Pós-graduação em Ciência da Computação

"ARAPONGA: Uma Ferramenta de Apoio a Recuperação de Informação na Web voltado a Segurança de Redes e Sistemas"

Thiago Gomes Rodrigues

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife, Março/2012

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PROGRAMA DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

THIAGO GOMES RODRIGUES

"ARAPONGA: UMA FERRAMENTA DE APOIO A RECUPERAÇÃO DE INFORMAÇÃO NA WEB VOLTADO A SEGURANÇA DE REDES E SISTEMAS"

DISSERTAÇÃO APRESENTADA AO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL À OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: Prof. Dra. Judith Kelner
CO-ORIENTADOR: Eduardo Luzeiro Feitosa

Recife, Março/2012

Catalogação na fonte

Bibliotecária Jane Souto Maior, CRB4-571

Rodrigues, Thiago Gomes

ARAPONGA: uma ferramenta de apoio a recuperação de informação na web voltado a segurança de redes e sistemas / Thiago Gomes Rodrigues. - Recife: O Autor, 2012.

xi, 69 folhas: il., fig., tab.

Orientador: Judith Kelner.

Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2012.

Inclui bibliografia e apêndice.

1. Redes de computadores. 2. Segurança de redes. I. Kelner, Judith (orientadora). II. Título.

0054.6 CDD (23. ed.) MEI2012 – 085

Dissertação de Mestrado apresentada por **Thiago Gomes Rodrigues** à Pós-Graduação em Ciênciada Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"ARAPONGA: Uma ferramenta de Apoio a Recuperação de Informação na Web voltado a Segurança de Redes e Sistemas",** orientada pela Profa. **Judith Kelner** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Paulo Romero Martins Maciel
Centro de Informática / UFPE

_____

Prof. Eduardo James Pereira Souto
Departamento de Computação / UFAM

_____

Profa. Judith Kelner
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 7 de março de 2012

_____

Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Agradecimentos

Primeiramente a Deus que por ter me agraciado com mais esta conquista e por ter me dado a oportunidade de passar momentos bons e ruins sempre me iluminando para conseguir aprender com cada situação vivida e poder dizer que nada como um dia após o outro.

Aos meus pais que sempre primaram pela minha educação, sempre tentando me entender e me direcionar para o caminho correto. À minha noiva, à minha irmã, cunhado e restante dos familiares, obrigado por todo apoio dado.

À minha orientadora a professora Judith Kelner, ao co-orientador Eduardo Feitosa e ao professor Djamel Sadok por terem me orientado, direcionado e cedido a infraestrutura do GRPT para que eu concluísse esta dissertação de mestrado. Sem a ajuda deles eu não teria conseguido.

A todos os meus amigos de trabalho e outros não mencionados que torceram pelo meu sucesso, obrigado.

Sumário

Lista de Figuras

Lista de Tabelas

Abreviações e Acrônimos

CVE – *Common Vulnerabilities and Explosures*

SA – *Security Alert*

SB – *Security Bulletins*

TA- *Technical Cyber Security Alert*

BGP – *Border Gateway Protocol*

ASN – *Autonomous System Number*

CVSS – *Common Vulnerability Scoring System*

OSVDB - *Open Source Vulnerability Database*

NDV - *National Vulnerability Database*

RDF - *Resource Description Framework*

Resumo

A área de segurança de redes de computadores e sistemas apresenta-se como uma das maiores preocupações atualmente. À medida que o número de usuários de computadores aumenta, cresce no número de incidentes de segurança.

A falta de comportamentos voltados à segurança, no que se refere a uso de hardware, e-mails ou configuração de programas são fatores facilitam a implantação de códigos maliciosos. O impacto da exploração de vulnerabilidades ou de falhas de softwares tem aumentado gradualmente e causado enormes prejuízos ao redor do mundo. A divulgação destas vulnerabilidades e boas práticas de segurança têm sido uma das soluções para este problema pois permitem que administradores de redes e sistemas consigam adquirir informações relevantes para mitigar o impacto de uma atividade maliciosa.

Ao notar que divulgar informações de segurança é uma das saídas para combater as atividades maliciosas e também para diminuir o impacto de uma exploração bem sucedida, várias organizações resolveram publicar este tipo de conteúdo. Estas bases encontram-se espalhadas em diferentes sítios Web, o que faz com que equipes de administração de redes e sistemas demore muito tempo buscando informações necessárias para a resolução dos seus problemas. Além disto, a exposição do conteúdo não é um fator preponderante para a solução dos problemas. Baseado neste cenário, este trabalho de mestrado se propõe a criar um sistema de apoio à recuperação de informação na Web voltado à segurança de redes e sistemas.

Palavras-chave: segurança de redes, vulnerabilidades, atividades maliciosas, crawler, extração de informação

Abstract

The area of computer networks and security systems is presented as a major concern of businesses today. Along with the increasing number of computer users was also growth in the number of security incidents.

The lack of safe practices in the use of hardware, or configuration of email programs facilitate the deployment of malicious code. The exploitation of vulnerabilities or software failures have caused massive damage around the world. As a solution to this problem, the disclosure of vulnerabilities and security best practices that allow network administrators and systems able to acquire information relevant to mitigate the impact of a malicious activity can lead to an entity.

Noting that disclosing information security was one of the ways to combat malicious activities and also to lessen the impact of a successful exploitation, was created several sources of information dissemination. These bases are scattered across different Websites, which makes network management teams and systems take a long time searching for information needed to solve their problems. Furthermore, mere exposure is not a major factor in solving the problems. Based on this scenario, this dissertation aims to create a system to support information retrieval and network security systems.

Keywords: network security, vulnerabilities, malicious activities, crawler, information retieval

1 Introdução

Hoje em dia é impossível negar que a Internet se tornou o meio de comunicação mais usado no mundo, capaz de combinar diferentes tipos de informação em tempo real. Entretanto, nos últimos dez anos, administradores de redes, gerentes de tecnologia da informação (TI) e especialistas em segurança vem percebendo o aumento do tráfego não desejado, não solicitado e por muitas vezes ilegítimo, que têm entre seus objetivos o uso não autorizado ou a inutilização de serviços.

As perdas financeiras causadas por tais práticas ao redor do mundo são milionárias. Somente em 2006, nos Estados Unidos, o prejuízo dos provedores de Internet chegou perto dos U$ 245 milhões de dólares [1]. No Brasil, o número de incidentes registrados no Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.Br) [2], de Janeiro a Setembro de 2011, passou de 318.000 (trezentos e dezoito mil), dos quais os mais frequentes são negações de serviço (DoS), varreduras (scan) e worms. No mesmo período, o Centro de Atendimento à Incidentes de Segurança (CAIS) da Rede Nacional de Pesquisa (RNP) [3] registrou mais de 308.000 (trezentos e oito mil) incidentes.

Grande parte deste problema é gerado pela existência de vulnerabilidades em hardware e software. Partindo do pressuposto que nada é totalmente seguro e que o número de vulnerabilidades conhecidas cresce todos os dias, a forma mais simples e economicamente viável é a divulgação das informações sobre essas vulnerabilidades. Neste contexto, o uso de bases de informação e sítios Web contendo estatísticas, descrições de vulnerabilidades e anomalias e informações de segurança têm surgido como prática comum para divulgação de incidentes, além de ser empregada por ferramentas segurança que visam impedir ou mitigar os danos causados pela exploração dessas vulnerabilidades.

A notoriedade deste tipo de "solução" pode ser facilmente observada pelo número de sítios Web e bases, de acesso público ou privado, presentes na Internet. VulDa [5], Cisco Security Center [6], Team Cymru[4], National Vulnerability Database (NVD) [7], Open Source Vulnerability Database (OSVDB)[8], Dragonsoft Vulnerability Database [9], ThreatExpert [10], ShadowServer [11] e IBM Internet Security Systems [12] são alguns exemplos. Tipicamente, estas bases publicam, com certa periodicidade, informações de segurança referentes a spam,

vulnerabilidades, atualizações de software, servidores com comportamentos maliciosos, *botnets*, vírus, *malwares*, entre outras.

Contudo, existem alguns obstáculos na publicação de tais conteúdos como, por exemplo, o tempo de análise da veracidade de um incidente; a falta de um padrão estrutural único (com campos pré-definidos e ordenados) para que publicações, independente do domínio, tivessem a mesma estrutura; e, também, a falta de sincronismo entre as diferentes bases, o que ocasiona, em muitos casos, repetições e discrepâncias de informações.

A fim de proporcionar uma melhor forma de gerenciamento do conteúdo de segurança publicado nestas diversas bases e otimizar o tempo de busca a informações, este trabalho propõe um sistema de apoio à recuperação de informação na Web (do inglês *Web-based Information Retrieval Support System* - WIRSS), chamado ARAPONGA, capaz de integrar os conteúdos de segurança disponíveis em domínios especializados, fornecendo uma única e direta fonte de acesso às informações. Mais especificamente, fornece características para lidar com questões de interoperabilidade e o uso integrado de recuperação de informação e ferramentas de tomada de decisão.

1.1 Objetivos

Este trabalho tem por objetivo a construção de um sistema de recuperação de informações na Web capaz de proporcionar a integração de informações de segurança e dar suporte a pesquisas avançadas. Chamado de ARAPONGA, a solução proposta, focada em vulnerabilidades, ataques, estatísticas, entre outros, é capaz de auxiliar gerentes de TI, administradores de rede e especialistas em segurança na atividade de obtenção de informações necessárias para a solução e/ou mitigação de incidentes de segurança.

Especificamente, este trabalho pretende:

- Criar múltiplos meios de aquisição do conteúdo (*Web*, desktop e outros sistemas);
- Prover aos usuários uma ferramenta de aquisição de conteúdo de segurança precisa e confiável;
- Propor e implementar um modelo diferenciado para indexação de conteúdo.

1.2 Estrutura da Dissertação

O restante desta dissertação está organizado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica sobre recuperação de informação na Web, os sítios Web e bases de dados públicas que contém informações de segurança sobre vulnerabilidades, vírus e estatísticas de ações maliciosas na Internet. Além de mostrar algumas ferramentas para a busca de vulnerabilidades bem como um levantamento do estado da arte sobre o assunto.

O terceiro capítulo apresenta o projeto e a implementação da solução proposta descrevendo características, requisitos e detalhes do desenvolvimento.

O quarto capítulo contém algumas avaliações sobre os resultados obtidos com a solução proposta levando em conta desempenho, completude e corretude das consultas, além da definição de métricas como o tamanho da base, precisão do indexador e uma demonstração do que pode ser considerado o vocabulário das publicações de segurança.

Por fim, no quinto capítulo, tem-se as conclusões desta dissertação e os possíveis trabalhos futuros.

2 Fundamentação Teórica

Este capítulo aborda aspectos relevantes para o melhor entendimento desta dissertação. A seção 2.1 trata da fundamentação teórica sobre a área de recuperação de informação na Web, incluindo alguns trabalhos relacionados. A seção 2.2 apresenta uma breve descrição dos elementos necessários para se realizar recuperação de informação na Web e suas características. A seção 2.3 apresenta uma breve descrição sobre sítios Web e bases de informação de segurança, mostrando os mais importantes e sumarizando suas características. Por fim, a seção 2.4 apresenta uma discussão sobre os sítios analisados.

2.1 Recuperação de Informação na Web

A constante busca por inovações e avanços tecnológicos resultou na riqueza de dados e informações difundidas pela Internet, especialmente após o estrondoso sucesso da *World Wide Web* (WWW). Porém, com o rápido crescimento da informação e a facilidade de acesso, uma questão tornou-se fundamental: como encontrar informação útil para construir o conhecimento? A resposta ideal, fundamentada pelo conceito no qual a Web semântica [39] foi criada, é que a resposta pode variar de acordo com o contexto da consulta ou de acordo com o usuário. Contudo, a resposta mais realista é o uso de um Sistema de Recuperação de Informação ou (do inglês *Information Retrieval System* - IRS) [24].

IRS é um nome genérico dado a um grupo de ferramentas dedicadas às tecnologias de dados, como banco de dados, para a manipulação seletiva e recuperação de grandes coleções de informação em diferentes formatos. Um IRS investiga diferentes aspectos da informação como, por exemplo, representação, armazenamento, organização e acesso. Um pressuposto central sobre as técnicas de recuperação/busca de informação é que os usuários saibam exatamente o que procuram.

Autores como Baeza-Yates e Ribeiro-Neto[24] e Yao [25] consideram um IRS como uma extensão ou evolução de um sistema de busca (do inglês *Data Retrieval System* - DRS). A razão é simples: ambos são focados apenas na funcionalidade de busca. Contudo, Yao afirma em [25]

que DRS lidam com desafios bem definidos, estruturados e problemas simples enquanto os IRS possuem desafios não tão bem definidos, dados semiestruturados ou sem nenhuma estrutura e tentam resolver problemas não tão simples. Fazendo uma analogia, um DRS funciona como um sistema de banco de dados que retorna exatamente o que foi consultado enquanto os IRS investigam alguns aspectos da informação como representação, armazenamento organização e acesso [40].

Mesmo sendo a evolução natural da área de recuperação de informação, juntamente com o crescimento da Web, os IRS passaram por problemas devido aos princípios e conceitos aos quais foram concebidos. Estes problemas, que foram citados no trabalho de Yao [25], são causados pela mudança das funcionalidades de busca e armazenamento, que antes eram executados apenas em documentos estruturados com o auxílio do usuário e, com a evolução, passaram a ser realizadas em modelos de dados (já que os documentos deixaram de ser estruturados) e sem a interação com o usuário. Em outras palavras, a tarefa de encontrar informações relevantes em documentos não estruturados requer algoritmos e estratégias mais complexas.

A fim de resolver este problema, Yao [25][26] propôs mudar o foco do IRS para o usuário, ou seja, o foco deixou de ser centrado no sistema e passou a ser centrado no usuário. Esta mudança culminou no que hoje é conhecido como Sistema de Suporte à Recuperação de Informação (do inglês *Information Retrieval Support System* - IRSS) [25]. O objetivo principal de qualquer IRSS é prover suporte a usuários dando a ferramenta necessária e linguagens que facilitem a tarefa de encontrar informações úteis que possam ser gerenciadas pelos seus usuários. Em outras palavras, um IRSS tem como foco principal a funcionalidade de suporte ao usuário ao invés de se preocupar com as funcionalidades de recuperação de informação.

Em um IRSS, os usuários têm um papel mais ativo e importante em comparação com as atividades exercidas nos sistemas predecessores. Eles podem, por exemplo, tomar decisões e encontrar informações úteis em vários níveis e estágios do processo de recuperação da informação. Com a pesquisa e navegação exploratória, os usuários podem determinar a relevância de cada item de informação. Além do mais, em algumas ocasiões, o usuário pode não desejar saber detalhes sobre os dados e sim apenas uma visão geral antes de fazer uma análise mais aprofundada. Ao contrário dos IRS que apresentam o resultado das buscas de forma

ranqueada, um usuário de um IRSS pode desejar usar uma forma gráfica para o retorno da busca, aumentando assim o nível da inferência e análise.

No entanto, no contexto da Web, o foco no usuário levanta algumas questões. Isto é especialmente pelo fato do usuário não saber exatamente o que está procurando, na medida em que bilhões de páginas da Web são atualizadas diariamente. Desta forma, os IRSS voltados a Web são conhecidos como Sistema Web de Suporte à Recuperação de Informação (do inglês *Web Information Retrieval Suport System* - WIRSS). De acordo com Hoeber [27], um WIRSS aplica métodos inteligentes e avançados de tecnologias baseadas na Web sobre o foco tradicional na busca automatizada em coleções de dados digitais, a fim de oferecer aos usuários uma informação mais específica do que eles necessitam, avaliando e explorando os resultados da busca e gerenciando as informações recuperadas.

Por exemplo, na resposta para a questão: Quantas vezes a Seleção Brasileira masculina ganhou a Copa do Mundo da FIFA? Ao longo dos resultados em busca na Internet, observariam-se relatos que a Copa de 2014 será no Brasil, que o Brasil participou da Copa da África e outras respostas, inclusive de sítios que não levaram em conta o início da frase. Por outro lado, um WIRSS com suporte a busca semântica retornaria a resposta cinco (5).

Dois exemplos, o Google Mapas[28] e o AllinOneNews [29], podem ilustrar estas funcionalidades na Web. O primeiro é um WIRSS heterogêneo que recebe consultas dos usuários em forma de endereço e retorna um mapa de alta resolução com opções de visualização do terreno em diferentes perspectivas, além da possibilidade de visualização do fluxo de tráfego nas vias em tempo real e imagens da vizinhança no navegador Web. O segundo é uma espécie de jornal com um motor (*engine*) baseado em meta-busca que integra uma fonte de origem de informações homogêneas onde a consulta do usuário é disparada para selecionar (baseado na consulta) um subconjunto de 10 a 20 mais prováveis fontes de uma lista de 1800 fontes de notícias. O resultado é misturado e ranqueado antes de ser apresentado [29].

No entanto, o desenvolvimento e implantação de um WIRSS introduz novos desafios como precisamente apontado por [27][30]. A principal delas é a evolução dos Motores de Busca (do inglês *Search Engines* – SE), baseados nos conceitos dos clássicos IRS, para os Motores de Suporte à Busca (do inglês *Search Support Engine* – SSE), focados em proverem diferentes funcionalidades de suporte a usuários finais. Autores como Zeng et al. [30] e Marchionini e

White [31] afirmam que os SSE devem oferecer, além da navegação típica e tradicional, funcionalidades voltadas ao apoio dos usuários como conhecimento da organização, descoberta e visualização. Para provar e avaliar tal abordagem, eles desenvolveram um SSE em camadas focado no gerenciamento de um conjunto de dados DBLP [32], conhecido como DBLP-SSE.

Outro desafio é a representação do resultado de uma busca Web. A típica representação que lista todos os resultados é bastante eficiente quando a informação que está sendo procurada é bem específica. Porém, quando a consulta criada pelos usuários é mal-definida, vaga ou ambígua, a lista provê poucos documentos e a maioria sem relevância. Tilsner et al. [33] implementaram um protótipo de busca Web baseada em agrupamento e visualização fuzzy, chamada *CubanSea*, capaz de prover uma nova representação dos resultados de uma busca, permitindo aos usuários terem um papel ativo no processo de busca através da seleção dos documentos nos agrupamentos fuzzy, conseguindo assim, a redução dos resultados não relevantes.

Recentemente, uma abordagem diferente foi proposta por Marchionini e White [31]. Eles introduziram o conceito de Sistema de Suporte à Busca de Informação (do inglês *Information Seeking Support System* – ISSS), que enfatiza a necessidade de deslocamento do foco do estudo de busca da informação para o apoio a busca. Os autores argumentaram que a busca de informação para o aprendizado, tomada de decisão, e outras atividades mentais similarmente complexas acontecem por repetidos períodos de tempo, e por isso requerem o desenvolvimento de novas soluções e serviços de suporte que ajudem os usuários a gerenciar, analisar e compartilhar o conhecimento adquirido.

Os ISSS atuais abrangem uma gama de funcionalidades. Shah [34] desenvolveu um framework, chamado *ContextMiner*, capaz de executar de forma automatizada a navegação em várias fontes Web e coletar dados, bem como informações contextuais. O *ContextMiner* pode analisar e agregar valor aos dados e seus contextos e continuamente os objetos digitais que possam interessar aos usuários ao longo do tempo. WolframAlpha [35] é conhecido como o mais famoso motor de busca semântica. Ele é capaz de responder a consultas diretamente pelo calculo do resultado a partir de documentos estruturados em vez de fornecer um grande número de ponteiros para documentos ou páginas Web, como é o caso dos típicos motores de busca existentes. *Relation Browser* [36][37] provê uma interface dinâmica que permite que usuários

possam explorar um conjunto de dados através do uso de uma navegação lapidada e busca por palavras-chave. Desenvolvida como um Applet Java, é focado na compreensão entre os itens em uma coleção e na exploração dos espaços da informação (por exemplo, um conjunto de documentos ou páginas Web).

2.2 Elementos para Recuperação de Informação na Web

Recuperação de Informação é uma área de pesquisa dedicada a tecnologias para manipulação e recuperação de grandes coleções de informação em diferentes formatos de apresentação. Tipicamente, investiga formas de representação, armazenamento, organização e acesso a itens de informação de modo a permitir ao usuário fácil acesso à informação na qual está interessado através de consultas. Entretanto, quando a necessidade por informações é aplicada na realização de atividades como aprendizado, tomada de decisão e outras atividades mentais complexas que ocorrem ao longo do tempo, a recuperação é necessária, mas não suficiente [42]. A solução é mudar essa busca de informações dos motores de busca que fornecem itens discretos como respostas as consultas para ferramentas e serviços que suportem pesquisas interativas e reflexivas ao longo do tempo e do modo colaborativo.

Esta subseção faz uma caracterização da área de recuperação de informação, apresenta os conceitos envolvidos, discutindo os desafios presentes e, por fim, descreve alguns trabalhos cujos resultados são encorajadores para a pesquisa nessa área.

2.2.1 Web Crawlers

Os *Web Crawlers*, também conhecido como *robots, ant, spider, wanderers, walkers, knownbots* ou *bot*, são programas responsáveis por percorrer a Web e baixar (download) páginas a serem usadas por sistema de busca [41].

Normalmente, um *Web Crawler* inicia o processo de navegação na Internet com um grupo inicial de URLs armazenado em uma estrutura de dados chamada *seeds*. A medida que acessa a URL, o *Web Crawler* faz o download da página pertencente a essa URL e analisa todas as URLs encontradas nessa página afim selecionar e armazenar essas novas URLs na lista de páginas a visitar. Esse processo é repetido até satisfazer a condição de parada do *Web Crawler*.

Contudo, seu funcionamento enfrenta três importantes problemas: o grande número de páginas, a velocidade com que estas páginas são atualizadas e, com o advento da Web 2.0, a geração de páginas dinâmicas. Uma vez que a Web possui uma grande quantidade de páginas, um *Web Crawler* pode apenas baixar uma pequena porção, o que faz com que seu funcionamento seja norteado pelo estabelecimento de prioridades relativas ao que baixar. Já a rápida atualização de conteúdo aumenta a probabilidade do *Web Crawler* baixar conteúdo desatualizado. Por fim, a geração dinâmica de páginas diminui o número possível de combinações do que baixar, visto que páginas dinâmicas não têm HTML como conteúdo e sim referências da estrutura dinâmica.

O comportamento de um *Web Crawler* é baseado em uma série de políticas de implementação que visam melhorar seu rendimento. Estas políticas estão relacionadas ao comportamento do *Web Crawler* quando está em ação como, por exemplo, que links visitar primeiro, o que fazer quando encontrar uma página já baixada, se vai executar em paralelo ou se vai seguir as políticas criadas pelo robots.txt[1] de cada domínio [43]. O uso de políticas permite a classificação dos *Web Crawlers* em três tipos: *Retricting Followed Links*, *Path-ascending crawling* e *Focused Crawler*.

Restricting Followed Links

Restricting Followed Links é um tipo de *Web Crawler* que busca somente links nas páginas HTML [46]. Basicamente, este tipo tenta encontrar o máximo de referências possíveis usando estratégias como, por exemplo, procurar apenas por URLs que terminam com .html, .htm, .asp, .aspx, .php ou com "/".

Existem muitos empecilhos neste tipo de abordagem, uma vez que uma escolha errada na estratégia de mapeamento dos links das páginas pode levar a requisições infinitas de páginas. Um exemplo acontece com URLs que tem em seu nome o símbolo "?", um claro indicativo de que o conteúdo é construído dinamicamente.

Este tipo de *Web Crawler* é bastante usado para verificar se os links das páginas continuam funcionando.

[1] Robots.txt são arquivos criados em sítios Web para controlar as ações de dos robôs (*robot*) de busca, ditando seu comportamento no domínio.

Path-Ascending Crawling

Path-Ascending Crawling é um tipo de *Web Crawler* que busca encontrar todos os recursos de um determinado sítio Web [46]. Basicamente, utiliza o link passado como referência e tenta extrair o máximo de páginas navegando pelos diretórios da URL. Supondo que a URL http://www.cin.ufpe.br/~tgr/arquivos/mestrado/index.html seja passada, o *Web Crawler* procurará arquivos no index.html, www.cin.ufpe.br/~tgr/arquivos/mestrado/, www.cin.ufpe.br/~tgr/arquivos/, www.cin.ufpe.br/~tgr/ e, por fim, www.cin.ufpe.br/.

Este tipo de *Web Crawler* pode ser usado quando se deseja transferir todo o conteúdo de um site.

Focused Crawling

Focused Crawling é um tipo de *Web Crawler* que busca páginas que tenham conteúdo inserido dentro de um tópico ou vários tópicos previamente determinados [50][51]. Podem ser usadas em seu funcionamento abordagens que usam apenas os nomes dos links para decidir se vão baixar a página ou não, bem como abordagens que usam uma medida de similaridade entre o conteúdo do HTML das páginas baixadas com os conteúdos das páginas ainda não visitadas para decidir se baixa ou não a página.

2.2.2 Mecanismos de Buscas

Mecanismos de Busca, também chamados de *Search Engines*, são aplicações utilizadas para buscar grande quantidade de informações na Web. Tipicamente, tais buscas são realizadas via Web browsers, onde, após a requisição inicial o mecanismo busca as informações utilizando técnicas particulares e retorna as referências a documentos que melhor satisfazem a consulta.

Tradicionalmente, mecanismos de busca são projetados de forma modular visando isolar atividades e funções específicas. De modo geral, o primeiro módulo de um mecanismo de busca é o módulo de coleta de páginas (*crawler*), responsável por navegar pela Web e montar um repositório com as páginas visitadas e selecionadas. O próximo módulo é o de indexação. Tipicamente executado após a finalização da coleta pelo *crawler*, analisa o conteúdo de cada página armazenada no repositório, cria um conjunto de palavras-chave (índice) que identificam o conteúdo da página e associa, em um banco de dados, a URL na qual cada palavra-chave

ocorre. Os métodos de indexação variam de acordo com a utilidade e as técnicas aplicadas por cada mecanismo de busca. Finalmente, os módulos de consulta e ranking (ordenação) recebem as requisições de usuários e as processam para retornar, de maneira ordenada, os documentos que melhor satisfazem essas requisições pelas consultas que foram processadas pelo módulo de consulta.

A subseção seguir apresenta os módulos básicos de um mecanismo de busca, exceto pelo o módulo de *crawler* que foi apresentado e discutido anteriormente.

Módulo de Indexação

Índices são descritos como palavras cuja semântica representam o principal assunto do documento. Sendo assim, a indexação de informação realizada neste módulo corresponde à representação de informações de páginas Web por termos de índice.

Entre as técnicas de indexação mais utilizadas em mecanismos de busca na Web encontram-se:

a) *Inverted files* - um mecanismo de indexação orientado a palavras, o qual armazena as diferentes palavras encontradas no texto e suas ocorrências;

b) *Suffix arrays* - tratam o conteúdo textual dos documentos como uma única cadeia de caracteres (*string*) e cada posição da palavra como um termo de índice;

c) *Signature files* - um mecanismo de indexação orientado a palavras manipuladas em tabelas de tipo *hash*. O texto é dividido em blocos de palavras e a cada bloco é aplicado uma função *hash* cujo resultado será o identificador desse bloco (*signature*).

Em [44] Kobayashi e Takeda apresentam algumas das principais características e funcionalidades de módulos de indexação utilizadas pelos mecanismos de busca. São elas:

- *Indexação manual ou humana*: especialistas no conteúdo a ser indexado organizam e compilam os diretórios e os índices da maneira que facilite as consultas. Por essa razão esse tipo de indexação é ainda considerado o mais preciso de todos os métodos.;

- *Indexação inteligente ou baseada em agentes*: são compostas por "agentes" computacionais que selecionam páginas, indexam-as, criam índices e armazenam as informações importantes para posterior recuperação da informação.;
- *Indexação baseada em metadados, RDF e anotação*: a indexação é feita considerando exclusivamente metadados.

Módulo de Busca e Ordenação

O módulo de busca e ordenação está extremamente relacionado com o modo com que as páginas foram indexadas, uma vez que nem todos os tipos de busca podem ser usados em qualquer sistema. Uma consulta passada a um mecanismo de busca é conhecida como *query* e representa a necessidade de informação do usuário. Uma consulta pode ser baseada em:

- **Palavras-Chave:** permite o ordenamento das respostas segundo a função de relevância adotada pelo mecanismo de busca. Pode ser construída baseada em palavras isoladas, baseada no contexto ou com junções booleanas. Seu objetivo é recuperar todos os documentos que contêm ao menos uma das palavras da consulta e em seguida, os documentos recuperados são ordenados e retornados ao usuário.
- **Casamento de Padrão:** permite realizar o "casamento" com strings ao invés de apenas palavras isoladas. Estas consultas podem ter um padrão simples (quando é apenas uma palavra, um prefixo, um sufixo, substring ou intervalo) ou um padrão complexo (que pode ser uma expressão regular). O objetivo deste tipo de consulta é encontrar documentos que contêm segmentos de texto que casam com o padrão da consulta e, para realizar tal tipo de busca, a lista de índices invertidos não é suficiente para uma recuperação eficiente.
- **Estrutura:** permite ao usuário realizar buscas a campos específicos das páginas. Por exemplo, um usuário que deseja procurar por páginas que no título aparece "Vulnerabilidade" recebe do mecanismo de busca somente páginas que contêm a string "Vulnerabilidade" em seu título.

Embora não sejam considerado um módulo, vários mecanismo de busca utilizam repositórios para armazenar as páginas manipuladas. Segundo Arasu et al. [45], esses repositórios devem possuir as seguintes características:

- **Método duplo de acesso às informações armazenadas**: acesso randômico, para ser usado rapidamente pelo módulo de busca e acesso por fluxo, para ser usado por indexadores para processar e analisar as páginas em volume;
- **Manipulação de grande volume de atualizações**, pois esses repositórios devem ser capazes de adicionar, de atualizar, e de reorganizar facilmente informações enviadas por *crawlers*;
- **Controle de páginas obsoletas**, pois um repositório deve ser capaz de detectar e remover páginas que não são utilizadas;
- **Escalabilidade**, durante a distribuição de repositórios através de clusters de computadores e de unidades de armazenamento distintas.

2.3 Sites de Vulnerabilidade

Esta seção tem como principal objetivo descrever os sítios Web estudados. Um maior detalhamento das informações observadas pode ser encontrado no Apêndice.

2.3.1 OSVDB

O *Open Source Vulnerability Database* (OSVDB) [8] é uma base de acesso gratuito, criada com o objetivo de fornecer à comunidade de segurança informações precisas, atualizadas, detalhadas e imparciais sobre vulnerabilidades.

OSVD possui mais de 77.000 relatos de vulnerabilidades e mais de 4.700 pesquisadores para fornecer uma atualização diária da base. Cada vulnerabilidade possui um identificador único, chamado OSVDB ID, formado apenas de números, incrementados a cada nova adição de vulnerabilidade. Este repositório pode ser acessado de várias formas. A mais usual e direta é via sítio Web (http://www.osvdb.org), onde podem ser feitas consultas de diferentes modos como, por exemplo, pelo identificador (OSVBD ID), pelo conteúdo da página, pelo "criador" do produto, entre outras. Outra opções de acesso incluem a cópia do repositório nos seguintes formatos: XML, CVS, MySQL e SQlite.

Porém, apesar de existirem milhares de pesquisadores ao redor do mundo ajudando a manter esta base, existem publicações que não possuem seus campos completamente preenchidos. Este problema já foi detalhado no trabalho de Borba [14], onde uma solução também foi apresentada.

2.3.2 US-CERT

O *United States Computer Emergency Readiness Team* (US-CERT) é uma organização governamental, mantida pelos Estados Unidos, que publica informações sobre vulnerabilidades, *exploits*, atualização de software, notificações de atividades maliciosas e práticas de segurança. Estas publicações acontecem periodicamente e são divididas em três categorias: Alertas Técnicos (*Techinical Alerts*) [15], para atender usuários que desejam conhecer detalhadamente os nuances técnicos; os boletins (*Bulletins*)[16], para usuários que desejam apenas obter um resumo das publicações; e os alertas (*Alerts*) [18], para os usuários que desejam obter informações detalhadas sem tanto conteúdo técnico.

O US-CERT organiza também uma base de vulnerabilidades chamada KB-CERT [17]. Criada em 2000, contém mais de 2800 relatos de vulnerabilidades. É uma base pública e aceita relato de qualquer pessoa, sendo somente publicado após a validação dos relatos.

Para todas as publicações controladas pelo US-CERT, todos os campos da publicação possuem informações.

2.3.3 NDV

A *National Vulnerability Database* (NDV) [7] é uma base de dados mantida pelo *National Institute of Standards and Technologies* (NIST), que contém mais de 49.000 publicações sobre vulnerabilidades, com uma média diária de 10 novas inserções.

A NDV é uma base sincronizada com o CVE (*Common Vulnerabilities and Exposures*) e que utiliza os mesmos nomes do CVE em suas publicações. Além desta ligação, ela fornece um sumário para todas as vulnerabilidades contidas no CVE, ligações externas a outros sistemas e correções (*patches*). Além desta sincronização, o NDV provê um esquema de ranqueamento das vulnerabilidades baseado no nível de periculosidade que ela representa. Este ranqueamento é formado por valores que variam de 0 (zero) a 10 (dez), onde quanto mais próximo de zero menos severo é o resultado da exploração da vulnerabilidade e quanto mais próximo de dez mais severo

é o impacto da exploração da vulnerabilidade. Este escore é chamado de *Common Vulnerability Scoring System* (CVSS) e provê a possibilidade de uma análise quantitativa das vulnerabilidades [15].

2.3.4 CISCO Security Center

O *Security Intelligence Operations* é uma base mantida pela empresa CISCO [6] e foi criada no intuito de informar, proteger e responder seus clientes sobre as ameaças e vulnerabilidades. Além destas publicações, a CISCO provê soluções que ajudam a proteger e mitigar as ameaças na rede. Esta base possui mais de 4.900 (quatro mil e novecentas) publicações que referenciam as bases de vulnerabilidades da CVE e da Security Focus (BugTrack Id).

2.3.5 Team Cymru

O *Team Cymru* [4] é uma organização norte americana, sem fins lucrativos, mantida por um grupo de pesquisadores que tem como motivação principal fazer da Internet um ambiente seguro. Este grupo é formado por pessoas que trabalham com e muito próximo de comunidades de segurança. A base mantida pelo *Cymru* contem informações sobre BGP *(Border Gateway Protocol),* estatísticas sobre ASN (*Autonomous System Number*) e malwares, além de provê serviços de mapeamento de IP para ASN, registro de *malware*, entre outros.

2.3.6 Dragonsoft Vulnerability Database

A Dragonsoft [9] possui uma base de vulnerabilidades e alertas de software compatível com a CVE. A Dragonsoft é uma empresa criada em 2001 com foco no desenvolvimento e pesquisa de produtos que avaliam as vulnerabilidades presentes, possui mais de 4.500 relatos de vulnerabilidade e está disponível também em Chinês.

2.3.7 ThreatExpert

ThreatExpert [10] é uma base de segurança que contém informações sobre *malware* e *adware*. Esta base serve como repositório para uma aplicação desktop que procura ameaças no computador. Esta base está disponível também para acesso Web, onde todo o conteúdo pode ser obtido detalhadamente.

2.3.8 ShadowServer

A fundação Shadowserver [11] foi criada em 2004 com o objetivo de entender e ajudar a por um fim no crime via Internet. A fundação conta com inúmeros profissionais de segurança que trabalham voluntariamente para manter a base atualizada com estatísticas sobre ASN, *malware*, *scan*, DDoS, vírus, entre outras.

2.3.9 Secunia

A Secunia [13] foi criada em 2002 a partir de capital privado de empresas interessadas em criar uma base sobre publicações de segurança em softwares. A empresa conta com mais de 150 empregados de mais de 22 nacionalidades diferentes para manter o repositório atualizado e desenvolver aplicações de segurança em diferentes linguagens de programação para diferentes ambientes de execução.

2.3.10 ATLAS

O ATLAS (*Active Threat Level Analysis Network System*) [69] é um sistema, desenvolvido pela Arbor Networks, que disponibiliza uma grande variedade de informações de ameaças a sistemas de computação. Tais informações contêm: resumos de ameaças, ranking de ataques da Internet, índice de riscos de vulnerabilidades, índice de ameaças e um mapa global de ameaças atualizado em tempo real. Estas informações são atualizadas a cada 24 horas com completude e sempre estão ligadas a uma referência CVE.

Contudo, parte do conteúdo do ATLAS é restrito, somente acessado por usuários cadastrados. Para confecção deste trabalho, uma conta de acesso foi concedida gentilmente e gratuitamente.

2.4 Discussão

Visando melhorar a compreensão sobre os sites Web apresentados, esta dissertação exibe, na Tabela 2.1, um comparativo entre elas. Para tanto, algumas características interessantes e comuns a todas serão tomadas como base:

- **Quantidade de informações registradas**: permite mensurar a quantidade de URLs que poderão visitadas por um sistema de busca de informações e também ajuda a mensurar o espaço necessário para indexar todo o conteúdo.
- **Tipo de acesso**: indica se o acesso as informações é simples, direto, possível de diferentes modos. Para um sistema de busca esta métrica é interessante porque indica a necessidade da criação ou utilização de mecanismo de autenticação para acesso do conteúdo.
- **Atualização**: indica a periodicidade de atualização da base de informações. Esta métrica é importante porque permite mensurar o intervalo de tempo no qual um sistema de busca deve visitar as páginas de um referido domínio.
- **Completude**: indica se as informações registradas estão completas ou não. Esta métrica pode servir de indicador do grau de confiabilidade e usabilidade das informações contidas nessa base.
- **Uso de padrões**: permite identificar se os conteúdos estão seguindo algum tipo de padrão para divulgação das informações. Esta métrica é interessante porque permite que sites que com o mesmo "perfil" sejam tratados de forma semelhante.

Tabela 2.1: Comparação entre os sítios Web avaliados.

	Número de Publicações	Acesso Gratuito	Taxa de Atualização	Completude	Uso de padrões	Permite crawlers
OSVDB	>58000	Total (Web/BD)	Diária	Parcial	Próprio (OSVDB ID)	Sim
Secunia	>35000	Parcial (Web)	N/I	Total	Próprio	Não
US-CERT	>2500	Total (Web)	N/I	Total	Próprio	Sim
NDV	>45000	Total (Web)	Diária	Total	CVE	Sim
Atlas	N/I	Parcial (Web)	Diária	Total	Próprio	Não
Shadow Server	N/I	Total (Web)	Diária	Total	Não	Parcialmente
Team Cymru	N/I	Total (Web)	Diária	Total	Não	Não
Threat	N/I	Total	N/I	Total	Não	Parcialmen

Expert		(Web)				te
Dragon Soft	N/I	Total (Web)	N/I	Total	CVE	Sim

Em avaliação preliminar dos cinco sítios focados em vulnerabilidades (OSVDB, Secunia, US-CERT, DragonSoft e NVD), no que se refere ao número de publicações, é natural afirmar que o OSVDB é o melhor representante, uma vez que disponibiliza suas informações de forma gratuita, em diferentes formatos, com atualizações diárias. Contudo, apesar de realmente conter o maior o número de publicações (quantidade de vulnerabilidades) registradas, o aspecto completude das informações age como fator limitante. As analises feitas por Borba [14] mostram que, em 05 de Junho de 2009, a base do OSVDB continha 54004 registros dos quais apenas 12407 estavam completos (restando 41597 registros incompletos), o que é uma taxa muito alta.

Embora apresentem um menor número de publicações, as bases da Secunia, US-CERT, DragonSoft e NVD possuem seus atrativos. A US-CERT, NVD e DragonSoft permitem acesso gratuito às suas bases, cuja completude é visível em qualquer consulta. Contudo, a Secunia, DragonSoft e US-CERT não indicam claramente o tempo de atualização de suas bases. Além disso, a base da empresa Secunia é a única entre as cinco cujo acesso é limitado.

Entre os sítios descritos, os únicos voltados para divulgação de resultados e estatísticas de anomalias ocorridas na Internet são: ATLAS, Team-Cymru e ShadowServer. Assim como o Secunia, grande parte de suas informações do ATLAS só podem ser acessadas por usuários cadastrados além de não permitir a coleta de conteúdo por indexadores por exemplo, *crawlers*. Em compensação suas informações são atualizadas a cada 24 horas. A completude também é um dos pontos fortes do ATLAS. O Team-Cymru possui acesso gratuito Web, várias atualizações diárias porém não permite que seu conteúdo seja obtido por *crawlers*. Já o ShadowServer possui atualizações diárias, sua base Web permite acesso gratuito e permite que *crawlers* possam obter o conteúdo.

Dentre os sítios listados há apenas um voltado à publicação de Malware e Adware que é o ThreatExpert. Sua base Web é gratuita e o conteúdo pode ser obtido por meio de *crawlers*.

Apesar das informações estarem publicadas na Web, não significa que o conteúdo pode ser obtido e muito menos publicado em outros domínios. Para isso existem as políticas de privacidade e uso das instituições e também existem regras que devem, obrigatoriamente, ser seguidas pelos indexadores de conteúdo automatizados. Estas regras vão desde instruções de não

obtenção do conteúdo que estão presentes nos arquivos robots.txt e nas meta-tags até divulgação de sumários das páginas no formato RDF (*Resource Description Framework*) para que indexadores voltados àquele tipo de conteúdo possam encontrar mais facilmente os documentos.

Os fatores preponderantes na escolha das bases de dados foram: a permissividade ou não a obtenção de conteúdo por indexadores (*crawlers*) e também a política de privacidade e uso das informações de cada base. Como pode ser visto na Tabela 2.1, as bases da Secunia, Atlas e TeamCymru não permitem obtenção de conteúdo por *crawlers* e, portanto, não foram estudadas detalhadamente. As demais bases, não possuem ou possuem poucas restrições sobre a aquisição de conteúdo de forma automatizada e as políticas de privacidade e uso fazem apenas a exigência que as páginas sejam devidamente referenciadas.

3 Solução Proposta e Implementação

Este capítulo descreve a solução proposta nesta dissertação e sua implementação. Primeiro, uma descrição geral da solução é apresentada em detalhes, incluindo a ideia dos Templates. Em seguida, cada um dos componentes (módulos) do protótipo será explicado e, por fim, o processo de funcionamento e integração entre os módulos será detalhado.

3.1 Solução Proposta

O ARAPONGA foi projetado com o objetivo de minimizar o tempo para a aquisição de conteúdos de segurança e por isto concentra informações sobre vulnerabilidades (em software, hardware e sistemas), ataques, *botnets*, *spam*, entre outras atividades maliciosas em uma base única, o que permite a utilização dessas informações por operadores humanos (administradores de redes e sistemas, gestores de TI, especialistas em segurança) e/ou outros sistemas em menos tempo que se comparado a sistemas de busca com informações mais amplas como o Google, por exemplo. Na prática, ARAPONGA pode ser encarado como uma solução de software capaz de, baseada em consultas gerais ou estruturadas (diferenciadas), exibir um conteúdo focado para a área de segurança de redes e sistemas.

A ideia central é utilizar os conceitos de recuperação de informação na Web para extrair o todo o conteúdo das páginas coletadas, que contenham informações de vulnerabilidades e atividades maliciosas, aumentando a precisão de consultas e diminuindo o tempo de procura por este tipo de informação.

Semelhante aos tradicionais WIRSS, o ARAPONGA, apresentado na Figura 3.1, é composto por quatro módulos: coleta, indexação/adequação, busca e ordenação e interface.

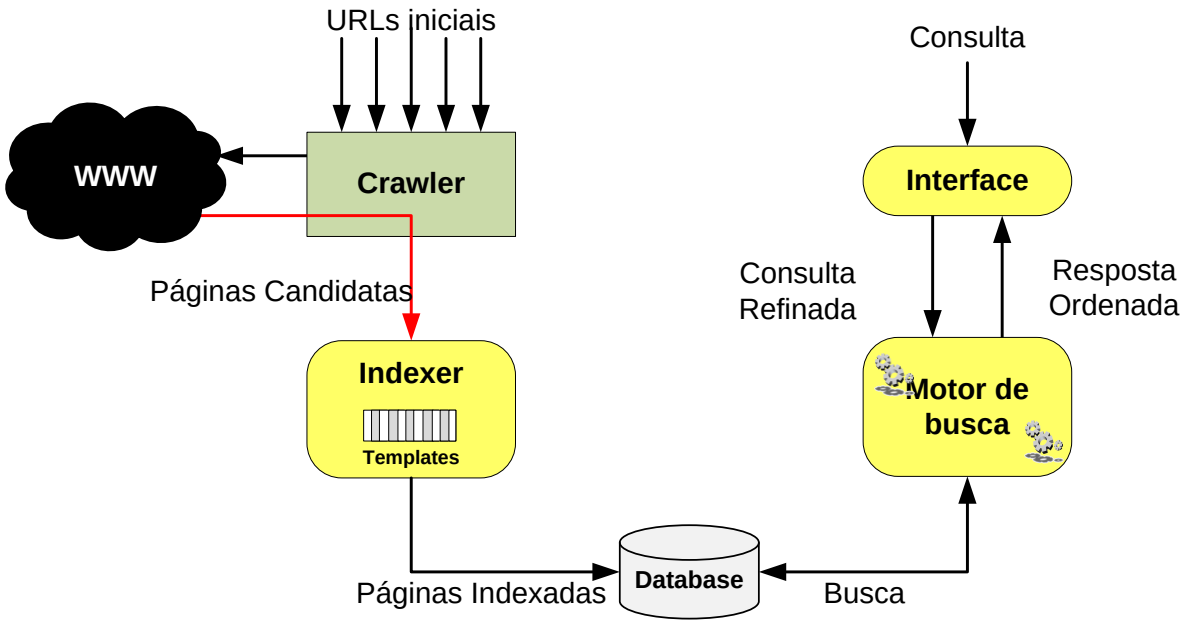

Figura 3.1: Visão Geral do ARAPONGA

- **Coleta:** Por se referir a um sistema em que todas as informações da base de dados são de sítios Web, a coleta é feita por uma ferramenta *crawler* cujas rotinas para coleta de documentos são implementados eficientemente e também respeitando as políticas de bom comportamento de robôs de busca (políticas das *meta-tags* e arquivo *robots.txt*, por exemplo). Outra característica desejada pela coleta é a possibilidade de autenticação, além do controle do fluxo da coleta com parâmetros de URLs iniciais, quantidades de *links* por página coletados (limitadores de amplitude) e limite de quantidade de *links* visitados (limitadores de profundidade).
- **Indexação e Adequação:** Permite a identificação do conteúdo extraído (palavras-chave, *timestamp*, *hash* do conteúdo), eliminação de páginas com o mesmo conteúdo e a indexação apenas de páginas com conteúdos relevantes.
- **Busca e Ordenação:** Permite ao usuário a possibilidade de realizar diferentes tipos de busca, sempre ordenadas de forma decrescente de ranking para que os documentos mais relevantes para a consulta sejam exibidos primeiro.
- **Interface:** É o meio com o qual o usuário interage com o sistema. Por se tratar de um WIRSS, este módulo deve ser capaz de permitir que o usuário tenha acesso às informações de diferentes modos. Além das consultas comuns em sistemas de busca Web (que retornam links), há a possibilidade de fazer consultas que restrinjam o espaço amostral do resultado, consultas que geram XML, consultas com os campos de resposta selecionados pelo usuário, exibição visual da estrutura da base de dados e interfaces de consulta via console e a por outros sistemas.

3.2 Componentes do ARAPONGA

Esta subseção tem como objetivo demonstrar detalhadamente cada módulo do ARAPONGA.

3.2.1 Módulo de Coleta

O Módulo de Coleta, também chamado de *crawler*, é responsável pela aquisição de páginas Web. Baseado em uma lista contendo URLs iniciais (focadas em sítios Web que divulgam informações de vulnerabilidades e estatísticas sobre ataques e anomalias Internet), o módulo busca em cada página visitada referências para outras páginas. Uma vez que problemas relativos à quantidade e qualidade da informação coletada são comuns nesse tipo de aplicação, o módulo utiliza limitadores de profundidade, que evitam a coleta indefinida de páginas a partir das *urls* iniciais; limitadores de amplitude, que restringem o número de *links* por páginas que podem ser referenciadas; e filtros de URL, consultados todas as vezes que uma nova página está para ser coletada a fim de evitar que páginas de domínios que não contenham as informações desejadas sejam coletadas.

3.2.2 Módulo de Indexação e Adequação

O Módulo de Indexação e Adequação (*Indexer*) recebe as páginas coletadas pelo módulo de busca, cria identificadores de conteúdo do documento e os adiciona à base de dados de conteúdo indexado (*index files*). A indexação é feita armazenando-se todo o conteúdo da página com o identificador principal "*content*" e outros identificadores como, por exemplo, a URL no campo "URL", a marca de tempo (*timestamp*) da página no campo "*tstamp*", entre outros campos que são usados no controle interno do indexador. Após estes passos, inicia-se a rotina de adequação do conteúdo, que visa melhorar a indexação (não indexando apenas pelo conteúdo das páginas).

Para tanto, faz-se uso de modelos (*templates*) para determinar quais partes de uma página devem ser indexadas com palavras-chaves diferentes, possibilitando assim, buscas diferenciadas. Estes *templates* são criados através de uma interface gráfica em duas etapas. Na primeira etapa o usuário escolhe o arquivo exemplo, as *tags* HTML (*h1, h2, h3, h4, title, div, table, TR e spam*) que separam os blocos de conteúdo a serem indexados e criam um identificador único para o *template*. Na segunda etapa o usuário escolhe, dentre os blocos de código extraídos, os que serão indexados e com quais palavras-chaves serão identificados.

Após o processo de identificação e escolha dos campos, o *template* acaba de ser criado e sua configuração é persistida em dois arquivos XML que foram criados para armazenar a configuração de todos os *templates* do ARAPONGA. Este processo de criação de *templates* feito graficamente é muito útil porque torna possível que usuários com pouco conhecimento em programação possam agregar valores aos seus sistemas de recuperação de informação ao manipular de forma detalhada das informações. Este módulo também executa a seleção de páginas que não serão indexadas porque não apresentam um conteúdo relevante na solução do problema.

3.2.2.1 Templates

O processo de extração e indexação de informação é o direcionador das funcionalidades de todo e qualquer sistema de recuperação de informação, de modo que quanto maior o controle sobre a informação, mais funcionalidades podem ser ofertadas.

A estratégia adotada para a criação do extrator e indexador de informação está diretamente ligada à estrutura dos documentos da base de dados. O extrator pode funcionar com classificação manual, semi-automatizada e totalmente automatizada. As estratégias de indexação manual são bastante efetivas porém com um baixo rendimento, uma vez que depende diretamente da intervenção humana. Na estratégia semi-automatizada o extrator de informação pode ser guiado por *templates* (construídos por humanos) ou por algoritmos inteligentes de reconhecimento de padrão. Na estratégia automatizada o extrator de informação é implementado através de algoritmos complexos de reconhecimento de padrão que podem ser supervisionados ou não.

Para este trabalho foi adotada uma abordagem semi-automatizada com avaliação de resultados supervisionada que faz uso de *templates* que pudessem variar de acordo com o tipo de página. Template se refere a um modelo de representação no conteúdo que será coletado e indexado. No contexto em questão, se refere à forma como está estruturado o conteúdo de uma página Web. Um template pode ser construído a partir do conteúdo (se for um documento estruturado) ou a partir da estrutura do documento (se for um conteúdo semi-estruturado).

3.2.3 Módulo de Interface

O Módulo de Interface é responsável pela comunicação entre usuários (humanos ou softwares) e o sistema. Neste módulo são definidas as regras para as consultas e para as respostas. Todas as

consultas são enviadas para o módulo de busca e ordenação, que retorna respostas ordenadas baseadas no ranking de cada página. Este módulo de interface apresenta três possibilidades de acesso entre os usuários e o ARAPONGA que são, uma interface gráfica (Web e desktop), uma interface de terminal e uma interface para outros sistemas.

3.2.4 Módulo de Busca e Ordenação

O Módulo de Busca e Ordenação é responsável por receber a consulta e retornar o objeto da consulta de forma ordenada. Ele é dividido em três sub-módulos: tradutor de consultas (*Query Parser*), ranqueamento (*Ranking*) e organizador (*Organizer*). O primeiro recebe consultas em linguagem natural oriundas do módulo de interface, as transforma em consultas aceitas pelo sistema (baseada no tipo de consulta e na consulta propriamente dita digitada pelo usuário), busca as informações na base de dados e repassa as páginas retornadas na consulta para o sub-módulo de ranqueamento. O sub-módulo de ranqueamento é responsável por quantificar a relevância da consulta em relação aos documentos retornados (levando em conta a posição da palavra procurada no documento, se tiver no título ou se a palavra estiver em negrito, aquele documento receberá maior nota que um documento cuja palavra aparece apenas no meio do conteúdo, por exemplo) e retorná-los para o módulo de interface para exibição. O sub-módulo organizador é responsável em organizar a resposta para o módulo de interface. A organização da resposta vai depender da natureza da consulta e do usuário. Como o ARAPONGA provê suporte a usuários humanos e outros sistemas, nas consultas executadas por usuários humanos, por exemplo, uma consulta pode retornar gráficos, lista de páginas ou XML, e exibidos em uma interface gráfica, enquanto para outros sistemas o retorno pode ser apenas um XML que deve ser interpretado pelo sistema gerador da consulta.

3.3 Funcionamento

Para tornar mais claro o processo de funcionamento do ARAPONGA, uma descrição completa de todo processo é exemplificada a seguir.

Em primeiro lugar é preciso entender que os módulos de coleta e indexação e adequação funcionam em conjunto, um após a execução do outro, e de forma *off-line*, ou seja, o processo desde a coleta a preparação da base aprimorada é realizado isolado, acontecendo todos os dias às 03:00 horas da manhã.

Em linhas gerais, quando um usuário ou outro sistema deseja efetuar uma consulta ao ARAPONGA, esta solicitação passa por uma série de etapas (módulos) até que haja o resultado à pergunta como pode ser observado pelo diagrama de atividades presente na Figura 3.2.

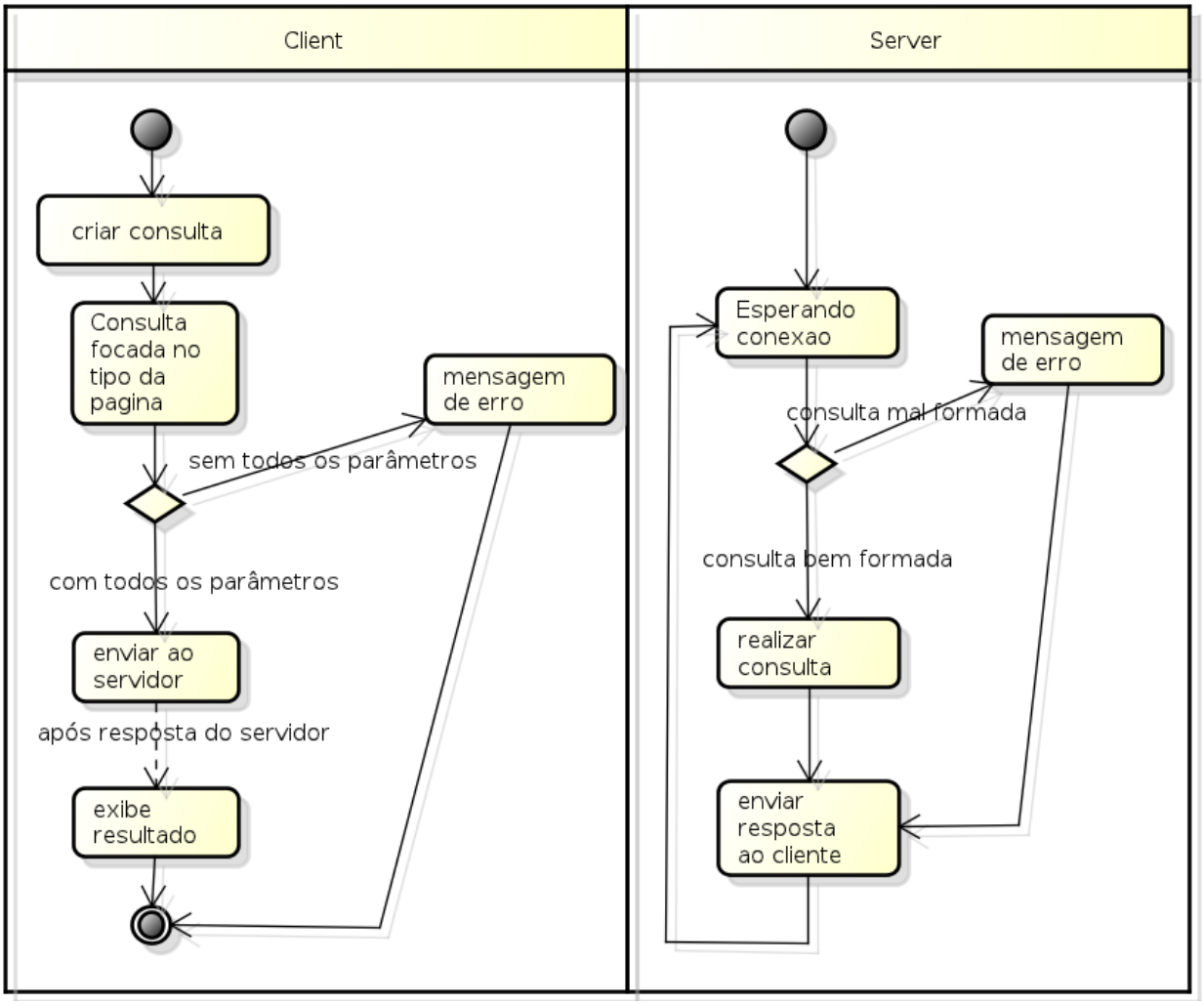

Figura 3.2. Diagrama de Atividades de uma consulta

Tomando como exemplo uma consulta referente a informações sobre ataques, *botnet*, vulnerabilidades, *spam* e boletins que envolvam o protocolo TCP na porta 80, a consulta gerada é a seguinte:

tcp/80 –focus Bulletin,Alert,Spam,Vulnerability,Attack,Botnet

A consulta é gerada via o módulo de interface, onde são analisados os números de parâmetros. Caso estejam de acordo com os tipos de consulta definidos, a consulta é enviada para o módulo de busca e ordenação.

No módulo de busca e ordenação, a consulta passa pelo processo de validação, onde são retiradas as *StopWords* e a consulta é então enviada para o motor de busca que fará a pesquisa nas páginas da base. A ideia é verificar quais páginas contém o valor descrito na consulta (no caso *tcp/80*) de acordo com o parâmetro especificado (*–focus Bulletin,Alert,Spam,Vulnerability,Attack,Botnet*). Após encontrar páginas que seguem estas regras, as mesmas são ranqueadas e é construída uma lista ordenada de modo decrescente de acordo com o valor do ranqueamento.

Por conseguinte, o resultado da busca é enviado para o módulo de interface que apenas a encaminha para o cliente.

3.4 Implementação

Antes iniciar a explicação sobre a implementação deste trabalho, se faz necessário esclarecer alguns pontos importantes e decisivos no projeto e desenvolvimento do ARAPONGA: a escolha do conteúdo e do *crawler*.

A escolha do tipo de conteúdo foi necessária para definir quais eram os sítios Web mais adequados à aquisição de informações sobre vulnerabilidades e estatísticas sobre tráfego e anomalias da Internet, visto que existem dezenas senão centenas de locais na Internet com este tipo de conteúdo. Após uma avaliação em termos de relevância e completude das informações, período de atualização e facilidade de acesso, foram definidos os seguintes sítios Web: Secunia, US-CERT (http://www.us-cert.gov), US-CERT (http://www.kb.cert.org), SecurityFocus, Nacional Vulnerability Database e Dragonsoft para boletins e relatórios de vulnerabilidades; ATLAS, ShadowServer e Team Cymru para estatísticas da Internet; e ThreatExpert para malware, trojans e outras ameaças.

A escolha do *crawler* que coleta as informações também teve que ser bem estudada, uma vez que se descobriu durante o projeto que alguns domínios, inclusive um dos escolhidos para a análise (ATLAS), apesar de possuírem acesso livre para coleta via robôs, certos conteúdos (informações extras sobre um determinado endereço IP envolvido em ataques DDoS, por exemplo) necessitam de autenticação. Desta forma, foi identificado que o software de *crawler* a ser usado no trabalho deveria ter como uma de suas características a capacidade de autenticação. Após uma avaliação preliminar, foram escolhidos três soluções para teste: WIRE [19], Heritrix

[20]e Nutch[21]. Desta forma, um estudo mais detalhado, incluindo até a instalação, foi executado e como resultado o *crawler* Nutch foi escolhido. A Tabela 3.1 ilustra essa avaliação.

Tabela 3.1: Comparativo entre os três Web crawlers testados.

Características	WIRE	Heritrix	Nutch
Instalação (Dificuldade)	Média	Alta	Baixa
Módulo de Autenticação	Não	Sim	Sim
Linguagem de implementação	C/C++	JAVA	JAVA
Integração direta com o Lucene	Não	Não	Sim

3.4.1 Módulo de Coleta

Para a implementação do módulo de coleta foi utilizado o *crawler* Nutch, conforme explicado na seção anterior. O Nutch, projetado e criado pela Apache, é uma mecanismo (*engine*) de busca Web que utiliza a biblioteca de busca Lucene para armazenar/buscar o conteúdo Web baixado. É um mecanismo de busca difundido em escala global por ter seu código fonte aberto, ganhando assim maior confiabilidade em relação a outros sistemas de busca.

Dentre as várias características do Nutch, pode-se citar a capacidade de:

- Localizar bilhões de páginas por mês;
- Manter o índice destas páginas;
- Pesquisar este índice mais de 1000 vezes por segundo;
- Prover resultados de alta qualidade;
- Operar com o menor custo possível.

Desenvolvido em JAVA, o Nutch apresenta simplicidade na atividade de modificação do seu código para adequação do comportamento as necessidades de coleta. O código é bem documentado.

Além das vantagens oferecidas, o Nutch obedece dois aspectos de "boa conduta" na área de recuperação de informação: respeita o que está publicado no arquivo robots.txt (presentes na raiz de cada domínio) e as META-TAGs[2] dos HTMLs das páginas.

[2] META-TAGs são palavras reservadas do HTML, "etiquetas", que entre outras coisas descrevem o conteúdo do sítio para os crawlers.

Como o objetivo era prover aos usuários um conteúdo mais completo de segurança da forma mais adequada, as políticas padrões estabelecidas por todos os Web *crawlers* que são considerados de "boa conduta" foram mantidas e os domínios que tinham alguma restrição sobre a aquisição de conteúdo de forma automatizada foram excluídos das URLs iniciais e caso algum link de alguma página coletada aponte para estes domínios os mesmos não serão coletados pelo *crawler*. Portanto, os domínios nos quais os conteúdos são coletados contém explicitamente em suas políticas de privacidade ou não contém em seus conteúdos nenhuma restrição sobre a aquisição/uso do conteúdo, desde que não seja para fins lucrativos (como é o caso) restando os domínios do TeamCymru, DragonSoft, ThreatExpert, Nacional Vulnerability Database, SecurityFocus e Shadowserver.

3.4.2 Módulo de Indexação e Adequação

A ferramenta escolhida para indexar o conteúdo coletado foi o Lucene [22]. Também desenvolvido pela Apache, é uma biblioteca de busca de texto de alto rendimento, com código fonte aberto e é recomendado para sistemas que precisam fazer buscas em textos completos.

Já para a parte de adequação foram utilizadas a API do Lucene e a biblioteca Jericho HTML parser [23] para extração do HTML dos conteúdos das páginas.

Em linhas gerais, o módulo de indexação e adequação é responsável por traduzir o conteúdo da base coletada e indexada (pelos dois módulos anteriores) para a base gerada pelo ARAPONGA. Seu funcionamento pode ser dividido em quatro etapas:

1. Consiste no uso da API do motor de busca do Lucene para a aquisição de todas as páginas baixadas e indexadas pelo *crawler* e o encaminhamento para a segunda etapa;
2. Consiste no processo de exclusão de páginas que estão relacionadas nos arquivos *excludedTags.xml* e *excludedUrls.xml* que são configuradas pelo administrador do ARAPONGA. Com o intuito de restringir a base do sistema, apenas com páginas que tenham conteúdos relevantes, passam pelo processo de exclusão, ou seja, apenas páginas que não estejam listadas em ambos os arquivos são encaminhadas para a próxima etapa;
3. Consiste na comparação das URLs de cada página com um conjunto de URLs pré-definidas, visando à identificação de modelos (*templates*) que ajudam a referenciar a página. Uma vez que um *template* é encontrado, a página tem seu conteúdo extraído e identificado e, cada bloco de informação tem associado a si uma palavra-chave. Caso

a página não tenha um *template* identificado, ela será apenas referencia pelo seu conteúdo. Após este processamento a página é encaminhada para a quarta etapa;

4. Consiste na adição dos identificadores de *timestamp*, *title*, *URL*, além dos identificadores de controle interno do sistema de busca/indexação como, por exemplo, um *hash* de todo conteúdo (para a fácil identificação de mudança de conteúdo).

Após estas quatro etapas, a página está pronta para ser indexada.

Criação de Templates

Com a análise realizada sobre a estrutura das páginas das bases citadas, notou-se uma variação do *template* no que se refere ao inter-domínio e, em alguns casos, no intra-domínio. Estas variações estruturais podem ser facilmente contornadas ao se construir, para cada grupo, um *template* específico. Para melhorar contextualizar as variações que ocorrem dentro do mesmo domínio, pode-se citar o caso das publicações da US-CERT sobre Alertas Técnicos [15] e Alertas Não-Técnicos[18], como pode ser observado na Figura 3.3.

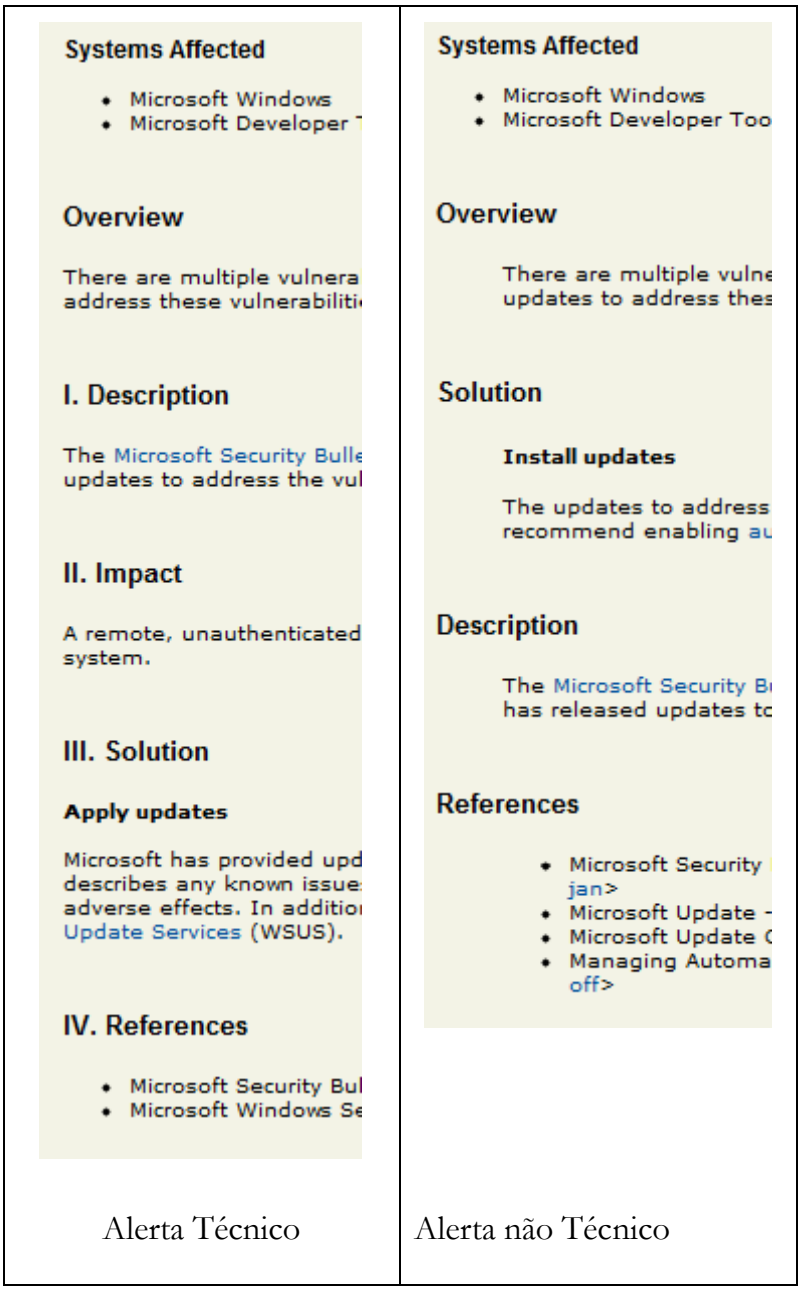

Figura 3.3. Diferenças entre Alerta Técnico e Alerta Não-Técnico

Percebe-se que na página da esquerda (Alerta Técnico) existem os campos *Description* e *Impact*, que não existem na página da direita (Alertas não Técnicos). Contudo, dentro do mesmo tipo de página (Alertas Técnicos e não Técnicos) não existe variação de estrutura, o que torna possível a construção de apenas um *template* para cada grupo de publicação.

Após fazer a combinação do conteúdo da página e sua palavra-chave, o documento é indexado. Após a indexação, o espaço amostral do conteúdo é subdividido em fatias menores

melhorando o tempo de resposta a consultas (pois não há a necessidade de buscar em todo conteúdo da página) e a precisão da resposta. Uma representação visual da diferença do espaço de busca no processo normal e na busca com *templates* é apresentada na Figura 3.4.

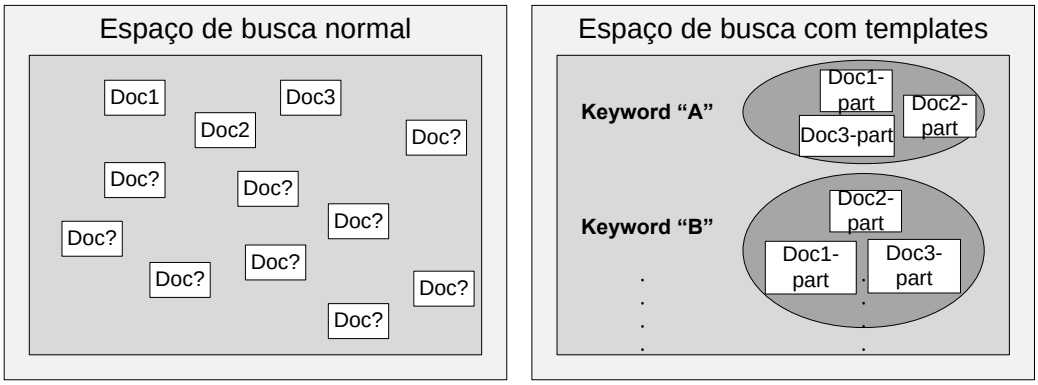

Figura 3.4. Diferenças entre espaços de busca

No intuito de deixar clara a ideia por trás dos *templates*, a Figura 3.5 ilustra um exemplo de *template* do grupo de páginas de Alertas Técnicos da US-CERT. O conteúdo textual selecionado por um retângulo corresponde a palavras-chave e o conteúdo selecionado por um retângulo de bordas arredondadas corresponde ao conteúdo indexado com a palavra-chave. Na Tabela 2.2 pode ser vista a descrição do significado de cada palavra-chave.

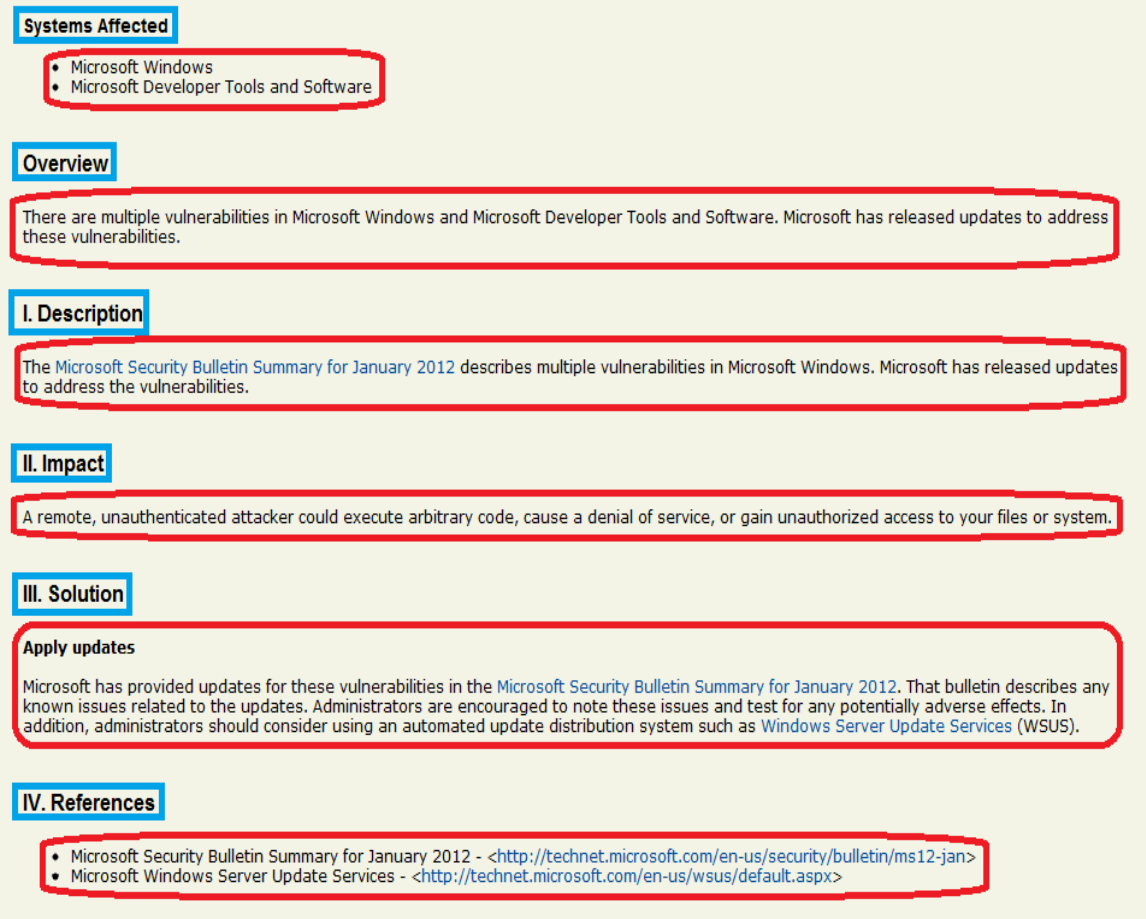

Figura 3.5. Funcionamento do Template para Alertas Técnicos

Tabela 3.2. Palavras-chave do Template de Alerta Técnico

Palavras-chave	**Descrição**
Systems Affected	Uma lista de todos os sistemas afetados pelo alerta.
Overview	Uma visão em linhas gerais sobre o que é o alerta.
Description	Uma descrição detalhada do alerta.
Impact	Impacto caso o alerta seja explorado por algum atacante.
Solution	Solução para o problema.
References	Referências para outros sites que estão relacionados ou com o problema ou com a solução.

O uso de *templates* permite que algumas funcionalidades de busca possam ser desenvolvidas, uma vez que consegue "reconhecer" que parte de texto deve ser indexado com sua referida palavra-chave. Por exemplo, se um usuário deseja saber informações sobre

vulnerabilidades no Microsoft Windows 7, uma busca normal resultaria em páginas contendo ameaças e vulnerabilidade, cujo um dos sistemas afetados é o Microsoft Windows 7, e páginas que apenas citam o Microsoft Windows 7. Com o uso dos *templates*, essa mesma consulta pode ser mais específica através de um parâmetro de busca (campo "sistemas afetados", por exemplo), onde somente seriam retornados os documentos que têm o Microsoft Windows 7 como sistemas afetados.

Como resultado da prática do uso de *templates*, tem-se a melhoria de desempenho e ganhos no tempo de resposta e processamento. É importante ressaltar que o processo de exibição dos resultados é o mesmo para uma consulta simples e avançada, onde os documentos encontrados são ranqueados, ordenados e exibidos. Outro fator que não pode ser deixado de mencionar é que uma página indexada com *template* também tem indexado todo o seu conteúdo e não somente os pedaços extraídos pelo *template*. O que pode ser notoriamente percebido pelo usuário é o número de páginas retornadas em uma consulta simples e uma avançada além de outros resultados que serão listados na seção de resultados.

3.4.3 Módulo de Busca e Ordenação

O módulo de busca e ordenação também foi implementado em Java e utiliza a API do Lucene para buscar as páginas que são relevantes à referida consulta. Basicamente, qualquer consulta (*query*) passa por um processo de eliminação de palavras (*StopWords*) e o resultado deste pré-processamento é enviado ao motor de busca do Lucene. Desta forma, inicia-se o processo de comparação da consulta com os documentos da base. Depois de efetuadas todas as comparações, os documentos são ranqueados e ordenados em ordem decrescente levando em conta o valor no qual o documento foi valorado.

3.4.4 Módulo de Interface

O módulo de interface fornece dois tipos de saída (ou interfaces). A primeira utiliza uma Interface Gráfica do Usuário (GUI) e é indicada para consulta dos operadores humanos do sistema (administradores de segurança e gerentes de TI, por exemplo). A Figura 3.6 ilustra a GUI de consulta. A segunda é operada via linha de comando e foi elaborada visando à consulta por outros sistemas como, por exemplo, um sistema de tomada de decisão querendo obter a indicação de que determinado endereço IP está envolvido em SPAM ou DNS fast-flux domain.

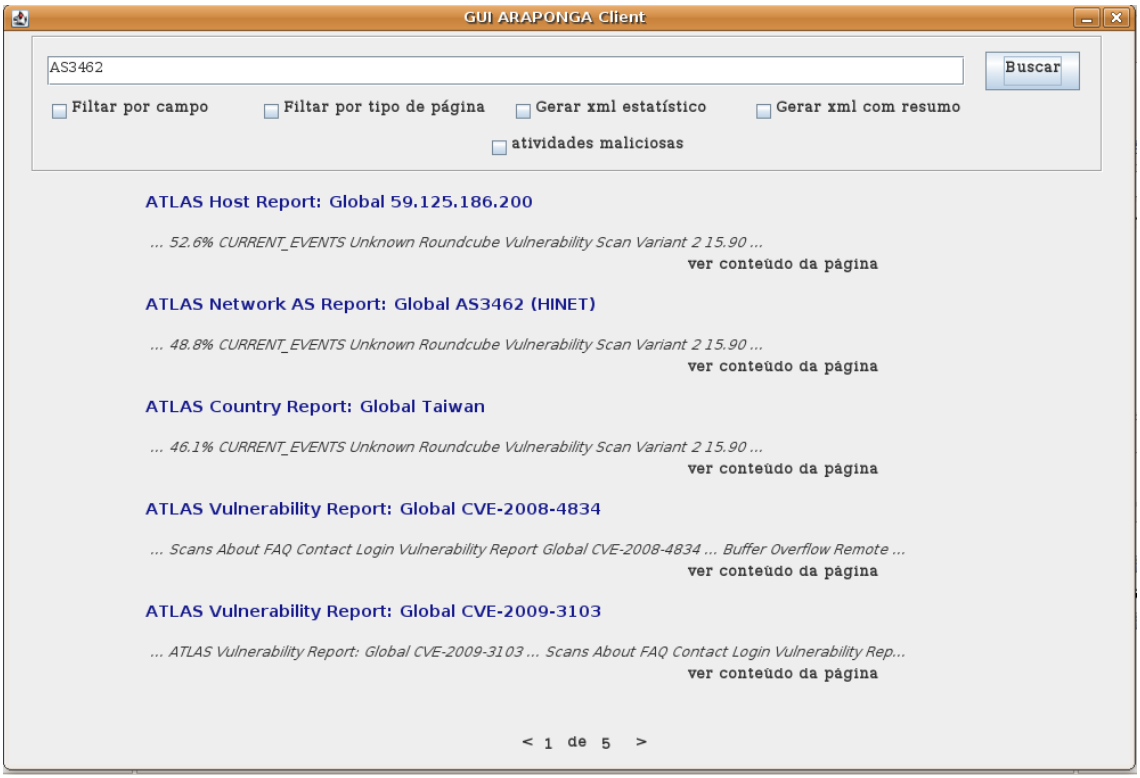

Figura 3.6. Exemplo da GUI de consulta.

Ambas as implementações foram desenvolvidas em Java, utilizando a API do Lucene para acessar os documentos indexados. Vale ressaltar que não há diferença de resultado e nem nos tipos de consulta que podem ser feitas usando o console ou a GUI.

Os tipos de consulta oferecidos pelo módulo de interface estão listados a seguir:

- **Geral** – utiliza apenas um parâmetro, a(s) palavra(s) a ser(em) buscada(s), percorrendo o conteúdo de todas as páginas. Um exemplo desta consulta é a busca por páginas que contenham a palavra "botnet";
- **Focada no tipo da página** – utiliza três parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) de foco "***-focus***" e o tipo de página em deve ser pesquisada. Este tipo de consulta percorre somente as páginas com tipo igual ao definido. Um exemplo desta consulta é a seguinte busca: SQLinjection ***-focus*** *Alert,bulletin,vulnerability*, onde somente as páginas do tipo alerta, boletim e vulnerabilidade serão consultadas;
- **Focada no campo da página** – utiliza três parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) do campo "***-field***" e o campo a ser considerado na

pesquisa. Este tipo de consulta percorre somente as páginas que possuem o campo igual ao definido. Um exemplo desta consulta é a seguinte busca: AS4134 ***–field*** *ASN*, onde somente as páginas que contenham o campo ASN serão consultadas;

- **Sim/Não** – utiliza três parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) "***-malicious***" e o tipo *YES/NO*. Este tipo de consulta percorre todas as páginas a procura da(s) palavra(s) buscada(s). O diferencial desta consulta em relação à consulta geral é que ela tem como retorno "YES", caso o que se procura esteja relacionado a qualquer tipo de página que descreve atividade maliciosa, ou "NO", caso contrário. Este tipo de consulta é bastante útil para averiguar determinadas situações com, por exemplo, se um servidor SMTP está listado em alguma *black list* ou *white list*, entre outras. Um exemplo desta consulta é a seguinte busca: AS4134 ***–malicious*** *YES/NO*;
- **Resumo de vulnerabilidade** – utiliza três parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) "***-summary***" e o endereço canônico do nome do arquivo que será salvo. Este tipo de consulta percorre todas as páginas a procura da(s) palavra(s) buscada(s), retornando um arquivo XML contendo o nível de criticidade ou severidade da vulnerabilidade pesquisada e quantas vezes a vulnerabilidade obteve este nível, as datas de aparição e, por fim, os locais onde essas vulnerabilidades poderiam ser atacadas. Este tipo de consulta é bastante útil porque permite traçar um perfil da vulnerabilidade. Um exemplo desta consulta é a seguinte busca: Microsoft ***–summary*** */home/trodrigues/summary*;
- **Resumo de ataques** – utiliza três parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) "***-statistic***" e o endereço canônico do nome do arquivo que será salvo. Este tipo de consulta percorre todas as páginas a procura da(s) palavra(s) buscada(s), retornando um arquivo XML contendo o identificador da vulnerabilidade (padrão CVE), a classificação e a descrição deste ataque. Este tipo de consulta é bastante útil porque permite traçar a abrangência de um ataque. Um exemplo desta consulta é a seguinte busca: Microsoft ***–statistic*** */home/trodrigues/statistic*;
- **Focada no campo e no tipo da página** – utiliza cinco parâmetros: a(s) palavra(s) a ser(em) buscada(s), o identificador (*tag*) "***-field***", o(s) campo(s) buscado(s), o identificador (*tag*) "***-focus***" e o(s) tipo(s) de pagina(s). Este tipo de consulta realiza

uma busca focada nos campos listados e apenas nos tipos de páginas definidas como parâmetro. Um exemplo desta consulta é a seguinte busca: *Microsoft* ***–field*** *high_v* ***-focus*** *Bulletin,Alert*;

Vale ressaltar que foram usadas algumas técnicas de mineração de dados tanto no conteúdo da página, com o objetivo de estruturarem os conteúdos para consulta, separando cada campo identificado nos *templates* como relevante e indexando-os com os valores da coluna "Campo" demonstrado nas tabelas de *templates*, em apêndice. Também foram aplicadas técnicas de mineração nas consultas (remoção de *StopWords*) para possibilitar o uso de consultas por estrutura, consultas gerais ou consultas com filtragem de domínio.

4 Avaliações e Resultados

Este capítulo mostrará o ambiente em que a solução foi criada e exibirá alguns resultados contrastando-os a fim de avaliar a importância dos mesmos.

4.1 Métricas de Desempenho

É de suma importância que métricas de desempenho sejam estabelecidas para que se possa mensurar a qualidade da solução proposta. A ferramenta proposta uso de algumas melhorias que podem ser mensuradas por diferentes métricas citadas em [24] que serão descritas a seguir.

4.1.1 Precisão

A precisão (do inglês *precision*) [65] é uma métrica que é medida sobre o resultado da consulta. É definida através da proporção do total da cardinalidade do conjunto de documentos recuperados relevantes pelo total da cardinalidade do conjunto de documentos recuperados (Fórmula 4.1).

$$\boldsymbol{Precisão} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{recuperados}}|} \qquad \textbf{(4.1)}$$

4.1.2 Abrangência

A abrangência (do inglês recall) [65]é uma métrica que está diretamente relacionada à variedade dos documentos da base. Deste modo, pode-se dizer que quanto mais abrangente a base, mais variados são os assuntos presentes na mesma e quanto menos abrangente a base, mais especializada em um assunto ela é.

A métrica abrangência é a proporção entre a cardinalidade do conjunto de documentos relevantes recuperados em uma consulta pela cardinalidade do conjunto dos documentos relevantes na base (Fórmula 4.2).

$$\boldsymbol{Abrangência} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{relevantes}}|} \qquad \textbf{(4.2)}$$

Taxa de acerto na extração do conteúdo

A taxa de acerto na extração do conteúdo é uma métrica muito importante para a validação do extrator. Como se trata de conteúdo textual fica muito difícil construir um analisador contextual para coletar automaticamente os valores de acerto/erro, por isso, a métrica taxa de acerto será construída a partir da observação de vários dias de coleta em diferentes meses.

Esta métrica é a que deve ser observada com mais atenção, pois a extração do conteúdo e indexação do mesmo com suas respectivas palavras-chave é o que possibilita as buscas avançadas e todas as outras funcionalidades que o ARAPONGA provê a seus usuários.

Esta métrica tem como espaço amostral um valor entre 0 e 1 que será obtido pela proporção da cardinalidade do conjunto de documentos extraídos corretamente pela cardinalidade do conjunto total de documentos, como podem ser visto na Fórmula 4.3 abaixo.

$$\boldsymbol{Tx\ Extra}\text{ção}\boldsymbol{o} = \frac{|\boldsymbol{N_{extração\ correta}}|}{|\boldsymbol{N_{total\ documentos}}|} \qquad \textbf{(4.3)}$$

Tamanho da base

O Tamanho da base é a métrica que valida a eficiência e eficácia do processo de exclusão de documentos por *tags* e por URLs (que são os predecessores da indexação). Esta medida é um dos parâmetros mandatórios no que se refere ao ganho de qualidade mensurado pela solução proposta uma vez que, se com a eliminação de páginas a taxa de acerto se mantiver constante ou melhorar e houver redução do espaço em disco usado, o processo de exclusão de páginas proposto provê um ganho. Caso haja uma queda na taxa de acerto, a solução de exclusão de documentos não provê o resultado esperado.

Os quatro tipos de base que são:

- Base simples: sem nenhum processo de exclusão de página e uso de *templates*;
- Base com exclusão simples: com o pré-processamento de exclusão por URLs;
- Base com exclusão dupla: com o pré-processamento de exclusão por URLs e por *tags*;
- Base ARAPONGA: com exclusão por *tags* e URLs, além do uso de *templates*.

A ideia de coletar o tamanho da base em Mb é mensurar o ganho do pré-processamento de exclusão de páginas e quantificar se, no processo de indexação com *templates*, há um aumento expressivo no espaço em disco.

4.2 Ambiente de Experimentação

Na construção e testes do ARAPONGA, foi utilizado um computador com processador Intel Core i7, 8 Gbytes de memória RAM e HDD de 1 TB. O sistema operacional utilizado foi o Ubuntu 11.10. O ambiente de rede do Grupo de Pesquisa em Redes em Telecomunicações (GPRT) da Universidade Federal de Pernambuco (UFPE) foi utilizado por fornecer um link de acesso a Internet de 100Mbps com o PoP-PE (Ponto de Presença da RNP).

4.3 Resultados

4.3.1 Número de elementos da base

A avaliação do número de elementos na base foi executado durante 8 meses (de Abril a Novembro de 2011) e tem por objetivo analisar o número de páginas com que a estrutura de busca trabalha, além de mostrar a diferença entre as bases acessadas via Web e a base gerada.

Basicamente, este experimento consiste na execução do módulo *crawler*, operando de forma a capturar no máximo 150 referências (*links*) por página e com profundidade na árvore de busca de até 10 referências.

A Tabela 4.1 mostra o número de documentos indexados em cada base, os documentos não indexados e aqueles indexados sem nenhum *template*.

Tabela 4.1. Documentos na base por mês

	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Páginas indexadas na base Tradicional	42792	31363	56321	47954	37810	42740	48417	51191
Páginas indexadas na base aprimorada	4443	3125	4103	2629	2801	2999	3138	4767
Páginas indexadas com template	3621	2529	3110	2021	2025	2092	2216	1622
Páginas indexadas sem template	822	596	993	608	776	907	922	3145
Páginas não indexadas	38349	28238	52218	45325	35009	39741	45279	46424

Como pode ser observado, existem diferenças entre os valores da base tradicional e aprimorada. Tais diferenças se devem ao esquema de filtragem realizada pelo módulo de adequação (*Text Parser*) que além de comparar as páginas com os *templates* criados, extrai informações mais detalhadas e as indexa com identificadores. Este processo também executa filtragem de páginas por URLs e por conteúdo de forma que páginas com conteúdos irrelevantes não sejam indexadas ou, caso sejam indexadas, seu número ficou o mais próximo possível de zero.

4.3.2 Tamanho da base

O experimento de tamanho da base foi realizado durante os 8 meses em que o *crawler* coletou páginas com o objetivo mensurar a eficiência do processo de filtragem de páginas realizada no módulo de adequação e quanto de espaço em disco é usado no processo de indexação do conteúdo extraído ao se fazer uso dos *templates*.

Para que tivéssemos a noção exata destes custos, no processo de indexação da base ARAPONGA, foram-se criadas mais três bases que são referentes às três etapas de pré-

processamento dos documentos até se chegar à base final do ARAPONGA. Estas bases foram nomeadas da seguinte maneira:

- **Base Simples:** todo o conteúdo é indexado sem passar por nenhum tipo de filtragem e/ou extração de conteúdo;
- **Base com 1 exclusão:** o conteúdo passa pelo processo de exclusão por URLs e não é encaminhado para o extrator;
- **Base com 2 exclusões:** o conteúdo passa pelo processo de exclusão por URLs e por TAGs e não é encaminhado para o extrator;
- **Base ARAPONGA:** o conteúdo passa pelo processo de exclusão por URLs e por TAGs e é encaminhado para o extrator onde, se a página tiver um *template* correspondente, serão adicionados ao documento os textos e suas referidas palavras-chaves.

  A evolução destas bases pode ser vista na Tabela 4.2.

Tabela 4.2. Evolução das bases por mês em Mb

	Abril	**Maio**	**Junho**	**Julho**	**Agosto**	**Setembro**	**Outubro**	**Novembro**
Base Simples	2080,6	3520.2	4386.7	2847.3	3600	5700	3500	4400
Base com 1 exclusão	98.7	123.9	142.7	98.3	76.4	103.6	154.9	63.9
Base com 2 exclusões	58.5	78.7	98.2	68	49.6	89.3	120.9	59.8
Base ARAPONGA	100.28	137.08	155.1	123.9	92.8	96.7	164.8	67

Como pode-se notar pela tabela, existe uma diferença notória no espaço em disco ocupado pela **Base Simples** em relação às outras bases. O processo de filtragem de páginas não relevantes é responsável pela economia de bastante espaço em disco além de aprimorar a precisão do sistema. Em relação à **Base ARAPONGA**, mesmo não tendo o menor espaço em disco, ela é a mais efetiva porque além de ter o mesmo conteúdo que a **Base com 2 Exclusões**, ela possibilita as consultas diferenciadas. Os ganhos destas consultas diferenciadas serão apresentados nas próximas Seções.

4.3.3 Teste de rendimento

O teste de rendimento entre as duas bases (**Base com 2 Exclusões** e **Base ARAPONGA**) do sistema é bastante relevante, pois exprime, em números, o ganho de rendimento de uma busca

comum para uma busca diferenciada. É importante ressaltar que de todos os tipos de buscas possíveis pelo módulo de interface, não foram analisadas aquelas que geram XML como resultado nem as do tipo Yes/No e nem a consulta que procura informações dos softwares instalados na máquina cliente, pois suas respostas não se enquadram nas métricas empregadas.

Para realização deste experimento foi considerado que todas as páginas indexadas são relevantes.

Basicamente, o experimento consiste de duas consultas: a primeira do tipo focada no campo e a segunda é focada no campo e no tipo de páginas. Para tanto, foi considerada a base dos 8 meses de coleta, contendo **21268** documentos na base mantida sem adequação, dita **Base com 2 Exclusões**, (criada pelo Nutch e Lucene) e **21268** documentos na base mantida com adequação, dita aprimorada, (processada após ser gerada pelo Nutch e Lucene).

A **consulta #1** buscou por referências a alguma publicação envolvendo o software "*Internet Explorer"*. Desta forma, a consulta gerada foi à seguinte: "*Internet Explorer" –field systems_affected,Vulnerable*.

Como resultado, a consulta aplicada na **Base com 2 Exclusões** retornou 297 páginas com informações sobre o software *Internet Explorer*, onde apenas 103 descreviam vulnerabilidades com alto grau de severidade. Sendo assim, a precisão da consulta nesta base é de 34,68%, ou seja, dos 297 documentos recuperados, apenas 103 eram relevantes de um total de 297 documentos recuperados.

$$\boldsymbol{Precisão} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{recuperados}}|} = \frac{\mathbf{103}}{\mathbf{297}} = \mathbf{0,3468}$$

A abrangência dessa consulta da base é de 0,48%, uma vez que dos 297 documentos retornados, apenas 103 eram relevantes de um universo de 21268 documentos.

$$\boldsymbol{Abrangência} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{relevantes}}|} = \frac{\mathbf{103}}{\mathbf{21268}} = \mathbf{0,0048}$$

Em relação à consulta na **Base ARAPONGA**, a precisão foi bem próxima de 100% uma vez que foram retornados 103 documentos, sendo 100 relevantes, e a abrangência foi de 0,47%.

$$\boldsymbol{Precisão} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{recuperados}}|} = \frac{\mathbf{100}}{\mathbf{103}} = \mathbf{0,9708}$$

$$\boldsymbol{Abrangência} = \frac{|\boldsymbol{N_{recuperados} \cap N_{relevantes}}|}{|\boldsymbol{N_{relevantes}}|} = \frac{\mathbf{100}}{\mathbf{21268}} = \mathbf{0,0047}$$

A Tabela 4.3 ilustra os valores encontrados na avaliação da Consulta #1.

Tabela 4.3. Resultado das métricas para Consulta #1.

	Abrangência	Precisão
Base com 2 Exclusões	0.0047	0.3367
Base ARAPONGA	0.0047	0.9708

Na **Consulta #2,** a busca foi por informação de produtos da empresa Adobe. A intenção é descobrir se há alguma publicação de softwares produzidos pela Adobe e exibir apenas as publicações da base de dados da US-CERT. Desta forma, a consulta gerada foi à seguinte: *Adobe –focus US-CERT.*

Como resultado, a consulta aplicada na **Base com 2 Exclusões** retornou 412 páginas com contendo softwares da Adobe, onde apenas 64 continham informações sobre softwares da Adobe e eram da base de dados da US-CERT. Sendo assim, a precisão da consulta nesta base é de 15,53%, ou seja, dos 412 documentos recuperados, apenas 64 eram relevantes de um total de 412 documentos recuperados. A abrangência é de 0,300 %, uma vez que dos 412 documentos retornados, 65 eram relevantes de um universo de 21268 documentos.

Em relação à consulta na **Base ARAPONGA**, a precisão foi de 98,48% uma vez que foram retornados 66 documentos, sendo 64 relevantes, e a abrangência foi de 0,300%, uma vez que 64 eram relevantes de um universo de 21268 documentos

A Tabela 4.4 ilustra os valores encontrados na avaliação da Consulta #2.

Tabela 4.4. Resultado das métricas para Consulta #2.

	Abrangência	Precisão
Base com 2 exclusões	0.00300	0.1577
Base ARAPONGA	0.00300	0.9848

De modo geral, observando-se os resultados obtidos é possível notar que o processo de indexar as páginas Web utilizando *templates* possibilitou a criação de consultas diferenciadas e aumentou a precisão do sistema para bem próximo de 100%, que era o objetivo a ser alcançado.

4.3.4 Taxa de acerto do extrator

A taxa de acerto do extrator é uma métrica muito importante para todo o processo de indexação porque se a extração não tiver um índice muito alto de acerto, todas as consultas diferenciadas não irão funcionar adequadamente e, por conseguinte, a melhor escolha de base seria a base com 2 exclusões.

Outro problema que se pode ter na criação dos *templates* é se a página usada como exemplo para o administrador criar as palavras-chaves não for a que melhor representa o grupo de páginas ou se as páginas do domínio tiverem uma variação muito grande na sua estrutura, o que impossibilita o processo de extração proposto já que o extrator leva em conta a estrutura das páginas. Se, no mínimo, algum destes problemas citados ocorrerem, o conteúdo que será indexado pela rotina agregada ao *template*, muito provavelmente nem fará sentido com a palavra-chave que o representa.

O processo de validação do extrator teve que ser de forma visual. Como o extrator leva em consideração a estrutura das páginas, a quantidade de texto presente entre as *tags* não faz nenhuma diferença desde que a árvore estrutural seja mantida. A fim de validar a estratégia de extração do conteúdo usando a árvore estrutural da página, para cada documento submetido ao processo de extração de conteúdo foi-se gerado outro documento contendo a palavra-chave e o texto que será indexado com a referida palavra-chave de tal modo que basta apenas observar os documentos gerados para ver se a extração ocorreu devidamente. Além da base de segurança que foi coletada pelo *crawler*, foi testada também uma parte da base do ClueWeb que foi gentilmente cedida de um projeto da UFAM.

Na Tabela 4.5 são exibidos os valores referentes à indexação da base de segurança e da base ClueWeb.

Tabela 4.5. Indexação Base de Segurança e Base ClueWeb

	Base ARAPONGA	**Base ClueWeb**
Extraídos corretamente	11280	35
Extraídos incorretos	361	9

Total de documentos	11641	44

Calculando a taxa de acerto do extrator para as bases ARAPONGA e ClueWeb tem-se:

$$TX_{extração\ ARAPONGA} = \frac{|N_{extração\ correta}|}{|N_{total\ documentos}|} = \frac{11280}{11641} = 0.96$$

$$TX_{extração\ ClueWeb} = \frac{|N_{extração\ correta}|}{|N_{total\ documentos}|} = \frac{35}{44} = 0.79$$

Como pode ser notado, a taxa de acerto do extrator na base ARAPONGA foi maior que a taxa da base ClueWeb. Tal fato se deve à baixa variação da estrutura do HTML nas páginas da base do ARAPONGA o que não é observado na base do ClueWeb onde as páginas possuem algumas variações que prejudicam o processo de extração. Uma forma de melhorar a taxa de acerto seria fazer mais um *template* que reconhecesse as páginas com estruturas diferentes.

4.4 Resultados de Outras Funcionalidades

Esta Seção exemplifica as funcionalidades que não possuem resultados que possam ser comparados pelas métricas estabelecidas.

4.4.1 Consulta com resposta resumida

Este tipo de consulta tem como resposta apenas partes do texto pré-determinadas pelo usuário. Vamos supor que o usuário deseja consultar vulnerabilidades que envolvam o Apache e apenas obter o conteúdo presente no campo "*description*" das páginas retornadas, a consulta gerada seria: *Apache –result description.*

Como respostas desta consulta teríamos documentos contendo apenas o conteúdo presente no campo "*description*" das páginas encontradas.

Este tipo de consulta possibilita a construção de uma funcionalidade que foi adicionada como trabalhos futuros chamada: Revista de Vulnerabilidades. Esta funcionalidade seria configurada pelo usuário e, periodicamente, o mesmo receberia uma espécie de "revista digital" contendo as informações por ele configuradas.

4.4.2 Resumo da base de dados

Este tipo de consulta retorna um arquivo de extensão ".xml" semi-estruturado contendo a quantidade de documentos presentes na base separados por domínio e tipo de página.

A Figura 4.1 exemplifica o resultado da consulta de todos os documentos presentes na Base ARAPONGA (*ARAPONGA –knbase /home/trodrigues/knbase*).

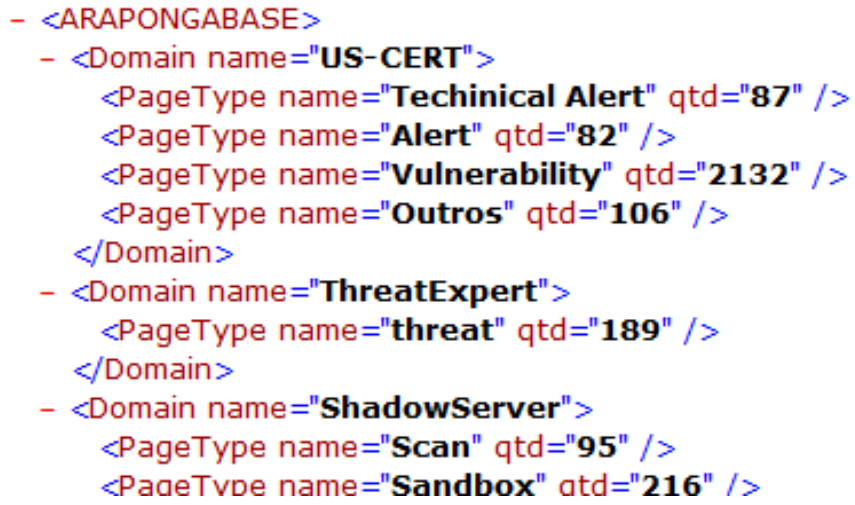

Figura 4.1. Resultado da consulta de resumo da Base ARAPONGA.

4.4.3 Resumo de uma consulta Sim/Não (Yes/No)

Neste tipo de consulta, o resultado retornado é uma resposta simples de "*Yes*" quando há a existência da consulta na base de dados e "*No*" caso contrário.

Como exemplo, supõe-se que a consulta deseja receber informações se o AS36666 está ou estava em uma lista de ASNs envolvidos em atividades maliciosas. A consulta gerada é a seguinte: *AS36666 –* ***result*** *YES/NO.* Se o ARAPONGA retornou "*Yes"* como resposta significa que há relatos do AS36666 na lista de ASN caso retorne "*No*" significa que o AS36666 não está na lista.

4.4.4 Vocabulário de segurança

Um dos principais métodos usados para melhorar a representação de um documento é a retirada das palavras mais comuns a todos os documentos da base para que as palavras que fiquem em cada documento sejam capazes de tornar fácil a identificação do mesmo.

Como a biblioteca do Lucene nos permite adicionar palavras à *stopList* padrão (lista das palavras mais comuns em Inglês), foi criada a *ARAPONGAStopList* que contêm as palavras mais comuns em Inglês (*stopList* padrão do Lucene) e as mais comuns nas publicações de segurança.

Na tabela 4.6 segue uma parte das palavras que fazem parte da *ARAPONGAStopList.*

Tabela 4.6. Dez primeiras palavras adicionadas à StopList

Palavra	Número de Aparições
Kernel	455859
Server	188437
Production	165253
Application	135476
Software	93321
Standard	71325
Unrecognized	57561
Developer	45705
Programs	39053
System	31999

4.4.5 Lista de softwares instalados que precisam de atualização

Desenvolvida primeiramente para Linux, esta funcionalidade tem como principal foco encontrar publicações de segurança relacionadas a softwares ou bibliotecas instaladas no computador do cliente.

O processo de criação da lista de softwares que precisam ser observados por terem alguma publicação de vulnerabilidade passa por três etapas. Na primeira etapa, todos os softwares e bibliotecas instaladas são listados e encaminhados para a etapa seguinte. Na segunda etapa, a lista é encaminhada para o processo de busca onde, se tiver alguma publicação, são agrupadas e passadas para a próxima etapa. Na etapa final, o cliente recebe uma lista com os softwares ou bibliotecas que tiveram ao menos uma vulnerabilidade publicada e pode acessar o conteúdo das mesmas para obter mais detalhes de como proceder.

5 Conclusões e Trabalhos Futuros

Esta dissertação de mestrado apresentou uma ferramenta de atualização de concentração de informação de segurança de redes e sistemas baseado na extração de informação de conteúdo Web adquirido em bases de dados de empresas e instituições renomadas e conceituadas no ramo da segurança de redes e sistemas. A aquisição destas informações na Web é feita através de um *crawler* que adquire todo o conteúdo das páginas HTML e indexam em uma base de dados para serem posteriormente acessadas com uma interface para a Web e outra interface para outros sistemas acessarem. No processo de criação da base de dados que é acessada por outros sistemas como o foco era uma informação direcionada e com conteúdo restritamente útil para o outro sistema, técnicas de mineração de dados, *templates*, consultas direcionadas foram criadas a fim de atender esta necessidade.

Foi criado nesta dissertação um estudo detalhado sobre algumas bases de dados e sites que divulgam informações de atividades maliciosas ou de vulnerabilidades onde cada base ou site tinha uma característica própria. Na escolha das bases usadas neste projeto o que mais pesou para a escolha foi a completude das informações divulgadas e o conteúdo divulgado.

Adicionalmente a este trabalho foi criada uma aplicação cliente que se conecta ao módulo usado por outros sistemas e acessa via console ou via GUI as informações cadastradas. Estas duas interfaces cliente só se diferenciam em um tipo de consulta diferenciada que é a que responde Sim ou Não apenas por não fazer sentido exibir em uma interface gráfica um conteúdo diferente do que foi modelado para a interface gráfica.

5.1 Contribuições

As principais contribuições desta dissertação de mestrado foram:

- A concentração das informações divulgadas nas várias bases de dados que publicam informações de vulnerabilidades e atividades maliciosas em um só lugar podendo ser acessadas via Web e/ou por outros sistemas auxiliando assim os administradores de redes ou sistemas automatizados a obterem informações

relevantes que facilitarão na tomada de decisão na hora de atualizar ou bloquear ou liberar uma aplicação;

- Um extrator de conteúdo focado na estrutura do documento que pode ser programado graficamente pelo usuário e pode se comportar de maneiras diferentes, dependendo apenas da forma que foi configurado;
- A publicação de dois artigos, um em congresso de segurança (SBSeg) em 2010 [68], e outro em uma conferência internacional sobre internet (WWW/Internet) em 2011 [67];

5.2 Trabalhos futuros

Como trabalhos futuros, podem ser relacionados às seguintes tarefas:

- A construção de um sistema de recomendação na busca baseado na proximidade das palavras ou sensível ao contexto. Como exemplo desta consulta por proximidade pode-se citar uma consulta erroneamente feita pela palavra "*Nicrosoft*" então o sistema é capaz de perguntar se a consulta desejada foi pela palavra "*Microsoft*".

- Estudo sobre a melhor forma de agrupamento da base para consultas usando, por exemplo, lógica Fuzzy ou redes neurais.

- A criação de uma Revista de Vulnerabilidades na qual o usuário selecionará que informações e em que freqüência deseja receber tais informações.

- Construção de uma versão da ferramenta de busca de software ou bibliotecas que precisam de atualização para Windows, MAC, Android e IOS.

Referências

[1] M. Morin, "The Financial Impact of Attack Traffic on Broadband Networks," *IEC Annual Review of Broadband Communications*, pp. 11-14, 2006.

[2] CERT.br. (2012) Computer Emergency Response Team Brazil. [Online]. http://www.cert.br

[3] CAIS. (2012) RNP's Security Incident Response Team. [Online]. http://www.rnp.br/cais

[4] Team Cymru. (2012) Team Cymru [Online]. http://www.team-cymru.org/

[5] IBM. (2012) VulDa: A Vulnerability Database. [Online]. http://domino.watson.ibm.com/library/cyberdig.nsf/a3807c5b4823c53f8525656100632 4be/4cc8fa2ee3af7fc9852567280039a299?OpenDocument

[6] CISCO. (2012) Cisco Security Center. [Online]. http://tools.cisco.com/security/center/home.x

[7] NIST. (2012) National Vulnerability Database (NVD). [Online]. http://nvd.nist.gov

[8] OSVDB. (2012) Open Source Vulnerabilities Database. [Online]. http://www.osvdb.org

[9] Dragonsoft Vulnerability Database. (2012) Dragonsoft Vulnerability Database [Online]. http://vdb.dragonsoft.com/

[10] ThreatExpert. (2012) ThreatExpert [Online]. http://www.threatexpert.com/

[11] ShadowServer. (2012) ShadowServer [Online]. http://www.shadowserver.org/wiki/pmwiki.php/Stats/Statistics/

[12] IBM Internet Security Systems. (2012) IBM Internet Security Systems [Online]. http://www.iss.net/threats/ThreatLis.php/

[13] Secunia (2012) Secunia [Online]. http://secunia.com

[14] Luis. O. C Borba, "Um esquema de divulgação sobre informações de vulnerabilidades," Universidade Federal de Pernambuco, Recife, Trabalho Final de Graduação 2009.

[15] US-CERT. (2012) Technical Alerts. [Online]. http://www.us-cert.gov/cas/techalerts

[16] US-CERT. (2012) Security Bulletins. [Online]. http://www.us-cert.gov/cas/bulletins/

[17] US-CERT. (2012) KB-CERT. [Online]. https://www.kb.cert.org/vuls

[18] US-CERT. (2012) Alerts. [Online]. http://www.us-cert.gov/cas/alerts/

[19] Cwr.cl. (2012) Web Information Retrieval Environment - WIRE. [Online]. http://www.cwr.cl/projects/WIRE/

[20] Heritrix. (2012) Heritrix. [Online]. http://crawler.archive.org/

[21] Apache. (2012) Nutch. [Online]. http://lucene.apache.org/nutch/

[22] Apache. (2012) Lucene. [Online]. http://lucene.apache.org/java/docs/

[23] Jericho HTML Parser. (2012) Jericho HTMLl Parser. [Online]. http://jericho.htmlparser.net/docs/index.html

[24] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley, New York, 1999.

[25] Y. Yao. Information Retrival Support System. In IEEE World Congress on Computational Intelligence, pages 773-778, 2002.

[26] J. Yao and Y. Yao. Web-based Information Retrieval Support System: Building Research Tools for Scientists in the New Information Age. In IEEE/WIC International Conference on Web Intelligence, pages 570-573, 2003.

[27] O Hoeber. Web Information Retrieval Support Systems: The Future of Web Search. In 2008 International Workshop on Web Information Retrieval Support Systems, pages 29-32, 2008.
[28] Google. Google Maps. Available from http://maps.google.com. 2011.
[29] AllInOneNews. Available from http://www.allinonenews.com. 2011.
[30] Y. Zeng, Y. Yao, and N. Zhong. DBLP-SSE: A DBLP Search Support Engine. In IEEE/WIC/ACM International Conference on Web Intelligence, 2009.
[31] G. Marchionini and R. W. White. Information Seeking Support System. IEEE Computer, 42(3):30-32, March 2009.
[32] M. Ley, "The DBLP computer science bibliography: Evolution, research issues, perspectives. In 9th International Symposium of String Processing and Information Retrieval, 2002, pp. 1-10.
[33] M. Tilsner, O. Hoeber, and A. Fiech. CubanSea: Cluster-Based Visualization of Search Results. IEEE Computer, 42(3):108-112, March 2009.
[34] C. Shah. ContextMiner: Explore Globally, Aggregate Locally. IEEE Computer, 42(3):94, March 2009.
[35] WolframAlpha. Available from http://www.wolframalhpa.com. 2009.
[36] R. Capra and G. Marchionini. Faceted Exploratory Search Using the Relation Browser. In NSF Workshop on Information Seeking Support Systems, pages 81-83, 2009.
[37] R. Capra and G. Marchionini. Faceted Browsing, Dynamic Interfaces, and Exploratory Search: Experiences and Challenges. In Workshop on Human-Computer Interaction and Information Retrieval (HCIR 07), pages 7-9, 2007.
[38] P. Hayes and B. McBride. RDF semantics. Available from http://www.w3.org/TR/rdf-mt. 2004.
[39] Souza, Renato R. and Alvarenga, Lilia. A Web Semântica e suas contribuições para a ciência da informação, Brasília, v. 33, n. 1, p. 132-141, jan./abril 2004.
[40] Souza, Renato R. Sistemas de recuperação de informações e mecanismos de busca na *web*: panorama atual e tendências, Perspect. ciênc. inf. vol.11 no.2 Belo Horizonte May/Aug. 2006.
[41] Candy Schwartz. Journal of the American Society for Information Science Volume 49, Issue 11, pages 973–982, 1998
[42] Ryen W. White and Resa A. Roth, "Exploratory Search: Beyond the Query-Response Paradigm," *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 1, no. 1, pp. 1-98, 2009.
[43] Robotstxt.org. (2009) Robots Exclusion. [Online]. http://www.robotstxt.org/
[44] M. Kobayashi and K. Takeda, "Information retrieval on the web," ACM Computing Surveys, vol. 32, no. 2, p. 144–173, Jun 2000.
[45] J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan A. Arasu, "Searching the web," ACM Transaction on Internet Technology, vol. 1, no. 1, pp. 2-43, Ago 2001.
[46] S.S. Dhenakaran and K. Thirugnana Sambanthan. WEB CRAWLER - AN OVERVIEW, International Journal of Computer Science and Communication, Vol. 2, No. 1, January-June 2011, pp. 265-267.
[47] Bidoki, Yazdani et el, "FICA: A fast intelligent crawling algorithm", Web Intelligence,
[48] IEEE/ACM/WIC International conference on Intelligent agent technology, Pages 635-641, 2007.
[49] Junghoo Cho, Hector Garcia-Molina, Lawrence Page, |Efficient crawling through URL ordering", 7 th International WWW Conference , April 14-18, Brisbane, 1998.

[50] Zheng, Chen, "HAWK: a Focused crawler with content and link analysis", E-business engineering, 2008, ICEBE'08, IEEE international conference, pages 677-680, Oct 2008.
[51] Zheng, Zhaou ET el, "URL Rule based focused crawler", E-business engineering, ICEBE'08, IEEE international conference, Oct 2008, pages 147-154, 2008.
[52] G. Yang, I. V. Ramakrishnan, and M. Kifer. On the complexity of schema inference from web pages in the presence of nullable data attributes. In Proceedings of the 12th International Conference on Information and Knowledge Management, pages 224–231. ACM Press, 2003.
[53] J.-K. Min, J.-Y. Ahn, and C.-W. Chung. Efficient extraction of schemas for xml documents. Information Processing Letters, 85(1):7–12, 2003
[54] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large Web sites. In Proceedings of the 27th International Conference on Very Large Data Bases, pages 109–118, Rome, Italy, 2001.
[55] V. Crescenzi, G. Mecca, and P. Merialdo. Wrapping-oriented classification of Web pages. In Proceedings of the 2002 ACM Symposium on Applied Computing, pages 1108–1112. ACM Press, 2002.
[56] L. Arllota, V. Crescenzi, G. Mecca, and P. Merialdo. Automatic annotation of data extraction from large Web sites. In Proceedings of the International Workshop on the Web and Databases, pages 7–12, San Diego, USA, 2003.
[57] Reis, D. Golgher, P., Silva, A., Laender, A. Automatic Web news extraction using tree edit distance, WWW'04, 2004.
[58] SODERLAND, S. Learning information extraction rules for semi-structured and free text. Machine Learning 34, 1-3 (1999), 233-272.
[59] FREITAO, D. Machine Learning for Information Extraction in Informal Domains. Machine Learning 39, 2/3 (2000), 169-202.
[60] ADELBERG, B. NoDoSE - A tool for semi-automatically extracting structured and semistructured data from text documents. In Proceedings of the ACM SIGMOD International Conference on Management of Data (Seattle, WA,1998), pp. 283-294.
[61] LAENDER, A. H. F., RIBEIRO-NETO, B., AND DA SILVA., A. S. DEByE - Data Extraction By Example. Data and Knowledge Engineering 40, 2 (2002), 121-154.
[62] RIBEIRO-NETO, B . , LAENDER, A. H. F., AND DA SILVA, A. S. Extracting semi-structured data through examples. In Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management (Kansas City, MO, 1999), pp. 94-101.
[63] EMBLEY, D.W., CAMPBELL, D.M., JIANO, Y. S., LIDDLE, S. W . , KAI N G , Y., QUASS, D., AND SMITH, R. D. Conceptual-model-based data extraction from multiple-record Web pages. Data and Knowledge Engineering 31, 3 (1999), 227-251
[64] ABASOAL, R., AND SANOHEZ, J. A .X-tract: Structure extraction f r o m botanical textual descriptions. In Proceeding off the String Processing & Information Retrieval Symposium and International Workshop on Groupware, SPIRE/GRIWG (Cancún,México,1999), pp. 2-7.
[65] C. J. V. Rijsbergen. Information Retrieval, 2nd edition. Dept. of Computer Science, University of Glasgow, 1979
[66] Charles T. Meadow,Bert R. Boyce,Donald H. Kraf. Text Information Retrieval Systems, third edition. Academic Press, United Kingdom, 2007.

[67] FEITOSA, E. L. ; Rodrigues, T. G. ; KELNER, J. ; SADOK, D. . ARAPONGA: A WIRSS for Internet Anomalies and Vulnerabilities. In: IADIS International Conference on WWW/Internet, 2011, Rio de Janeiro. IADIS International Conference on WWW/Internet. Lisboa : IADIS, 2011. v. 1. p. 90-98.

[68] Rodrigues, T. G. ; FEITOSA, E. L. ; SADOK, D. ; KELNER, J. . Uma Ferramenta de Suporte a Recuperação de Informação na Web focada em Vulnerabilidades e Anomalias Internet. In: X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2010, Fortaleza. X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg). Porto Alegre : SBC, 2010. v. 1. p. 227-240.

[69] ATLAS. (2012) ATLAS [online]. http://atlas.arbor.net/

Apêndice – Templates

Este tópico explanará as características comuns nos padrões de cada site de divulgação de vulnerabilidade que foram estudados. Estes padrões que foram descobertos foram chamados de *templates* e nos próximos tópicos serão exibidos com uma breve descrição sobre cada campo.

Secunia Advisores

Campo	Descrição
Secunia	Identificador único da publicação
Release date	Data de lançamento
Popularity	Detalha os produtos afetados
Critical	Mostra quão crítico é a vulnerabilidade
Impact	Como afeta os sistemas
Where	Local do sistema afetado
Solution Status	O que foi feito para resolver o problema, se foi criado um patch ou uma nova versão
Software	Softwares afetados
CVSS Score	Pontuação para o impacto que esta vulnerabilidade ocasiona
Content	Conteúdo da publicação com informações de descrição, solução, referências externas e CVE's referentes ao assunto

KB-CERT

Campo	Descrição
Overview	Resumo da vulnerabilidade
Description	Descrição da vulnerabilidade
Impact	Impacto de um ataque explorando a vulnerabilidade
Solution	Como sanar o problema
Systems Affected	Sistemas afetados pela vulnerabilidade
Referencies	Links externos que contêm informações sobre a vulnerabilidade
Credit	Quem descobriu/solucionou a vulnerabilidade
Other Information	Informações das datas de publicação, solução e ultima atualização, ID da CVE e NVD referente à vulnerabilidade

US-CERT

As páginas da US-CERT contêm três padrões diferentes, um para cada tipo de divulgação de informação.

Alerta Técnico [5]

Campo	Descrição
Systems Affected	Lista com os sistemas afetados.
Overview	Resumo da vulnerabilidade.
Description	Descrição da vulnerabilidade
Impact	Impacto de um ataque explorando a vulnerabilidade.
Solution	Como sanar o problema.
Referencies	Links externos que contêm informações sobre a vulnerabilidade.

Boletim [6]

Campo	Descrição
Vendor-Product	Faz uma ligação entre o produto e o fabricante.
Description	Descreve brevemente o comportamento da vulnerabilidade.
Published	Data em que a vulnerabilidade foi publicada.
CVSS Score	Divulga uma pontuação para a vulnerabilidade de acordo com a severidade.
Source & Patch Info	Links externos para mais informações sobre a vulnerabilidade.

Alerta Simples [7]

Campo	Descrição
Systems Affected	Lista com os sistemas afetados
Overview	Resumo da vulnerabilidade
Solution	Como sanar o problema
Description	Descrição da vulnerabilidade
Referencies	Links externos que contêm informações sobre a vulnerabilidade

ATLAS

Campo	Descrição
ID	Identificador do elemento na tabela
Attacks per Subnet	Número médio de ataques que cada sub-rede sofreu
Percentage	Porcentagem no número de ataques em relação a todos os ataques

Além das informações contidas nas tabelas, há uma parte da página com o campo "BACKGROUND" que exibe informações detalhadas e referências externas sobre a ameaça.
As páginas da Arbor exibem mais informações quando o usuário está "logado". Estas informações podem ser visualizadas ao clicar nos Identificadores das Tabelas e contêm informações detalhadas sobre o elemento selecionado como, por exemplo, ao clicar em um identificador que está na tabela de ASN, são exibidas mais informações sobre aquele ASN que ao usuário não "logado".

ShadowServer

ASN

Campo	Descrição
By Number	Tabela dos ASN ordenados pelo maior número de comandos e controles ativos e inativos (C&C)
By Highest Closed	Tabela contendo ASNs que têm os C&C inativos ordenados pela maior porcentagem de inatividade
By Lowest Closed	Tabela contendo ASNs que têm os C&C ativos ordenados pela menor porcentagem de inatividade

Viruses

Campo	Descrição
Linux Vendor List	Lista de todos os antivírus para Linux que fizeram atualização
Windows Vendor list	Lista de todos os antivírus para Windows que fizeram atualização

Malware

Campo	Descrição
Charts	Tabela contendo as estatísticas de implantação de Malware (email, IRC, sandbox etc) em vários períodos (24 horas, 48 horas, 10 dias, 30 dias, até 2 anos)

Nacional Vulnerability Database (NDV)

CVE Vulnerability

Campo	Descrição
Original Release Date	Data de cadastro da vulnerabilidade
Last Revised	Ultima revisão da vulnerabilidade
Source	Onde a vulnerabilidade foi cadastrada pela primeira vez
Overview	Resumo da vulnerabilidade
References to Advisories, Solutions, and tools	Referências para a solução da vulnerabilidade

National Checklist Program (NCP)

Campo	Descrição
Checklist Results	Tabela contendo os produtos, categoria dos produtos, data de publicação, nome da lista e recursos