Uma Ferramenta de Suporte a Recuperação de Informação na Web focada em Vulnerabilidades e Anomalias Internet

Thiago Gomes Rodrigues[1], Eduardo Luzeiro Feitosa[12], Djamel Sadok[1], Judith Kelner[1]

[1]Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – Cidade Universitária - Recife - PE

[2]Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM)
CEP 69077-000 Campus Universitário - Manaus - AM

{tgr, elf, jamel, jk}@cin.ufpe.br

Resumo. *Hoje em dia, administradores de rede e gerentes de TI têm utilizado sítios Web, que contêm informações sobre vulnerabilidades e estatísticas do tráfego Internet, para manterem-se atualizados sobre a atual situação de segurança da Internet e sistemas e, consequentemente, tentarem minimizar o impacto causado por ataques e anomalias. Este trabalho apresenta um sistema de apoio à recuperação de informação na Web (WIRSS) focado em recolher informações sobre segurança disponíveis na Internet, chamado ARAPONGA. Através do uso desta ferramenta é possível concentrar e condensar melhor informações sobre vulnerabilidades e estatísticas do tráfego Internet.*

Abstract. *In present days, network administrators and TI managers must rely on Web sites containing vulnerabilities reports and Internet traffic statistics to keep up to date about current security threats and incidents in an attempt to minimize the impact of attacks and anomalies. This paper presents a Web Information Retrieval Support System (WIRSS) focused on gathering information about Internet security events, called ARAPONGA. By using this tool, we are able to better concentrate, relate and present information on vulnerabilities unwanted traffic statistics.*

1. Introdução

A última década presenciou o aumento do tráfego Internet não desejado, não solicitado e muitas vezes ilegítimo. Apesar de estar relacionado a atividades como spam e ataques de negação de serviço, grande parte do tráfego Internet não desejado envolve diretamente vulnerabilidades em software, sistemas e serviços.

Uma vez que garantir a inexistência de vulnerabilidades é praticamente impossível, estar ou ficar atualizado sobre as atuais e reais questões de segurança é a melhor solução. É neste contexto que o uso de bases de informação e sítios Web sobre vulnerabilidades, anomalias e segurança da informação surgem como solução de segurança simples, comum e prática e tem sido utilizada na construção de sistemas de detecção de intrusão e

ferramentas de verificação de vulnerabilidade. A relevância deste tipo de solução é facilmente comprovada pela existência de dezenas de bases de dados e sítios Web, tais como *Cisco Security Center* [Cisco 2009], *National Vulnerability Database* [NIST 2009], *Secunia Advisories* [Secunia 2009] e *Open Source Vulnerability Database* [OSVDB 2009]. Contudo, problemas como a falta de coordenação, o uso de diferentes formatos de apresentação e classificação da informação, além da não apresentação de dados estatísticos e/ou correlação de eventos fazem com que as equipes de segurança precisem gastar recursos e tempo navegando na Web e tentado filtrar o que pode ser relevante.

A fim de solucionar este problema, este trabalho propõe um sistema de busca na Web capaz de proporcionar a integração de informações de segurança, além dar suporte a pesquisas avançadas. Este sistema é chamado de **ARAPONGA**.

O resto deste artigo explica a utilidade do ARAPONGA. Primeiramente, conceitos básicos e trabalhos relacionados são apresentados com o objetivo de diferenciar esta solução a partir de outras propostas. Em seguida, uma visão geral do projeto é apresentada, incluindo uma descrição detalhada dos componentes e sua implementação. Depois, uma avaliação inicial é apresentada para validar a ferramenta desenvolvida. Por último, algumas considerações são feitas.

2. Conceitos Básicos e Trabalhos Relacionados

É notória a enorme riqueza de dados e informações acumuladas pela humanidade. No entanto, com o enorme sucesso da Web aliado ao rápido e fácil acesso a informação, o modo de como encontrar conhecimento e informação útil tornou-se relevante. A resposta usual é a utilização de um sistema de recuperação de informação (do inglês *Information Retrieval System*- IRS).

IRS é uma denominação genérica para uma classe de ferramentas dedicadas à manipulação e recuperação de grandes volumes de informação em diferentes formatos de apresentação como bancos de dados. Normalmente, um IRS investiga diferentes aspectos da informação tais como representação, armazenamento, organização e acesso. Contudo, o pressuposto básico de seu funcionamento é que os usuários saibam exatamente o que querem. No entanto, em um cenário como a Internet os usuários frequentemente esquecem ou ignoram este princípio uma vez que se deparam com bilhões de páginas atualizadas quase diariamente.

Para resolver este problema, Yao e Yao [Yao e Yao 2003] propuseram mudar a filosofia (foco) do IRS, que era voltada ao sistema (*system centric*) e de voltado recuperação (*retrieval centric)*, para voltada aos usuários (*user centric*), e ao suporte a recuperação (*support centric*). Isso culminou com o conceito de suporte a recuperação de informação (do inglês *Information Retrieval Support Systems*- IRSS). O principal objetivo de qualquer IRSS é dar suporte aos usuários, fornecendo os meios necessários, ferramentas e linguagens para facilitar a tarefa de encontrar informações úteis aos usuários. Em outras palavras, IRSS tem seu foco nas funcionalidades de suporte aos usuários.

No contexto Web, IRSS são conhecidos como sistema de suporte a recuperação de informação na web (do inglês *Web-based Information Retrieval Support Systems* -

WIRSS). Segundo [Hoeber 2008], WIRSS aplicam métodos inteligentes e tecnologias baseadas na Web sobre o foco tradicional de consulta automatizada em coleções digitais para ajudar os usuários a especificar suas necessidades de informação, avaliar e explorar os resultados da pesquisa e gerir as informações que querem encontrar. Por exemplo, um WIRSS pode oferecer meios para responder a perguntas como: "Quantas vezes o Brasil ganhou a Copa do Mundo?" Entre os resultados retornados existirão aqueles que refletem o fato de que a Web sabe, por exemplo, que o Brasil sediará a Copa do Mundo de futebol de 2014 e que a Copa do Mundo de 2010 foi na África do Sul. Por outro lado, para um WIRSS com suporte a busca semântica, a resposta seria cinco (5).

Embora sejam apontados como a evolução natural na área de recuperação da informação, WIRSS introduzem novos desafios como precisamente apontados por [Yao e Yao 2003] e [Hoeber 2008]. O principal deles é a evolução dos tradicionais motores de busca (do inglês *Search Engines* - SE), baseado nos conceitos clássicos de IRS, para motores de suporte a busca (do inglês *Search Support Engines* - SSE), focados em fornecer diferentes funcionalidades de suporte aos usuários. Autores como [Zeng *et al.* 2009] e [Marchionini e White 2009] afirmam que uma SSE, além de oferecer típicas e tradicionais funções de busca e navegação, também deve fornecer funcionalidades de apoio, tais como organização, descoberta e visualização do conhecimento. Outro desafio está relacionado com a representação dos resultados. A típica representação baseada em listas de resultados só é efetiva quando a informação solicitada é bem específica. Entretanto, quando as consultas feitas pelos usuários são mal definidas, vagas ou até ambíguas, a lista fornece pouco ou nenhum suporte ao usuário na descoberta de informações relevantes. Um novo método para representar os resultados de pesquisa, proposto por [Tilsner *et al.* 2009], permite que os usuários tenham um papel ativo no processo de busca, fazendo seleções de alto nível usando clusters *fuzzy* de documentos e, assim, reduzindo o número de documentos irrelevantes dentro da lista de resultados.

Recentemente, uma metodologia diferente proposta por [Marchionini e White 2009] introduziu o conceito de sistema de suporte a busca de informação (do inglês *Information Seeking Support System* - ISSS), que enfatiza a necessidade de mudar da busca por informação para o apoio a busca de informação. Os autores argumentam que a busca da informação para aprendizagem, tomada de decisões e atividades mentais complexas que ocorrem durante longos períodos de tempo exigem o desenvolvimento de novas soluções especialmente concebidas para serviços de apoio aos usuários. Atualmente, trabalhos envolvendo ISSS abrangem uma vasta gama de funcionalidades como ContextMiner [Shah 2009], WolframAlpha [WolframAlpha 2009] e Relation Browser [Capra e Marchionini 2009].

3. ARAPONGA

A adoção de tecnologias de busca Web fez com que as pessoas esperassem acesso imediato e fácil a qualquer tipo de informação. Um caso especial é a segurança da informação. Embora haja um grande número de sítios Web construídos para gerenciar relatórios de vulnerabilidades, listas de computadores maliciosos (tipicamente contendo endereços IP, servidores DNS, domínios e ASN), gráficos e até estatísticas, os administradores de rede e

gerentes de TI são forçados a realizar exaustivas buscas para fundamentar qualquer informação.

Para resolver este problema, este trabalho desenvolveu uma ferramenta, **ARAPONGA**, baseada nos conceitos básicos da WIRSS e ISSS, capaz de proporcionar: (i) integração de conteúdo Web em um único lugar, (ii) um poderoso motor apoio a pesquisa focada em segurança, (iii) um acesso unificado e simples, com suporte para expressões lógicas e (iv) interfaces para lidar tanto com operadores humanos como sistema externos (outros motores de busca e ferramentas de tomada de decisão, por exemplo).

Mais especificamente, ARAPONGA oferece dois tipos de funcionalidades de apoio: apoio a buscas refinadas e apoio a análise do conhecimento. O primeiro descreve um conjunto de buscas que podem ser executadas sobre o conteúdo coletado de segurança da informação. Esta funcionalidade pode ser comparada com as tradicionais buscas disponíveis na maioria dos motores de busca da Internet. No entanto, ARAPONGA emprega o conceito de modelos (*templates*) de indexação com o objetivo de extrair melhor e de modo mais específico o conteúdo. Consequentemente, os resultados da busca tornam-se mais coerentes e pertinentes com os interesses dos usuários já que existe uma grande diversidade de complementos para ajudar na busca, tais como palavras-chave, tipos de páginas e domínio de conhecimento. Os *templates* serão explicados na próxima seção.

Por esta razão, ARAPONGA é capaz de responder questões específicas como:

- O meu endereço IP/servidor de domínio/ASN está relacionado a algum tipo de atividades suspeitas ou maliciosas, intrusão ou ataque relatado na Internet?
- Existe algum novo tipo de ataque relacionado a um tipo específico de serviço, porta ou protocolo?
- Qual a última aparição de uma anomalia ou vulnerabilidade?
- Quais são as vulnerabilidades relacionadas a um produto específico (software ou hardware) ou determinado fabricante?

A segunda funcionalidade é o apoio à análise do conhecimento, permitindo a construção de vários tipos de estruturas de conhecimento e gráficos estatísticos para representar um domínio específico. Por exemplo, ARAPONGA pode modelar e apresentar a distribuição de frequência de uma determinada vulnerabilidade. É importante ressaltar que esta funcionalidade é centrada no usuário uma vez que o sistema gera resultados de análises pertinentes para compreensão de ameaças de segurança em um determinado contexto.

3.1 Templates

Normalmente, as tecnologias de coleta Web (*crawling*) realizam a extração direta de conteúdo de páginas, sem considerar as palavras-chave presente no código-fonte HTML. Consequentemente, os índices gerados pelo processo de indexação são comuns ou genéricos, uma vez que se baseiam apenas no conteúdo ou tema das páginas.

Para resolver este problema e também melhorar os resultados do processo de indexação, ARAPONGA emprega o conceito de *templates*, um modelo para representar

conteúdos e fornecer padrões de visualização. No ARAPONGA, os *templates* têm palavras-chave relevantes que seguem um padrão pré-definido. Normalmente, os *templates* podem ser criados para representar um sítio ou domínio completo na medida em que os conteúdos e as estruturas são muitas vezes repetidos. No entanto, cada vez que um sítio ou domínio introduz estruturas diferentes, vários *templates* precisam ser gerados para representar essas diferenças.

Uma vez que os *templates* tentam estabelecer um relacionamento entre URLs e palavras-chave, após o processo de indexação, ao invés de existir um único espaço de busca, múltiplos espaços de busca são criados para diversidade, aumentando a probabilidade de um determinado tópico ser relevante. A Figura 1 exemplifica a diferença entre um espaço de busca normal e um espaço de pesquisa com *templates*.

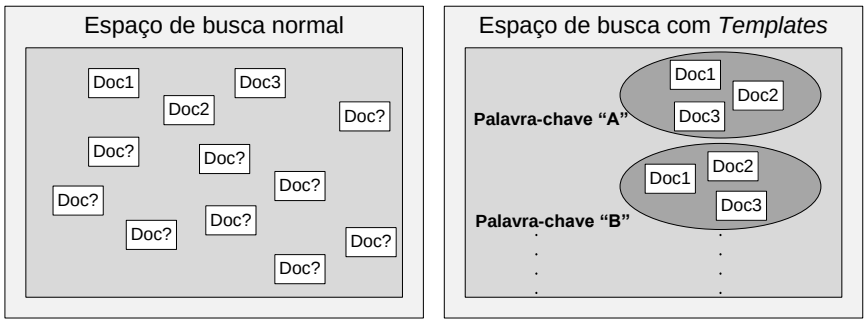

Figura 1. Comparação entre os espaços de busca normal e com templates.

O uso de *templates* oferece a possibilidade de identificar conteúdos mais específicos e indexá-los com uma palavra-chave que o identifica. Em comparação com um processo de indexação comum, onde o espaço de busca gerado é composto por todos os conteúdos, o processo de indexação usando *templates* gera um espaço de busca mais reduzido, composto pelo conteúdo indexado por palavras-chave.

Como resultado, o uso de *templates* permite um ganho de desempenho em termos de tempo de resposta e processamento. É importante ressaltar que o processo de pesquisa no espaço de busca normal e no com *templates* é o mesmo (em relação à pesquisa, triagem e classificação da página), o que realmente muda é o espaço de busca e o número de páginas retornadas.

3.2 Arquitetura

ARAPONGA tem uma arquitetura modular para facilitar modificações e adição de novos componentes. A Figura 2 mostra os componentes arquiteturais do ARAPONGA.

- **Crawler** – responsável por coletar páginas Web, metadados associados e informações contextuais.
- **Indexador**– responsável por indexar o conteúdo coletado pelo *Crawler*. Entretanto, ao invés de executar apenas uma indexação tradicional (baseada no conteúdo, URL, marcas de tempo e outros identificadores), este componente faz uso de *templates* para extrair diferentes identificadores e, consequentemente, permitir buscas

refinadas. O *Indexador* também é responsável por selecionar páginas Web que não serão indexadas por não apresentarem conteúdo relevante.

- **Motor de Busca** – responsável por receber consultas e retornar as respostas de maneira ordenada. Este componente executa duas atividades distintas: busca (*searching*) e ranqueamento (*ranking*).
- **Interface** – responsável por fazer a ligação (*front-end)* entre os usuários (operadores humanos e sistemas externos) e ARAPONGA (motor de busca), fornecendo diferentes tipos de interface de entrada e saída para execução de consultas e exibição das respostas. Este componente realiza duas funções principais: manipulação de consultas e visualização das saídas. A primeira é responsável por traduzir as consultas originais para o formato aceitável pelo motor de busca. A última exibe os resultados das consultas na forma de páginas Web, gráficos e listas XML ordenadas.

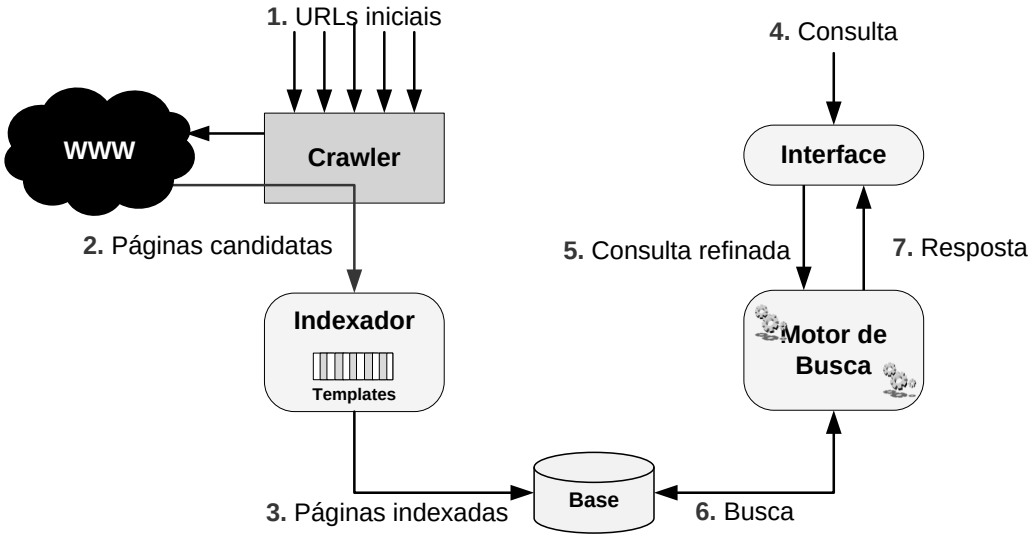

Figura 2. Visão geral da arquitetura ARAPONGA

3.3 Interação entre os Componentes

A interação entre os componentes do ARAPONGA podem ser divididos em duas fases. A primeira é o processo de coleta e indexação de páginas Web envolvendo o *Cralwer* e o indexador (etapas 1, 2 e 3 na Figura 2.). Na etapa 1, o *Crawler* é alimentado com as URLs iniciais (ou domínios) indicados como fontes de informação para as vulnerabilidades e anomalias. Na etapa 2, as páginas coletadas (páginas candidatas) são salvas e o *Indexador* usa os *templates* com indexar todo o conteúdo. Como resultado, na etapa 3, as páginas úteis e relevantes são guardadas na base de arquivos indexados.

A segunda fase envolve a consulta (etapas 4, 5, 6 e 7). Para ajudar o leitor ganhar familiaridade com o ARAPONGA, o seguinte exemplo é apresentado. Um gerente de TI, desejando saber se existem alertas TCP na porta 80 relacionadas a ataques, spam e *botnets* no período de 01 a 07 de Janeiro de 2010, faz uma consulta. Ao receber essa consulta, o componente de *Interface* a traduz e formula uma nova consulta de acordo com parâmetros

pré-definidos. Se alguma parte da pergunta não é entendida, isto é, não pode ser traduzida, a pergunta é descartado e uma mensagem de erro é retornada. Caso contrário, a nova consulta é gerada (**TCP 80** *–pageType* **attack,botnet,spam** *–date* **01/01/2010** *to* **01/07/2010)** e enviada ao motor de busca.

No motor de busca, a consulta passa por um processo de remoção de palavras irrelevantes (*stop-words*). Após isto, o motor de busca inicia a procura em todos as páginas Web armazenadas e indexadas, com o objetivo de verificar quais contém alguma informação relevante para a consulta, ou seja, TCP 80 em documentos relacionados a ataques, *botnet* ou spam, entre os dias 01 e 07 de Janeiro de 2010. Os documentos encontrados são listados e ordenados (em ordem decrescente de relevância) e então encaminhados para o componente Interface.

4. IMPLEMENTAÇÃO

Antes de iniciar a descrição da implementação, é necessário esclarecer alguns aspectos.

Primeiro, a ferramenta Nutch [Apache Nutch 2010], um *Web Crawler* de código aberto e capaz de executar funções como coleta e análise (*parsing*) de conteúdo em HTML e em outros formatos, foi escolhida para ser o componente *Crawler*. Desenvolvido em Java, Nutch utiliza a biblioteca Lucene [Apache Lucene 2010] para indexar conteúdos e tem sido amplamente usado em mecanismos de busca e indexação.

Outra escolha de projeto diz respeito à definição de onde coletar os conteúdos. Após um estudo que considerou métricas como relevância e completude da informação, tempo de atualização e acesso ao conteúdo, ARAPONGA utiliza dezenas de sítios Web que oferecem informações de vulnerabilidades, segurança e alertas sobre anomalias de tráfego. Entre os mais importantes, destacam-se ATLAS [Arbor Networks 2009], Secunia [Secunia 2009], US-CERT [US-CERT 2009] e Team Cymru [Team Cymru 2009].

4.1 Componentes

4.1.1 Crawler

No ARAPONGA, Nutch desempenha a função do componente *Crawler*. Para lidar com questões como a quantidade e a qualidade das informações coletadas, Nutch faz uso de filtros e limitadores. Por exemplo, já que a Internet tem um enorme número de páginas e uma rápida atualização, a probabilidade do conteúdo das páginas coletadas estarem desatualizadas é alta. Para evitar estes problemas, são empregados limitadores de profundidade e de amplitude, para evitar grandes desvios do ponto de partida e para restringir o número de link em cada página que podem ser referenciados. Além disso, filtros de URL também são usados para delimitar uma URL específica para consulta.

4.1.1 Indexador

Para implementar o *Indexador* foram utilizados: a linguagem JAVA (versão 1.6); a biblioteca Lucene [Apache Lucene 2010], um motor de busca de alta performance escrito

em Java que oferece simples eficientes algoritmos de busca; e Jericho HTML Parser [Jericho 2010], uma biblioteca Java usada para manipulação de documentos HTML.

Basicamente, a operação de indexação pode ser dividida em três passos:

- O Lucene recebe todo o conteúdo (páginas Web) coletado pelo *Crawler*.
- Para cada página, Jericho faz comparações entre os títulos das páginas e alguns identificadores predefinidos com o objetivo de identificar *templates* que ajudem a identificar a referência da página. Se algum *template* é encontrado, a página tem seu contexto extraído, identificado e cada bloco de informação (parte do conteúdo) é associado a uma palavra-chave. Por outro lado, se nenhum *template* for encontrado, a página é indexada de modo normal, apenas pelo conteúdo.
- Por fim, Lucene adiciona identificadores de *timestamp*, título e URL para controle interno do sistema.

Após estes passos, o Lucene indexa a página.

4.1.3 Motor de Busca

Similarmente aos outros componentes, o motor de busca também é implementado em Java.

O motor de busca responsável pelas funcionalidades de busca e ranqueamento, além da análise dos resultados da busca. Para executar essas atividades, também faz uso do Lucene. Basicamente, depois de receber a consulta, o motor de busca inicia o processo de comparação com todos os documentos indexados. Em seguida, os documentos retornados são ranqueados, ordenados e enviados ao componente *Interface*.

4.1.4 Interface

Interface é o último, mas não menos importante, componente do ARAPONGA. Responsável por intermediar a comunicação entre usuários e o motor de busca, ele fornece diferentes tipos de entradas e saídas para execução das consultas e exibição das respostas. Para atender tais funcionalidades, as seguintes consultas são suportadas:

1. *Simples*, que representa uma consulta comum composta por uma única entrada (o termo de interesse) e que é procurada em todo o conteúdo indexado. Essa consulta também permite o uso de intervalos de tempo (parâmetro ***-date***) para restringir a resposta. Como resposta, uma lista ordenada é retornada. *botnet* ou *Internet Explorer –date 04/01/2010 to 04/06/2010* são exemplos dessa consulta.
2. *Avançada*, que representa uma consulta detalhada e resultante do uso dos *templates*. Esta consulta permite a junção de diferentes parâmetros de entrada, objetivando restringir o espaço de busca e fornecer respostas mais corretas. Ela é composta pelo termo de interesse e algum(ns) dos seguintes parâmetros.
    a. field – indica que o termo de interesse deve ser obrigatoriamente encontrado junto com a(s) palavra(s)-chave especificada(s). No exemplo *AS3462 -field ASN date 02/01/2010 to 02/08/2010*, o parâmetro *-field* ASN indica que o termo AS3462 somente será procurado em documentos indexados onde a palavra-chave ASN está presente.

b. pageType – indica que o termo de interesse deve ser obrigatoriamente encontrado em páginas Web cujo tipo combine com o especificado como entrada. O exemplo *Storm Worm -pageType alerts,bulletin* exprime a obrigatoriedade do termo Storm Worm ser encontrado em páginas classificadas como alertas e boletins.

c. domain – indica que o termo de interesse deve ser obrigatoriamente encontrado em páginas Web cujo domínio combine com o especificado como entrada. No exemplo *Microsoft -domain secunia.com,cert.org*, o termo Microsoft somente será procurado em páginas Web pertencentes aos domínios especificados.

O resultado de uma consulta avançada é uma lista ordenada contendo onde (URLs) o termo foi encontrado. Percebe-se que as palavras-chave são extraídas de cada página durante o processo de indexação feito de acordo com os *templates* específicos.

3. *Maliciosa*, que representa uma consulta refinada (avançada). Seu objetivo é identificar se uma determinada entrada está relacionada com alguma atividade maliciosa. Ela recebe duas entradas: o termo de interesse e o parâmetro *-malicious*. A primeira entrada representa um endereço IP, um servidor, um domínio ou um ASN. Já a segunda deve ser acompanhada por uma lista de palavras-chave (*attacks*, spam, *phishing*, *botnet*, entre outras) que devem ser pesquisadas. O resultado é uma lista ordenada. Contudo, a lista é ordenada alfabeticamente de acordo com a atividade maliciosa encontrada. Este tipo de consulta é valiosa para investigar situações como, por exemplo, se um servidor SMTP está listado em algum sítio de blacklist. *gprt.ufpe.br -malicious spam,fastflux,CC,attack -date 01/04/2010 to 06/04/2010* é um exemplo desta consulta.

5. AVALIAÇÃO E RESULTADOS INICIAIS

5.1 Ambiente de teste

Todos os estágios de desenvolvimento, avaliação e testes do ARAPONGA foram realizados nos laboratórios do GPRT (Grupo de Pesquisa em Redes e Telecomunicações) da UFPE (Universidade Federal de Pernambuco). O ambiente de teste foi composto por um computador Intel Core2Duo T5300, 2 Gb de RAM, e 250 Gb de HDD, executando a distribuição do Linux Ubuntu 8.04. Quanto a conexão de rede, o GPRT mantém uma ligação de 1 Gbps como PoP-PE (ponto de presença da RNP).

Todos os experimentos foram conduzidos usando um conjunto de páginas coletadas desde 04 de Janeiro de 2010 até 06 de Junho de 2010, com a ferramenta Nutch capturando 150 referencias por página e com profundidade de 20 referencias.

5.2 Métricas de Desempenho

No intuito de avaliar os resultados do ARAPONGA, foram adotadas métricas tradicionais da área de recuperação da informação baseadas no conceito de relevância. As métricas são:

- **Precisão** – definida com a divisão dos documentos recuperados que são relevantes por todos os documentos recuperados.

$$\text{Precisão} = \frac{N_{recuperadas} \cap N_{relevantes}}{N_{recuperadas}}, \quad (1)$$

- **Abrangência** – definida com a divisão dos documentos relevantes recuperados pelo número total de documentos relevantes à consulta.

$$\text{Abrangência} = \frac{N_{recuperadas} \cap N_{relevantes}}{N_{relevantes}}, \quad (2)$$

De modo geral, precisão é o percentual de itens recuperados que são relevantes. Abrangência é o percentual de itens que foram recuperados. Por exemplo, uma consulta com precisão igual a 0,7 significa que 70% dos itens recuperados são relevantes, enquanto que uma consulta com abrangência igual a 0,7 significa que somente 70% dos itens são ou podem ser relevantes.

5.3 Resultados dos Experimentos

Para extrair resultados visíveis, todas as páginas Web coletadas foram indexadas de duas formas: uma usando o ARAPONGA e outra usando somente o Lucene. O objetivo foi estabelecer uma comparação entre eles e, assim, provar a importância e eficiência dos *templates* do ARAPONGA. A análise dos resultados revela uma diferença marcante entre as páginas indexadas pelo Lucene, a partir de agora referida como base geral, e as páginas indexadas pelo ARAPONGA. Enquanto a base geral gerencia 47.993 páginas, ARAPONGA gerencia 27.921 páginas. Tal diferença é atribuída diretamente à utilização de *templates*, que permite à extração de informações mais detalhadas e evita a indexação de páginas irrelevantes.

A Figura 3 mostra claramente a diferença entre as páginas indexadas pelo Lucene e ARAPONGA, em janeiro de 2010.

5.3.1 Desempenho

Para avaliar o desempenho, uma consulta pelo termo Microsoft com o parâmetro *-field* associado a *vulnerability* e *high severity* foi realizada. A idéia era encontrar graves vulnerabilidades envolvendo produtos Microsoft.

Como resultado, a consulta aplicada na base geral retornou 4.875 páginas que continham referências para Microsoft, onde apenas 25 páginas descreviam vulnerabilidades de alta gravidade. Assim, a precisão nesta base foi de 0,0051%, ou seja, apenas 25 páginas foram relevantes de um total de 4.875 documentos recuperados. A abrangência foi de 0,000521%, devido ao fato de apenas 25 páginas foram relevantes de um total de 47.993 documentos.

$$\text{Precisão} = \frac{4875 \cap 25}{4875} = \frac{25}{4875} = 0{,}0051$$

$$\text{Abrangência} = \frac{3309 \cap 25}{47993} = \frac{25}{47993} = 0{,}000521$$

Quanto à base ARAPONGA, a consulta também retornou 25 páginas contendo referências para Microsoft com vulnerabilidades de alta gravidade. Assim, a precisão alcançada na base foi de 100%. A abrangência foi de 0,000895%, uma vez que 25 páginas relevantes foram recuperadas de 27.921 disponíveis.

$$\text{Precisão} = \frac{25 \cap 25}{25} = \frac{25}{25} = 1$$

$$\text{Abrangência} = \frac{27921 \cap 25}{27921} = \frac{25}{27921} = 0.000895$$

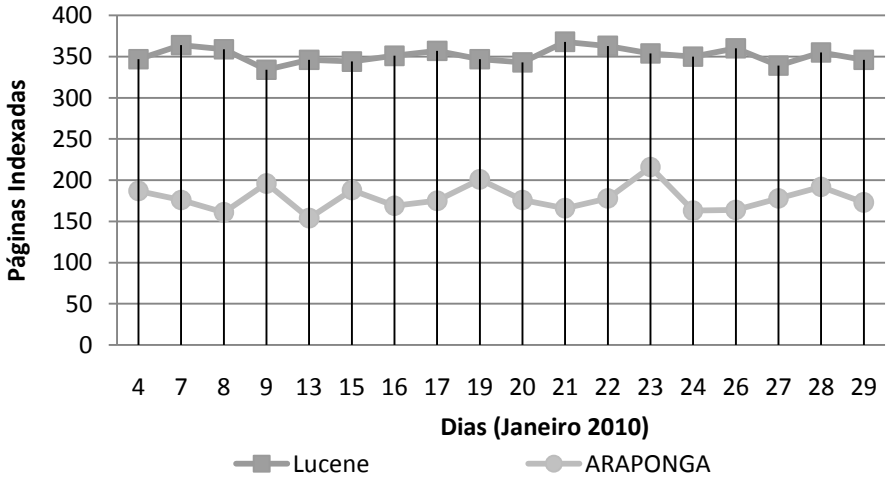

Figura 3. Comparação entre as bases do Lucene e ARAPONGA

5.3.2 Suporte a funcionalidades

Para avaliar o suporte às funcionalidades, duas consultas consideradas relevantes para operadores de rede e gerentes de TI foram realizadas.

A primeira consulta visa identificar ASes brasileiros relacionados com tráfego malicioso e anômalo. O resultado para a consulta *ASN Brazil –malicious ASN –date 01/04/2010 to 01/04/2010* é um grafo. A Figura 4 ilustra essa estrutura.

Já a segunda consulta mostra a evolução da informação recolhida. Seu objetivo é mostrar uma cronologia das aparições de determinado termo. A Figura 5 mostra as vulnerabilidades encontradas para o termo Microsoft Internet Explorer durante o período de 01 de Janeiro até 31 de Maio de 2010 (*Internet Explorer -field system_affected,software -date 01/01/2010 to 05/31/2010).*

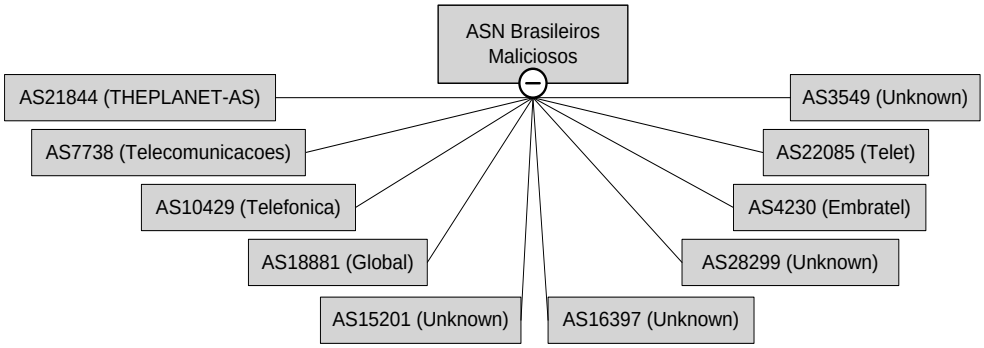

Figura 4. Grafo para ANS brasileiros listados em atividades maliciosas.

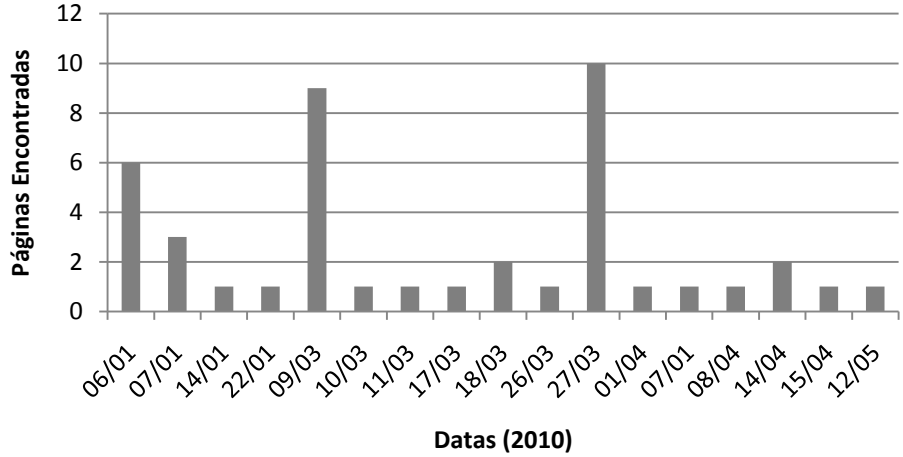

Figura 5. Linha de tempo das vulnerabilidades do Microsoft Internet Explorer

5.3.3 Caso de uso

Para finalizar a avaliação do ARAPONGA, uma experiência final foi realizada. Supondo que um gerente de TI desejos para descobrir se um determinado endereço IP do seu domínio está relacionado com as atividades anômalas, uma consulta típica seria a seguinte: *190.2.29.193 -malicious vulnerability -urlSummary URLSummary.xml*, onde o último parâmetro irá gerar a resposta (uma lista de URLs) no formato XML. O resultado é apresentado na Figura 6.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <URL_Summary>
  <url>http://atlas.arbor.net/vuln/CVE-2009-3103</url>
  <url>http://atlas.arbor.net/vuln/CVE-2008-4834</url>
  <url>http://atlas.arbor.net/vuln/CVE-2007-5381</url>
  <url>http://atlas.arbor.net/vuln/CVE-2007-0247</url>
 </URL_Summary>
```

Figura 6. Exemplo de uma resposta em formato XML

Com base nesta resposta, o gerente de TI decide realizar uma consulta que resuma todas as vulnerabilidades listadas nas URLs da resposta anterior. Essa nova consulta tem

duas entradas: o termo de interesse (cada URL retornada na Figura 6) e o parâmetro -*summary_vulnerability*.

Essa consulta percorre todo o conteúdo indexado a procura do termo, retornando um arquivo XML contendo as seguintes informações conforme a Figura 7.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Summary_Vulnerability>
- <Title ID="CVE-2009-3103">
  - <Date></Date>
  <Description>Array index error in the SMBv2 protocol implementation</Description>
  <Where>From network</Where>
  <Services>TCP/445</Services>
  -<Products></Products>
  - <References></References>
  </Title>
- <Title ID="CVE-2007-0247">
  - <Date></Date>
  <Description>squid/src/ftp.c allows remote FTP servers to cause a DoS</Description>
  <Where>From network, remote network</Where>
  <Services>TCP/3128</Services>
  <Products>Squid 2.6.STABLE6</Products>
  - <References></References>
  </Title>
  </Summary_Vulnerability>
```

Figura 7. Exemplo de uma consulta com o parâmetro -summary_vulnerabilities.

6. CONCLUSÃO

Este artigo apresentou uma ferramenta projetada para obter informações de vulnerabilidades e estatísticas de tráfego anômalo na Internet. ARAPONGA é a implementação de um prova de conceito que combina o uso de técnicas de mineração de dados e modelos (*templates*) para expandir a capacidade de indexação sobre informações de segurança e, consequentemente, permitir consultas diferenciadas e mais focadas.

Para este fim, ARAPONGA fez as seguintes contribuições. Os conceitos de WIRSS e ISSS são aplicados para fornecer o suporte a funcionalidades que vão além de um sistema tradicional de busca e indexação. ARAPONGA concentra informações de segurança disponíveis em muitos locais em apenas uma base, que contém apenas dados relevantes. Dezenas de sítios Web sobre vulnerabilidades e tráfego Internet foram avaliadas em termos de completude e acesso ao conteúdo e somente as mais adequadas para integrarem a esta solução foram escolhidas.

ARAPONGA não pretende fornecer um mecanismo de busca semântica, mas um pequeno passo é dado por este trabalho em direção a recuperação, monitoração, gerenciamento e visualização orientada ao usuário de informação disponível na Web.

Como trabalhos futuros, pretende-se:

- Desenvolvimento de um esquema para geração automática dos *templates*;

- Adoção de novas técnicas que possam ser agregadas aos *templates*, visando melhorar a indexação e busca das informações;
- Estudo sobre a usabilidade da interface, com usuários reais, visando avaliar questões como eficiência, taxa de erros e grau de satisfação.

7. REFERÊNCIAS

Apache Lucene. (2010). Lucene Java. http://lucene.apache.org/java/docs.

Apache Nutch. (2010). *Nutch*. http://lucene.apache.org/nutch.

Arbor Networks. (2010) *Atlas*. http://atlas./arbor.net.

Capra, R. e Marchionini, G. (2009) Faceted Exploratory Search Using the Relation Browser. In *NSF Workshop on Information Seeking Support Systems*, pp. 81--83.

CISCO. (2009) *Cisco Security Center*. http://tools.cisco.com/security/center/home.x.

Hoeber, O. (2008) Web Information Retrieval Support Systems: The Future of Web Search. In *2008 International Workshop on Web Information Retrieval Support Systems*, páginas 29-32.

Jericho. (2010) Jericho HTML Parser. http://jericho.htmlparser.net/docs/index.html.

Marchionini, G. e White, R. W. (2009) Information Seeking Support System. *IEEE Computer*, 42 (3), páginas 30-32.

NIST. (2009) *National Vulnerability Database (NVD)*. http://nvd.nist.gov.

OSVDB. (2009) *Open Source Vulnerabilities Database*. http://www.osvdb.org.

Secunia. (2009) *Secunia Advisories*. http://secunia.com/advisories.

Shah, C. (2009) ContextMiner: Explore Globally, Aggregate Locally. *IEEE Computer*, página 94.

Team Cymru. (2009) Team Cymru. http://www.team-cymru.org/.

Tilsner, M., Hoeber, O., e Fiech, A. (2009) CubanSea: Cluster-Based Visualization of Search Results. *IEEE Computer*, vol. 42, no. 3, páginas 108-112.

US-CERT. (2009) *US-CERT*. https://www.us-cert.gov.

WolframAlpha. (2009) *WolframAlpha*. http://www.wolframalhpa.com.

Yao, J. T. e Yao, Y. Y. (2003) Web-based Information Retrieval Support System: Building Research Tools for Scientists in the New Information Age. In *IEEE/WIC International Conference on Web Intelligence*, páginas 570-573.

Zeng, Y., Yao, Y., e Zhong, N. (2009) DBLP-SSE: A DBLP Search Support Engine. In *IEEE/WIC/ACM International Conference on Web Intelligence*.