		0		0
6.2.3.2	Cross-Site Scripting													
6.2.3.2.1	Reflected cross-site scripting	5	4	20	4	20	4	20	4	20	5	25	4	20
6.2.3.2.2	Persistent cross-site	5	4	20	4	20	4	20	4	20	4	20	4	20

	scripting													
6.2.3.2.3	DOM-based cross-site scripting	0		0		0		0		0		0		0
6.2.3.3	Cross-frame scripting	0		0		0		0		0		0		0
6.2.3.4	HTML injection	5	0	0	0	0	0	0	5	25	0	0	1	5
6.2.3.5	Cross-site request forgery	5	0	0	0	0	2	10	0	0	0	0	2	10
6.2.3.6	Flash-Related Attacks													
6.2.3.6.1	Cross-site flashing	0		0		0		0		0		0		0
6.2.3.6.1	Cross-site scripting through flash	0		0		0		0		0		0		0
6.2.3.6.1	Phishing/URL redirection through flash	0		0		0		0		0		0		0
6.2.3.6.1	Open cross-domain policy	0		0		0		0		0		0		0
6.2.4	Client-side Attacks													
6.2.4.1	Format string attack	0		0		0		0		0		0		0
6.2.4.2	LDAP injection	4	0	0	0	0	0	0	5	20	0	0	0	0
6.2.4.3	OS command injection	5	0	0	0	0	4	20	5	25	0	0	0	0
6.2.4.4	SQL injection	5	5	25	5	25	5	25	5	25	5	25	5	25
6.2.4.4.1	Blind SQL injection	4	4	16	4	16	4	16	4	16	4	16	4	16
6.2.4.5	SSI injection	1	0	0	5	5	1	1	5	5	1	1	2	2
6.2.4.6	XPath injection	1	0	0	0	0	5	5	5	5	0	0	0	0
6.2.4.7	HTTP header injection/response splitting	0		0		0		0		0		0		0
6.2.4.8	Remote file includes	0		0		0		0		0		0		0
6.2.4.9	Local file includes	0		0		0		0		0		0		0
6.2.4.10	Potential malicious file uploads	0		0		0		0		0		0		0
6.2.5	Information Disclosure													
6.2.5.1	Directory indexing	4	5	20	5	20	5	20	5	25	5	25	5	0

6.2.5.2	Information Leakage												
6.2.5.2.1	Sensitive information in code comments	0		0		0		0		0		0		0
6.2.5.2.2	Detailed application error messages	0		0		0		0		0		0		0
6.2.5.2.3	Backup files	0		0		0		0		0		0		0
6.2.5.2.4	Include file source code disclosure	0		0		0		0		0		0		0
6.2.5.3	Path traversal	0		0		0		0		0		0		0
6.2.5.4	Predictable resource location	0		0		0		0		0		0		0
6.2.5.5	Insecure HTTP methods enabled	0		0		0		0		0		0		0
6.2.5.6	WebDAV enabled	0		0		0		0		0		0		0
6.2.5.7	Default web server files	0		0		0		0		0		0		0
6.2.5.8	Testing and diagnostics pages	0		0		0		0		0		0		0
6.2.5.9	Front page extensions enabled	0		0		0		0		0		0		0
6.2.5.10	Internal IP address disclosure	0		0		0		0		0		0		0
6,3	Testing Customization													
6.3.1	Modify existing tests			0		0		0		0		0		0
6.3.2	Create new tests			0		0		0		0		0		0
Section Score			*101*		*106*		*137*		*186*		*112*		*98*	
7	**Command and Control**													
7,1	Scan Control Capabilities													
7.1.1	Schedule scans	0		0		0		0		0		0		0
7.1.2	Pause and resume scans	0		0		0		0		0		0		0
7.1.3	Vew real-time status	0		0		0		0		0		0		0

7.1.4	Define re-usable scan configuration templates	0		0		0		0		0		0		0
7.1.5	Run multiple scans simultaneously	0		0		0		0		0		0		0
7.1.6	Support multiple users	0		0		0		0		0		0		0
7.1.7	Remote/distributed scanning	0		0		0		0		0		0		0
7,2	remote/distributed scanning													
7.2.1	Client application with GUI	0		0		0		0		0		0		0
7.2.2	Command line interface	0		0		0		0		0		0		0
7.2.3	Web-based interface	0		0		0		0		0		0		0
7,3	Extensibility and Interoperability													
7.3.1	Scan API	0		0		0		0		0		0		0
7.3.2	Integrates with bug-tracking systems	0		0		0		0		0		0		0
Section Score			*0*		*0*		*0*		*0*		*0*		*0*	
8.1.2	Technical Detail Report													
8.1.2.1	Full request and response data	5	5	25	5	25	5	25	5	25	5	25	5	25
8.1.2.2	List of all hosts and URLs	0		0		0		0		0		0		0
8.1.3	Delta Report	0		0		0		0		0		0		0
8.1.4	Compliance Report													
8.1.4.1	OWASP Top 10	0		0		0		0		0		0		0
8.1.4.2	WASC Threat Classification	0		0		0		0		0		0		0
8.1.4.3	SANS Top 20	0		0		0		0		0		0		0
8.1.4.4	Sarbanes-Oxley (SOX)	0		0		0		0		0		0		0

8.1.4.5	PCI DSS	0		0		0		0		0		0		0
8.1.4.6	HIPAA	0		0		0		0		0		0		0
8.1.4.7	GLBA	0		0		0		0		0		0		0
8.1.4.8	NIST 800-53	0		0		0		0		0		0		0
8.1.4.9	FISMA	0		0		0		0		0		0		0
8.1.4.10	PIPEDA	0		0		0		0		0		0		0
8.1.4.11	Basel II	0		0		0		0		0		0		0
8,2	Advisories For Each Unique Vulnerability Type													
8.2.1	Vulnerability description	0		0		0		0		0		0		0
8.2.2	CVE or CWE ID	0		0		0		0		0		0		0
8.2.3	Severity level	0		0		0		0		0		0		0
8.2.4	CVSS version 2 Score	0		0		0		0		0		0		0
8.2.5	Remediation guidance	0		0		0		0		0		0		0
8.2.6	Remediation code example(s)	0		0		0		0		0		0		0
8,3	Report Customization													
8.3.1	Add custom notes	0		0		0		0		0		0		0
8.3.2	Mark vulnerabilities as false positives	0		0		0		0		0		0		0
8.3.3	Adjust the risk level													
8.3.3.1	CVSS score	0		0		0		0		0		0		0
8.3.3.2	Severity level or other risk quantifiers	0		0		0		0		0		0		0
8.3.4	Report vulns according to content location	0		0		0		0		0		0		0
8.3.5	Ability to include customizations	0		0		0		0		0		0		0
8,4	Report Format													
8.4.1	PDF	0		0		0		0		0		0		0
8.4.2	HTML	0		0		0		0		0		0		0
8.4.3	XML	0		0		0		0		0		0		0

8,5	Vendor Feedback	0		0		0		0		0		0		0
Section Score			*25*		*25*		*25*		*25*		*25*		*25*	
9	**Custom Criteria** Use this section for custom criteria you may have for your organization													
9,1	Concordante com OWASP Top 10 2010	5	5	25	4	20	5	25	5	25	5	25	4	20
9,2	Actividade e atualizações recentes	5	0	0	0	0	5	25	5	25	5	25	3	15
9,3	Suporte a novas tecnologias	4	2	8	1	4	5	20	5	20	5	20	3	12
9,4	Facilidade de interacção com criadores	4	1	4	1	4	5	20	5	20	5	20	4	16
9,5	Apoio e suporte por parte de uma organização	4	0	0	0	0	5	20	5	20	5	20	5	20
Section Score			*37*		*28*		*110*		*110*		*110*		*83*	

Classificações finais	
Classificação	***Web scanner***
604	Grendel-Scan
620	Paros Proxy
734	Skipfish
791	w3af
717	Websecurify
622	RatProxy