Uma Plataforma de Modelagem Colaborativa de Ontologias para Análise de Programas Maliciosos

Antonio Carlos de Marchi

Maio/2017

Dissertação de Mestrado em Ciência da Computação

Uma Plataforma de Modelagem Colaborativa de Ontologias para Análise de Programas Maliciosos

Esse documento corresponde a Dissertação apresentada à Banca Examinadora para a defesa de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista.

Campo Limpo Paulista, 12 de Maio de 2017.

Antonio Carlos de Marchi

Rodrigo Bonacin (Orientador)

PÁGINA DE APROVAÇÃO

Faculdade Campo Limpo Paulista - FACCAMP

Uma Plataforma de Modelagem Colaborativa de Ontologias para Análise de Programas Maliciosos

Antonio Carlos de Marchi

Campo Limpo Paulista, 12 de maio de 2017.

BANCA EXAMINADORA

Prof. Dr. Rodrigo Bonacin

(Orientador – FACCAMP)

Prof^{a}. Dr^{a}. Ana Maria Monteiro

(FACCAMP)

Prof. Dr. André Ricardo Abed Grégio

(UFPR)

FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Câmara Brasileira do Livro, São Paulo, Brasil.

Marchi, Antonio Carlos de

Uma plataforma de modelagem colaborativa de ontologias para análise de programas maliciosos. Campo Limpo Paulista, SP: FACCAMP, 2017.

Orientador: Profº. Dr. Rodrigo Bonacin

Dissertação (Programa de Mestrado em Ciência da Computação) – Faculdade Campo Limpo Paulista – FACCAMP.

1. Desenvolvimento colaborativo. 2. Crowdsourcing. 3. Análise de Malware. 4. Ontologias. I. Bonacin, Rodrigo. II. Campo Limpo Paulista. III. Título.

CDD-005.43

Resumo: A análise de programas maliciosos (*malware*) é um desafio de difícil solução, em função do alto número e variedade de *malware* produzidos diariamente. Além disso, os códigos maliciosos atuais possuem notável capacidade de mutação automática, o que torna as técnicas de detecção atuais, baseadas em assinaturas de código, obsoletas. Por meio da análise dos comportamentos dos *malware* existentes, podemos detectar suas variantes e predizer variantes futuras, independentemente de suas codificações internas, assim esses comportamentos podem ser utilizados por especialistas em engenharia de conhecimento e transformados em classes e regras em ontologias *Web*. Nesta dissertação, é apresentado um modelo de *crowdsourcing* para evolução de ontologias que representam o comportamento de *malware*, este modelo é materializado em um sistema que implementa uma plataforma colaborativa. Nesta plataforma, os usuários podem evoluir ontologias e cooperar relatando comportamentos suspeitos de softwares que serão analisados e classificados como maliciosos ou não maliciosos. O objetivo é explorar mecanismos de *crowdsourcing* para criar de maneira colaborativa novas classes e regras capazes de identificar cada vez mais programas potencialmente maliciosos. Com o desenvolvimento de um sistema de fácil utilização espera-se contribuir com um modelo que possa ser utilizado em outros sistemas de análise de *malware*, bem como responder rapidamente a novas variedades de *malware*. O sistema foi analisado por especialistas do domínio com o objetivo de validar e apontar seu potencial e limitações.

Palavras-chave: *Desenvolvimento Colaborativo, Crowdsourcing, Análise de Malware, Ontologias.*

Abstract: *The analysis of malicious programs (malware) is a challenge of hard solution, since there is a huge number and variety of malicious programs produced every day. Besides, new malicious codes have automatic mutation capabilities, turning the current detection techniques, which are based on code signatures, in obsolete ones. By means of behavior analysis of existing malware, independently of internal codifications, we are able to detect and predict new variants. Knowledge engineering experts can also model ontology classes and rules to represent malware behavior and their variations. In this dissertation, we present a crowdsourcing model for the evolution of ontologies that represent malware behavior, this model implemented in a system based on collaborative platform. Users are able to evolve ontologies and cooperate by sharing suspect behaviors, which will be analyzed and classified as malicious (or not) by using the proposed collaborative platform. The object is to make use of crowdsourcing mechanisms to create collaboratively new classes and rules able to identify potential malicious programs. With a user-friendly system, we expect to contribute with a model that can be used in other malware analysis systems, as well as to quickly respond to new malware variants. The system was analyzed by domain experts with the objective of validating it and pointing out its potentials and limitations.*

Keywords: *Collaborative Development, Crowdsourcing, Malware Analysis, Ontology.*

SUMÁRIO

1. Introdução 1

1.1 Contexto e Motivação 1

1.2 Problemática e Justificativa 4

1.3 Objetivos, Contribuições e Métodos 6

1.4 Estrutura da Proposta 8

2. Referencial Teórico e Metedológico 10

2.1 Segurança em Sistemas de Informação 10

2.1.1 Ameaças, Ataques e Prevenção 11

2.1.2 *Malware* 13

2.2 *Web* Semântica e Ontologias 16

2.3 Sistemas Colaborativos e *Crowdsourcing* 20

3. Trabalhos Relacionados 22

3.1 Cenários da Pesquisa Realizada 23

3.2 Análise e Discussão Sobre os Artigos Relacionados 30

3.2.1 Uso de Ontologias na Modelagem e Análise de *Malware* 30

3.2.2 Uso de Técnicas de *Crowdsourcing* na Modelagem de Ontologias 36

3.2.3 Uso de Técnicas de *Crowdsourcing* na Análise e Detecção de *Malware* 39

3.3 Considerações Finais da Análise de Trabalhos Relacionados 44

4. Um Modelo de Plataforma Colaborativa Para Evolução de Ontologias de Análide de *Malware* 45

4.1 Design do Modelo de *Crowdsourcing* 45

4.2 Descrição do Modelo de *Crowdsourcing* 49

5. Desenvolvimento do Sistema CrOMa (*Crowdsourcing* Para Ontologias de *Malware* .. 57

5.1 Descrição da Arquitetura do Sistema. 58

5.2 Tecnologias Utilizadas 65

5.3 Descrição das Interfaces do Sistema 66

6. Avaliação e Discussão 80

6.1 Método de Avaliação 80

6.2 Resultados 83

6.3 Discussão 94

7. Conclusão 98

7.1 Contribuições 98

7.2 Trabalhos Futuros 100

7.3 Considerações Finais 101

Referências 102

Apêndice I – Artigo Publicado 111

Apêndice II – Artigo Aceito 112

LISTA DE TABELAS

Tabela 3.1 - Quantidade de artigos selecionados para busca: *Analysis Malware*. 24

Tabela 3.2 - Quantidade de artigos selecionados para busca: *Analysis Behavior*. 24

Tabela 3.3 - Quantidade de artigos selecionados para busca: Ontology. 25

Tabela 3.4 - Quantidade de artigos selecionados para busca: *Crowdsource(ing)* Ontology 25

Tabela 3.5 - Quantidade de artigos selecionados para busca: *Crowdsource(ing)* e *Malware*.....26

Tabela 3.6 - Quantidade de artigos selecionados para busca: Ontology *Malware*.....26

Tabela 3.7 - Quantidade de artigos selecionados para busca: Ontology, *Malware* and *Crowdsource (ing)*.....27

Tabela 3.8 - Síntese dos Resultados por Base de Indexada.....27

Tabela 3.9 - Títulos dos Artigos, Base Indexada e Palavras Chaves Utilizadas 29

Tabela 3.10 - Pontos Fortes e Limitações de Trabalhos sobre Ontologias e *Malware* 35

Tabela 3.11 - Pontos Fortes e Limitações de Trabalhos sobre *Crowdsourcing* e Ontologias.....39

Tabela 3.12 - Pontos Fortes e Limitações de Trabalhos sobre *Crowdsourcing* e *Malware* 43

Tabela 4.1 - Utilização e ação dos Especialistas com as Contribuições.....56

Tabela 6.1 - Média de Porcentagem das 5 perguntas sobre as funcionalidades do sistema 88

Tabela 6.2 - Média de Porcentagem das 5 perguntas sobre a usabilidade do sistema.....91

Tabela 6.3 - Avaliador e Tipos de Respostas 92

LISTA DE FIGURAS

Figura 2.1 - Arquitetura da *Web* Semântica 17

Figura 2.2 - Exemplo de hierarquia de classes 19

Figura 4.1 - Modelo Conceitual de *Crowdsourcing* 46

Figura 4.2 - Diagrama de cadastro do Usuário 49

Figura 4.3 - Diagrama de acesso do usuário iniciante 50

Figura 4.4 - Diagrama de acesso do usuário médio 51

Figura 4.5 - Diagrama de acesso do usuário avançado 52

Figura 4.6 - Diagrama de acesso do usuário expert 53

Figura 4.7 - Diagrama de acesso do usuário moderador 54

Figura 5.1 - Visão Geral da Arquitetura do Sistema CrOMa 61

Figura 5.2 - Diagrama com os módulos do sistema 65

Figura 5.3 - Site de apresentação do Sistema CrOMa 67

Figura 5.4 - Tutoriais de uso do site do Sistema CrOMa 68

Figura 5.5 - Página de Cadastro de Usuários 69

Figura 5.6 - Liberação de Usuários 69

Figura 5.7 - Recebimentos de contrasenha via *SMS* 70

Figura 5.8 - Página de *Login* 70

Figura 5.9 - Menu principal do Sistema CrOMa 71

Figura 5.10 - Página inicial do Fórum de Discussão 72

Figura 5.11- D*ownload* Versões 72

Figura 5.12 - *Upload* de Documentos 73

Figura 5.13 - Interface de exclusão de versões.....73

Figura 5.14 - Visualizações das contribuições efetuadas pelo usuário.....74

Figura 5.15 - Contribuição de regras.....74

Figura 5.16 - Exclusão de regras 75

Figura 5.17- Interface para avaliar regras.....75

Figura 5.18 - Visualização do ambiente de aprovações 76

Figura 5.19 - Contribuição de classes.....76

Figura 5.20 - Visualização de classes.....77

Figura 5.21 - Interface edição das contribuições de classes 77

Figura 5.22 - Visualização de classes para aprovações.....78

Figura 5.23 - Visualização das aprovações de classes.....78

Figura 6.1 - Primeira parte do Formulário de Avaliação (Perfil Profissional).....81

Figura 6.2 - Segunda parte do Formulário de Avaliação (Funcionalidade do Sistema)...82

Figura 6.3 - Terceira parte do Formulário de Avaliaçao (Usabilidade do Sistema).....82

Figura 6.4 - Grau de Instrução dos Participantes.....83

Figura 6.5 - Área de atuação e experiência em segurança da informação.....84

Figura 6.6 - Respostas para a pergunta 3.1 do formulário.....85

Figura 6.7 - Resposta para a pergunta 3.2 do formulário 85

Figura 6.8 - Respostas para a pergunta 3.3 do formulário.....86

Figura 6.9 - Respostas para a pergunta 3.4 do formulário.....87

Figura 6.10 - Respostas para a pergunta 3.5 do formulário.....87

Figura 6.11 - Respostas para a pergunta 4.1 do formulário.....88

Figura 6.12 - Respostas para a pergunta 4.2 do formulário.....89

Figura 6.13 - Respostas para a pergunta 4.3 do formulário.....89

Figura 6.14 - Respostas para a pergunta 4.4 do formulário.....90

Figura 6.15 - Respostas para a pergunta 4.5 do formulário.....90

Figura 6.16 - Resposta para a pergunta 3.6 92

Figura 6.17 - Resposta para a pergunta 4.6 93

Figura 6.18 - Resposta para a pergunta 5.1 93

Figura 6.19 - Classificação das respostas da pergunta 5.2 94

Glossário

AIDE - *Advanced Intrusion Detection Environment*

API - *Application Programming Interface*

BET- *Behavior Extraction Tool*

Crowd MAV- *Applying Crowdsourcing to Machine Learning Anti-Virus System*

CSS - *Cascading Style Sheets*

CUD - *Crowdsource for URL Spam Detection*

DDoS - *Distributed Denial of Service*

DoS - *Denial of Service*

FTP - *File Transfer Protocol*

IP- *Internet Protocol*

IRC - *Internet Relay Chat*

ISO - *International Organization for Standardization*

HTML - *Hyper Text Markup Language*

HTTPS - *Hyper Text Transfer Protocol Secure*

JXTA - (*juxtapose) Language and Platform Independent Protocol for P2P*

LLT- *Log Loader Tool*

MP - Conjunto de Subprocesso

MVC - *Model View Controller*

OWL - *Web Ontology Language*

PM - Processo de Malware

PHP - *Hypertext Preprocessor*

P2P- *Peer to Peer*

RAT- *Risk Analysis Tool*

RDF - *Resource Description Framework*

SMS - *Short Message Service*

SPAM - *Sending and Posting Advertisement in Mass*

SNOMED CT - Nomenclatura sistematizada de termos clínicos de medicina

SWRL - *Semantic Web Rule Language*

XML - *eXtensible Markup Language*

WC3 - Consórcio *Word Wide Web*

Web (www)-*World Wide Web*

Dedicatória

Dedico este trabalho à minha mãe Maria D. C. de Marchi (*in memorian*), pelo maior exemplo de amor e perseverança; ao meu pai José A. de Marchi, pela sua imensa generosidade; à minha esposa Vanessa P. C. de Marchi, pelo apoio incondicional e paciência me ajudando a percorrer este caminho; à minha irmã Valéria A. M. Silveira, pelo apoio direto e indireto, à Deus que me deu sabedoria e oportunidades; e aos meus filhos Vinicius, Miguel e Maria Gabriella os maiores presentes da minha vida.

Agradecimentos

Ao Prof. Dr. Rodrigo Bonacin (orientador), pela confiança e oportunidade de trabalhar ao seu lado e me ajudar a crescer pessoalmente e profissionalmente compartilhando seu enorme conhecimento, por acreditar em meu potencial de uma maneira que eu mesmo não acreditava, pela paciência nos momentos de desânimo, excelente orientação e pela grande amizade que formamos neste período.

A meu amigo, companheiro e irmão Edemar Mendes Perciani, que através de nossas longas conversas conseguimos manter a alegria e a estabilidade neste momento difícil, uma amizade que se transformou em irmandade.

Aos meus companheiros de mestrado que durante estes 3 anos compartilhamos angústias, dificuldades, mas também muitas alegrias e troca de conhecimentos, momentos maravilhosos que dividimos juntos em especial à Bruno Alves, Leonardo A. Cortez, Humberto Zanetti, James Mendes U. Martins, Maurício Mendes Faria e Ricardo J. Araújo.

Aos meus professores de graduação Ms. Anselmo Couto, Ms. Sergio Furgeri e Dr. Leandro Marques pelos ensinamentos que moldaram o que sou hoje e principalmente pelo incentivo de seguir a carreira acadêmica.

A todos os alunos de graduação, aos mestres e doutores que contribuíram fazendo a avaliação do sistema, que foi de suma importância para a conclusão da dissertação.

A todos os funcionários da FACCAMP, principalmente aos professores do mestrado os quais passaram grande conhecimento e aprendizado em especial a Prof ª. Drª. Ana Maria Monteiro pelos diálogos sempre proveitoso e incentivador, pela boa educação e pela alegria sempre presente em seu rosto, e ao Prof. Dr. Marcelo de Paiva Guimarães que com seu senso crítico me ajudou a entender melhor um programa de mestrado e pelas informações as quais me passou ajudando no meu crescimento, meu muito obrigado.

As secretárias do mestrado da FACCAMP Bruna, Talita e Tatiane que com muita disposição sempre resolveram os problemas burocráticos da melhor e mais rápida maneira possível.

Ao Prof. Dr. André Grégio e a Profª. Drª. Ana Maria Monteiro por aceitarem o convite de participar da banca da minha defesa.

A todos que direta e indiretamente participaram desta minha conquista gostaria de agradecer um a um mais com tantas pessoas se torna impossível.

Meu muito obrigado a todos.

“Talvez não tenhamos conseguido fazer o melhor, mas lutamos para que o melhor fosse feito, não somos o que deveríamos ser e não somos o que iremos ser, mas graças a Deus não somos o que éramos. ”

Martin Luther King

1. Introdução

A integração e interoperabilidade de sistemas computacionais pela Internet transformaram seus usuários em potenciais vítimas para diversos tipos de ataques. Os programas maliciosos (*malware*) atuais têm características que englobam diversos tipos de comportamentos em uma única execução, não sendo, portanto, mais passíveis de classificações tradicionais que os dividem em vírus, cavalos de *tróia*, *worms*, etc. A análise de *malware* requer uma profunda compreensão de seus possíveis comportamentos, de forma detalhada o suficiente para separar programas suspeitos em benignos ou maliciosos (Grégio e outros, 2014).

Sistemas *antivírus* puramente baseados em assinaturas não são suficientes para detectar novas variantes de *malware*, que utilizam métodos como a ofuscação e variações dinâmicas de códigos. Portanto, a análise do comportamento de *malware*, se faz necessária para a construção de sistemas mais seguros, bem como o desenvolvimento de futuras ferramentas de detecção. Ontologias podem ser utilizadas para descrever o conhecimento sobre o comportamento dos softwares maliciosos atuais, uma vez que elas são estruturas para representação de conhecimento por meio de uma modelagem explícita de uma conceptualização compartilhada (Guarino, 1998). Entretanto, a manutenção e evolução de ontologias na velocidade requerida para a análise de *malware* é um desafio. Uma alternativa viável é a utilização de *crowdsourcing* e contar com o auxílio da inteligência coletiva para evoluir a ontologia. *Crowdsourcing* é um termo recente que engloba diversas práticas ou qualquer atividade colaborativa visando criação e inovação baseada na internet (Estellés-Arola e González-Ladrón-de-Guevara, 2012). Assim, esta dissertação propõem-se a concepção e construção de uma plataforma colaborativa baseada em mecanismos de *crowdsourcing* para evolução de ontologias que modelam o comportamento de *malware*.

1.1. Contexto e Motivação

Os riscos para usuários de sistemas computacionais estão cada vez mais evidentes; atualmente são comuns, por exemplo, roubo de informações em documentos, credenciais bancárias, invasão de privacidade, entre outras ameaças. Boa parte do problema é decorrente do aumento do número e complexidade de *malware*, bem como mudanças no

comportamento dos usuários. *Cybers* criminosos hoje são “profissionais” organizados, com grande potencial para realização de crimes (Al-Bataineh e White, 2012).

Isso também se traduz em uma ameaça na dimensão social, uma vez que as pessoas estão cada vez mais dependentes de sistemas computacionais para as tarefas do dia a dia, assim como as organizações estão dependentes de transações em rede. Dados da Febraban (2014), por exemplo, indicam que 47% das transações bancárias do ano de 2013 foram feitas usando Internet ou *mobile banking*, isso representa um aumento de 27% em relação ao ano de 2009.

Neste trabalho, são consideradas *malware* quaisquer aplicações ou programas que possuem a finalidade de agir de forma maliciosa. *Malware* é uma definição genérica que engloba todo tipo de ameaça em informática (Gavrilut e outros, 2009). *Malware* são ameaças contínuas contra a segurança de sistemas conectados, para evitar a detecção e análise, atacantes evoluem constantemente os códigos maliciosos (Filho e outros, 2009).

O aumento da interação de dispositivos via Internet, junto com a quantidade de serviços e falta de conhecimento adequado por parte dos usuários contribuem para ataques por meio de *malware* (Filho e outros, 2010). A motivação por trás desses ataques inclui violar a integridade, confidencialidade ou disponibilidade de um sistema; tais fatores aliados à complexidade dos sistemas computacionais atuais contribuem com o aumento da incidência de ataques.

Milhares de códigos maliciosos são criados diariamente e para tanto, se faz necessário analisá-los. A análise do código e comportamento malicioso visa o entendimento profundo do funcionamento de um *malware* (Filho e outros, 2009), e a partir desta análise é possível projetar sistemas mais seguros e nos proteger de ataques futuros. Além disto, para realizar detecção de *malware* é necessário conhecer seus possíveis comportamentos e assim concluir se um dado programa é benigno ou malicioso. Existem *malware* complexos, difíceis de serem identificados. Em função de suas tecnologias avançadas, contornam os mecanismos de segurança para atingirem seus objetivos. A análise de comportamento malicioso ainda permite visualizar comportamentos suspeitos de software ditos como “não maliciosos”.

O Stuxnet (Tecmundo, 2010), é um exemplo da complexidade de ataques por *malware*, nesse caso, não se sabe quando começaram os ataques e muito menos quem

foram os responsáveis. Havia apenas suposições sobre um grupo de profissionais, pois a ação exigiu conhecimento profundo em várias áreas, que usuários domésticos não possuíam (Tecmundo, 2010). O Stuxnet foi um dos ataques mais sofisticado já identificado, atingindo as instalações nucleares iranianas (Masood e Anwar, 2011). De acordo com dados da *Symantec®*, atacou cerca de 100.000 computadores e a maior incidência foi no Irã com 58% das ocorrências, mas também houve ataques na Indonésia, Índia e Azerbaijão (Falliere e outros, 2014). Algum tempo depois foi identificada uma variação do Stuxnet chamada de Duqu (Bencsáth e outros, 2012). Esta variação tinha como alvo as centrífugas da usina de enriquecimento de urânio de Natanz. O ataque bloqueava a saída de gás das centrífugas fazendo a pressão subir e danificar o equipamento de modo que os engenheiros não percebiam o que havia algo errado (GizModo Brasil, 2013).

Existem muitos casos de *malware* que causaram impactos aos usuários domésticos, o *malware* "melissa" foi um exemplo típico. Este *malware* é distribuído como anexo de um e-mail, que ao ser aberto copia os primeiros 40 endereços do usuário e reenvia o *malware*, chega à caixa de entrada do usuário de várias formas com assunto "Minhas imagens", "Mensagem importante", etc., sem texto algum, mas com um anexo (Security-faqs, 2010).

Em todos os lugares onde se utilizam sistemas computacionais usuários estão expostos às ameaças, que vem de várias formas com os mais variados propósitos, portanto a *cyber* segurança torna-se cada vez mais necessária. Devido à complexidade do comportamento de novos *malware*, foi construído um sistema que implementa uma plataforma colaborativa onde usuários poderão contribuir para melhorar a análise de *malware*, aumentando e aprimorando assim as regras, classes e axiomas lógicos já existentes na ontologia. Nessa plataforma *crowdsourcing*, o usuário especialista (ou não) pode contribuir com seu conhecimento.

1.2. Problemática e Justificativa

Mecanismos de defesas *anti-malware* são essenciais para conter as ameaças nas redes. A maioria desses mecanismos é baseada em assinaturas para a detecção de *malware* (Martins e outros, 2014). Nesses mecanismos é comum fazer uso de uma base de dados contendo trechos de código retirados de cada *malware*, que associada a um processo de busca de programas suspeitos, permitindo a identificação de códigos maliciosos (Ask, 2006). Os mecanismos de defesas deveriam também se concentrar em outra questão fundamental para segurança, que é identificar vulnerabilidades que possam ser exploradas por *malware*, existem mecanismos que realizam esta tarefa, porém nem sempre são eficazes, além da propriedade de mutação que esses códigos possuem. Uma vez que a defesa baseada em trechos de códigos é limitada ao combater *cybers* criminosos, pois, os novos programas maliciosos têm uma estrutura bem projetada com elaboração personalizada que podem burlar os sistemas de detecção baseado na análise de códigos.

Os novos invasores, projetistas de novo *malware*, não procuram os "furtos" de maneira rápida, mas de forma mais lenta eles ocupam dispositivos dentro de uma rede (Cisco, 2014), para poder atacar de forma contínua. Atacantes agem para evitar a detecção de assinaturas com *malware* muito complexos e com uma ampla gama de técnicas de ofuscação de código para dificultar a detecção. Isso nos leva a uma problemática relacionada ao estudo do comportamento de *malware* para minimizar as vulnerabilidades dos sistemas computacionais.

Quando um *malware* não tem uma assinatura conhecida, nos resta conhecer seu comportamento dentro do sistema. Um dos métodos para a solução desse problema é o levantamento das ações no sistema da vítima (Filho e outros, 2010), isto é, a análise do comportamento do programa para identificar suas ações suspeitas. A análise de comportamento de *malware* é um passo importante para auxiliar na descoberta de padrões suspeitos, mas ao mesmo tempo é uma tarefa difícil, pois depende do monitoramento do sistema da vítima durante a infecção (Grégio e outros, 2014).

Ao comparar a sequência de ações de um código, é possível detectar variantes de classes de *malware*, dificultando os artifícios dos códigos maliciosos para burlar a detecção. Um dos mecanismos mais usados é o metamorfismo, que é a alteração dos códigos dos *malware* à medida que a infecção ocorre (Grégio e outros, 2014). Existem

ainda outras formas de ofuscação como os *malware* polimórficos, *malware* oligomórfico, *malware* cifrados, entre outras. Tais artifícios dificultam em muito a detecção, principalmente via mecanismos que utilizam somente assinaturas.

Neste contexto, denominações e classificações de *malware* "tradicionais" (*e.g.*, Vírus, Cavalo de *Tróia*, *Worm*) não são mais relevantes, pois a classificação pela ação individual do *malware* pode ser ofuscada e alguns *malware* podem ter ações variadas concomitantes. Outro fator que se deve levar em consideração é que *malware* utilizam mecanismos para se atualizar automaticamente para uma nova versão em curtos períodos de tempo (Gavrilut e outros, 2009).

Mitre (2010) destaca que se faz necessária a construção de uma linguagem comum sobre os conceitos de *malware* e segurança em geral, para assim, desenvolver um entendimento comum na área. Para a representação do conhecimento da área é necessária a construção de mecanismos complexos, tais como ontologias. O uso de regras e mecanismos de inferência lógica, como a linguagem *SWRL*[1] (*Semantic Web Rule Language*), ainda pode viabilizar a especificação do comportamento complexo e dinâmico dos *malware* atuais, tal como os destacados nos últimos parágrafos.

Para a modelagem, se faz necessário um esforço conjunto que vai além da percepção individual de um único pesquisador. Além disto, o caráter dinâmico do comportamento dos *malware* torna necessário o contínuo aprimoramento dos modelos de representação de conhecimento e regras. Tais problemas podem ser minimizados com técnicas que viabilizam o trabalho colaborativo, como por exemplo, técnicas de *crowdsourcing*. Nesta dissertação, utilizou-se como ontologia inicial uma previamente construída (Grégio e outros, 2016), com regras para análises de programas potencialmente suspeitos. Para sua modelagem, foram analisados comportamentos *malware* para descrever uma hierarquia e elaborar novos conceitos além do compartilhamento de informações por meio de uma base compartilhada.

Muitos problemas podem ser resolvidos utilizando redes colaborativas, por exemplo, por meio de *crowdsourcing* onde é possível formar grupos de trabalho onde sejam feitas propostas em paralelo pelos diversos especialistas da área, acelerando assim a descoberta de comportamentos maliciosos. Partindo deste pressuposto, o autor desta

[1]http://www.w3.org/Submission/SWRL/

dissertação desenvolveu uma plataforma de *crowdsourcing* voltada para análise de *malware* utilizando ontologias. Com a interação com a ontologia, por exemplo, usuários poderão postar ações suspeitas de programas que serão analisadas e incorporadas na ontologia. A próxima Seção destaca os objetivos, métodos empregados e contribuições desta dissertação.

1.3. Objetivos, Contribuições e Métodos

Neste trabalho buscamos responder a pergunta de pesquisa: "Como utilizar mecanismos de *crowdsourcing* para possibilitar o compartilhamento de informações e evolução colaborativa de ontologias e regras para análise comportamental de *malware*? ".

Conforme destacado anteriormente, foi desenvolvido um modelo e um sistema que implementa uma plataforma de *crowdsourcing* para que pessoas possam fornecer ideias, postando comportamentos suspeitos de software maliciosos. Esta plataforma tem como foco os profissionais de segurança de informação e usuários com interesse na análise de *malware*. A atividade colaborativa ajudará gradativamente a aumentar e deixar mais precisas as regras da ontologia compartilhada. Assim, pretende-se, em longo prazo, contribuir para aumentar a capacidade de analisar o comportamento de programas maliciosos; identificando novos comportamentos maliciosos em execuções de programas usando de regras de inferência e assim calcular os níveis de ameaça. Está fora do escopo deste trabalho, o desenvolvimento de classificadores binários, *antivírus* e sistemas seguros, mas o resultado gerado pelo "conhecimento" compartilhado na plataforma serve de base para o desenvolvimento futuro dessas soluções.

A hipótese é que com ontologias podemos descrever um modelo flexível onde usuários poderão representar atividades suspeitas de software por meio da elaboração de novos conceitos e regras em uma plataforma de *crowdsourcing*. Desta forma, será possível melhorar a qualidade e detalhamento da ontologia coma colaboração e conhecimento compartilhado. Propostas para análise de *malware* utilizando tecnologias baseado em *Web Semântica* são pouco comuns. Espera-se, portanto, com este trabalho contribuir no desenvolvimento de novas abordagens para análise de *malware*, por meio de uma plataforma colaborativa onde usuários poderão contribuir com especialistas de *malware* para atualizações constantes e aumentar o entendimento dos usuários sobre o problema.

Como as características de *malware* mudam em uma velocidade surpreendente, em função de novas técnicas adotadas, a utilização de estratégias de *crowdsourcing* pode diminuir o tempo entre o surgimento de um novo *malware* e a identificação de novos comportamentos suspeitos, assim, quando novos comportamentos são descobertos por um especialista, ele (a) pode rapidamente compartilhar este conhecimento com os demais e contribuir para o modelo via plataforma de *crowdsourcing*.

Atualmente, há um interesse crescente no desenvolvimento de sistemas baseados na representação do conhecimento, por meio de abordagens baseada em ontologias (Huang e outros, 2013). Com a utilização do *crowdsourcing* é esperado um aumento da base de conhecimento. Segundo Chi (2012), *crowdsourcing* tem sido proposto para a realização de trabalhos que exigem grandes esforços humanos para o desenvolvimento de tarefas especificas como o desenvolvimento de uma grande base de conhecimento. Partindo da questão de pesquisa principal desta dissertação (descrita no início desta Seção) pode-se destacar as seguintes metas:

- **Meta 1**: Elaboração de um modelo de *crowdsourcing* que inclui o processo decisório para definir quais novas classes e regras deverão ser incorporadas a ontologia ou não, esse processo possui participação humana onde os especialistas em segurança da informação analisam as contribuições e verificam se estas podem ser inseridas ou não na estrutura da ontologia;
- **Meta 2**: Desenvolvimento de sistema que implementa uma plataforma de *crowdsourcing* com várias formas de contribuições facilitadas, onde usuários e especialistas podem trabalhar de maneira colaborativa a ontologia existente via software, além da implantação de segurança no sistema para evitar danos na estrutura da ontologia utilizada;
- **Meta 3**: Avaliação da eficácia do modelo e sistema produzido com base na análise das respostas de potenciais usuários, que serão apresentadas no Capítulo 6.

Com uma plataforma de *crowdsourcing* espera-se ter uma solução eficiente onde usuários poderão contribuir com a ontologia com ideias que serão validadas por usuários mais experientes, garantindo assim um processo controlado para a evolução da ontologia; onde os usuários postarão comportamentos suspeitos de aplicações que serão analisados e

classificados pelos participantes como maliciosos ou não, além de atribuir o grau de possíveis danos por eles causados.

Nesse processo, usuários enviam dados sobre software e relatam execuções suspeitas, onde passam pela verificação da ontologia e pelas regras preexistentes, bem como por outros usuários experientes que definem se esta ação é passível de classificação. Esta última análise é feita manualmente por especialistas em ontologia e *malware*.

Uma vez atingidas estas metas, do ponto de vista científico, são esperadas as seguintes contribuições principais desta dissertação: (1) melhor entendimento sobre o potencial do uso de ontologias e regras semânticas na análise *malware*, usuários do sistema poderão ter um melhor conhecimento sobre as tecnologias, e assim auxiliar no desenvolvimento de pesquisas relativas ao tema; e (2) avanço no conhecimento de como plataformas de *crowdsourcing* podem contribuir para a evolução de modelos com utilização de ontologias, para se criar novos modelos conceituais dentro de um domínio e facilitar na organização de informações sobre incidentes relativos a *malware*. Do ponto de vista prático, este trabalho de pesquisa pode contribuir para aumentar, em longo prazo, a eficácia dos mecanismos de análise de programas potencialmente suspeitos. Esta abordagem mostra que ontologias podem serem utilizadas em conjunto com regras para análise de *malware*, que futuramente podem serem utilizadas para a construção de ferramentas eficientes de análise comportamental de programas maliciosos.

1.4. Estrutura da Proposta

O restante dessa proposta está estruturado da seguinte forma:

- O Capítulo 2 apresenta o referencial teórico e metodológico empregado nesta dissertação, incluindo segurança em sistemas de informação, *Web semântica* e sistemas colaborativos e *crowdsourcing*;
- O Capítulo 3 apresenta uma pesquisa sistemática e análise do estado da arte de trabalhos relacionados ao tema de pesquisa em foco;
- O Capítulo 4 apresenta o modelo conceitual do *crowdsourcing* incluindo o design do modelo e a descrição desse modelo;

- O Capítulo 5 descreve o desenvolvimento do sistema, incluindo a arquitetura do sistema, as tecnologias utilizadas e a descrição das interfaces do sistema, bem como a metodologia empregada na construção;
- O Capítulo 6 apresenta a avaliação do sistema e discussões sobre os resultados obtidos;
- O Capítulo 7 apresenta a conclusão do trabalho, incluindo as considerações finais e os trabalhos futuros.
- O Apêndice I apresenta o resumo do artigo: Grégio, A., Bonacin, R., de Marchi, A. C., Nabuco, O. F., & de Geus, P. L. (2016). *An ontology of suspicious software Behavior*. *Applied Ontology*, pp 29-49;
- O Apêndice II apresenta o resumo do artigo: de Marchi, A. C., Grégio, A., Bonacin, R. (2017). *Enhancing the Creation of Detection Rules for Malicious Software though Ontologies and Crowdsourcing*. (*WETICE*, 2017), Polônia.

2. Referencial Teórico e Metodológico

Neste Capítulo são apresentados conceitos e métodos relacionados à segurança em sistemas de informação (Seção 2.1), tecnologias e padrões da *web semântica* (Seção 2.2), e conceitos e tecnologias para o desenvolvimento de sistemas colaborativos e de *crowdsourcing* (Seção 2.3).

2.1. Segurança em Sistemas de Informação

Segurança da Informação pode ser definida como a proteção dos sistemas de informação contra acesso não autorizado ou alteração das informações, seja em armazenamento, processamento ou de trânsito, incluindo as medidas necessárias para detectar essas ameaças (The Agence, 2000).

Com o advento da Internet houve um crescimento significativo do número de ataques à segurança dos sistemas de informação. Isso nos mostra que os esforços para conter os ataques aos sistemas computacionais são insuficientes (Kropiwiec e de Geus, 2004). No contexto atual, novos mecanismos de segurança da informação são essenciais para proteger a informação.

Segundo Yao e Wei (2014), várias invasões bem-sucedidas revelaram a vulnerabilidade das arquiteturas de segurança em sistemas de informação. Entende-se como segurança de sistemas de informação a proteção contra negação de serviço; intrusão; modificação desautorizada de dados e informações armazenadas em processamento (Simião, 2009).

Garantir a segurança de informações digitais é uma preocupação global, pois ameaças não são restritas a fronteiras de países ou regiões, sejam em organizações (públicas ou privadas), empresas, universidades e cidadãos (Pereira, 2005). Quando uma nova ferramenta capaz de reprimir um tipo de ameaça é desenvolvida, outras formas de ameaças são concebidas, concretizando assim um cenário de aumento de ataques à segurança da informação.

Fundamentalmente a Segurança da Informação está calcada em três princípios básicos (Maia, 2013):

- ***Confidencialidade***: o acesso à informação deve ser concedido apenas a quem esteja autorizado;
- ***Integridade***: a integridade da informação deve ser preservada, isto é, a não corrupção dos dados desde o envio até a chegada no destino;
- ***Disponibilidade***: a informação deve estar sempre disponível quando um usuário ou sistema autorizado solicitar.

De acordo com os padrões de segurança da informação, da série de normas ISO/IEC 27000, esses três atributos são os principais, porém, existem outros, tais como: *Autenticidade* que é quando a informação é de fonte conhecida e não foi modificada durante algum processo e *Irretratabilidade* que é a impossibilidade de negar a autoria sobre uma transação efetuada.

2.1.1. Ameaças, Ataques e Prevenção

É necessário conhecer as fragilidades e vulnerabilidades que possam existir dentro do sistema, e assim, ter a dimensão das consequências. Com isto espera-se ter tomadas de decisões certas e definitivas para a garantia de segurança desses sistemas.

As ameaças estão ligadas as falhas de segurança dos sistemas de informações, apontando para possíveis danos. Os ataques variam desde curiosidade, conhecimento do sistema invadido e também por ganhos financeiros através de extorsão, espionagem, roubo de informações confidenciais, etc. (Pinheiro, 2007). Na Internet, indivíduos podem se passar por outra pessoa, o anonimato existe em muitos casos, e se aproveitando disso, com conhecimentos amplos e com computadores com muitos recursos, muitos praticam invasões e crimes cibernéticos. A situação é tão crítica que a adoção de recursos adicionais para identificação de usuários se torna necessários, tais como mecanismo de autenticação e certificação digital.

São muitas as ameaças em sistemas de informação, que vão desde as vulnerabilidades dos sistemas até o fator humano. Segundo Marciano (2006), algumas das principais ameaças aos sistemas de informação são: atos cometidos com software maliciosos; falhas ou erros humanos; falhas de software; atos de espionagem ou invasão; e, sabotagem ou furtos.

Nos últimos anos, a frequência e a sofisticação dos ataques aumentou drasticamente. *Malware* são projetados para de forma oportunista para explorar automaticamente deficiências técnicas e falhas dos sistemas de informação, os ataques têm por objetivo alvos específicos (Kiesling e outros, 2014). Os ataques são disparados contra os sistemas por pessoas, pode ser, por exemplo, um funcionário insatisfeito com a empresa, pessoas visando lucro através de roubo de dados ou até mesmo alguém querendo apenas ser conhecido, mostrando seu conhecimento.

Dentre os diferentes tipos de ataques, existem os que têm como finalidade tornar um sistema indisponível, pois além de deixá-lo indisponível, um usuário mal-intencionado também pode tentar se passar pelo alvo (Michelin, 2015). Este ataque é conhecido como DoS (*Denial of Service*), existem muitos outros tipos de ataques (Kargl e outros, 2001). Segundo Al-Bataineh e White (2012) uma possível classificação de ataques é: ataques por publicidade, negação de serviço (DoS), ataques internos, e ataques destrutivos.

A implantação de técnicas para proteção das informações é muito importante para usuários domésticos e essencial para empresas. A prevenção implica no uso de ferramentas e pode ocasionar custos. Evitar vírus, por exemplo, é mais recomendado do que tentar se livrar deles quando o computador estiver infectado (Tanenbaum, 2010). Entretanto, é impossível proteger totalmente um sistema de informações, pois novas ameaças aparecem a cada minuto.

De acordo com Coelho e outros (2012), a segurança das informações é aumentada utilizando algumas proteções no sistema como: proteção de hardware (proteção física de acesso ao hardware do sistema), proteção de perímetro (proteção da rede contra intrusões, *firewall* e *routers*) e proteção de arquivos e dados (processo de autenticação dos usuários).

Estar conectado à Internet traz benefícios, entretanto existem problemas identificados como “problemas de entradas” que incluem invasores e pragas virtuais, e também os “problemas de saída” onde podem ser interceptados dados confidenciais. A instalação de *firewalls* (Pinheiro, 2004) e *antivírus* são mecanismos muito utilizados para minimizar esses problemas.

Os *antivírus* tipicamente baseiam-se em um banco de dados de assinaturas para caracterizar um *malware* conhecido, contém seu próprio banco de dados que são atualizados frequentemente através da Internet. Entretanto, quando um *malware* não

possui uma assinatura é necessária uma análise para entender como o seu comportamento afeta o sistema, para enfim desenvolver uma assinatura, para poder ser detectado e combatido (Ceron e outros, 2009).

2.1.2. *Malware*

Malware é a principal ameaça a usuários de computadores, ao serem conectados em rede os computadores podem ser infectados por programas maliciosos a qualquer momento. Segundo Afonso e outros (2011), um *malware* pode ser entendido como um programa desenvolvido para realizar algo ilícito, que modifica a operação legítima de um sistema computacional de modo a violar sua integridade, confidencialidade ou disponibilidade realizando alguma ação maliciosa dentro de um computador ou sistema.

Muitos desenvolvedores de *malware* são programadores com vasta experiência, enquanto outros utilizam "*kits*" de *malware* disponíveis na Internet, por exemplo, comprando na *Dark Web*[2]. Assim, usuários com pouco conhecimento também podem utilizar estes *malware* de maneira visual para enviar programas maliciosos e fazer invasões em sistemas, eles inserem nos códigos ferramentas *anti-análise* para dificultar o trabalho do analista de *malware*. Um software que permite o controle não autorizado de um sistema obviamente é mal-intencionado, mas um software que exibe anúncios não é estritamente prejudicial, a menos que invada a privacidade e colete informações pessoais sem o conhecimento do usuário (*e.g,spyware*) (Blount e outros, 2011).

Ataques de *malware* aumentam muito rapidamente, as ferramentas de detecção também evoluíram muito, entretanto ainda são insuficientes para combater os códigos maliciosos. Por exemplo, uma análise da empresa *G Data Security Labs* na qual somente no primeiro semestre de 2014 descobriram 1,8 milhões de novos casos de *malware*, o que resulta em um novo *malware* a cada 8,6s (GData, 2014)[3].

---

[2]https://hack4fun.club/2017/03/02/trojans-publicitarios-e-vendas-de-kits-de-malware-na-dark-web/

[3]*https://www.gdatasoftware.com/newsroom/news/article/g-data-new-computer-malware-every-86-seconds*.

Programas maliciosos podem mudar conforme necessidade e ambiente. Para escapar de detectores de *malware*, muitos usam algum método de ofuscação para mudar seu verdadeiro propósito (Armoun e Hashemi, 2012). Com essa intenção, ele converte um programa para uma nova versão diferente ao torná-los funcionalmente iguais uns aos outros (mesmo comportamento com outro código), fazendo que passe pelas barreiras de detecção e continue seu propósito dentro do sistema (You e Yim, 2010). Já comportamentos de *malware* são ações executadas pelos programas maliciosos e assim sabemos os danos causados independente de seu código (Grégio e outros, 2014). Alguns sistemas (*e.g.,SCARECROW*) enviam aos usuários um resumo com informações como nome, comportamento e gravidade quando um *malware* é detectado, aumentando assim a consciência dos usuários finais (Barakat e outros, 2013).

Na abordagem "tradicional" os *malware* são classificados por tipos. Os tipos mais comuns são segundo Maziero (2014):

- **Vírus**: É um programa ou parte de programa de computador que se propaga através de auto replicação inserindo cópias de si mesmo em outros arquivos, utilizam para a infecção também e-mail, redes p2p, entre outras. Para serem acionados é necessária a execução do mesmo por um usuário;
- ***Worm***: Similares aos vírus na forma de infecção, porém não necessita infectar arquivos e nem da execução de um usuário para disseminar o ataque. Proliferam com uma velocidade maior que os vírus, um *worm* modifica o sistema operacional do computador infectado para serem inicializados como parte do processo. *Worm* são independentes de outros programas e utilizam as vulnerabilidades do sistema;
- ***Keyloggers***: São programas que capturam teclas pressionadas e registram em um arquivo, essas informações são enviadas para um atacante. São utilizados para espionagem e principalmente para o roubo de senhas. Alguns *keyloggers* são tão sofisticados que apenas registram informações quando o usuário acessa um site seguro e capturam números e contas, senhas e outras informações antes que elas sejam criptografadas;
- ***Rootkit***: São utilizados por invasores após corromper o sistema, depois de controlar o sistema, o invasor mantém acesso ao sistema sem conhecimento

de outros usuários, incluindo o administrador. São difíceis de identificar porque eles atuam no nível de *kernel* para ocultar sua presença. Seu objetivo é permanecer no sistema, mas alguns podem conter *backdoors* que lhe permite voltar quando achar conveniente. Em caso de infecção por *rootkit*, o mais indicado é formatar e reinstalar o sistema porque é impossível ter certeza que todos seus componentes foram removidos;

- ***Trojan***: São os conhecidos "cavalos de *tróia*" e tem dupla finalidade, uma atividade benigna conhecida do usuário e outra maliciosa executada sem que o usuário perceba. Para serem executados eles despertam a curiosidade do usuário e depois comprometem o sistema, não se replicam de forma automática e infectam programas úteis do computador. Muitos cavalos de *tróia* são utilizados para a proliferação de outros programas maliciosos como vírus e *worm*, quando isso acontece são chamados de *droppers*;
- ***Spyware***: Sua infecção é semelhante à do cavalo de *tróia*, não se replicam, porém, exploram as vulnerabilidades dos navegadores. Tem como objetivo o lucro financeiro como propagandas não solicitadas e até roubo de informações pessoais e sigilosas;
- ***Backdoor***: Permite acessar um computador usando meios normais. Permitem abrir portas onde invasores podem acessar o sistema posteriormente. Podem ser inseridos manualmente no código do software hospedeiro e avisa quando vai sendo instalado, ou podem se comportar como *worm* que integram o processo de inicialização da máquina e se espalham;
- ***Exploit***: Exploram as vulnerabilidades do sistema para realizar a infecção, podem estar incorporados em *worm* e *trojans* ou podem ser usados em ataques manuais através de ferramentas. Podem não ser maliciosos, alguns são projetados por pesquisadores para explorar as vulnerabilidades dos sistemas e trabalhar em melhorias;
- ***Packet Sniffer***: Age capturando pacotes da rede local ou do próprio computador buscando informações sigilosas como senhas. Capturam

informações antes da criptografia ou após os pacotes serem descriptografados.

Entretanto, atualmente é difícil "enquadrar" um exemplar de *malware* em uma única classe, devido à evolução desses códigos e a facilidade de se adicionar novas funcionalidades (Grégio e outros, 2014). Nesta dissertação é explorada a construção colaborativa de modelos e regras que analisam e classificam o *malware* de acordo com seu comportamento independente de seu tipo. Os comportamentos conhecidos estão inseridos dentro da ontologia através de regras.

2.2. *Web Semântica* e Ontologias

Semântica pode ser entendida como o estudo do significado e as relações sobre os significados (Morris, 1938), a Semântica se refere ao estudo do significado, indica o que algo explica. Diferentemente da Sintaxe, que se debruça sobre as estruturas ou padrões formais do modo como algo é expresso enquanto a sintaxe se ocupa mais no formato e formalizações, e a semântica no significado e interpretação. Segundo Berners-Lee (2001) a *Web Semântica* irá permitir que máquinas interpretem a semâmtica dos documentos.

A *Web Semântica* não é uma *Web* separada, mas uma extensão da atual. Nela a informação é dada com um significado bem definido, permitindo melhor interação entre os computadores e as pessoas (Berners-Lee e outros, 2001). A arquitetura da *Web Semântica* é composta por uma "pilha de protocolos". A Figura 2.1 apresenta a proposta inicial de Berners-Lee para a arquitetura da *Web Semântica*.

Segundo Dias e Santos (2003), um problema a ser enfrentado é que os computadores na *Web*, atualmente, têm papel somente no direcionamento e entrega de informações. Eles não são capazes de interpretar o conteúdo das páginas, porque essa informação está estruturada para utilização pelas pessoas e não por máquinas. Nessa visão, hoje temos uma *Web* de documentos e não de informações.

É possível dizer que a Internet é uma das maiores fontes de informações que existe atualmente. Nela é possível pesquisar sobre os mais variados temas com apenas alguns cliques. Um dos desafios é tornar a informação dentro da rede compreensível, vinculando

os dados, fazendo uma padronização onde as máquinas possam "conversar" entre si. Para isso, seria necessária a inclusão de dados em modelos que todos os sistemas possam entender.

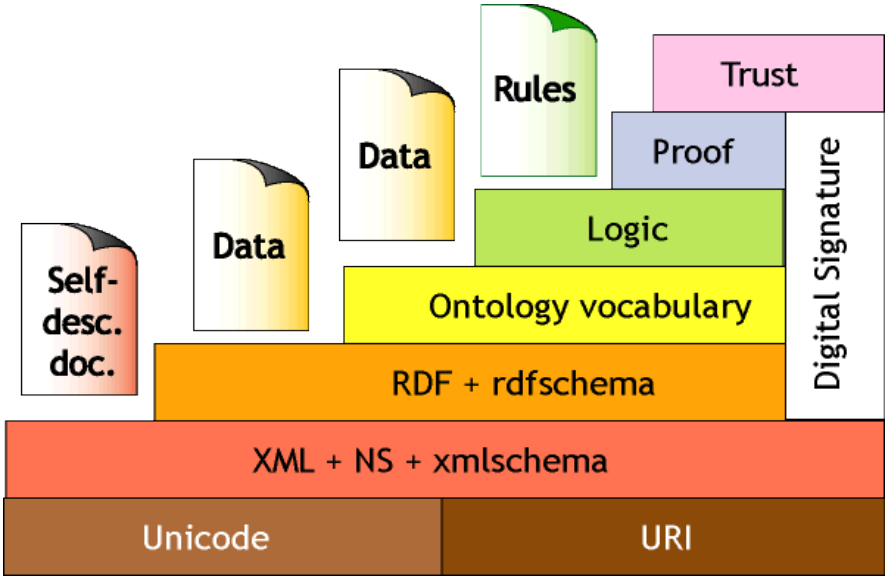


Figura 2.1: Arquitetura da Web Semântica[4].

Nessa arquitetura destacamos aqui as principais linguagens/padrões da *Web Semântica* utilizadas neste trabalho. O *XML* (*eXtensible Markup Language*), o *RDF* (*Resource Description Framework*).

A *OWL* (W3C, 2012) utiliza a lógica descritiva para a construção de ontologias, utiliza um mecanismo de inferência (*Reasoner*) para verificar a hierarquia de classes e a consistência da ontologia. *OWL 2.0* é o padrão mais recente para a construção de ontologias desenvolvida pelo *WC3* (Horridge e outros, 2008), promovendo a interoperabilidade e compartilhamento entre aplicativos.

A *SWRL* fornece um conjunto de regras avançadas para serem trabalhadas juntas com ontologias *OWL*. *SWRL* é uma linguagem que permite o desenvolvedor criar regras que possam ser expressas no formato *OWL* recomendada pelo *WC3* (Horrocks e outros, 2004). Ela é uma linguagem de padrão emergente onde regras são descritas com antecedente e consequente (Hwamg e outros, 2008). Em *SWRL*, o antecedente é referido como a regra de corpo e a consequente é referida como a cabeça da regra. Cada regra

[4]http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

consiste em um conjunto possivelmente vazio de átomos, uma regra pode ser lida como se o antecedente é verdadeiro, então a consequente também deve ser verdade (Golbreich, 2004).

A ideia básica da *SWRL* é estender *OWL* com um formato padrão de regras mantendo a compatibilidade com as demais tecnologias da *Web Semântica*. Ou seja, ela é uma extensão da semântica e a sintaxe descrita em *OWL* (Wang e He, 2006).

Em *SWRL* os predicados são definidos pela ontologia em *OWL* e as variáveis são indicadas com um ponto de interrogação (*e.g.*,*?x*)(Horrocks e outros, 2004). Um exemplo de uma regra para *malware* seria:

ProcessAction(?x), Registry(?y), hasTarget(?x, ?y), processActionName(?x,"WRITE"), targetName(?y, ?z). contains(?z, sharedaccess\\paremeters\\firewallpolicy\\standardprofile\\donotalloweexceptions")-> ShutdownSystemFirewall(?x), riskLevel(?x, "4")

Esta regra significa que toda ação de um processo do tipo escrita que tenha como alvo um registro que contenha a seguinte *string* no seu nome "sharedaccess\\paremeters\\firewallpolicy\\standardprofile\\donotalloweexceptions", então essa ação é classificada como uma ação de desligamento do *Firewall* e possui um risco associado ao nível 4.

A Figura 2.2 apresenta um exemplo de hierarquia das classes em OWL utilizando a ferramenta Protégé para descrição de comportamento de *malware*.

Segundo Grégio e outros (2014), SuspiciousExecution é a classe principal da ontologia. As relações entre as classes são as seguintes:

- Cada SuspiciousExecution deve ser associado a um SuspiciousSoftware;
- Todas as instâncias do SuspiciousSoftware são executadas em um Sistema;
- Um SuspiciousExecution contém um conjunto de ProcessActions;
- Uma instância ProcessAction está associada a um timestamp, a um processo (com um nome /SourceObject (por exemplo, um ApplicationProcess que o executa) e um TargetObject (por exemplo, Mutex, Rede, Registry, SystemFile ou ApplicationProcess);

- Uma instância de ProcessAction pode ser vinculada a uma SuspiciousBehaviour, que pode ser uma ou mais eventos (AttackLaunchingEvent, EvasionEvent, RemoteControlEvent, SelfDefenceEvent, StealingEvent, SubversionEvent)

A Figura ilustra a hierarquia de classes SuspiciousBehaviour expandida, contendo todas as subclasses (Eventos) e os comportamentos pertencentes a um ou mais desses eventos. Esses comportamentos podem então ser especializados em atividades suspeitas, que fornecem mais detalhes sobre ações potencialmente perigosas realizadas durante o ciclo de vida da infecção.

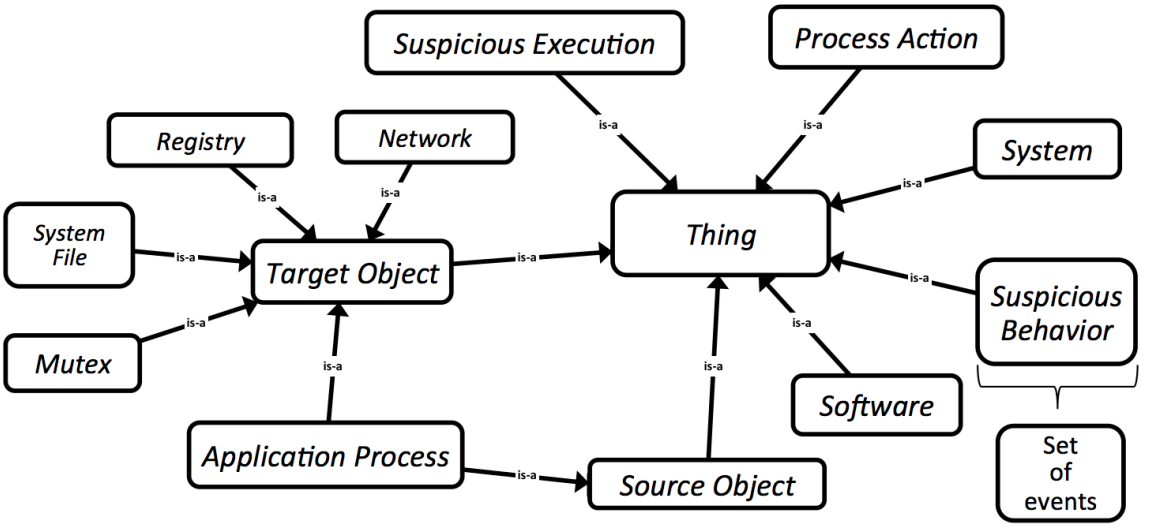


Figura 2.2: Exemplo de hierarquia de classes.

Fonte: Grégio e outros, 2014.

Mesmo com toda a expressividade que a linguagem *OWL* oferece com seus construtores para a definição de ontologias, não se pode expressar todo o conhecimento de um domínio somente com ela (Bastos, 2013), para este fim será utilizado à linguagem *SWRL*. O *SWRL* caminha para ser a linguagem de regra padrão da *Web Semântica*.

Um desafio é manter essas ontologias atualizadas, mudanças e novos termos devem ser adicionados na ontologia ficando difícil apenas para os especialistas manter essas atualizações. Segundo Braun e outros (2007), as pesquisas concentram-se apenas no desenvolvimento e criação de ontologias, sem olhar sua manutenção que são cruciais para os usuários a utilizarem.

Uma técnica que apoiaria a atualização e evolução de ontologia é o *crowdsourcing*. Técnicas computacionais não substituem completamente o trabalho dos especialistas na modelagem de ontologias, mas a proposta do *crowdsourcing* pode resolver alguns problemas difíceis e às vezes complexos. *Crowdsourcing* tem o potencial de agregar conhecimento e a inteligência de um grande número de pessoas através da colaboração (Lin e outros, 2010). Assim, nesta dissertação é apresentado um sistema de *crowdsourcing* especialmente construído para a evolução de ontologias e regras que modelam comportamento de *malware*.

2.3. Sistemas Colaborativos e *Crowdsourcing*

O termo *crowdsourcing* (ou *crowdsource*), inicialmente proposto em Howe (2006), é um termo para descrever atividades de terceirização para o desenvolvimento de algo via Internet. É uma forma de colaboração solidamente paralela, com baixo investimento. Muitas tarefas triviais para pessoas continuam a desafiar os programas de computadores mais complexos; ou seja, são tarefas que não podem ser informatizadas e a atribuição dessas tarefas para funcionários inviabiliza o custo de um projeto (Yuen e outros, 2011).

O *crowdsourcing* foi proposto para tornar a mão de obra mais eficiente e reduzir os custos de produção (Howe, 2009). *Crowdsourcing* é um processo onde o trabalho é dividido para um grupo indefinido de pessoas. A *Web* simplifica enormemente a tarefa de recrutar grupos de trabalhadores, um exemplo de sucesso é a *Wikipédia* (Wang e outros, 2012). A *Wikipédia*, por exemplo, já alcançou a marca de mais de 25 milhões de colaboradores. Os "usuários" cooperaram para se criar uma das maiores bases de conhecimento online que conhecemos[5].

[5] https://en.wikipedia.org/wiki/Special:Statistics

O sistema operacional Linux foi desenvolvido e continua sendo melhorado com o conhecimento de muitos internautas. Assim, o Linux pode, de certa forma, ser classificado como um produto do modelo de *crowdsourcing*. *Crowdsourcing* pode ser pensado como um movimento em prol de software livre, com o desenvolvimento de sistemas por voluntários. Nesses casos, é possível observar como o apoio de muitas pessoas pode facilitar o desenvolvimento constituindo assim um grande processo (Cocate e outros, 2014). Geralmente, no *crowdsourcing* a contribuição não é formal como em um emprego, ainda que os envios sejam feitos de forma regulamentada o trabalho depende do colaborador (Butinhão e Soares, 2014).

Crowdsourcing evoluiu rapidamente, sendo atualmente utilizado para indicar várias atividades realizadas na Internet. Pode-se destacar a resolução de problemas distribuídos, colaboração em massa, computação humana e resolução de problemas como atividades "típicas" de *crowdsourcing* (Davis, 2011).

O *crowdturfing* é uma forma de *crowdsourcing*, mas utilizado para ações de marketing. Segundo Marcelo Lopes (2012), as ações de publicidade coletiva, iniciou-se de forma privada, as empresas utilizam muitos profissionais para desenvolver estas plataformas, estas empresas perceberam que poderiam ganhar dinheiro utilizando a coletividade como uma forma de *marketing* e expondo melhor o produto.

O *crowdturfing* também é conhecido como plataformas abertas e usuários ao acessarem alguns anúncios podem estar sujeitos à *malware*, não existe um controle sobre os anunciantes e desenvolvedores de anúncios sendo assim, podem conter programas maliciosos. Segundo Porto (2012), quando as pessoas buscam informações sobre a plataforma, elas encontram as melhores possíveis, pois os próprios desenvolvedores publicam o máximo de mensagens de apoio em diversos lugares mostrando a qualidade e a honestidade destas empresas. Cada mensagem rende alguns centavos para o autor, que na maioria das vezes, ganha por produção, ou seja, quanto mais postar, mais recebe. Sabe-se que as pessoas confiam em opiniões nas redes sociais, mesmo que sejam de estranhos, assim o *crowdturfing* ganha escala e cresce.

3. Trabalhos Relacionados

Esta Seção apresenta os trabalhos relacionados, que foram divididos de acordo com combinações de conceitos utilizados nas buscas. Para tanto, tabelas foram criadas para sintetizar os resultados e, por fim foi feita uma comparação entre as propostas analisando os pontos fortes e fracos de cada trabalho relacionado.

Os trabalhos analisados foram selecionados por meio de uma busca sistemática da literatura existente. Para tanto, foram analisados artigos próximos a nossa investigação que envolve a construção de ferramentas de *crowdsourcing* para a modelagem colaborativa de ontologias e regras, e o uso dessas para análise de *malware*.

Com esse objetivo, foram realizadas buscas nas bases científicas e análise sistemática da literatura utilizando a busca manual e seleção de artigos relevantes com o tema. As buscas e seleção de artigos foram realizadas em cinco passos detalhados a seguir.

Primeiramente, foram realizadas buscas usando palavras chaves nas bases cientificas: *IEEEXPLORE*[6], *ACMDL*[7], *GOOGLE SCHOLAR*[8], *SCIENCE DIRECT (ELSIEVER)*[9] *e SPRINGER*[10].As buscas foram realizadas entre 12/01/2015 a 05/09/2015. As seguintes combinações de palavras chaves foram utilizadas como critérios para buscas, sem restrição de data de publicação:

- *Analysis Malware*;
- *Malware Behavior*;
- *Ontology*;
- *Crowdsource(ing) Malware*;
- *Ontology Crowdsource(ing)*;
- *Ontology Malware*;
- *Malware, Ontology and Crowdsource(ing).*

[6] ieee.org/ieeexplore
[7] dl.acm.org/
[8] scholar.google.com.br/
[9] www.sciencedirect.com/
[10] http://link.springer.com/

O processo de pesquisa retornou vários resultados nas diversas bases científicas pesquisadas. No segundo passo do processo de seleção, foi realizada a leitura de títulos e palavras chaves. Nos casos de retorno de milhares de artigos, a leitura foi realizada até se obter um número substancial de trabalhos seguidos sem relevância, como critério de parada foi utilizado uma página com 20 artigos seguidos sem relevância para a pesquisa. Assim, em alguns casos foram lidos todos os títulos e palavras chaves retornados na busca, em outros foi realizada a leitura em apenas algumas páginas, os critérios são destacados nas tabelas apresentadas na Seção 3.1.

O próximo passo foi à leitura dos resumos dos artigos e descarte dos artigos menos relevantes de acordo com a proximidade de assunto e contribuições destacadas pelos autores (critério de descarte/inclusão). Logo após, foi realizado o *download* dos artigos e procedida à leitura em duas fases, primeiramente uma leitura "parcial" para identificar a relevância. Por fim, foi realizada a seleção para leitura integral e análise detalhada do artigo.

A seguir, na Subseção 3.1, são exibidos os resultados quantitativos de artigos selecionados e escolhidos em cada passo do estudo. Na Seção 3.2 são apresentados os trabalhos com maior interseção com a nossa proposta de pesquisa.

3.1. Cenários da Pesquisa Realizada

As Tabelas (3.1 a 3.7), apresentadas nesta seção, apresentam a quantidade de artigos selecionados em cada passo do desenvolvimento da pesquisa, conforme destacado a seguir:

a) Quantidade de artigos selecionados de acordo com as palavras-chaves de busca;
b) Critério de leitura inicial de títulos e palavras chaves;
c) Quantidade de artigos selecionados a partir das palavras chaves para leitura de resumos e descartes;
d) Quantidade de artigos que foram obtidos para identificar e relevância (em leitura parcial);
e) Quantidade de artigos selecionados para análise detalhada;
f) Artigos selecionados como relacionados e discutidos na seção 3.2.

Tabela 3.1: Quantidade de artigos selecionados para busca: *Analysis Malware*.

ANALYSIS MALWARE	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
IEEE XPLORE	1.130	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	27	27	18	5
ACM DL	6.243	ANÁLISE POR TÍTULOS, PRIMEIRAS 7 PAGINAS.	9	3	2	1
GOOGLE SCHOLAR	43.100	ANÁLISES POR TÍTULOS, PRIMEIRAS 10 PÁGINAS E LOCAL DE PUBLICAÇÃO.	18	18	12	0
SCIENCE DIRECT (ELSIEVER)	2.883	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS. CONSIDERADOS ARTIGOS COM ACESSO ABERTO PARA PRÓXIMA FASE.	5	3	3	1
SPRINGER	4.010	ANÁLISE POR TÍTULOS DAS 12 PRIMEIRAS PÁGINAS. CONSIDERADOS ARTIGOS COM ACESSO ABERTO PARA PRÓXIMA FASE.	9	1	1	1
TOTAIS	**57.366**		**68**	**52**	**36**	**8**

Tabela 3.2: Quantidade de artigos selecionados para busca: *Malware Behavior*.

MALWARE BEHAVIOR	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
IEEE XPLORE	473	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	14	4	3	5
ACM	4.260	ANÁLISE POR TÍTULOS DAS PRIMEIRAS 14 PÁGINAS.	6	0	0	0
SCHOLAR	30.500	ANÁLISE POR TÍTULOS DAS PRIMEIRAS 18 PÁGINAS E LOCAL DE PUBLICAÇÃO.	13	3	2	0
SCIENCE DIRECT (ELSIEVER)	1.891	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS. CONSIDERADOS ARTIGOSCOM ACESSO ABERTO PARA PRÓXIMA FASE.	11	0	0	0
SPRINGER	3.105	ANÁLISE POR TÍTULOS DAS 10 PRIMEIRAS PÁGINAS. CONSIDERADOS ARTIGOS COM ACESSO ABERTO PARA PRÓXIMA FASE.	16	2	2	1
TOTAIS	**40.229**		**60**	**9**	**7**	**6**

Tabela 3.3: Quantidade de artigos selecionados para busca: *Ontology*.

ONTOLOGY	(A)	(B)	(C)	(D)	(E)	(F)
IEEE XPLORE	17.438	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	26	11	8	5
ACM	57.185	ANÁLISE POR TÍTULOS DAS PRIMEIRAS 8 PAGINAS.	12	6	4	0
SCHOLAR	1.380.000	ANÁLISE POR TÍTULOS DAS PRIMEIRAS 18 PÁGINAS E LOCAL DE PUBLICAÇÃO.	28	27	15	0
SCIENCE DIRECT (ELSIEVER)	35.728	ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS.CONSIDERADOS ARTIGOSCOM ACESSO ABERTO PARA PRÓXIMA FASE.	12	3	3	0
SPRINGER	96.940	ANÁLISE POR TÍTULOS DAS10 PRIMEIRAS PÁGINAS. CONSIDERADOS ARTIGOSCOM ACESSO ABERTO PARA PRÓXIMA FASE.	10	5	4	0
TOTAIS	**1.587.291**		**88**	**52**	**34**	**5**

Após o trabalho preliminar de busca apresentado nas Tabelas de 3.1 a 3.3, a análise procedeu com mais termos de pesquisas; filtrando assim os resultados para chegarmos aos trabalhos existentes na literatura mais próximos de nossa proposta. Para tanto, foram utilizadas combinações dos termos *crowdsource(ing)*,ontology e *malware* conforme apresenta as Tabelas 3.4, 3.5 e 3.6.

Tabela 3.4: Quantidade de artigos selecionados para busca: *Crowdsource(ing) Ontology*.

CROWDSOURCE+ ONTOLOGY	(A)	(B)	(C)	(D)	(E)	(F)
IEEE XPLORE	1	ANÁLISE POR TÍTULOS.	1	1	1	1
ACM	43	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	7	1	1	1
SCHOLAR	7.200	ANÁLISE POR TÍTULOS DAS 8 PRIMEIRAS PÁGINAS E LOCAL DE PUBLICAÇÃO.	20	8	6	3
SCIENCE DIRECT (ELSIEVER)	18	ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS.	10	3	1	0
SPRINGER	44	ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS.	1	1	1	1
TOTAIS	**7.306**		**39**	**14**	**10**	**6**

Tabela 3.5: Quantidade de artigos selecionados para busca: *Crowdsource(ing) Malware.*

CROWDSOURCE MALWARE	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
IEEE XPLORE	1	ANÁLISE POR TÍTULOS.	1	1	1	2
ACM	16	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	6	2	2	2
SCHOLAR	928	ANÁLISE POR TÍTULOS DAS PRIMEIRAS 10 PÁGINAS E LOCAL DE PUBLICAÇÃO.	10	7	5	0
SCIENCE DIRECT (ELSIEVER)	4	ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS.	3	2	1	0
SPRINGER	5	CRITÉRIO ANÁLISE POR TÍTULOS. CONSIDERADOS ARTIGOSCOM ACESSO ABERTO PARA PRÓXIMA FASE.	2	1	1	1
TOTAIS	**954**		**22**	**13**	**10**	**5**

Tabela 3.6: Quantidade de artigos selecionados para busca: *Ontology Malware.*

ONTOLOGY MALWARE	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
IEEE XPLORE	18	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	13	12	8	6
ACM	206	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	10	3	1	0
SCHOLAR	2.170	ANÁLISE POR TÍTULOS DAS8 PRIMEIRAS PÁGINAS E LOCAL DE PUBLICAÇÃO.	12	9	6	0
SCIENCE DIRECT (ELSIEVER)	84	ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS.	8	1	1	0
SPRINGER	158	CRITÉRIO ANÁLISE POR TÍTULOSDE TODOS OS ARTIGOS. CONSIDERADOS ARTIGOSCOM ACESSO ABERTO PARA PRÓXIMA FASE.	10	6	4	1
TOTAIS	**2.636**		**53**	**31**	**20**	**7**

Tabela 3.7: Quantidade de artigos selecionados para busca: *Ontology, Malware and Crowdsource(ing)*.

CROWDSOURCE MALWAREONTOLOGY	**(A)**	**(B)**	**(C)**	**(D)**	**(E)**	**(F)**
IEEE XPLORE	20	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS.	13	10	9	6
ACM	0	SEM RESULTADOS.	0	0	0	0
SCHOLAR	138	ANÁLISE POR TÍTULOS DE TODOS OS ARTIGOS E LOCAL DE PUBLICAÇÃO.	10	9	7	0
SCIENCE DIRECT (ELSIEVER)	0	SEM RESULTADOS.	0	0	0	0
SPRINGER	1	CRITÉRIO ANÁLISE POR TÍTULOS.	1	1	0	0
TOTAIS	**159**		**24**	**20**	**16**	**6**

A Tabela 3.8 apresenta uma síntese com todos os dados colhidos durante nossa revisão sistemática da literatura para cada base indexada de artigos e palavras chave utilizadas.

Tabela 3.8: Síntese dos Resultados por Base de Indexada.

BC	**PC**	**AR**	**AL**	**AB**	**AP**	**AI**
IEEE XPLORE	*ANALYSIS MALWARE*	1.130	21	27	9	18
	MALWARE BEHAVIOUR	473	14	4	1	3
	ONTOLOGY	17.438	26	11	3	8
	CROWDSOURCE/ONTOLOGY	1	5	6	1	8
	CROWDSOURCE/MALWARE	1	15	5	0	5
	ONTOLOGY/MALWARE	18	13	12	4	8
	CROWDSOURCE/MALWARE/ONTOLOGY	20	13	11	4	6
ACM DIGITAL LIBRARY	*ANALYSIS MALWARE*	6.243	21	27	9	18
	MALWARE BEHAVIOUR	4.260	6	0	0	0
	ONTOLOGY	57.185	12	6	2	4
	CROWDSOURCE/ONTOLOGY	43	7	1	0	1
	CROWDSOURCE/MALWARE	16	6	0	0	0
	ONTOLOGY/MALWARE	206	10	3	1	2
	CROWDSOURCE/MALWARE/ONTOLOGY	0	0	0	0	0
	ANALYSIS MALWARE	43.100	11	18	6	12

GOOGLE SCHOLAR	*MALWARE BEHAVIOUR*	30.500	13	3	1	2
	ONTOLOGY	1.380.000	28	27	12	15
	CROWDSOURCE/ONTOLOGY	7.200	20	8	2	6
	CROWDSOURCE/MALWARE	928	10	7	2	5
	ONTOLOGY/MALWARE	2.170	12	9	3	6
	CROWDSOURCE/MALWARE/ONTOLOGY	138	10	9	7	5
SCIENCE DIRECT (ELSIEVER)	*ANALYSIS MALWARE*	2.883	5	3	0	3
	MALWARE BEHAVIOUR	1.891	16	2	0	2
	ONTOLOGY	35.728	11	0	0	0
	CROWDSOURCE/ONTOLOGY	18	12	3	0	3
	CROWDSOURCE/MALWARE	4	8	2	1	1
	ONTOLOGY/MALWARE	84	8	1	0	1
	CROWDSOURCE/MALWARE/ONTOLOGY	0	0	0	0	0
SPRINGER	*ANALYSIS MALWARE*	4.010	9	1	1	0
	MALWARE BEHAVIOUR	3.105	16	2	2	0
	ONTOLOGY	96.940	10	5	4	1
	CROWDSOURCE/ONTOLOGY	44	0	0	0	0
	CROWDSOURCE/MALWARE	5	2	1	1	0
	ONTOLOGY/MALWARE	158	10	6	4	2
	CROWDSOURCE/MALWARE/ONTOLOGY	1	0	1	0	1
TOTAL	**ARTIGOS PESQUISADOS**	**1.695.941**	**380**	**221**	**80**	**146**

BC: Base Científica;

PC: Palavra-chave;

AR: Quantidade de Artigos retornados;

AL: Resumos Lidos de artigos não utilizados;

AB: Artigos Feito o Download;

AI: Artigos Lidos Integralmente;

AP: Artigos Lidos Parcialmente.

A Tabela 3.9 apresenta os artigos resultantes de nossas buscas que mais se aproximam da nossa proposta, estes artigos serão discutidos na próxima seção. Podemos ressaltar que alguns foram encontrados em mais de um dos termos utilizados nas buscas, resultando numa diferença entre artigos selecionados com a quantidade explícita nas tabelas feitas para correlacionar as pesquisas (*i.e.*, lá existem duplicações).

Tabela 3.9: Títulos dos Artigos, Base Indexada e Palavras Chaves Utilizadas.

DESCRIÇÃO DO ARTIGO	BASE PUBLICADA	PALAVRAS CHAVES						
ARTIGO CIENTÍFICO	BASE	A	B	C	D	E	F	G
Ontology for Malware Behavior: a Core Model Proposal (Gregio e outros , 2014)	*IEEE Xplore Digital Library*	0	x	x	0	0	x	x
Identification of malware activities with rules (Jasuil e outros 2014).	*IEEE Xplore Digital Library*	0	x	x	0	0	x	x
USPAM-A User Centric Ontology Driver Spam Detection System (Shoiab e Farrop, 2015).	*IEEE Xplore Digital Library*	x	0	x	0	0	x	x
Ontology-Based Intelligent System for Malware Behavioral Analysis (Huang e outros, 2010).	*IEEE Xplore Digital Library*	x	x	x	0	0	x	x
Anticipating Dormant Functionality in Malware: A Semantics Based Approach (Noor e Abbas, 2013).	*IEEE Xplore Digital Library*	x	x	0	0	0	x	x
Fuzzy Markup Language for Malware Behavioral Analysis (Huang e outros, 2013).	*Springer Link*	x	x	0	0	0	x	0
Behavior based Malware Analysis and Detection (Wu e outros, 2011).	*IEEE Xplore Digital Library*	x	x	0	0	0	0	0
An Ontology for Malware Analysis (Mundie e Mcintire, 2013).	*IEEE Xplore Digital Library*	x	0	x	0	0	x	x
Crowdsourcing semantic content: a model and two applications (Di Lorio e outros, 2010).	*IEEE Xplore Digital Library*	0	0	0	x	0	0	0
Computation and Crowdsourcing Method sand Extraction Ontological Structure from Folksonomy (Lin e Davis, 2010).	*Springer Link*	0	0	0	x	0	0	0
Ontological Services Using Crowdsourcing (Lin e outros, 2010).	*Google Scholar*	0	0	0	x	0	0	0
Ontology Quality Assurance with the Crowd (Mortensen e outros, 2013).	*Google Scholar*	0	0	0	x	0	0	0
Crowdsourcing the Verification of Relationship in Biomedical Ontologies (Mortensen e outros, 2013	*Google Scholar*	0	0	0	x	0	0	0
Conceptual Modeling Principles for Crowdsourcing (Lukyanenko e Parsons, 2012).	*ACM DL Digital Library*	0	0	0	x	0	0	0
Putting Out a HIT: Crowdsourcing Malware Installs (Kanich e outros, 2011).	*ACM DL Digital Library*	x	0	0	0	x	0	0
CUD: Crowdsourcing for URL Spam Detection (Hu e outros, 2011).	*ACM DL Digital Library*	0	0	0	0	x	0	0
Crowdsource: Automated inference of high-level malware functionality from low-level symbols using crowd trained machine learning model (Saxe e outros, 2014).	*IEEE Xplore Digital Library*	0	0	0	0	x	0	x
A Crowdsourcing Approach to Protect Against Novel Malware Threads (Chistoforidis e outros, 2014).	*IEEE Xplore Digital Library*	0	0	0	0	x	0	x
An Architecture Utilizing the Crowd for Building an Antivirus Knowledge Base (Thuan e outros, 2014).	*Springer Link*	0	0	0	0	x	0	0
SCARECROW: Scalable Malware Reporting, Detection and Analysis (Barakat e outros, 2013).	*Google Scholar*	x	0	0	0	0	0	0

(A) Analysis *Malware*;

(B) *Malware* Behavior;

(C) Ontology;

(D) *Crowdsource* Ontology;

(E) *Crowdsource Malware*;

(F) Ontology *Malware*;

(G) **Ontology, *Malware* and *Crowdsource*.**

3.2 Análise e Discussão sobre os Artigos Relacionados

Conforme destacado no Capítulo 2, o termo *crowdsourcing*, pode ser visto como o esforço de múltiplas pessoas para a realização de uma ou mais tarefas em comum. Atualmente, em muitos trabalhos na Internet que exigem grande esforço humano é utilizada esta prática. Já o uso de ontologias nos proporciona métodos para análise de *malware*, pois elas permitem a criação de modelos lógicos, regras e axiomas. Pelo uso das representações semânticas, como o uso de ontologias, podemos criar uma base de conhecimento sobre o assunto, e com o uso de *crowdsourcing* aumentar e aprimorar as regras e nossa base de conhecimento de forma dinâmica e colaborativa. Destaca-se, entretanto, que nessa pesquisa nenhum dos artigos abordam de maneira conjunta ontologias, técnicas de *crowdsourcing* e análise de *malware*. Assim, esta Seção está subdividida da seguinte maneira: a Subseção 3.2.1 apresenta trabalhos relacionados que propõem o uso de ontologias para a análise de *malware*, a Subseção 3.2.2 apresenta trabalhos relacionados que propõem o uso de *crowdsourcing* na modelagem de ontologias, e a Subseção 3.2.3 apresenta trabalhos relacionados que propõem o uso de *crowdsourcing* para melhoria na análise ou detecção de *malware*.

3.2.1. Uso de Ontologias na Modelagem e Análise de *Malware*

A literatura recente tem mostrado alternativas para o uso de ontologias na modelagem e análise de *malware*. Jasuil e outros (2014), por exemplo, propuseram o desenvolvimento de uma ontologia para análise e detecção de *malware* utilizando regras *SWRL* e a linguagem *OWL2*. Para tanto, são verificados os comportamentos dos códigos maliciosos.

O sistema proposto por eles filtra os processos através de centenas de milhares de eventos regulares identificando os processos suspeitos.

As maiorias dos ataques bem-sucedidos estão relacionadas à negação de serviços, fraudes financeiras, roubos de dados privados. O aumento do uso das redes sociais também contribui para os ataques, visto que os usuários aceitam arquivos, fotos, etc., vindo de perfis desconhecidos. De acordo com Jasuil e outros (2014), por serem normalmente fundamentado em assinaturas, programas de detecção como *antivírus* ficam desatualizados à medida que novos *malware* surgem com novas assinaturas. Assim sugerem uma técnica para detecção baseada nos comportamentos dos códigos dentro do sistema.

Partindo desta pressuposição que a assinatura é desconhecida, Jasuil e outros (2014) propõem que o sistema rastreie comportamentos suspeitos dentro do sistema da vítima através da filtragem dos eventos com a utilização de sensores para identificação de atividades hostis. Esses eventos são comparados com os já armazenados a fim de descobrir semelhanças entre eles.

O trabalho de Jasuil e outros (2014) demonstram a fragilidade dos sistemas *antivírus* baseados em assinaturas e também a possibilidade de utilização de uma classificação baseada na análise de comportamento suspeito. Entretanto, a classificação dos *malware* em *trojan,* cavalo de *tróia,* etc., utilizada pelos autores, podem gerar problemas na análise de comportamento; uma vez que os códigos maliciosos modernos são multiuso e podem conter comportamento de várias formas diferentes. O artigo não explica detalhadamente como os sensores são utilizados para rastrear os eventos e desta forma classificá-los.

Já Shoaib e Farooq (2015) propõem um sistema baseado em ontologias para classificar *spams*. O objetivo é evitar a alta taxa de mensagens enviadas para o lixo, que são na verdade mensagens legítimas e a dificuldade do sistema de aprender de acordo com o perfil do usuário. O sistema apresentado por Shoaib e Farroq (2015) utiliza uma ontologia para modelar os interesses de um usuário de acordo com seu perfil, e através dessas informações classifica uma mensagem suspeita de *spam* como boa ou ruim. O sistema proposto utiliza uma classificação baseada na atribuição de pesos; estabelecendo pela ontologia a relação semântica com o conteúdo das mensagens e os interesses de

acordo com o perfil do usuário para diminuir a alta taxa de falsos positivos. O sistema apresentou melhorias na recepção de mensagens pelos usuários, bem como para as agências de publicidades por permitir mensagens mais precisas ao consumidor final. Entretanto, o trabalho é dependente da avaliação de usuários que dão seu retorno em relação a regras predefinidas que categorizam as mensagens como abusivas, *malware* e fora do contexto. É importante também destacar que o objetivo difere do nosso trabalho, uma vez que a detecção de *spam* está fora do escopo proposto.

O trabalho de Huang e outros (2010) propõe um sistema inteligente para análise do comportamento de *malware*. Os autores capturaram o comportamento de *malware* e os analisaram com o objetivo de construir ontologias e regras apropriadas. A solução proposta é composta por três camadas: de aplicação, de comunicação e de conhecimento. A camada de conhecimento fornece uma base para as regras e ontologia de *malware*; a camada comunicação fornece as interfaces para interligar a camada de aplicação com a camada de conhecimento e a camada de aplicação é onde os usuários acessam via internet com o sistema que interage indiretamente com a ontologia de *malware*. A solução de Huang e outros (2010) utiliza a ferramenta *AIDE* (*Advanced Intrusion Detection Environment*) que compara a imagem de disco após a execução do *malware* com a imagem antes da execução e mostra o que foi alterado, apagado ou modificado. Entretanto, os autores apresentam apenas um exemplo superficial de sua arquitetura, não sendo assim capaz de identificar capacidade de trabalhar com classes complexas de *malware*.

A análise estática e manual não tem mais eficácia devido ao aumento crescente e diário da quantidade e complexidade de *malware* (Noor e Abbas, 2013), que inclui novas técnicas de ofuscação de código e criptografia. Por outro lado, a análise dinâmica analisa o comportamento exibido pelo código malicioso, mas não descobre sua verdadeira intenção quando consideramos funcionalidades dormentes (*i.e.*, o *malware* ainda não executou a funcionalidade). De acordo com Noor e Abbas (2013), nesta condição a melhor abordagem seria utilizar conexões de rede para correlacionar as funções e seus parâmetros como valor de registro, mudança de arquivos, etc.

Para ter a cobertura da intenção do *malware*, Noor e Abbas (2013) propõe uma abordagem utilizando a semântica baseada em modelos formais do software, através da observação do comportamento e do registro do *malware* em uma análise dinâmica, os

resultados são comparados com amostras de *malware* e só então passa pela ontologia. O papel da ontologia está em decifrar o algoritmo de reagrupamento do *malware* utilizando o mapeamento de seu comportamento, identificar o *malware* e o sistema. Com esta abordagem pode-se identificar vários subconjuntos de *malware* e atividades semelhantes que eles possuem. Entretanto, o trabalho não aborda problemas de escalabilidade, e custo alto de processamento, bem como destacamos a visão restrita quanto ao uso da ontologia, não considerando esta como um modelo em evolução passível de contribuições via *crowdsourcing* e ou mesmo provenientes de análise formais.

De acordo com Huang e outros (2013) existe um crescente interesse em sistemas baseados na representação do conhecimento. No entanto, as linguagens tradicionais que as ontologias não podem lidar com a imprecisão e incertezas que são características marcantes do mundo real. Eles destacam que o desenvolvimento de um sistema para pesquisar o comportamento de *malware* é crucial para prever ações de software malicioso e diminuir os danos causados. Neste contexto, é necessária uma ferramenta de análise de comportamento desenvolvida com uma ontologia capaz de entender, ler, decifrar, analisar pedaços de códigos dos atacantes. Para tanto, Huang e outros (2013) propõe o uso do sistema *TAWMAN*, que possui técnicas de inferência *fuzzy*. O sistema é capaz de comparar as imagens antes de serem executadas e após a execução, onde podem ser vistos mudanças em registros, também o acesso aos protocolos como o *FTP e IRC*. Entretanto, as análises foram limitadas, sem o uso de uma base própria de *malware* estando restrita a discussão de resultados de sistemas anteriores que utilizam essa arquitetura. O artigo também não apresenta uma solução completa, mas sim uma proposta para se construir uma plataforma de detecção.

Para Wu e outros (2011), mesmo que *malware* e suas variantes modificarem muito de assinaturas e também de conteúdo, elas possuem características de comportamento precisas capazes de indicar a verdadeira intenção. O método proposto por Wu e outros (2011), consiste em analisar o *malware* recolhido manualmente e extrair sua assinatura, e também utilizar as informações extraídas do seu comportamento durante sua execução. Na abordagem deles, o comportamento é definido por meio de operações básicas, como criar um arquivo, mudar uma chave, registro, etc. As operações básicas são os objetos de nível de sistema (arquivo, chave de registro, porta de rede, etc.). A solução tem potencial de analisar assinaturas e eliminar falso positivos de maneira mais eficiente, além de ser

um sistema que tem a capacidade de analisar um lote grande de *malware*. Entretanto, tem a capacidade reduzida de identificar novos comportamentos e consequentemente novos *malware*, pois os resultados mostram que utilizaram dois vírus conhecidos para análise, e também não tem a capacidade de identificar *malware* de "propagação rápida".

Mundie e Mcintiri (2013) destacam que a área de segurança da informação necessita de uma base científica para comunicar requisitos, treinamento de pessoal etc., bem como necessita uma linguagem comum entre eles. Para tanto os autores propuseram um dicionário e *taxonomia* para este fim, baseado em um modelo ontológico. Os autores ainda destacam que o uso da linguagem *OWL* pode ajudar na captura e representação de conhecimento complexo, e que este conhecimento pode ser utilizado em análises de *malware*. O conhecimento modelado atua como uma verificação de validade eliminando ambiguidade e erros, também melhora a troca de informações, ajudando a produzir uma padronização dos termos utilizados. Os termos utilizados pelos analistas são jargões que estão em evolução, mas não há normalização desses termos, mesmos os mais comuns, portanto, a construção de um vocabulário se torna necessária. Tal iniciativa pode servir para uniformização da linguagem e conhecimento da área, entretanto não foi desenvolvido um sistema para utilizar a ontologia. O foco do trabalho de Mundie e Mcintiri (2013) é a utilização por analistas de *malware* e não propriamente na análise do mesmo.

O trabalho apresentado por Grégio e outros (2014) serviu de inspiração para este trabalho. Nele foi proposta uma ontologia onde as classes foram derivadas da análise de comportamentos suspeitos que podem ser dinamicamente interpretados através de um conjunto de regras *SWRL*. Para lidar com comportamentos complexos, os autores do artigo utilizaram uma hierarquia de comportamento com categorias que identificam vários tipos de ações que podem ser executadas por programas maliciosos. O trabalho desconsidera a classificação tradicional existente, a ontologia modela padrões de comportamento do processo em alvos que podem ser, por exemplo, *mutex*, rede, registros, arquivos de sistema ou outras aplicações. Neste modelo, um processo malicioso não é "rotulado" simplesmente por possuir um comportamento suspeito. No lugar são propostos níveis de ameaças que são atualizadas utilizando regras *SWRL*. Entretanto, esta não é uma solução pronta, sendo necessário um extenso trabalho de aprimoramento da ontologia. O artigo também não apresenta modelo e sistemas que permitam a evolução da ontologia e regras de forma dinâmica.

A Tabela 3.10 apresenta os pontos fortes e limitações dos artigos analisados. Esses artigos tratam dos assuntos relacionados com *malware* por meio de ontologias e regras *SWRL* para detecção, análise de *malware* entre outros assuntos relevantes à nossa proposta. Os artigos relacionados e analisados nesta Subseção evidenciam a importância do uso de ontologias como alternativa ao surgimento e sofisticação contínua de *malware*. Eles incluem novas técnicas que dificultam a análise por assinaturas, tornando os sistemas *antivírus* baseados em assinaturas obsoletos. Assim novas técnicas de análise baseadas no comportamento são necessárias. O uso de ontologias foi aplicado à detecção em diversos contextos incluindo *malware* e *spam*. A padronização da linguagem também se destaca como elemento necessário para a área de segurança e a análise de *malware*. Os artigos não abordaram a evolução e o aprimoramento contínuo de ontologias de *malware*. Tal aprimoramento pode ser obtido por meio de técnicas como modelagem colaborativa, *crowdsourcing*, aprendizagem de máquina, entre outras. A próxima Subseção destaca artigos que abordam a modelagem de ontologias com técnicas de *crowdsourcing*.

Tabela 3.10. Pontos fortes e limitações de trabalhos sobre Ontologias e *Malware*.

TÍTULO	PONTOS FORTES	LIMITAÇÕES	REFERÊNCIA
Ontology for Malware Behavior: a Core Model Proposal	Apresenta ontologia criada para a análise de *malware*, desconsiderando as classes tradicionais existentes. Assim, os mais diversos tipos de códigos maliciosos podem ser analisados através das regras, independente das assinaturas que esses códigos possuem.	Não é uma solução pronta, sendo apenas uma base com uma ontologia que poderá ser utilizada para representar e classificar comportamentos suspeitos de forma precisa. Ainda é necessário identificar manualmente como novos *malware* serão analisados.	Grégio e outros (2014)
Identification of malware activities with rules	Mostram como os sistemas *antivírus* baseados em assinaturas estão cada vez menos efetivos, também propõe a utilização de uma classificação para o grau da ameaça encontrada.	O artigo não explica como será a utilização dos sensores para rastrear os eventos.	Jasiul e outros. (2014)
USpam – A User Centric Ontology Driven Spam Detection System	Apresenta modelos e soluções que resulta na melhora na recepção de mensagens pelos usuários, bem como melhoras para as agências de publicidades chegarem de uma forma mais precisa ao consumidor final.	A análise é dependente da avaliação dos usuários com base em regras pré-definidas. Esta avaliação ocorre de forma estática.	Shoaib e Farooq (2015)
Ontology-based Intelligent System for Malware Behavioral Analysis	O uso da ferramenta *AIDE*, ele compara a imagem de disco após a execução do *malware* com a imagem antes da execução e mostra se foi alterada, apagada ou modificada.	Apresenta apenas um exemplo superficial da arquitetura proposta. Não tem capacidade de trabalhar com classes complexas de *malware*.	Huang e outros (2010)

Anticipating Dormant Functionality in Malware: A Semantics based Approach	Através da semântica podem-se identificar os subconjuntos dos *malware* e as atividades semelhantes que eles possuem.	Pouca escalabilidade, e custo alto de processamento. Não consideram ontologias como modelos que são propícios à evolução do comportamento de *malware*.	Noor e Abbas, (2013).
Fuzzy Markup Language for Malware Behavioral Analysis	O sistema compara imagens antes e após a execução, assim é capaz de detectar mudanças em registros, também o acesso aos protocolos *FTP* e *IRC* que são informações de *login*s e quais arquivos são adicionados, alterados ou removidos.	Não utilizaram uma base própria de *malware*, mas sim resultados de sistemas anteriores que utilizam essa arquitetura. É apenas uma proposta para construir uma plataforma de detecção.	Huang e outros (2013)
Behavior based Malware Analysis and Detection	O artigo propõe uma forma dinâmica de analisar assinaturas e eliminar falsos positivos, além de um sistema que tem a capacidade de analisar lotes grandes de *malware*.	A capacidade reduzida de identificar novos comportamentos de *malware*, pois utilizaram dois vírus conhecidos para análise, e também não tem a capacidade de identificar *malware* de propagação rápida.	Wu e outros (2011)

3.2.2. Uso de Técnicas de *Crowdsourcing* na modelagem de Ontologias

A literatura apresenta diversos trabalhos sobre o uso de *crowdsourcing* para a modelagem de Ontologias, destacamos aqui (conforme método apresentado na Seção 3.1) os trabalhos considerados mais significativos a nossa proposta.

De acordo com Di Lorio e outros (2010) a maneira dos usuários utilizarem a *Web* mudou muito, usuários inexperientes podem publicar conteúdos com alguns cliques, graças a softwares de fácil utilização e gratuitos. Porém a criação de conteúdo para a *Web Semântica* utilizados para busca, classificação e raciocínio são difíceis e demorados (Di Lorio e outros, 2010). No artigo eles propõem uma ferramenta para simplificar a criação de artefatos da *Web Semântica* através do *Wiks*. Para tanto, utilizaram dois ambientes distintos, baseados em ontologias onde com ajuda aos usuários inexperientes podem criar modelos semânticos utilizando uma interface intuitiva. Estes modelos são totalmente transparentes para os usuários, onde eles podem criar e editar os casos relativos através do *Wiks*, mas nunca diretamente os modelos. A solução proposta também possui interface gráfica para ontologias. Assim, ela possui a capacidade de fornecer aos usuários interfaces simples e poderosas para manipulação de dados e utiliza uma ontologia consistente além de utilizar tecnologias estáveis que ajudam no desenvolvimento. Porém, o protótipo ainda é limitado, com poucos *widgets* na interface ontológica e um importador pesado através de *Web Service.*

Já Lin e Davis (2010) apresentam um sistema integrado para extrair estruturas ontológicas de *folksonomy* (marcação social). Para tanto, são utilizadas regras obtidas por uma ontologia superior em conjunto com um método de *crowdsourcing* para ajudar as buscas de usuários *online* com o objetivo de evoluir a ontologia, usando técnicas de mineração de dados com os termos relevantes. Entretanto, o sistema não possui uma implementação onde usuários possam utilizar diretamente a ontologia e, assim, enviar os dados para que posteriormente possam ser aproveitados (ou não) na evolução da ontologia.

Outros trabalhos, como o *Onto Assist* (Lin e outros, 2010), propõem serviços para evolução de ontologias utilizando *crowdsourcing*. O *Onto Assist* tenta capturar a intenção dos usuários através do uso da semântica, permitindo a evolução constante da ontologia. Uma grande quantidade de dados, para especificar as mudanças, é necessária para a evolução de ontologias. Esses dados vêm geralmente de especialistas em ontologia o que torna difícil manter essa evolução. Quando as técnicas computacionais não podem substituir especialistas, o *crowdsourcing* é um a alternativa para resolver problemas difíceis agregando conhecimento de um grande número de usuários *online* através de colaboração em massa. *Onto Assist* foi projetado para fornecer uma navegação simples e intuitiva além de ser um motor de busca de propósito geral; nele, os usuários têm acesso e podem acompanhar a melhoria dos serviços do sistema, isso ajuda a manter os usuários já existentes e atrair novos. Entretanto, o sistema é limitado pelo fato de necessitar de muitos trabalhadores, e também de serem recompensados monetariamente, ele necessita de uma forma para atrair os trabalhadores sem custo para continuar a pesquisa.

Mortensen e outros (2013) argumentam que a *Web Semântica* tem a capacidade e potencial de mudar a *Web*, porém enfrenta o desafio de agregar uma enorme base de conhecimento. Para resolver este problema, os autores propõem o uso de uma multidão de pessoas em um *crowdsourcing*. O objetivo foi desenvolver um *crowdsourcing* que agrega pessoas "trabalhadores" e utiliza métodos de engenharia de ontologias. Os autores ainda argumentam que com o uso da ontologia o aumento da complexidade é evidente, e os métodos automatizados para identificar os erros são limitados. Partindo deste pressuposto o *crowdsourcing* pode identificar erros significativos por meio de muitas pessoas trabalhando em conjunto. Entretanto, os artigos não representam estudos com sistemas para verificar a viabilidade da proposta.

Em outro trabalho, Mortensen e outros (2013) analisaram o desenvolvimento de ontologias grandes e complexas. Os autores propõem identificar erros em um subconjunto do *SNOMED CT* usando *crowdsourcing*, criando assim uma auditoria na ontologia existente. O sistema em questão utiliza um modelo de inferência que agrega respostas dos trabalhadores. Utilizando um parâmetro de credibilidade um trabalhador pode aperfeiçoar o modelo com regras corretas ou incorretas encontradas utilizando *crowdsourcing*. A proposta foi avaliada por meio de um subconjunto extraído do *SNOMED CT*, onde se sabia as partes corretas e as incorretas existentes. Este subconjunto foi colocado para os trabalhadores darem opiniões, a resposta foi passada por verificações para enfim completar os testes de qualificações dos trabalhadores selecionados do *crowdsourcing*. Entretanto, a ferramenta requer muitos passos manuais em função da forma de seleção dos trabalhadores e como se chega a uma proposta. Além disto, não é integrado com um modelador de ontologias.

Lukyanenko e Parsons (2012) mostram um estudo de como a modelagem conceitual pode englobar em conceitos de alto nível toda a informação sobre um domínio do qual se aborda, esta informação pode então ser especializada ou descoberta com ajuda do usuário. O artigo explora os desafios da modelagem conceitual, além do *crowdsourcing* usando fundamentos da ontologia para oferecer potenciais soluções. O produto de um *crowdsourcing* é a informação que é gerada pelo usuário com um propósito dentro do *crowdsourcing*, tornando possível coletar grande quantidade de dados através de mutirão. Ao contrário do ambiente corporativo, em *crowdsourcing* não há restrições sobre quem pode participar podendo ter um público bem mais amplo. Este estudo reforça a utilização de modelos conceituais em *crowdsourcing*, porém com a utilização de muitos usuários com pensamentos e abstrações da realidade diferentes o sistema será propenso a erros, porém nota-se que o domínio utilizado é pequeno, sendo necessário de uma maior abrangência de classes.

A Tabela 3.11 apresenta os pontos fortes e limitações dos artigos analisados. Esses artigos evidenciam a importância do estudo da modelagem de ontologias utilizando de *crowdsourcing*. Nestes estudos, *crowdsourcing* é utilizado como ferramenta para se criar ou acrescentar informações dentro de uma ontologia. Partem do pressuposto da necessidade de utilizar inteligência colaborativa para trabalhos de análise que envolve grande mão de obra e de inteligência coletiva. Esta parte da pesquisa demonstrou como o

conhecimento colaborativo contribui para o desenvolvimento de ontologias auxiliando-as a agregar novos saberes aumentando sua capacidade. A próxima subseção destaca estudos sobre análise de *malware* com *crowdsourcing.*

Tabela 3.11: Pontos fortes e limitações de trabalhos sobre Crowdsourcing e Ontologias.

TITULO	PONTOS FORTES	LIMITAÇÕES	REFERÊNCIA
Crowdsourcing semantic content: a model and two applications	O sistema destaca-se pela capacidade de fornecer aos usuários interfaces simples e poderosas e por utilizar uma ontologia consistente, além de utilizar tecnologias estáveis que ajudam no desenvolvimento.	Ainda é um protótipo com poucos *widgets* na interface ontológica e um importador em um *Web Service*.	Di Lorio e outros (2010)
Computational and Crowdsourcing Methods for Extracting Ontological Structure from Folksonomy	Explora como utilizar um modelo de *crowdsourcing* com usuários sem a participação de peritos em ontologias. Também apresenta um motor de busca semântica para integração.	Não tem uma implementação onde usuários possam utilizar diretamente a ontologia, sem enviar os dados para que posteriormente possam ser utilizados ou não na ontologia.	Lin e Davis (2010)
Ontological Services Using Crowdsourcing	Apresenta um sistema onde propõem a evolução da ontologia com o serviço de *crowdsourcing*.	Apresenta uso em domínios limitados do sistema, pelo fato de necessitar de muitos trabalhadores.	Lin e outros (2010)
Ontology Quality Assurance with the Crowd	Utiliza *crowdsourcing* para verificação de uma ontologia em grande escala.	Apresenta um método sem estudo empírico para verificar custo e desempenho.	Mortensen e outros (2013)
Crowdsourcing the Verification of Relationships in Biomedical Ontologies	Apresenta boa alternativa para a seleção dos trabalhadores, e também para como chegar a um consenso sobre uma resposta em um *crowdsourcing*.	Requer muitos passos manuais e não é integrado a um modelador de ontologias.	Mortensen e outros (2013)
Conceptual Modeling Principles for Crowdsourcing	Este estudo reforça a utilização de modelos conceituais em *crowdsourcing*,	O domínio utilizado é pequeno, sendo necessária uma abrangência, mais ampla de classes na gramatica utilizada.	Lukyanenko e Parsons (2012)

3.2.3. Uso de Técnicas de *Crowdsourcing* na Análise e Detecção de *Malware*

Esta Subseção apresenta os trabalhos considerados mais significativos para esta dissertação no que diz respeito ao uso de *crowdsourcing* para análise e detecção de *malware*. Kanich e outros (2011), por exemplo, exploram uma abordagem de estudo sobre mercados na Internet que oferecem serviços a criminosos, colocando em risco

computadores propensos a vulnerabilidades e a exploração. Dentro desses mercados o criminoso pode comprar acesso à uma ou mais máquinas, e é vendido acesso a *hosts*, tudo a um alto valor. Ainda sobre os trabalhadores deste submundo do *crowdsourcing*, que agem executando programas maliciosos em troca de pequenas recompensas. Foram testadas em máquinas de usuários de *crowdsourcing e* as informações foram coletadas de três fontes, *JavaScript*, relatórios do *antivírus*, e um programa executado pelo "trabalhador" do *crowdsourcing* fora do contexto do navegador. No trabalho deles foi utilizada uma compensação monetária para poder ter acesso à falha de máquinas de usuários, destaca-se a visualização de como máquinas são facilmente invadidas e infectadas, além disso, os *plug-ins* essenciais dos navegadores deixam as máquinas como potenciais vítimas dos atacantes. Entretanto, *antivírus* baseados em rede poderiam facilmente identificar o software malicioso nos casos apresentados, e também a técnica manual utilizada proposta resulta em baixa infecção diária.

O trabalho de Hu e outros (2011) mostra como a incidência de envios de *spam* pela Internet se tornou um problema sério em redes sociais, blogs, e-mail e fóruns *online*. Estes meios passaram a hospedar anúncios *spam*, *phishing* e *malware* que prejudicam as máquinas dos usuários. Os mecanismos de defesas, tais como *antivírus* e *firewall* se mostram limitados para conter essas ameaças, principalmente quando os atacantes empregam novas estratégias para ataques. O trabalho de pesquisa apresenta o sistema *CUD (Crowdsource for URL Spam Detection)* como ferramenta de detecção de apoio a ferramentas existentes, o sistema através de *crowdsourcing* utiliza a ajuda humana com análise de comentários de usuários na Internet para a detecção automática dessas pragas. O trabalho ressalta o motivo pelo qual algumas páginas da *Web* de *spam* possam ser facilmente detectadas por seres humanos, mas são de difícil detecção pelos computadores. O uso do *crowdsourcing* com inteligência humana melhora essa detecção, porém pelo fato de utilizar comentários em linguagem natural, novos *spam* com novas estratégias podem passar despercebidos pelo sistema e atingir o seu objetivo.

Christoforidis e outros (2014) mostram como no ambiente da Internet existem muitos alvos que executam softwares com as mesmas vulnerabilidades. Ao utilizar inteligência humana por meio de *crowdsourcing*, pode restringir essa vulnerabilidade por um determinado período. Os autores apresentam a análise do *Net Buckler*, um aplicativo que utiliza *crowdsourcing* para se defender contra *worms* na Internet. O sistema cria uma

rede *peer-to-peer* utilizando o framework *JXTA*[11], usuários são reunidos em pares e fazem as trocas de informações relacionadas ao tráfego da rede. Conforme a informação que eles têm, adicionam medidas de segurança; ao utilizar uma partilha de conhecimentos dos usuários da rede sobre possíveis ataques, aumentam a possibilidade de tomada de decisões corretas contra as ameaças. O aplicativo proposto possui uma interface gráfica que permite usuários interagir e controlar a aplicação, o usuário também pode decidir qual aplicação parar. Por exemplo, em caso de *worm*, a aplicação explicita o sistema operacional do *host* onde está sendo executado o aplicativo malicioso. Este é um sistema descentralizado, não necessita de alertas dos usuários que fazem as detecções. Entretanto, para a utilização do sistema o usuário tem que ter um bom conhecimento, por exemplo, mais é necessário conhecimento de implementação para detectar outros tipos de pragas virtuais, além disto, a autenticação é falha podendo usuários desonestos se passar por honestos e quebrar as barreiras de segurança imposta pelos outros usuários do sistema.

Para Thuan e outros (2014), a análise de *malware* baseada em comportamento tem maior capacidade de detectar vírus desconhecidos, por este motivo, os autores propuseram uma arquitetura utilizando *crowdsourcing* para a construção de uma base de conhecimento *antivírus*. Para tanto, foram introduzidos mecanismos para coleta das informações dos usuários (*feedback*), o uso de especialistas e *crowdsourcing* para a análise e classificação; sendo após aplicado algoritmo de aprendizado para classificação final. Este sistema foi chamado pelos autores de *Crowd MAV* (*Applying Crowdsourcing to Machine Learning Anti-Virus System*). Na aplicação de *crowdsourcing*, primeiro é aberta uma chamada pedindo para o usuário enviar insumos para o sistema, depois a chamada é enviada para os participantes do *crowdsourcing* e se aprovado é enviado uma chamada para novas participações. O sistema possui capacidade de lidar com uma quantidade grande de dados pelo uso do *crowdsourcing* e melhorias na técnica de detecção por análise de comportamento; porém, não está completamente claro como é feita atualização por meio de aprendizado de máquina, e também ao utilizar especialistas se torna uma abordagem cara.

Barakat e outros (2013) propõem uma abordagem para dar uma resposta eficiente aos utilizadores sobre arquivos suspeitos e assim reduzir o número de *malware*

[11] *The Language and Platform Independent Protocol for P2P Networking*, https://jxta.kenai.com/

desconhecidos. Este estudo propõe um analisador de *malware* automático e escalável, sendo capaz de analisar rapidamente e gerar um relatório para cada *malware* desconhecido detectado. O sistema *SCARECROW* tem o propósito de acelerar e automatizar o processo de detecção de *malware*, além de utilizar técnica *crowdsourcing* envolvendo o usuário para relatar arquivos suspeitos. O sistema utiliza os dois tipos de análise estática e dinâmica. Estática quando um *malware* ou sua assinatura já existe no banco de dados, dinâmica quando programas desconhecidos ou arquivos executáveis são suspeitos. Para tanto, é feita uma análise mais profunda com a participação do usuário do *crowdsourcing* e, após isto, é gerada uma resposta para o usuário final. Os usuários finais são capazes de obter opiniões de especialistas sobre qualquer programa que deseja baixar. O *SCARECROW* tem uma abordagem onde os usuários podem submeter arquivos suspeitos e receber uma resposta, essas respostas incluem todas as informações acerca do arquivo, por exemplo, se é um *malware* já detectado, ou uma nova amostra de *malware* ou não é software malicioso. Entretanto, o uso de especialistas encarece o sistema, além de serem necessárias máquinas potentes para a utilização em nuvem. Além disso, ao enviar o arquivo para uma agência de *cyber segurança* os custos se tornam altos e ocorre a demora no retorno das informações.

A Tabela 3.12 apresenta os pontos fortes e limitações dos artigos analisados nesta Subseção. Os artigos apresentam diferentes abordagens de utilização de *crowdsourcing* para agregar conhecimento coletivo para combater a complexidade dos novos *malware*. Ao aprofundar sistematicamente nesta abordagem podemos notar a importância dela para entender o funcionamento, agregar conhecimento e combater este problema que atinge os usuários de sistemas de forma geral. Os artigos remetem a necessidade de novas técnicas de detecção, de modelos dinâmicos e conhecimento intensivo que demanda grandes esforços. Do lado empírico, os estudos apresentados apontam *crowdsourcing* como alternativa para proporcionar meios para satisfazer tais necessidades.

Tabela 3.12. Pontos fortes e limitações de trabalhos sobre *Crowdsourcing* e *Malware*.

TÍTULO	PONTOS FORTES	LIMITAÇÕES	REFERÊNCIA
Putting Out a HIT: Crowdsourcing Malware Installs	O artigo expõe de forma clara como máquinas são facilmente invadidas e infectadas. Também apresenta como os *plug-ins* essenciais dos navegadores deixam as máquinas como potencial vítima dos atacantes. Relata ainda o *Mechanical Turk* que são trabalhos feitos em equipes (*crowdsourcing*), denominados como inteligência coletiva.	*Antivírus* baseado em rede poderia facilmente identificar o software malicioso nos casos apresentados sem a necessidade da inteligência coletiva, e também a técnica manual utilizada que pressupõe baixa infecção diária.	Kanich e outros (2011)
CUD: Crowdsourcing for URL Spam Detection	Apresenta como o uso do *crowdsourcing* com inteligência humana melhorara detecção de páginas *web* de *spam*.	Pelo fato de utilizar comentários em linguagem natural, novos *spams* com novas estratégias podem passar despercebidos pelo sistema e atingir o seu objetivo.	Hu, e outros (2011)
CrowdSource: Automated inference of high level malware functionality from low-level symbols using a crowd trained machine learning model	O artigo destaca a capacidade de inferência realizada automaticamente em dados extraídos a partir de binários de *malware*.	O sistema não tem um processamento dinâmico apenas extático, o que diminui a capacidade do sistema. O *crowdsourcing* é utilizado apenas no processo de treinamento.	Saxe, e outros (2014)
A Crowdsourcing Approach to Protect Against Novel Malware Threats	Este sistema descentralizado não necessita de alertas, pois os próprios usuários são capazes de fazer as detecções o que em segurança da informação pode ser uma desvantagem.	O usuário deve ter um bom conhecimento, pois necessita novas implementações para detectar outros tipos de *malware*. A autenticação é falha permitindo usuários mal-intencionados se passarem por honestos.	Christoforidis e outros (2014)
An Architecture Utilizing the Crowd for Building an Anti-virus Knowledge Base	Permite lidar com uma quantidade grande de dados pelo uso do *crowdsourcing*. Apresenta uma melhora na técnica de detecção por análise de comportamento.	Não está completamente claro como é feita atualização por meio de aprendizado de máquina, e também ao utilizar especialistas se torna uma abordagem cara.	Thuan,e outros (2014)
SCARECROW: Scalable Malware Reporting, Detection and Analysis	Os usuários podem submeter arquivos suspeitos e por meio de *crowdsourcing* de especialistas obter respostas que incluem todas as informações que acerca o arquivo.	O uso de especialistas encarece o sistema, além de serem necessárias máquinas potentes para a utilização em nuvem, quando ao enviar o arquivo para uma agência de *cyber* segurança os custos se tornam alto e pode ocorrer demora o retorno das informações.	Barakat e outros (2013)

3.3. Considerações Sobre a Análise dos Trabalhos Relacionados

A utilização de ontologias para análise de *malware* é uma linha recente de pesquisa, mas que possui potencial de avanço por meio de pesquisas em *Web Semântica. Crowdsourcing* se apresenta como uma alternativa para tarefas intensivas de mão de obra, tais como a modelagem de ontologias e análise e detecção de *malware*. Entretanto, conforme destacado, não foram encontrados na extensa literatura existente trabalhos que fazem uso de *crowdsourcing* em modelagens de ontologias para análise de *malware*. Destacamos assim artigos encontrados que se aproximam deste tema e que certamente serão relevantes ao posicionamento da nossa proposta.

Para tanto, este Capítulo descreveu os principais trabalhos relacionados que investigam os termos de pesquisa agrupados dois a dois, as propostas existentes remetem a análise da *malware* com ontologia e *crowdsourcing* para análise de *malware* e *crowdsourcing* com ontologia. Tais trabalhos apresentam novas soluções e clarificam desafios a serem abordados em cada caso.

Como destacado anteriormente, com esta proposta envolve *malware* e ontologias, o trabalho de Grégio e outros (2014) serve de ponto de partida para a solução.

Artigos da revisão bibliográfica apresentada também apontam a limitação baseada em análise de código e no uso dessas classificações.

Entretanto, este trabalho dá um passo à frente em relação à literatura analisada, ao considerar o *crowdsourcing* como elemento para tornar possível a evolução de ontologias e a análise de *malware*. Muitos dos trabalhos apresentados relatam experiências positivas nos dois casos. Esta proposta se diferencia das demais e tem o objetivo, em longo prazo, de contribuir para uma análise precisa e completa de *malware*, mesmo os com novos tipos de comportamentos, uma vez que ontologias são estruturas flexíveis passíveis de evolução por meio de *crowdsourcing*. Assim, os usuários além de agregar conhecimento poderão interagir diretamente com a ontologia, através de uma interface que facilitará esta tarefa.

Com base nesta análise, definimos um modelo para o desenvolvimento da nossa proposta para fazer a integração de ontologia com uma plataforma colaborativa para analisar comportamentos maliciosos. Tal modelo é apresentado no próximo Capítulo.

4. Um Modelo de Plataforma Colaborativa para Evolução de Ontologias de Análise de *Malware*

Segundo Grégio e outros (2016), a análise de comportamento é uma maneira que pode ser eficaz no descobrimento de padrões de sistemas maliciosos, e com o uso de ontologias é possível aprimorar a análise de comportamentos de programas. Este Capítulo apresenta um modelo de *crowdsourcing* onde usuários e especialistas no domínio podem compartilhar conhecimento sobre *malware* para evolução de ontologias, alimentando a análise de *malware* de maneira controlada e segura. A Seção 4.1 apresenta os requisitos e aspectos chaves do design do modelo de *crowdsourcing* proposto, que é descrito na Seção 4.2.

4.1. Design do Modelo de *Crowdsourcing*

O modelo foi construído levando em consideração os problemas de análise de softwares maliciosos e a evolução colaborativa de ontologias, mostrado na literatura estudada (*cf.*, Capítulos 2 e 3). Os trabalhos apresentados no Capítulo 3 mostram que este é um caminho viável para melhorar as técnicas de análise de *malware*, porém, como ontologias não são uma tecnologia de amplo conhecimento, inclusive entre os profissionais de segurança de informação, foi definida a construção de uma plataforma com mecanismos que facilitam a interação e evolução da ontologia. Nesta plataforma os usuários podem colocar dúvidas e questões, conhecendo ou não a utilização de uma linguagem de especificação de ontologias. Assim eles podem agregar conhecimento e auxiliar na construção de regras da ontologia, por meio do envio de contribuições para especialistas e moderadores, que a partir da análise dessas contribuições poderão manter a ontologia atualizada e aumentar a eficácia do sistema utilizando, da ajuda da inteligência coletiva proporcionada pela plataforma. A Figura 4.1 mostra um diagrama com o caminho de uma contribuição no modelo proposto, ou seja, as contribuições surgem da comunidade de *crowdsourcing*, então são submetidas para a aprovação e, por fim, são incorporadas à ontologia. A elipse maior mostra o *crowdsourcing* onde usuários podem fazer contribuições, as quais passam por aprovações e finalmente são inseridas como elementos da ontologia.

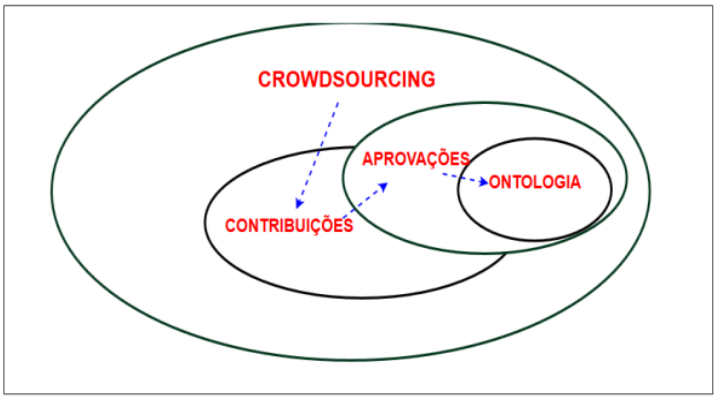


Figura 4.1: Modelo conceitual do *crowdsourcing*.

O modelo foi construído com base em dois aspectos principais:

- Revisão sistemática da literatura, conforme apresentada no Capítulo anterior.
- Consulta aos especialistas do domínio, um em segurança e outro em ontologias.

Durante a revisão da literatura, como citado, observamos não existir um modelo que abrange os três aspectos centrais desse modelo (*crowdsourcing*, ontologias e análise de *malware*). Entretanto, os artigos analisados apresentam muitos aspectos importantes a serem considerados neste modelo, bem como apresentam limitações a serem consideradas e a necessidade de adaptar as soluções ao foco deste trabalho.

Na revisão da literatura foi encontrada uma vasta massa de dados sobre as tecnologias utilizadas, *crowdsourcing*, comportamento de *malware* e ontologias, evidenciando que é um assunto complexo e que está em constante busca de soluções. De maneira breve destaca-se, neste trabalho, quatro requisitos chaves considerados no projeto do nosso modelo:

(1) *Necessidade de segurança nas próprias contribuições no crowdsourcing*. Embora trabalhos sobre *crowdsourcing* e ontologias apresentem modelos mais abertos de contribuição indiscriminada, os trabalhos voltados a segurança demonstram a necessidade de controlar estas contribuições de modo a não permitir que usuários desconhecidos (e mal-intencionados)

tirem proveito ou provoquem falhas no modelo. Portanto, devemos levar em consideração a segurança do sistema de modo a proteger e manter a qualidade e consistência da ontologia já existente, onde apenas os especialistas de domínio têm a permissão de modificá-la de modo direto. O usuário ao fazer o cadastro inicial, por exemplo, pode não ser liberado para utilização do sistema, isto é, ele só é liberado com a aprovação do moderador;

(2) *Necessidade de promover mecanismos indiretos e facilitados para contribuição na ontologia*. A revisão destacou uma distância entre as áreas de segurança e *Web Semântica*, de modo que ontologia não é maneira mais comum para o compartilhamento de conhecimento na área de segurança. Portanto, não podemos restringir a colaboração via especificação em uma linguagem formal de ontologias. Por exemplo, todos os usuários poderiam fazer suas contribuições livres por meio de discussões abertas em linguagem natural sem a necessidade de conhecimento algum em linguagens para especificação de ontologias. Usuários ainda poderiam, por exemplo, propor regras usando em texto descritivo em linguagem natural ou também em *SWRL* (caso tenham conhecimento). Em caso de textos em linguagem natural o usuário não necessita saber a sintaxe complexa das regras *SWRL*, bastando saber se expressar em forma de uma regra comum dos tipos *e, ..., então, ..., se ..., então*;

(3) *Necessidade de respeitar a estrutura da ontologia.* Ao mesmo tempo que devemos ter mecanismos para contribuição indireta (item anterior) alguns trabalhos sobre evolução colaborativa de modelos alertam para necessidade de manter uma estrutura consistente da ontologia. Como destacado anteriormente, a evolução da ontologia é complexa, mas com a utilização do *crowdsourcing* pode ser facilitada, com uma estruturação no sistema com tipos de contribuições de acordo com o nível de usuários. Assim é possível evitar "danos" (intencionais ou não) à ontologia;

(4) *Necessidade de recompensa*. A maioria dos trabalhos em *crowdsourcing* apresenta a necessidade de oferecer recompensa (monetária ou não) para os participantes, uma alternativa é trabalhar a pontuação e status do usuário

no sistema. O acúmulo de pontos no fórum e contribuições podem levar os usuários a subirem de nível, podendo evoluir dentro do sistema conforme o uso.

Adicionalmente à revisão da literatura, foram ouvidos especialistas em ontologias de *malware* e especialistas em segurança da informação de modo a coletar requisitos para o design do modelo de *crowdsourcing* proposto. Eles reforçaram a importância dos aspectos já levantados com a revisão da literatura, mas também destacaram aspectos adicionais, sendo os três principais:

(1) *O uso de ontologias OWL complementado com regras SWRL* como mecanismos para representação completa do conhecimento compartilhado. O modelo pode ser estruturado em função dos elementos que compõem estas linguagens, possibilitando, por exemplo, a inclusão de regras isoladas ou novas classes, propriedades e instâncias na ontologia;

(2) *Necessidade de controle de alterações e versões para ontologia*, bem como a possibilidade de desfazer alterações propostas. Muitos efeitos no modelo podem só ser perceptíveis em longo prazo, por exemplo, após a experimentação com ferramentas e validação com logs de execução de *malware*. Portanto, neste caso as opções de desfazer são essenciais;

(3) *Inclusão de ferramentas para download* que fazem uso da ontologia dentro do ambiente de *crowdsourcing*. É importante a validação dos modelos perante *logs* de execução por meio de ferramentas que fazem uso deles. Todas as ferramentas utilizadas para a execução da ontologia serão disponibilizadas para *download* onde os usuários poderão utilizar localmente em sua máquina, facilitando assim, o entendimento e melhorando o compartilhamento do conhecimento.

Por meio da consulta com especialistas foi definido um modelo de contribuição, onde usuários podem utilizar as informações contextuais para a definição de regras. A ideia é distribuir os tipos de contribuições de modo a aproveitar a inteligência coletiva que incluem ideias de pessoas que vão além dos especialistas de ontologias.

4.2. Descrição do Modelo de *Crowdsourcing*

Este modelo, portanto, visa possibilitar a criação de uma comunidade de *crowdsourcing* onde usuários propõem melhorias para a ontologia de *malware*. Nele, o sistema atua em conjunto com ferramentas que visam apoiar a seleção de informações sobre comportamentos suspeitos de programas utilizando-se de colaboração.

O acesso ao sistema, por motivos de segurança, é restrito a pessoas cadastradas. A Figura 4.2 mostra que o usuário realiza o cadastro, e passa pelo moderador onde é feita a liberação ou não do usuário.

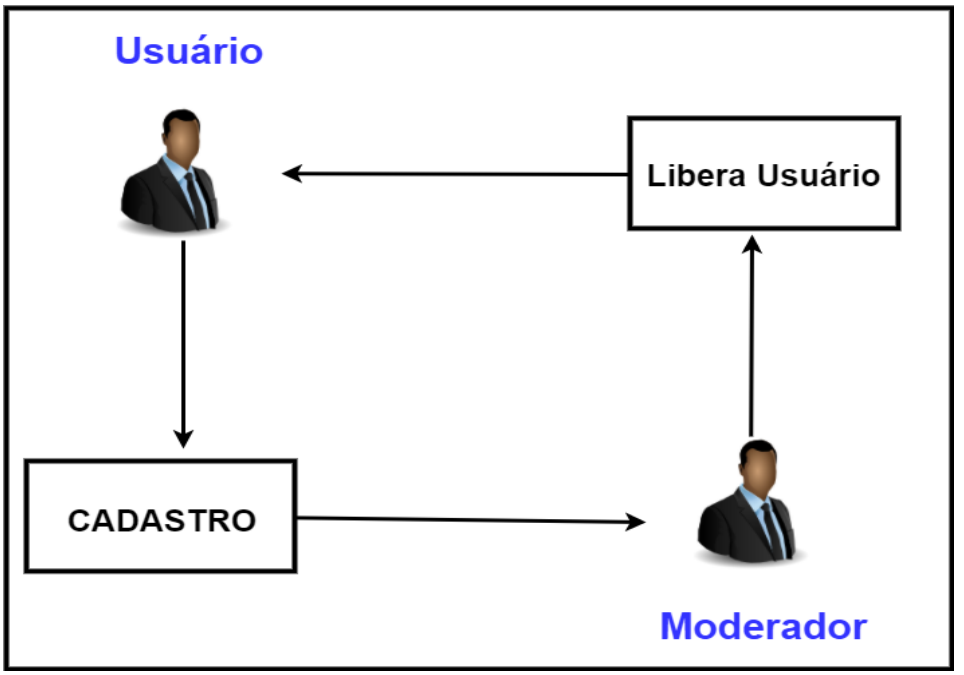


Figura 4.2: Diagrama de cadastro do Usuário.

Uma vez cadastrado, os usuários serão classificado como iniciantes. Ao todo, o modelo prevê cinco níveis diferentes de acessos além do administrador do sistema, conforme apresentado a seguir:

- *Iniciante*: Para o usuário iniciante é previsto o acesso ao sistema e a autorização para ler a ontologia e contribuir no fórum. Os comentários marcados por outros usuários como positivos servem de pontuação para subir de nível no sistema (mecanismo de recompensa). Estes usuários podem fazer *download* da ontologia e de documentos. A Figura 4.3 detalha as atividades que um usuário iniciante pode fazer no sistema. O usuário faz o cadastro e após aprovado, ter acesso às funcionalidades permitidas neste nível, sendo elas: fórum de discussão, d*ownload* de documentos relativos ao sistema e *download* de versões da ontologia;

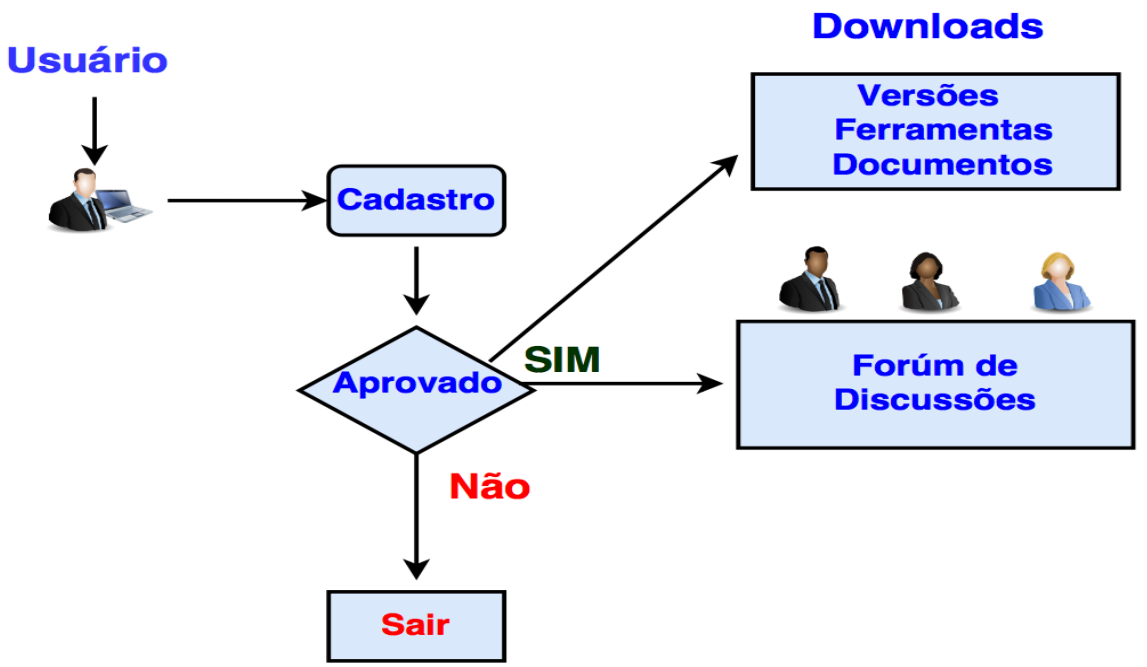


Figura 4.3: Diagrama de acesso do usuário iniciante.

- *Médio*: Um usuário se tornará médio com *likes* no fórum (foi inicialmente proposto 10), ou através de convite por um usuário *expert* ou do moderador. Este usuário pode fazer todos os acessos do usuário iniciante, mas também propor novas regras através de descrição em texto livre, ou em linguagem *SWRL* (se o usuário tiver conhecimento da linguagem). Este usuário também terá acesso a ferramentas de análise que faz uso da ontologia. A Figura 4.4 detalha as atividades que um usuário médio pode fazer no sistema, incluindo: fazer *download* de versões da ontologia, fórum de discussões e contribuição de regras;

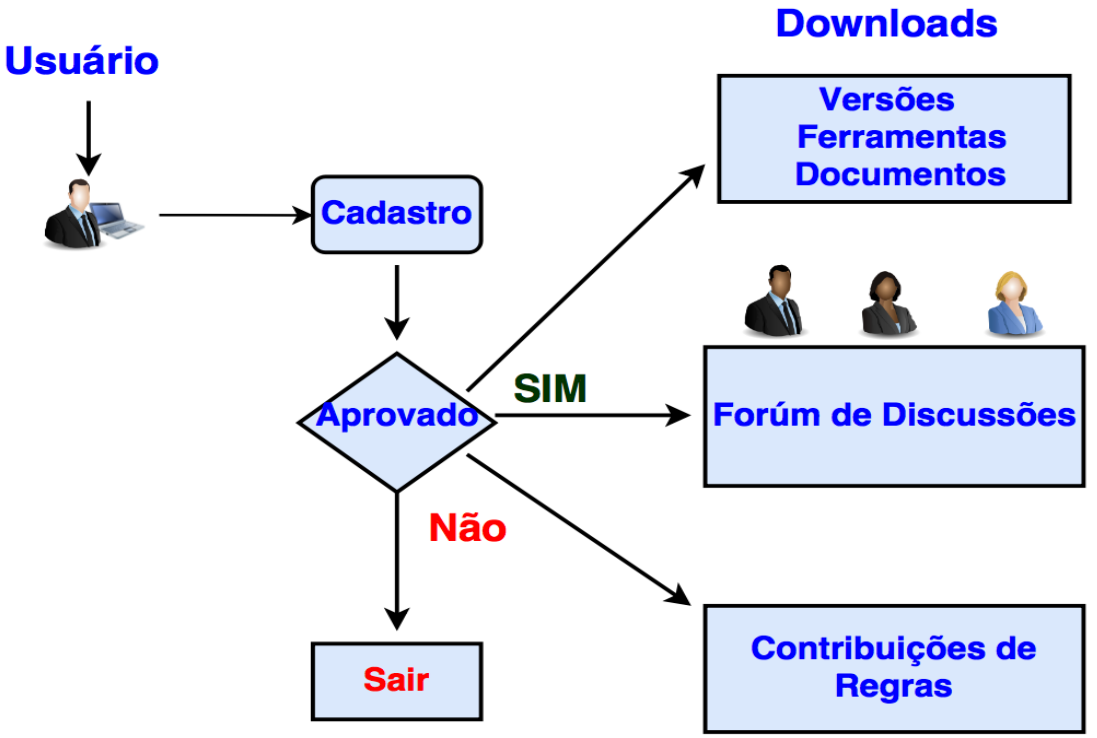


Figura 4.4: Diagrama de acesso do usuário médio.

- *Avançado*: Um usuário pode chegar ao nível de avançado apenas com o convite do moderador, sendo necessário já estar no nível médio. Além de ter acesso a todas as funcionalidades do nível médio, o usuário de nível avançado pode também propor novas regras na própria ontologia via sistema, estas mudanças devem ser aprovadas por um usuário *expert* ou moderador. Ele/ela também contribui com novos elementos da ontologia, como novas classes. A Figura 4.5 detalha as atividades que um usuário avançado pode fazer no sistema, incluindo fazer *download* das versões da ontologia, de ferramenta e documentos; *upload* de ferramentas e documentos; acesso a fórum de discussões; e contribuição de regras e contribuições de classes;

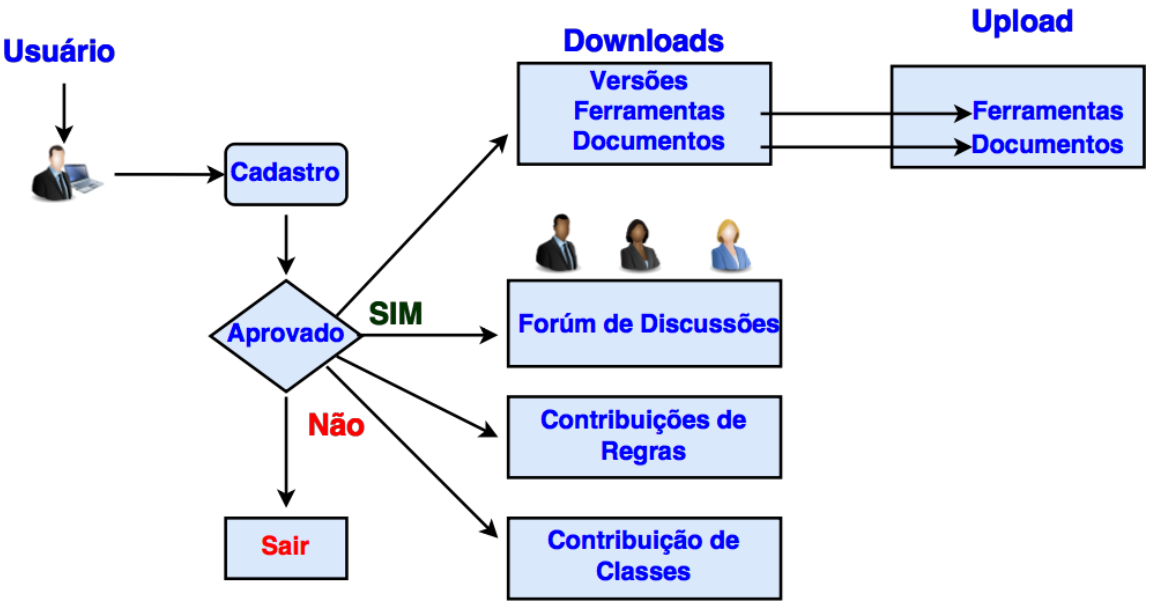


Figura 4.5: Diagrama de acesso do usuário avançado.

- *Expert*: Um usuário poderá chegar ao nível *expert* apenas por meio de convite do moderador (es), sendo necessário já estar no nível avançado para receber o convite. Além de ter acesso a todas as funcionalidades do nível avançado, o usuário de nível *expert* pode também propor mudanças na estrutura da própria ontologia via sistema. Mudanças de usuários *experts* podem refeletir diretamente na segurança do sistema pois, podem contribuir com uma nova versão da ontologia no controle de versão, sem necessidade de aprovação. Um usuário *expert* também contribui com os moderadores na administração do sistema e deve zelar pela segurança do sistema. Conforme apresentada na Figura 4.6 este usuário pode fazer *download* de versões, ferramentas e documentos; *upload* de versões, ferramentas e documentos; fórum de discussões; contribuição de regras e classes; aprovações de regras e classes; criar e editar grupos; e liberar ou bloquear usuários.

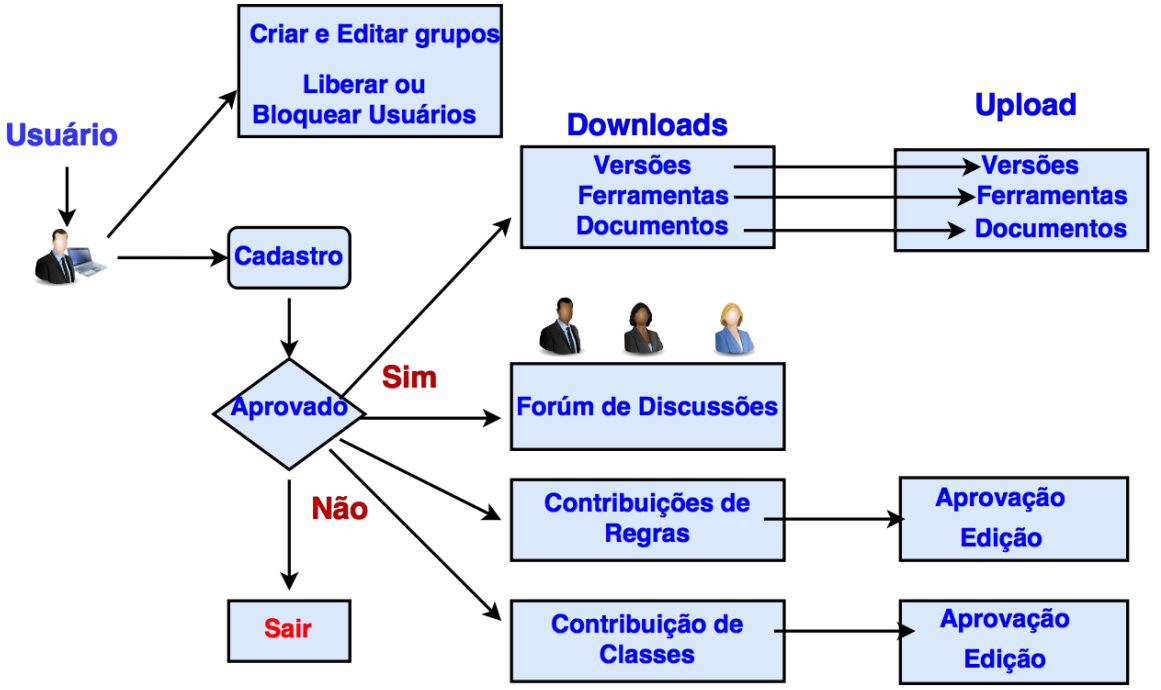


Figura 4.6: Diagrama de acesso do usuário *expert*.

- *Moderador*: O moderador é quem gerencia a plataforma. Um *expert* pode se tornar moderador à medida que o sistema cresça em número de usuários e necessite. Isto só ocorre via convite de outro moderador. Um moderador tem acesso a todas as funcionalidades, ele/ela poderá ainda excluir regras ou até uma nova versão da ontologia disponibilizada, sem necessidade de aprovação, ou mesmo, desfazer alterações de qualquer usuário. O moderador pode gerenciar usuários, eliminar usuários e tem acesso a todas as funcionalidades de configuração do sistema. A Figura 4.7 detalha as atividades que um moderador pode fazer no sistema, incluindo fazer *download*, *upload* e exclusão de ferramentas, versões e documentos; fórum de discussão; contribuição de regras e classes; aprovação e edição de regras e classes; criar e editar grupos; liberar ou bloquear usuários; e tem acesso aos registros de atividades.

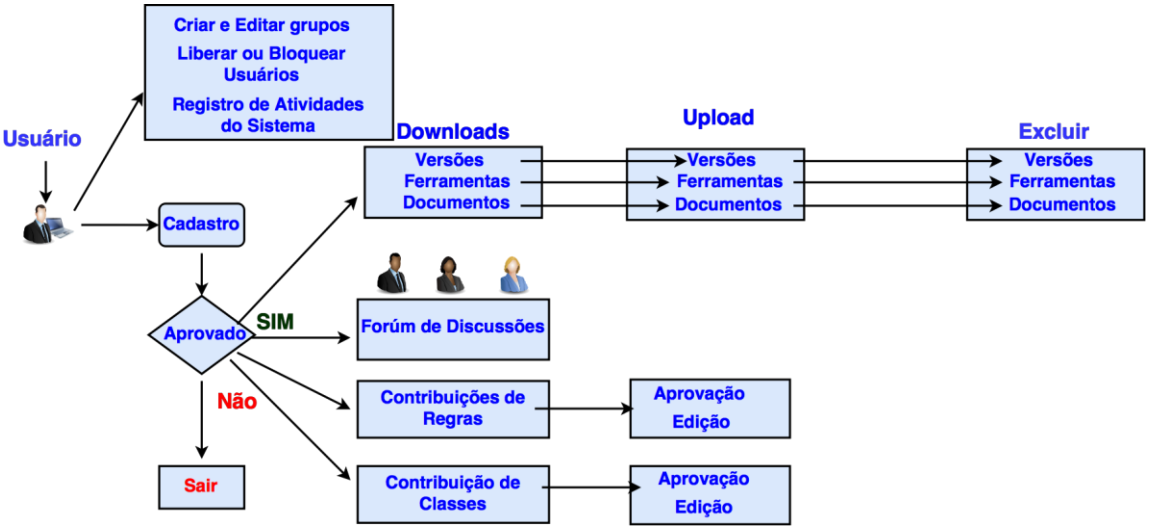


Figura 4.7: Diagrama de acesso do usuário moderador.

As formas de contribuições previstas no modelo de *crowdsourcing* proposto são:

- **Postagem no fórum**: Com o objetivo de proporcionar ao usuário um local onde possam ter discussões livres sobre qualquer assunto relativo à *malware*, o modelo prevê a construção de um fórum de discussão. Dentro do fórum não deve existir tomada de decisões, mas ser um ambiente livre onde pode-se melhorar o entendimento sobre comportamentos por meio da colaboração entre os participantes. Usuários podem avaliar as postagens e classificá-las como relevantes, garantindo pontuação e status aos participantes. Usuários mais avançados podem ainda orientar e ajudar outros os usuários do fórum. No modelo, qualquer um que tem a liberação para usar o sistema terá acesso ao fórum de discussões. O fórum ainda pode ajudar no aprendizado dos usuários iniciantes em ontologia;
- **Regras *SWRL***: Os usuários podem contribuir com alterações em regras já existentes na ontologia, com propostas de novas regras ou com textos explicativos sobre conceitos de regras, caso não tenham familiaridade com a linguagem. Quando o usuário que tiver permissão para incorporar regras novas na ontologia, isto será realizado por meio de controle de versão, evitando assim, perder versões estáveis da ontologia;
- **Classes para a ontologia**: usuários poderão contribuir com elementos da ontologia, tais como classes. Estas contribuições podem ser diretas por meio de controle de versão para usuários mais experientes, ou contribuições indiretas com sugestões em texto que são analisadas por usuários experientes;
- **Novas versões da ontologia**: usuários mais experientes poderão inserir novas versões da ontologia diretamente no controle de versão. Para tanto, podem ser utilizados ferramentas externas como o Protégé. Toda modificação na ontologia deve ser registrada como uma nova versão.

Conforme o modelo proposto, os especialistas avaliam e tomam as ações necessárias conforme a Tabela 4.1. A tabela específica como os especialistas agem ao analisar uma contribuição, incluindo verificar se ela descreve realmente um comportamento suspeito, verificar se a situação já foi reportada, e por fim, inserir a contribuição na ontologia.

Tabela 4.1: Utilização e ação dos especialistas com as contribuições.

DESCRIÇÃO	ACONTECIMENTO	SITUAÇÃO	AÇÃO
Postagem de respostas ou perguntas no fórum de discussão	Postagem descreve um comportamento de malware	Problema já existe ou alguma coisa similar	Encontrar a causa do problema, o tipo de ação e possíveis danos
Postagem de uma regra em texto descritivo	Ideia descreve um comportamento de malware	Já existe uma regra, para o problema ou para um comportamento similar	Criação da regra ou modificação de alguma existente
Postagem de uma regra em linguagem SWRL	Regra descreve um comportamento de malware	Já existe uma regra, para o problema ou para um comportamento similar	Análise e correção da regra (se necessário), e inserção na ontologia
Postagem de uma classe em linguagem OWL	Classe descreve um comportamento de malware	Já existe essa classe ou se relaciona com a ontologia	Análise e correção da classe (se necessário), e inserção na ontologia

Com estas formas de contribuições é esperada uma evolução compartilhada da ontologia, uma vez que, os usuários podem analisar os comportamentos de forma colaborativa, tornar a ontologia sempre atualizada com relação a novos *malware*, bem como manter sua qualidade e evolução constante. Por meio do controle de versão, todas as versões da ontologia ficam disponíveis, sendo necessário para a manutenção do sistema.

Neste Capítulo, foi apresentada a engenharia do modelo e a descrição conceitual do modelo proposto, incluindo as suas formas de acesso e níveis. No próximo Capítulo será apresentado o desenvolvimento do sistema.

5. Desenvolvimento do Sistema CrOMa (*Crowdsourcing* para Ontologia de *Malware*)

Conforme destacado anteriormente, ao utilizarmos ontologias podemos modelar comportamentos típicos de um programa malicioso, e consequentemente analisar ações indesejadas. Uma linguagem formal como a *OWL* é desejável para capturar programas com comportamentos complexos, tais com os comportamentos dos *malware* atuais (Mundie e Mcintire, 2013).

A pesquisa em que esta dissertação se insere iniciou-se com o estudo dos comportamentos de programas maliciosos para poder, assim, definir o comportamento destes softwares. Esses comportamentos foram utilizados para definir regras para serem analisadas e definir o nível de cada ameaça. Este trabalho iniciou-se por Grégio e outros (2014), com objetivo de criar medidas de prevenção.

A partir disto, foi desenvolvida uma ontologia (Grégio e outros, 2016) para modelar os conceitos utilizados e descrever as regras semânticas descritas em *SWRL* que são utilizadas para análise de *malware*.

Portanto esta dissertação situa-se em um projeto maior constituído de três fases:

1. Análise de comportamentos de softwares maliciosos;
2. Modelagem da Ontologia, para criação das regras e arquitetura básica;
3. Desenvolvimento de plataforma de *Crowdsourcing*, que é o foco principal desta dissertação.

Este Capítulo apresenta o sistema CrOMa, que implementa o modelo apresentado no Capítulo anterior. Nele os usuários podem interagir e sugerir aprimoramentos e alterações em uma ontologia compartilhada por meio de uma interface simples e intuitiva (se comparadas a descrições em *OWL* e *SWRL*). O sistema CrOMa é parte de uma arquitetura para análise de *malware* descrita em trabalhos anteriores (Grégio e outros, 2016), mas também pode ser utilizado de forma individual; não sendo, portando, dependente de ferramentas de análise ou de um modelo específico. A Seção 5.1 apresenta brevemente a arquitetura geral da solução e detalha a arquitetura do sistema CrOMa. A Seção 5.2 apresenta as tecnologias empregadas no desenvolvimento do sistema, e a Seção 5.3 descreve o funcionamento do sistema.

5.1. Descrição da arquitetura do sistema

Trabalhos anteriores descrevem uma ontologia inicial construída para a modelagem de comportamentos de *malware* (Grégio e outros, 2014), bem como um conjunto de ferramentas construídas para utilizar esta ontologia na análise de *malware* e ameaças em geral. Em Grégio e outros (2016) são detalhadas estas ferramentas, bem como uma avaliação da representatividade da ontologia por meio de um experimento que envolveu 2000 amostras de execuções de *malware* e 385 execuções únicas de software benigno.

Com o desenvolvimento do CrOMa, usuários e especialistas no domínio passarão a interagir de forma colaborativa para evoluir a ontologia proposta por meio de contribuições diretas e indiretas para a ontologia. Este sistema pode atuar em conjunto com ferramentas previamente construídas, ou seja, os participantes do *crowdsourcing* terão acesso às demais ferramentas construídas por meio do CrOMa. As ferramentas de apoio estão incorporadas no sistema para *download* para usuários autorizados conforme descrito no modelo apresentado no Capítulo anterior.

A seguir é apresentado uma breve descrição sobre o conjunto de ferramentas disponibilizadas para auxiliar os usuários que são compartilhadas no sistema CrOMa:

- **Ferramenta de Extração de Comportamento (*BET–Behavior Extraction Tool*).** Esta ferramenta centra-se na geração de estatísticas relacionadas com o número de ocorrências de um determinado comportamento modelado em amostras de *malware*. O usuário pode configurar o comportamento a ser analisado em múltiplos níveis hierárquicos, de acordo com a estrutura da ontologia. A *BET* lê um conjunto de execução de logs que já foram instanciados na ontologia pelo *LLT* (*Log Loader Tool*), o conjunto de regras *SWRL* e os axiomas de classe que determina possíveis comportamentos, e executa inferências utilizando a *API pellet* para determinar a ocorrência de cada comportamento no conjunto de amostras;
- **Ferramenta de Análise de Risco (*RAT – Risk Analysis Tool*).** Esta ferramenta concentra-se na análise dos logs para definir e apresentar estatísticas sobre execuções em função do comportamento observado, bem como os níveis de risco de forma individual para cada execução. A *RAT* lê

um conjunto de execuções já instanciadas na ontologia pelo *LLT*, o conjunto de Regras *SWRL* e axiomas de classe; incluindo aquelas que determinam os níveis de risco dos comportamentos apresentados por meio de inferências usando a *API pellet*. Ao utilizar a *RAT*, é possível visualizar estatísticas sobre os níveis de risco associados a execuções de *malware*;

- **Especificação de Regras e Comportamento**. Este componente fornece interfaces para especificar regras *SWRL*, novas classes e axiomas da Ontologia. Do ponto de vista conceptual, múltiplas interfaces podem implementar este componente. No momento, as regras são especificadas diretamente na interface da ferramenta *Protégé*. O ambiente de *crowdsourcing* proposto nesta dissertação visa expandir este componente;
- **Ferramenta Carregadora de Logs (*LLT - Log Loader Tool*).** Este componente lê os *logs* coletados, realiza a verificação de consistência, e inclui todos os indivíduos e suas respectivas propriedades para a ontologia. A *LLT* usa a versão *API OWL*(para *OWL*2.0) para manipular a ontologia;
- **Camada de Coleta de Logs de Execução (*Execution Log Collection Layer*)**. Inclui as ferramentas, infraestrutura e os procedimentos implementados para coletar dados sobre *malware* e execuções não maliciosas.
- ***LLT (Log Loader Tool):*** Uma ferramenta essencial na proposta da arquitetura e para outros componentes, que foi desenvolvida no contexto desta dissertação. Ela é capaz de alimentar ontologias com instâncias de comportamento de *malware* que serão utilizadas na avaliação da ontologia dentro da plataforma de *Crowdsourcing*.

A Figura 5.1 mostra a visão geral das funcionalidades do sistema CrOMa, destacando que as principais funcionalidades são as enumeradas do 3 a 10, conforme descritas a seguir:

(1) *Navegador* – o acesso ao sistema é via um navegador *Web*;

(2) *Interface Web Responsiva*– através da qual os usuários têm acesso a todas as funcionalidades do sistema, podendo, portanto, ter acesso em diferentes dispositivos;

(3) *Contribuição "Acesso Livre"* – contribuição por meio da utilização do fórum, esta é a primeira forma de contribuição prevista no modelo detalhado no Capítulo 4;

(4) *Formular/Enviar Resposta*– postar repostas para as perguntas enviadas no fórum e avalia-las com *likes*;

(5) *Gerenciador de Análise* – Especialistas em ontologias analisam as respostas e se necessário criam regras com essas contribuições, e inserem elas na ontologia;

(6) Acesso Ontologias – O acesso às versões de ontologia, ferramentas e documentos relativos;

(7) *Downloads, Uploads e Exclusão* – Funções de atualização de ontologias e ferramentas compartilhadas;

(8) *Contribuição Especialistas* – Meio de acesso para usuários que tem conhecimento em ontologias e regras, básico ou avançado;

(9) *Contribuir* – Onde se contribui com regras e classes da ontologia, em texto descritivo ou *SWRL/OWL*;

(10) *Gerenciar Aprovações* – Onde especialistas de domínio analisam as regras e/ou classes, podendo aprovar ou reprovar inserindo justificativa para tanto;

(11) *Administração do Sistema* – Apenas o administrador do sistema tem acesso diretamente ao servidor e ao banco de dados.

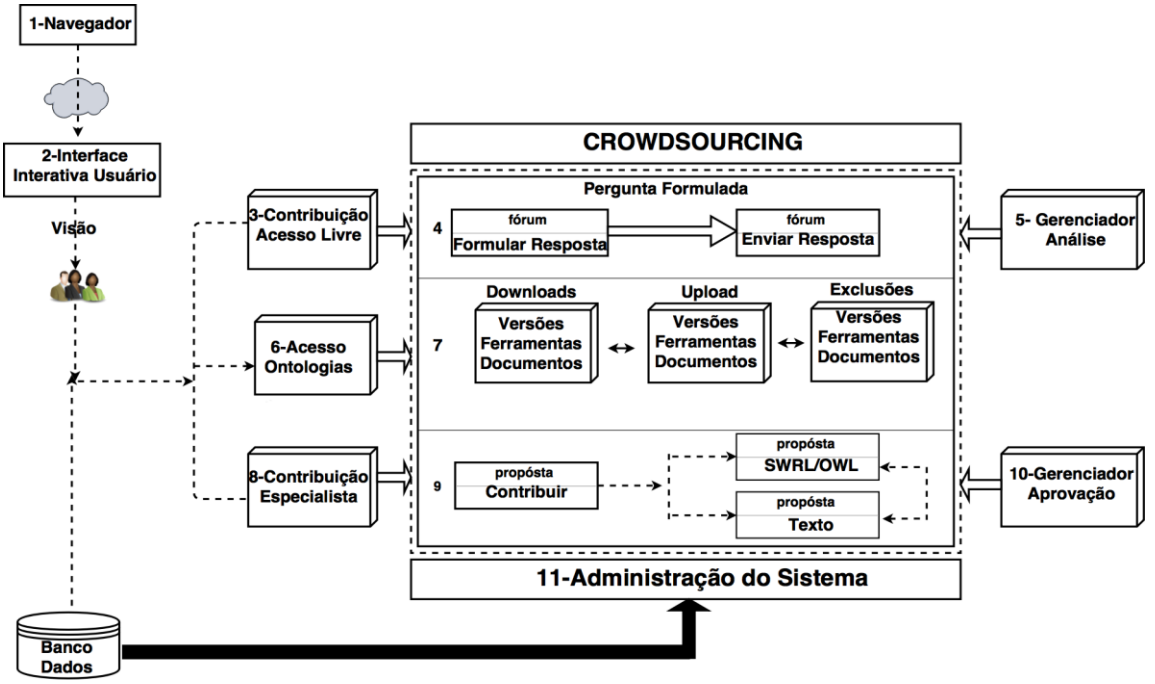


Figura 5.1: Visão Geral da Arquitetura do Sistema CrOMa.

Os módulos são todos hospedados em um servidor *Web*. O desenvolvedor trabalhou diretamente com os especialistas em ontologia para garantir um melhor acesso e compartilhamento das informações pelos usuários.

A Figura 5.2 mostra a divisão dos módulos desde o primeiro acesso passando por segurança até avaliação do sistema, a qual será usada no próximo Capítulo para análise dos dados coletados em um formulário, conforme descritos a seguir:

1- **Módulo de acesso -** o módulo de seguraça da plataforma onde usuários terão que fornecer dados incluindo um número de celular real para receber a liberação, minimizando o problema de usuários falsos que possam tentar fazer uso indevido das informações compartilhadas, ou mesmo colocar regras ou alterar a extrutura maliciosamente. As funcionalidades deste módulo são as seguintes:
   - Usuários acessam a plataforma de *crowdsourcing*;
   - Usuários fazem o cadastro dos dados;
   - Usuários recebem um *SMS* informando para aguardar confirmação e recebimento da senha;

- Moderador(es) recebem *SMS* informando que tem novos usuários para análise no sistema;
- Se aprovado pelo moderador, o usuário recebe uma senha para acesso ao sistema via *SMS*;

2- **Módulo de Contribuições -** Este módulo implementa os diversos tipos de contribuições que são acessíveis ao usuário conforme modelo descrito no Capítulo 4. As funcionalidades implementadas por este módulo são as seguintes:

- Fórum
  - O fórum é a única funcionalidade que o usuário iniciante tem acesso no módulo de contribuições;
  - Engloba inclusão de perguntas e respostas;
  - Realiza cálculo de mudança de níveis, onde o iniciante com 10 *likes* sobe para o nivel médio;
  - Especialistas de domínio podem filtrar e visualizar respostas para trabalha-lhas e transforma-las em regras para a ontologia.
- Regras
  - Usuários do nível médio e superiores têm acesso às contribuições de regras;
  - As regras podem ser inseridas como texto em linguagem natural, em liguagem *SWRL* ou ambas.
- Classes
  - Usuários de nível avançado e superiores podem fazer contribuições de classes;
  - As classes têm que ser fornecidas em linguagem *OWL*, assim, existe a necessidade do usuário de conhecer ontologias e a linguagem *OWL*.

3- **Módulo Dados do Usuário –** Módulo de controle e edição de dados do usuário, suas funções são:

- Todos os usuários do sistema tem acesso a este módulo;
- Neste módulo os usuários podem modificar seus dados de cadastro;
- A troca de senha também é implementada neste módulo.

4- **Módulo Formulário de Avaliação –** Este módulo foi implementado para facilitar a avalição do sistema conforme descrito no próximo Capítulo. Este módulo possui as seguintes características:

- Todos os usuários tem acesso a este módulo para realizar a avalição;
- O módulo contém um formulário de avaliação onde os usuários podem responder a perguntas de múltiplas escolhas e colocar críticas e sugestões em texto livre;
- O administrador pode ainda vizualizar os formulários de todos os usuários do sistema.

5- **Módulo de Downloads –** Módulo onde os usuários podem fazer *download* de diversas versões da ontologia. Os moderadores e especialistas podem disponibilizar outras versões, desfazer modifições e excluir versões da ontologia. Este módulo também possibilita o *download* e *upload* de ferramentas e documentos. As funcionalidades implementadas estão descritas abaixo:

- Todos os usuários tem acesso a este módulo, mas com diferentes níveis de permissões;
- Usuário iniciante pode fazer *download* das versões da ontologia e documentos;
- Usuário médio pode fazer *download* das versões da ontologia, ferramentas e documentos;
- Usuário avançado pode fazer *download* das versões da ontologia, ferramentas e documentos, e *upload* de documentos e ferramentas;
- Usuário *expert* pode fazer *download* das versões da ontologia, ferramentas e documentos, e *upload* das versões da ontologia, documentos e ferramentas;
- Usuário moderador pode fazer *download* das versões da ontologia, ferramentas e documentos, *upload* das versões da ontologia, documentos e ferramentas, e exclusão das versões da ontologia, documentos e ferramentas.

6 – Módulo de Armazenamento do Servidor – Módulo para uso exclusivo do administrador, os acessos dos usuarios são apenas via o móduo de segurança (não sendo permitido o uso deste módulo):

- Meio de acesso direto, pelo administrador, aos documentos, versões da ontologia e ferramentas armazenadas, além do banco de dados do sistema.

7 – Módulo de Análise a Aprovações – Módulo com funcionalidades para aprovação das contribuições de regras e classes, bem como funcionalidade para selecionar contribuições do fórum para tranformar em regras. As funcionalidades são:

- Apenas moderador(es) e *expert*(s) tem acesso a estemódulo;
- Possui funcionalidades para aprovar regras e classes, bem como para construir regras por meio das postagens do fórum;
- Ao aprovar (ou reprovar) uma regra, o sistema envia um *SMS* ao usuário que a propôs;
- Após aprovar uma regra o especialista a insere na ontologia e coloca uma nova versão dela no controle de versões.

8 – Módulo Segurança do Sistema – Este é o módulo reponsável pela segurança interna do sistema. É onde são controladas as permissões dos usuários, feitas as alterações de privilégio de grupos, exclusão de usuários, elevação ou rebaixamento do nivel de usários, liberação de usuários, e consulta os *logs* de acesso e ação do sistema, as principais funcionalidades são:

- Apenas os usuários moderador e *expert* tem acesso a este módulo;
- Moderadores podem cadastrar um novo grupo de usuários (novo nível de acesso para os usuários), além de editar e modificar autorização para estes grupos;
- Os moderadores realizam a análise dos cadastros dos usuários, podendo liberar, modificar o nível, bloquear ou excluir um usúario.
- Os moderadores ainda têm acesso ao registro de atividades.

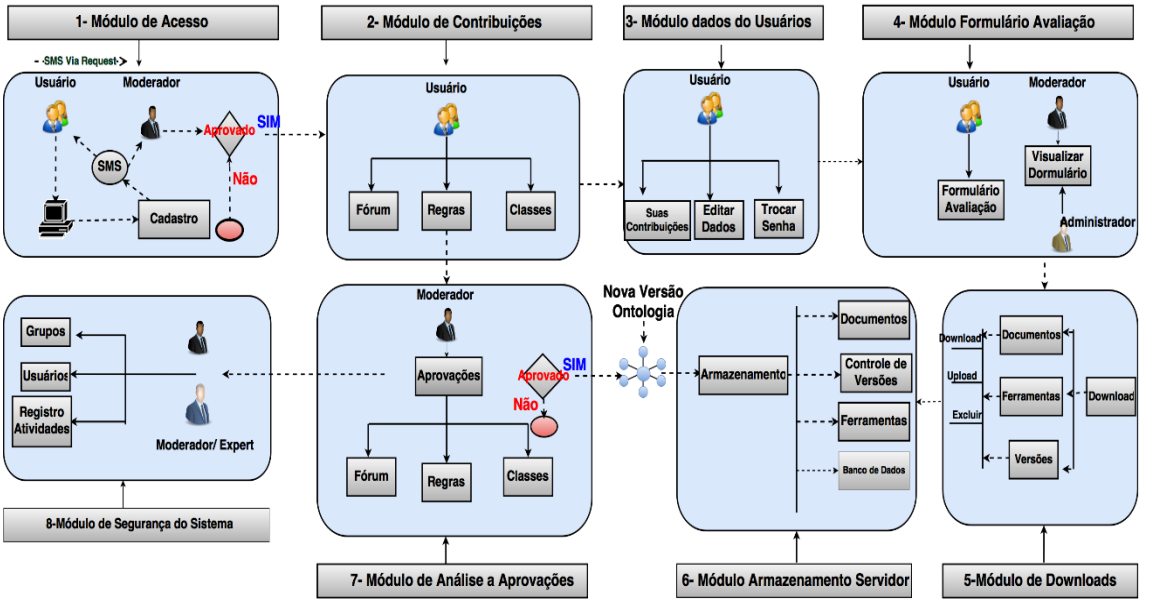


Figura 5.2: Digrama com os módulos do sistema.

Embora existam muitas outras formas para aprimorar a segurança de acesso. Esta dissertação provê um nível aceitável de segurança para não termos problemas com usuarios indesejados, mas está fora do escopo fazer pesquisa em "segurança em *crowdsourcing*".

5.2 Tecnologias Utilizadas

Durante o desenvolvimento do CrOMa foram utilizadas as seguintes tecnologias: *HTML*, *CSS*, *Java Script*, *Bootstrap*, *PHP*, *Ajax*, *o framework* Zender Optmizer, o *design pattern MVC (Model-View-Controller)* e o banco de dados *MYSQL*. O *PHP* foi a principal linguagem de programação utilizada, ela foi empregada para a implementação das páginas dinâmicas em conjunto com *HTML 5* (para especificação da estrutura estática das páginas), *CSS 3* (para definição dos estilos), *JavaScript* (para execução de *scripts* no servidor) e o framework *bootstrap* (Platas, 2013) para especificação de páginas responsivas. O *Zend Optimizer* foi utilizado para otimizar os códigos *PHP*. No

desenvolvimento também foi utilizado o padrão de projeto *MVC*, que é necessário para facilitar a manutenção e melhor organização da aplicação, separando o modelo (dados), visão (interface) e controle (lógica de execução) da aplicação.

Para acesso ao sistema é necessário apenas um navegador e Internet. O código do servidor foi desenvolvido totalmente em *PHP*, na parte de visão dos usuários, as páginas foram desenvolvidas utilizado a tecnologia *bootstrap*, que contém os códigos *HTML5* e *CSS* 3 padronizados, deixando assim, a aparência com melhor visualização em diversos dispositivos. Já o *Zend Optimize*r reconhece os códigos *PHP* e carrega otimizando as aplicações, compilando o *PHP* em tempo de execução e assim melhorando o desempenho dos sistemas.

5.3 Descrição das Interfaces do Sistema

Durante a concepção do sistema foi decidido pela construção de um *site*, que serve de ajuda e aprendizado na utilização do sistema CrOMa para os usuários. Este site exemplifica o uso das ferramentas, explica as tecnologias utilizadas, os níveis de acesso, além de conter uma base com vídeos explicativos sobre a utilização do sistema.

A Figura 5.3 mostra a interface do site www.projectweb-solutions.myscriptcase.com/cm com o menu com as opções supracitadas, que são descritas a seguir:

- Início – Uma breve explicação do sistema, bem como as ferramentas disponíveis e tecnologias utilizadas;
- Desenvolvedores – Os desenvolvedores do sistema, nome uma parte do *curriculum lattes*, com linha de pesquisa, endereço acadêmico além do *link* para redirecionamento para o *Lattes* da plataforma *CNPq*;
- Descrição do Sistema – A descrição das funcionalidades do sistema e imagens da interface do sistema;
- Níveis de Acesso – Explicação sobre os níveis de acesso que o sistema contém, e formas de chegarem a estes níveis;
- Artigo – Artigo publicado no *journal Applied Ontology* contendo o *abstract* e o *link* para acesso ao artigo;

- Tutoriais de Uso – Vídeos mostrando a utilização do sistema CrOMa, auxiliando em um melhor entendimento do usuário;
- Faccamp – Acesso direto ao site da Faculdade de Campo Limpo Paulista – FACCAMP-;
- Contato – Contato direto com o desenvolvedor através de *email*;
- Acesso ao Sistema CrOMA onde é redirecionado diretamente à página de *login*.

O usuário pode acompanhar o desenvolvedor diretamente em redes sociais no *facebook* e *twitter*.

Figura 5.3: Site de Apresentação do Sistema CrOMa.

Vale a pena ressaltar na Figura 5.4 os tutoriais de uso que contêm vídeos explicativos sobre o uso do sistema CrOMa, onde usuários aprendem as funcionalidades do sistema. O objetivo é deixar de forma clara as funcionalidades e os tipos de acesso do sistema.

Figura 5.4: Tutoriais de uso no site do Sistema CrOMa.

Um novo usuário deve iniciar o acesso ao sistema por meio de um cadastro que deve ser aprovado pelo moderador. A Figura 5.5 apresenta o formulário de cadastro no sistema, neste formulário são obrigatórios os seguintes campos: nome, *login*, data de nascimento, *RG*, endereço, número, bairro, e um número de celular que é utilizado como mecanismo de validação do usuário. No caso de um número de celular inválido, *CPF* ou

e-mail incorreto o sistema não permite o cadastro. Também é efetuada a consistência sobre *login* (nome de usuário) repetidos conforme destacado em vermelho na Figura 5.5.

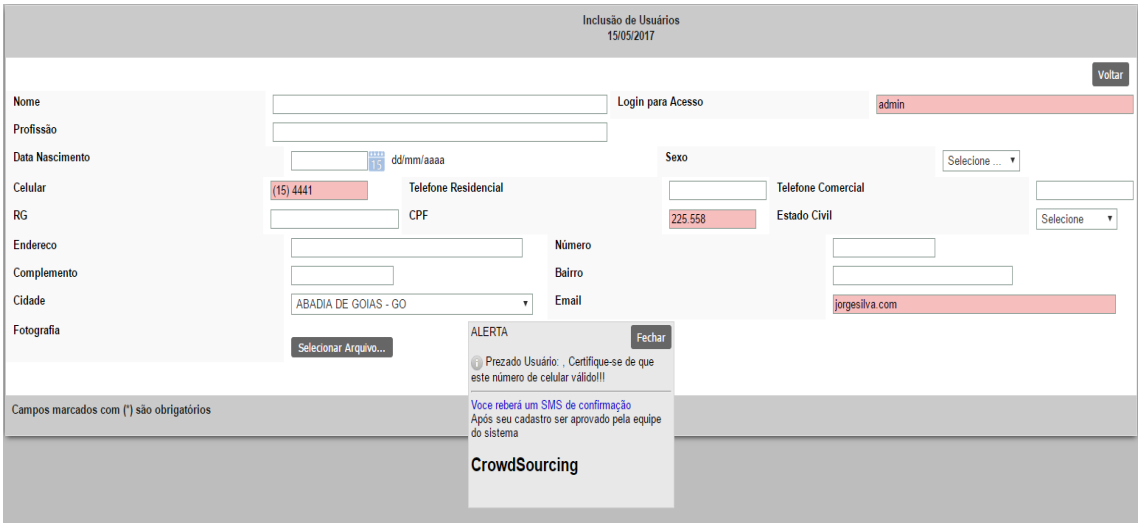


Figura 5.5: Página de cadastros de usuários.

A Figura 5.6 mostra o ambiente onde os especialistas podem fazer a liberação dos usuários do sistema, a partir de clicar em salvar o usuário que tinha feito o cadastro recebe uma senha no celular para acessar o sistema.

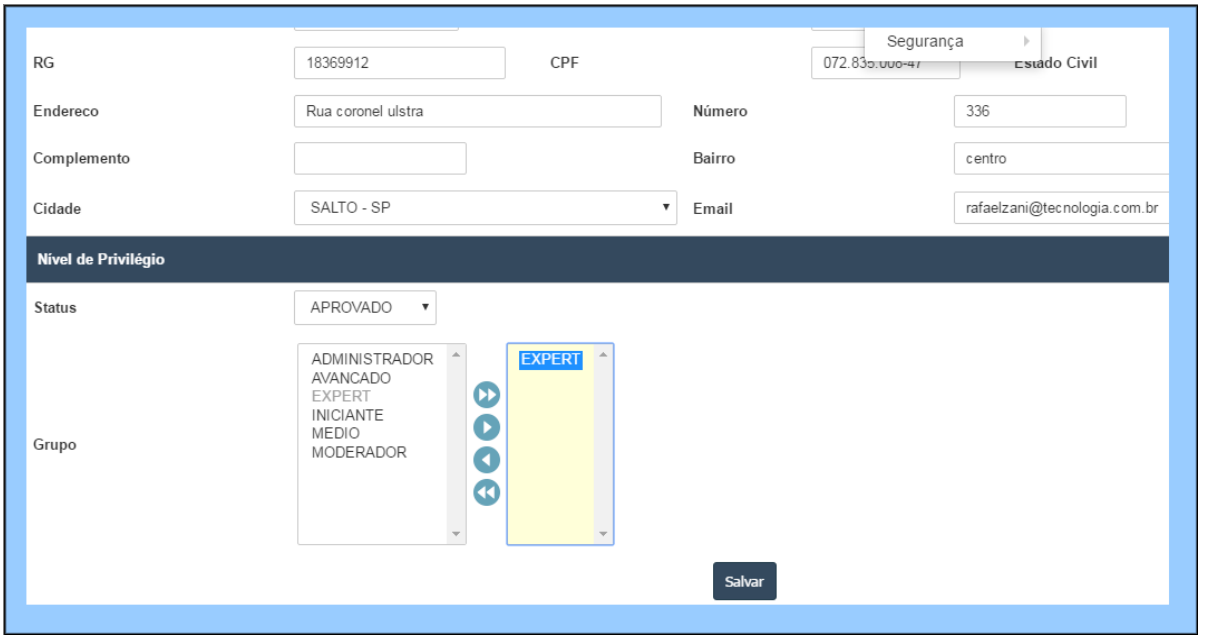


Figura 5.6 Liberação de usuários.

A Figura 5.7 apresenta uma contrassenha enviada para usuário via *SMS* após a análise do cadastro e liberação do moderador. Desta forma, o usuário é obrigado utilizar um celular válido, minimizando assim alguns problemas de acessos indesejados no sistema.

Figura 5.7: Recebimentos de contrassenha via *SMS*.

A seguir são apresentadas algumas imagens com as principais funcionalidades do sistema CrOMa. A Figura 5.8 apresenta a interface de *login* no ambiente de acesso ao sistema, contento a opção de "Novo Usuário" utilizada para submeter em seus cadastros no sistema.

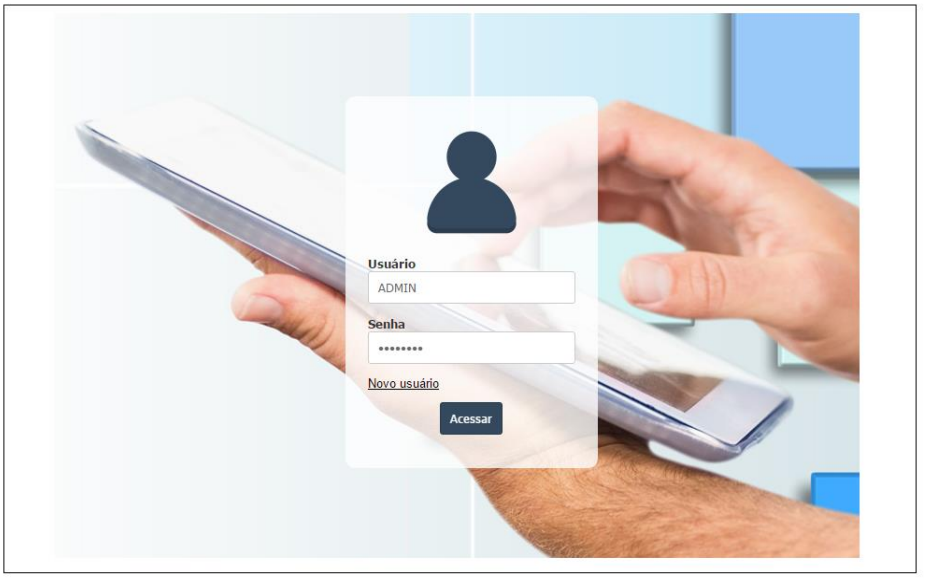


Figura 5.8: Página de Login.

A Figura 5.9 apresenta o menu principal do sistema CrOMa, onde estão disponíveis a seguintes opções: “Fórum” de discussões, que implementa um ambiente para colocar dúvidas e respostas; em “Propostas” encontra-se todas as opções relacionadas as contribuições de regras e classes e as suas respectivas aprovações; em “*Downloads*” os usuarios podem baixar, fazer *upload* e excluir versões da ontologia, ferramentas e documentos, de acordo com os respectivos privilégios; em “Administração do Sistema” estão as opções de segurança incluindo o acesso aos grupos do sistema, usuários, liberação de *SMS* e registro de atividades do sistema; em “Avaliação do Sistema” estão os formulários de avaliações com as perguntas relativas ao sistema, incluido críticas e sujestões; em “Meu perfil” os usuários podem visualizar suas postagens no fórum, editar seus dados e trocar sua senha de acesso; e por fim, a opção de “Sair”.

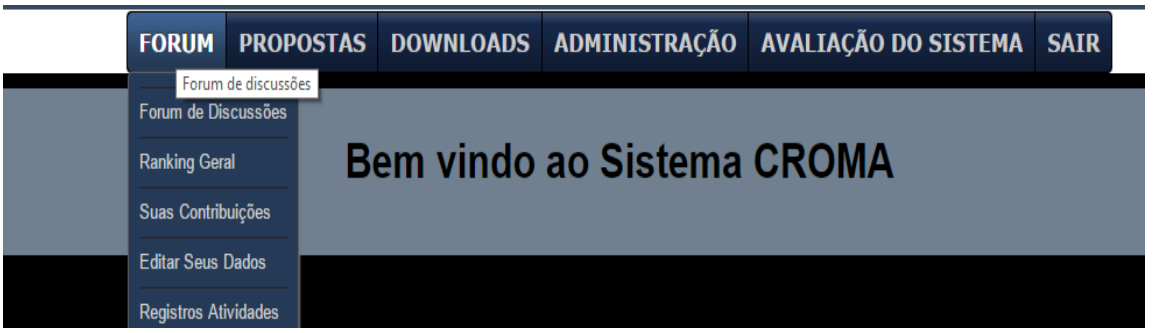


Figura 5.9: Menu principal do sistema CrOMa.

A Figura 5.10 apresenta o fórum de discussão do sistema CrOMa, onde encontram-se as perguntas postadas. Ao clicar em uma pergunta são abertas todas as respostas. No ambiente de respostas existe a opção de avaliá-las como positiva (*like*) ou negativa. Esta opção é utilizada para computar a pontuação para o usuário ir do nível iniciante para o médio.

Figura 5.10: Página Inicial do Fórum de discussões.

A Figura 5.11 apresenta a interface com controle de versões, onde usuários iniciantes, médios e avançados podem fazer *download* das versões da ontologia. Nela, usuários especialistas e moderadores também podem criar novas versões da ontologia e fazer *download* inclusive de documentos e ferramentas.

Figura 5.11: Download de versões.

A Figura 5.12 apresenta a interface do *upload* de documentos, o usuário que tiver permissão poderá utilizar tanto para documentos, versões e ferramentas.

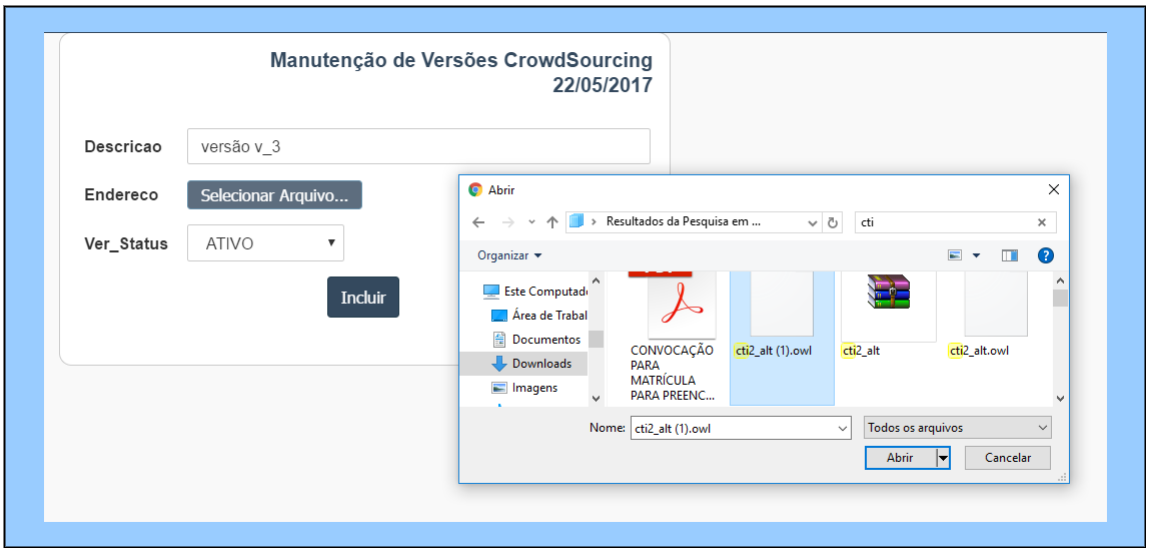


Figura 5.12: Upload de documentos.

A Figura 5.13 apresenta a interface onde moderadores podem excluir versões da ontologia. Em caso de alguma anomalia na versão ou uma regra inserida de forma errada os moderadores podem excluir uma versão ou restaurar a ontologia. Podem também fazer a exclusão de documentos e ferramentas.

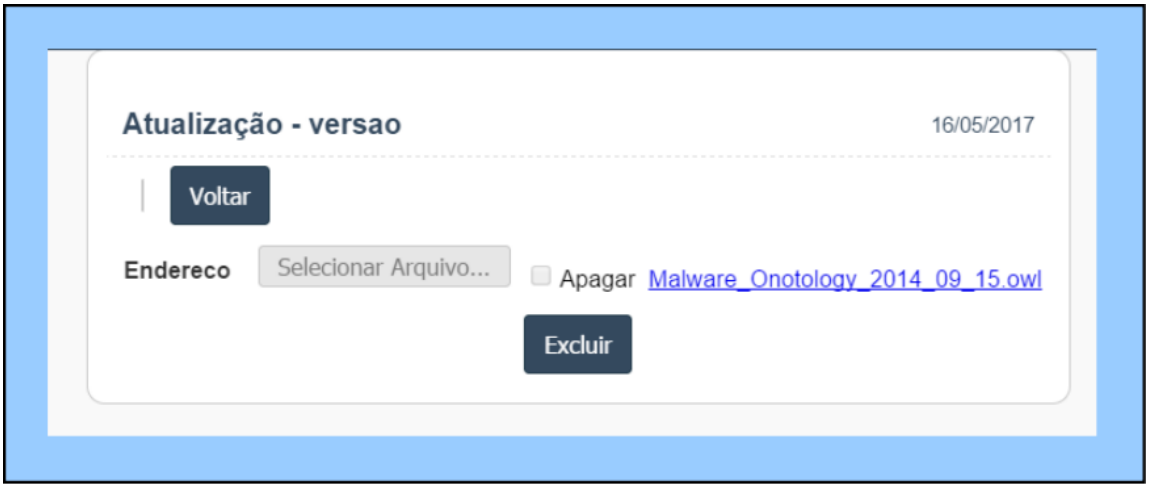


Figura 5.13: Interface de exclusão de versões.

A Figura 5.14 mostra a visualização do usuário das suas próprias contribuições de regras, podendo editá-las, por exemplo, se um moderador pedir uma justificativa mais detalhada ou for detectado algum erro no código *SWRL*.

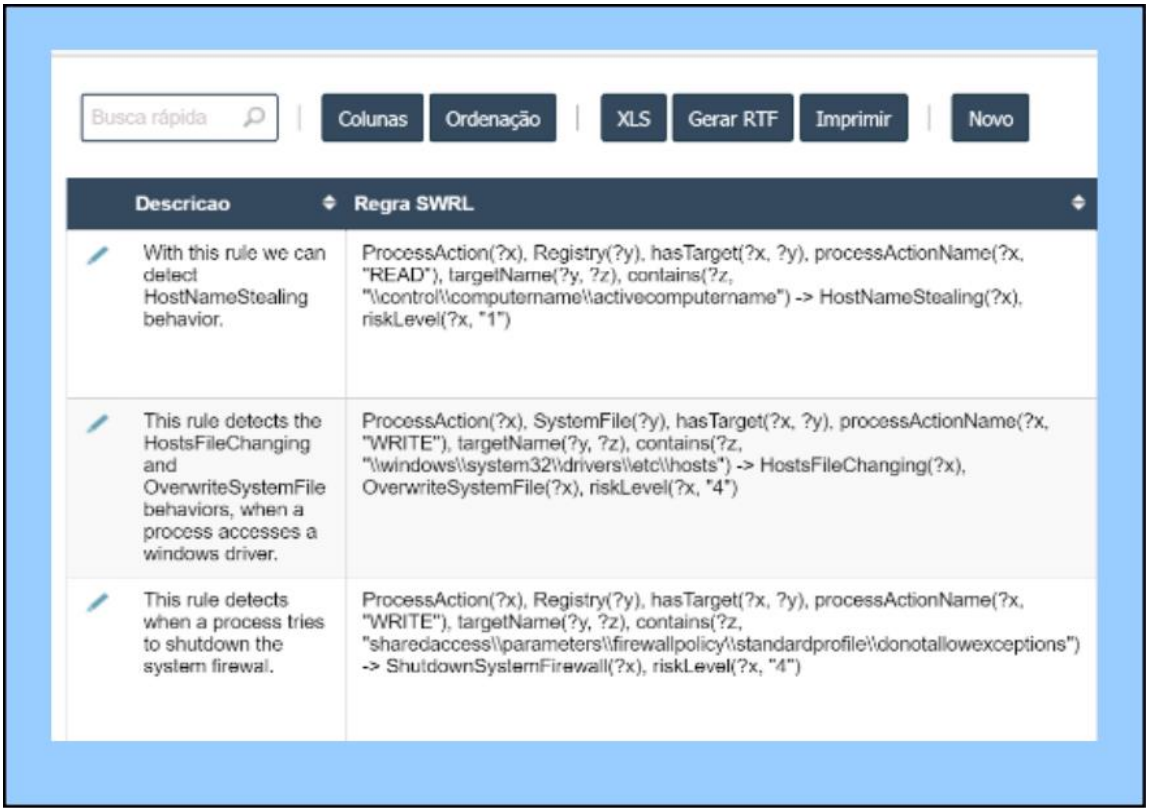


Figura 5.14: Visualizações das contribuições efetuadas pelo usuário.

A Figura 5.15 apresenta interface para contribuição de regras onde o usuário posta uma contribuição podendo ser descrita em linguagem natural ou/e em *SWRL*, havendo ainda um campo para justificar a inclusão.

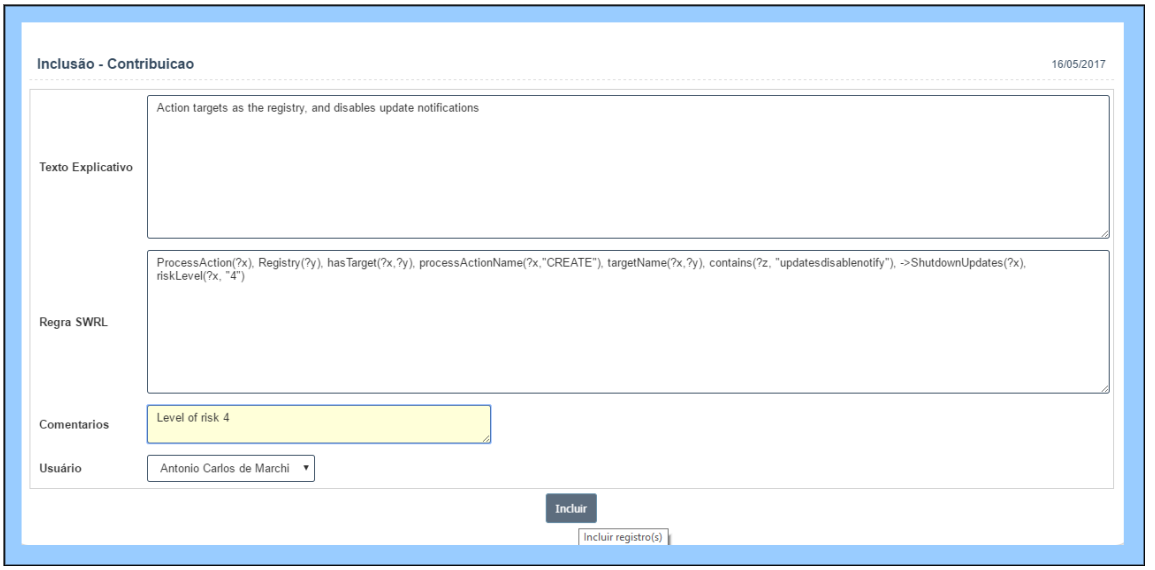


Figura 5.15: Contribuições de regras.

A Figura 5.16 apresenta a interface de exclusão da contribuição que podem ser feitas pelo próprio usuário que fez a contribuição ou pelos usuários avançados.

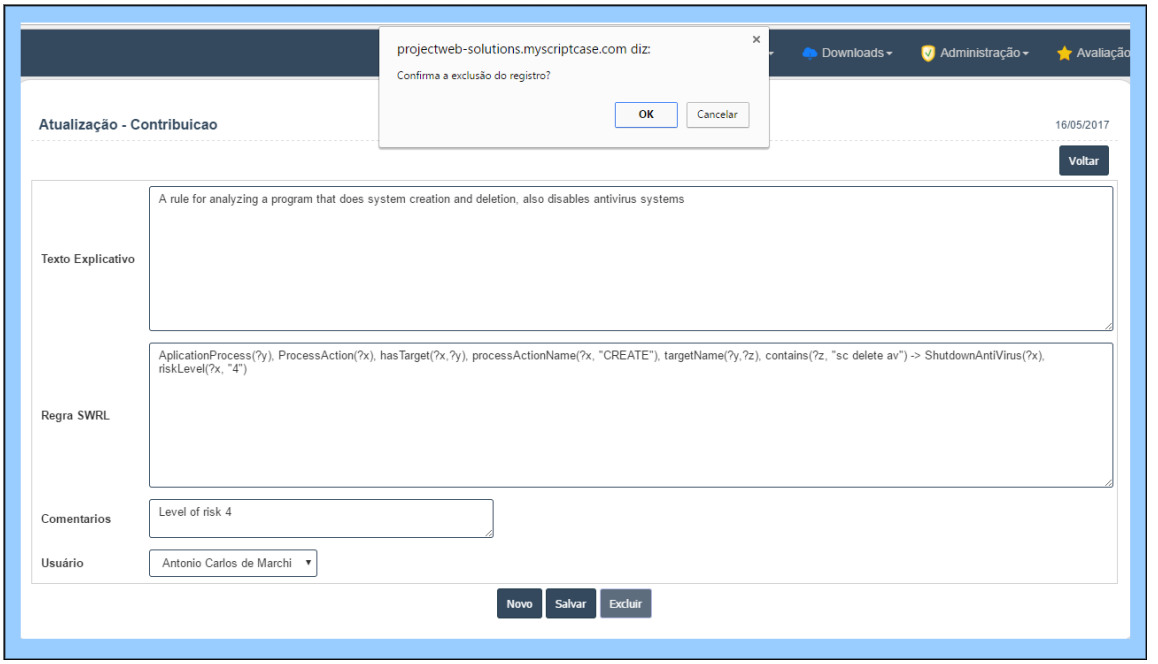


Figura 5.16: Exclusão de regras.

A Figura 5.17 mostra a interface de aprovações de regras onde o usuário avançado avalia a regra no lado esquerdo tem o botão aprovar.

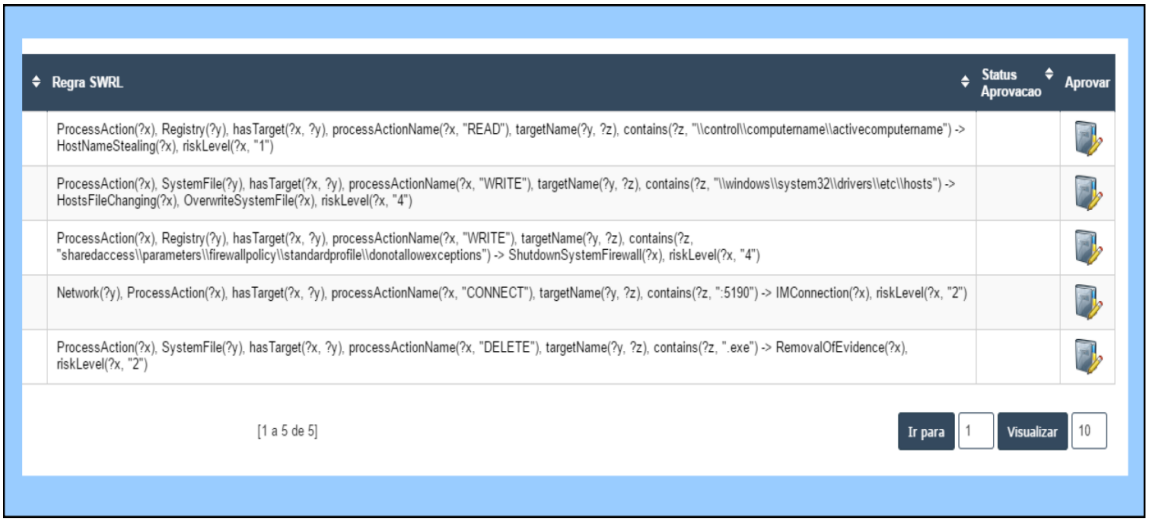


Figura 5.17: Interface para avaliar regras.

A Figura 5.18 mostra a interface para as aprovações onde o usuário pode aprovar ou reprovar as regras, quando o usuário inclui a aprovação o usuário que enviou a regra recebe um *SMS* avisando sobre o status da sua contribuição.

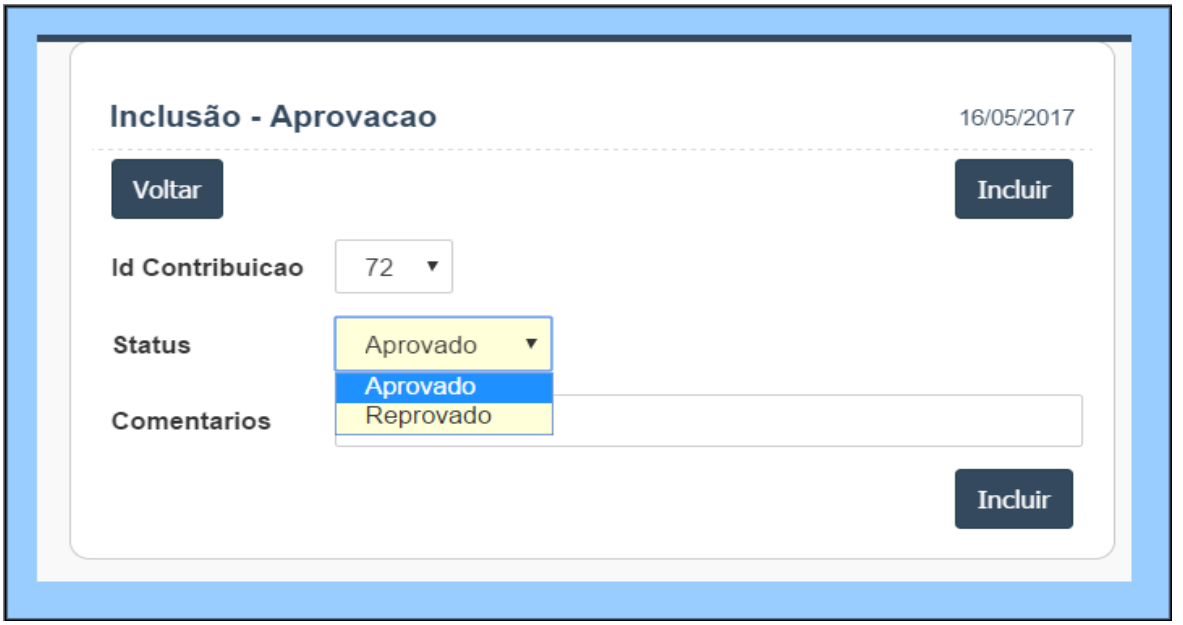


Figura 5.18: Visualização do ambiente de aprovações.

A Figura 5.19 apresenta interface para contribuição de classes onde usuários descrevem a proposta de novas classes que representam comportamentos de *malware*, junto com o campo de comentários que julguem apropriado.

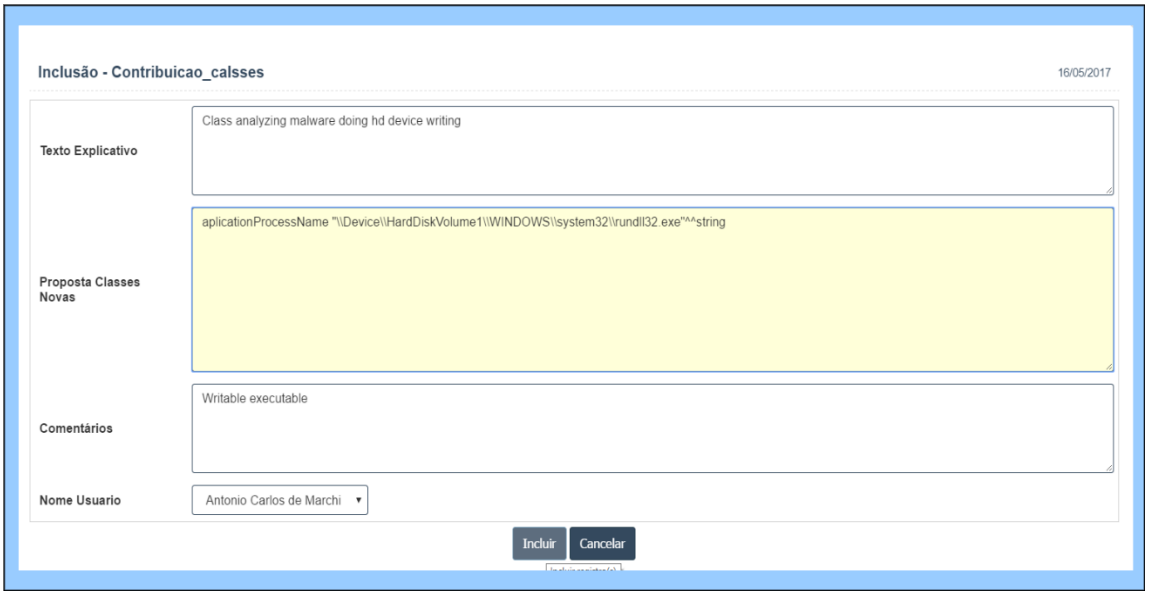
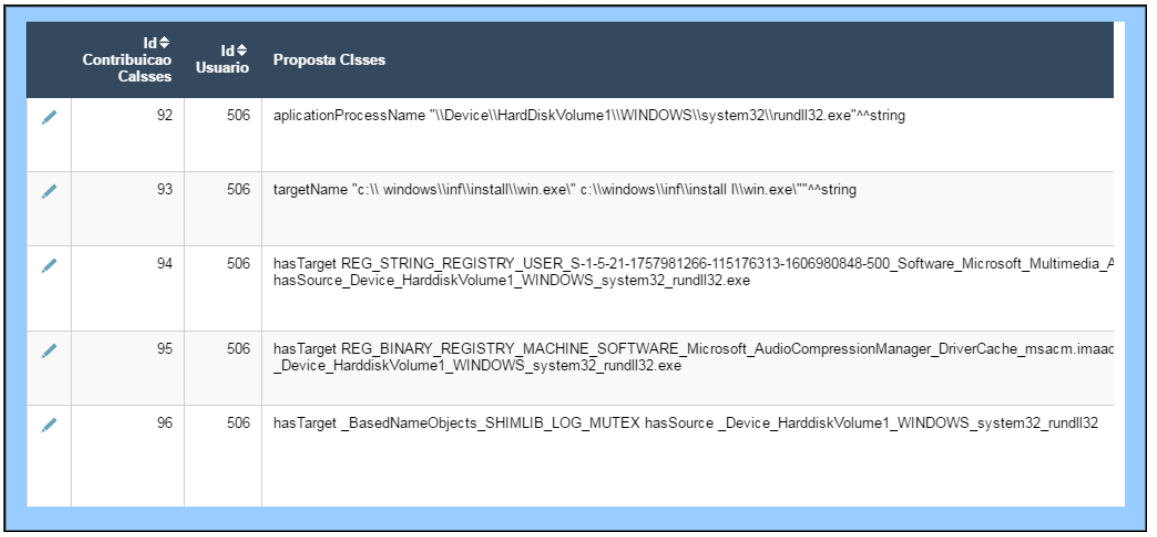


Figura 5.19: Contribuições de classes.

A Figura 5.20 mostra a interface das classes enviadas pelo usuário onde ele pode edita-las e visualizá-las.



	Id Contribuicao Calsses	Id Usuario	Proposta Clsses
	92	506	aplicationProcessName "\\Device\\HardDiskVolume1\\WINDOWS\\system32\\rundll32.exe"^^string
	93	506	targetName "c:\\ windows\\inf\\install\\win.exe\" c:\\windows\\inf\\install l\\win.exe\""^^string
	94	506	hasTarget REG_STRING_REGISTRY_USER_S-1-5-21-1757981266-115176313-1606980848-500_Software_Microsoft_Multimedia_A hasSource_Device_HarddiskVolume1_WINDOWS_system32_rundll32.exe
	95	506	hasTarget REG_BINARY_REGISTRY_MACHINE_SOFTWARE_Microsoft_AudioCompressionManager_DriverCache_msacm.imaac _Device_HarddiskVolume1_WINDOWS_system32_rundll32.exe
	96	506	hasTarget _BasedNameObjects_SHIMLIB_LOG_MUTEX hasSource _Device_HarddiskVolume1_WINDOWS_system32_rundll32

Figura 5.20: Visualização de classes.

A Figura 5.21 mostra a interface onde o usuário pode editar ou excluir suas contribuições de classes.

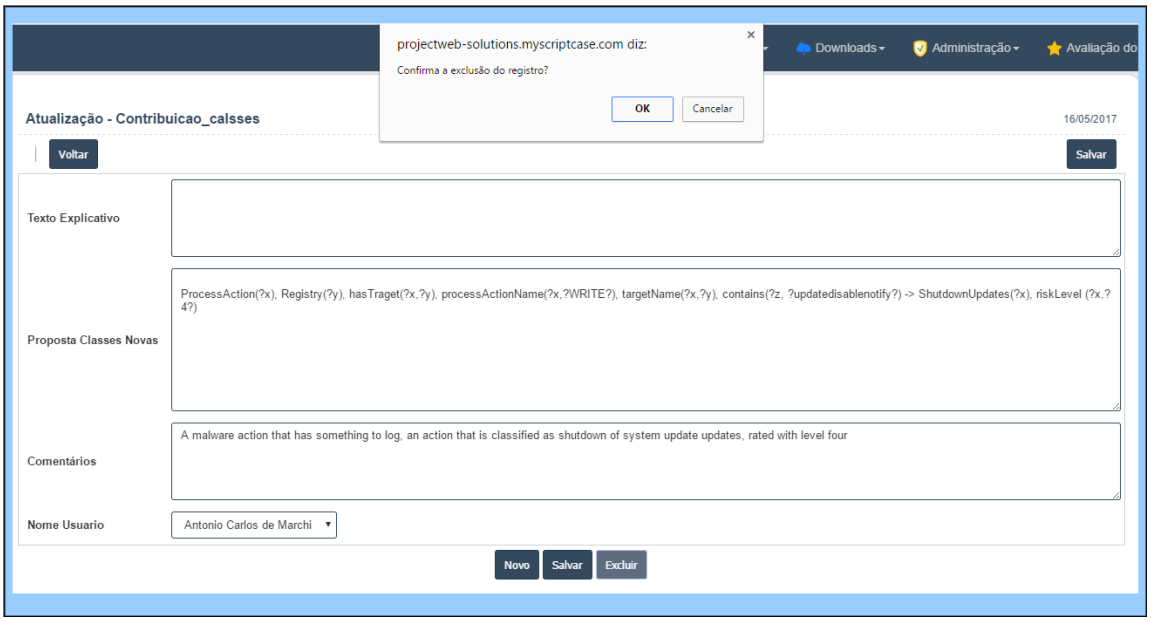


Figura 5.21: Interface edição das contribuições de classes.

A Figura 5.22 mostra a visualização dos usuários *expert*(s)/ moderadores(s) nas contribuições de classes dos usuários para aprovações.

Id Contribuicao Calsses	Id Usuario	Comentarios	Aprovar
92	Antonio Carlos de Marchi	Writable executable	
93	Antonio Carlos de Marchi	Writing and executable installation	
94	Antonio Carlos de Marchi	To run a multimedia file in system format	
95	Antonio Carlos de Marchi	Access to the record by reading	
96	Antonio Carlos de Marchi	Throws a mutex semaphore through a string	
97	Antonio Carlos de Marchi	Disables terminal control with TimeStanp by reading	
98	Antonio Carlos de Marchi	Uses TimeStamp to capture by reading with the Flash plugin	

Figura 5.22: Visualização de classes para aprovação.

A Figura 5.23 mostra a interface de aprovações ou reprovações de classes, onde o (s) moderador (es) e *expert* (es) utilizam após fazerem a análise das propostas enviadas pelos usuários e fazem as aprovações ou reprovações, e consequentemente o utilizador do sistema que fez a contribuição recebe um *SMS* com o status.

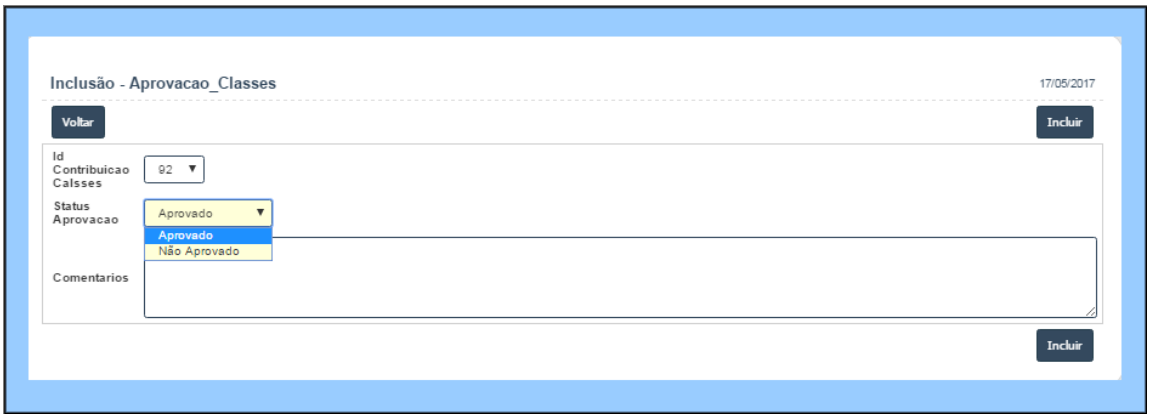


Figura 5.23: Visualização das aprovações de classes.

.

Neste Capítulo, foi apresentada a arquitetura, o desenvolvimento e funcionamento do sistema CrOMa. No próximo Capítulo são apresentadas a avaliação e discussão dos resultados obtidos com a participação de usuários especialistas, que testaram o sistema, suas funcionalidades e principais interfaces.

6. Avaliação e Discussão

Este Capítulo apresenta a avaliação do protótipo com os usuários, para tanto, especialistas nas áreas de ontologia e segurança utilizaram o sistema e emitiram suas opiniões por meio de um questionário que está incluindo dentro do sistema CrOMa. A exploração de vários cenários por usuários com diferentes níveis de conhecimento, formação e atuação na área de segurança da informação ou ontologia, propiciou uma avaliação preliminar do sistema e possibilitou a discussão dos resultados desta pesquisa.

A Seção 6.1 apresenta o formato do formulário e o método de avaliação utilizado, a Seção 6.2 apresenta os resultados e discute possíveis implicações e a Seção 6.3 discute os resultados da avaliação.

6.1. Método de Avaliação

O protótipo do sistema foi desenvolvido com a intenção de facilitar a interação entre usuários da plataforma, principalmente os que atuam na área de Segurança da Informação, conhecedores de ontologia e quem se dedica à pesquisa sobre comportamento de *malware*. Adicionalmente, o desenvolvedor do trabalho procurou tornar o sistema de fácil uso, abrangendo assim, o público que não se dedica a nenhuma dessas áreas, ou seja, a qualquer pessoa interessada no assunto. A avaliação contou com professores, alunos de pós-graduação e profissionais da área de *TI*, totalizando 8 especialistas.

Para possibilitar uma avaliação mais ampla por diferentes tipos de usuários, os avaliadores foram divididos em níveis diferentes de acessos, focando nos níveis com acesso a mais funcionalidades do sistema. Sendo, 1 moderador, 3 *experts*, 2 avançados e 2 iniciantes de acordo com a experiência e nível de entendimento da área.

O formulário possui: 8 perguntas sobre o perfil profissional do avaliador, incluindo dados pessoais, formação acadêmica (itens 1.1 a 1.4 da Figura 6.1) e atuação profissional (itens 2.1 a 2.4 da Figura 6.1); 5 perguntas sobre funcionalidade do sistema (itens 3.1 a 3.5 da Figura 6.2) com a escala *likert* (Likert, 1932), (concordo totalmente, concordo parcialmente, neutro, discordo parcialmente e discordo totalmente); 5 perguntas sobre a usabilidade do sistema (itens 4.1 a 4.5 da Figura 6.3) com a escala em 5 níveis (ótimo, bom, neutro, ruim e péssimo); e, 4 perguntas discursivas (Figura 6.16 a Figura 6.19), sendo uma sobre sugestões de funcionalidade, outra sobre usabilidade e outras duas sobre aspectos gerais.

Likert (1932), propôs uma avaliação utilizando uma escala com várias alternativas, facilitando a respostas para usuários com pouco conhecimento no tema principalmente na utilização do neutro.

Primeiramente foi dada uma breve explicação sobre o sistema para os avaliadores, então eles receberam uma senha previamente cadastrada. Então foi requisitado que explorassem o sistema e, ao final, respondessem o questionário. O procedimento foi realizado em cerca de uma hora em média. Os avaliadores não eram obrigados a responder todas as perguntas, por isto, algumas perguntas foram respondidas por 8 pessoas, enquanto outras por 7 ou 6.

O formulário foi construído com base em uma análise crítica entre o desenvolvedor, orientador e especialistas em segurança da informação. É necessário salientar que como um método de análise por meio de seleções de perguntas, não é possível afirmar com assertividade muitos fatores subjetivos, tais como, o entendimento correto das perguntas por parte dos usuários e o entendimento das tecnologias utilizadas.

As Figuras 6.1, 6.2 e 6.3 apresentam o formulário de avaliação, onde os avaliadores deixaram suas opiniões sobre a funcionalidade e usabilidade do sistema, além de questões discursivas com críticas e sugestões.

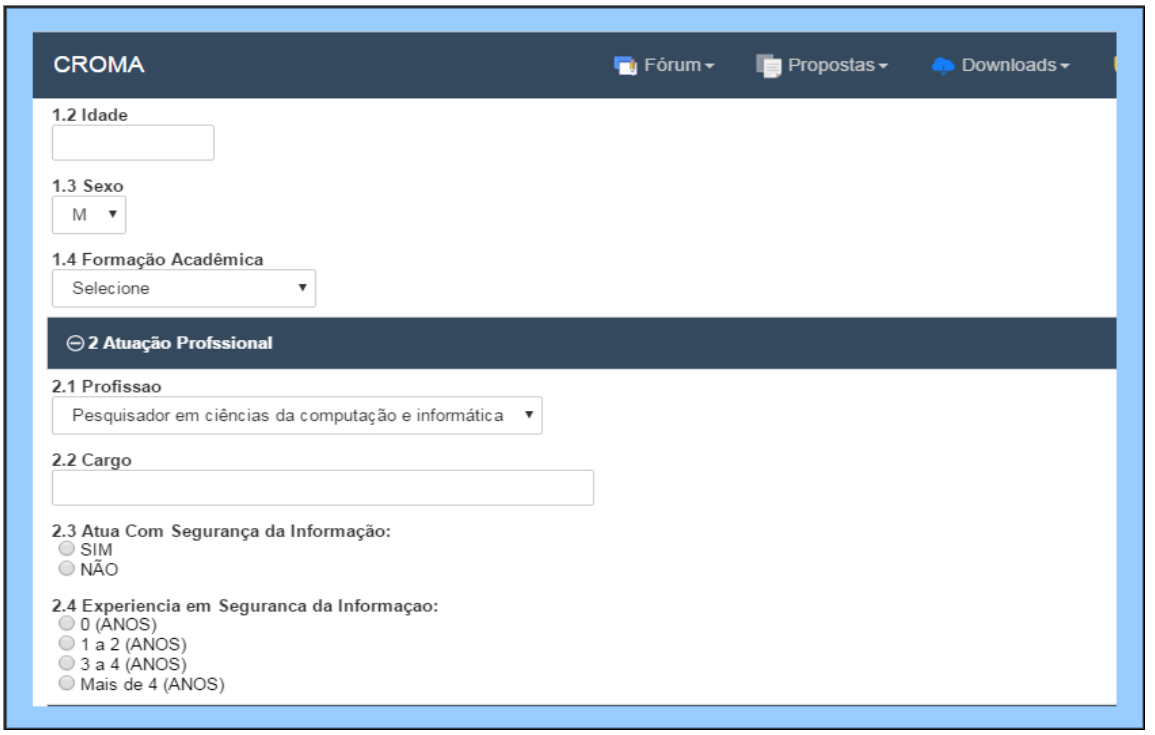
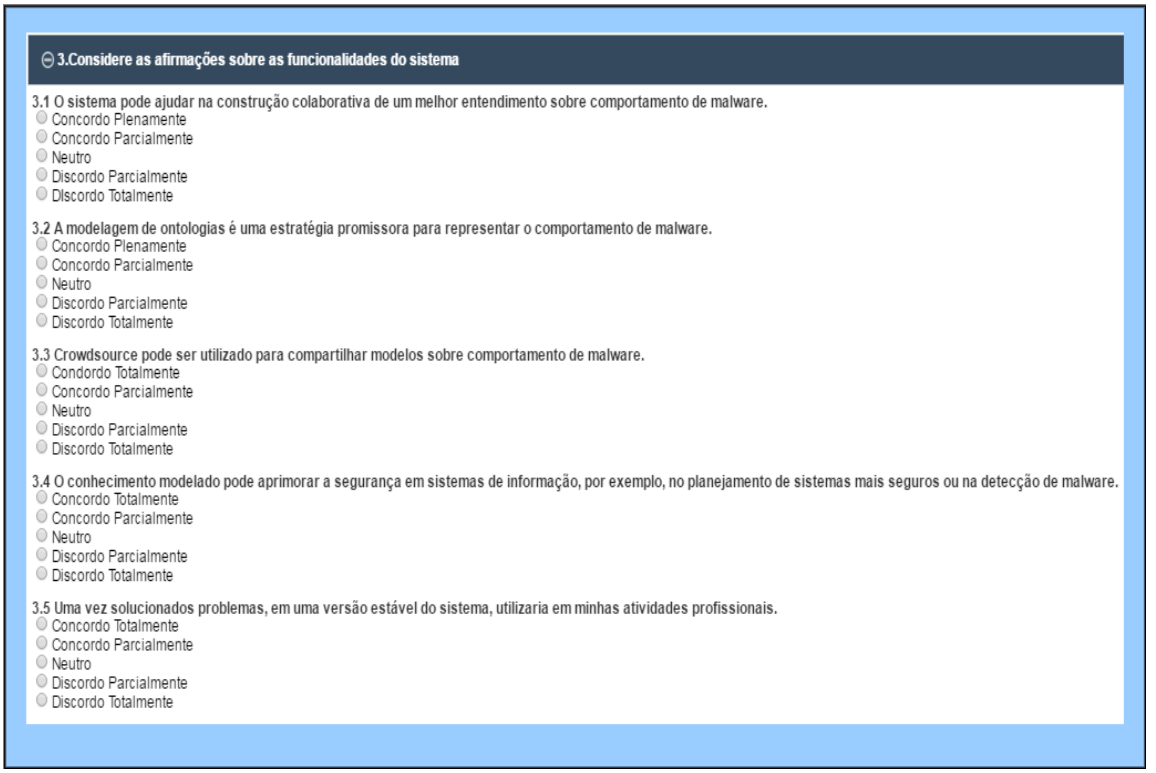


Figura 6.1: Primeira parte do Formulário de Avaliação (Perfil Profissional).



3.Considere as afirmações sobre as funcionalidades do sistema

3.1 O sistema pode ajudar na construção colaborativa de um melhor entendimento sobre comportamento de malware.
- Concordo Plenamente
- Concordo Parcialmente
- Neutro
- Discordo Parcialmente
- DIscordo Totalmente

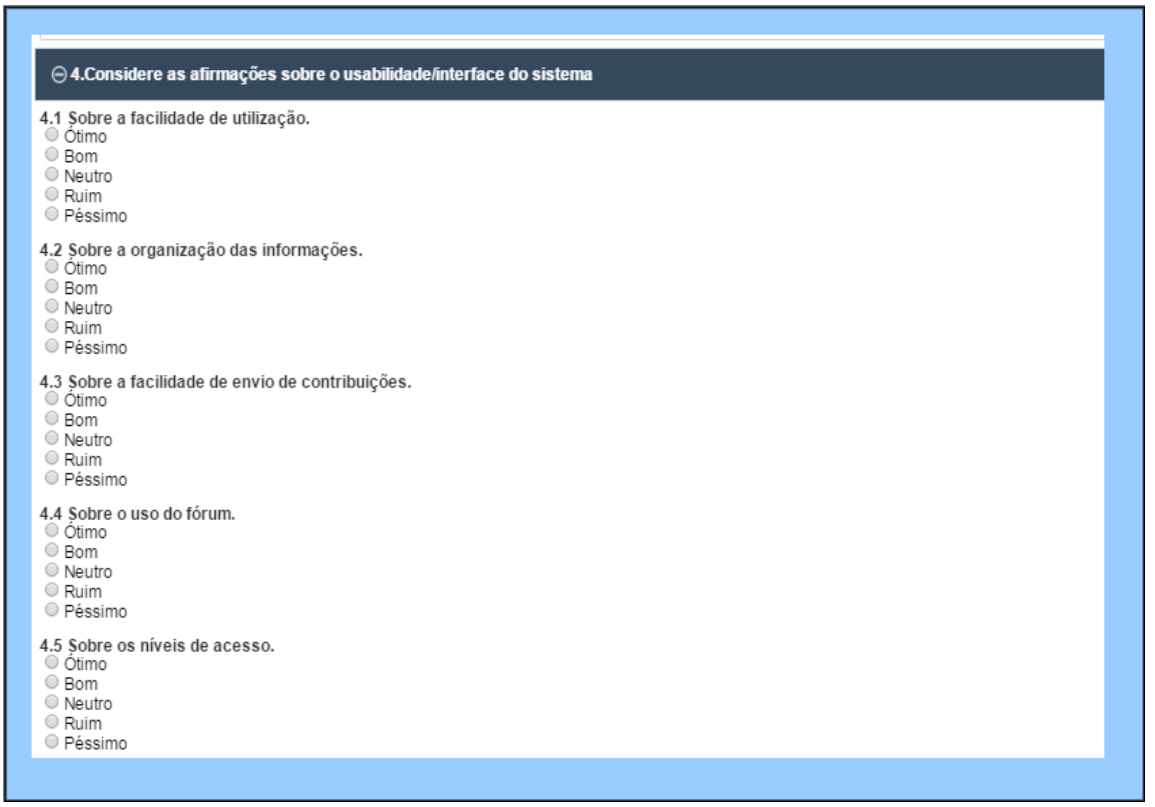
3.2 A modelagem de ontologias é uma estratégia promissora para representar o comportamento de malware.
- Concordo Plenamente
- Concordo Parcialmente
- Neutro
- Discordo Parcialmente
- Discordo Totalmente

3.3 Crowdsource pode ser utilizado para compartilhar modelos sobre comportamento de malware.
- Condordo Totalmente
- Concordo Parcialmente
- Neutro
- Discordo Parcialmente
- Discordo Totalmente

3.4 O conhecimento modelado pode aprimorar a segurança em sistemas de informação, por exemplo, no planejamento de sistemas mais seguros ou na detecção de malware.
- Concordo Totalmente
- Concordo Parcialmente
- Neutro
- Discordo Parcialmente
- Discordo Totalmente

3.5 Uma vez solucionados problemas, em uma versão estável do sistema, utilizaria em minhas atividades profissionais.
- Concordo Totalmente
- Concordo Parcialmente
- Neutro
- Discordo Parcialmente
- Discordo Totalmente

Figura 6.2: Segunda parte do Formulário de Avaliação (Funcionalidade do Sistema).



4.Considere as afirmações sobre o usabilidade/interface do sistema

4.1 Sobre a facilidade de utilização.
- Ótimo
- Bom
- Neutro
- Ruim
- Péssimo

4.2 Sobre a organização das informações.
- Ótimo
- Bom
- Neutro
- Ruim
- Péssimo

4.3 Sobre a facilidade de envio de contribuições.
- Ótimo
- Bom
- Neutro
- Ruim
- Péssimo

4.4 Sobre o uso do fórum.
- Ótimo
- Bom
- Neutro
- Ruim
- Péssimo

4.5 Sobre os níveis de acesso.
- Ótimo
- Bom
- Neutro
- Ruim
- Péssimo

Figura 6.3: Terceira parte do Formulário de Avaliação (Usabilidade do Sistema).

6.2. Resultados

O estudo do sistema foi realizado dentro do ambiente acadêmico das instituições de ensino superior da Universidade Federal do Paraná (*UFPR*), na Faculdade Campo Limpo Paulista (*FACCAMP*), e no Centro de Tecnologia da Informação Renato Archer (*CTI*). Assim, os avaliadores possuíam um grau de instrução diferenciado em relação à população em geral e possuíam conhecimentos específicos nas áreas relacionadas. A Figura 6.4 apresentam grau de instrução dos usuários responsáveis pelas respostas do questionário. Nessa figura podemos visualizar que o mestrado tem maior incidência com 50%, 25% especialização, 12,5% pós-doutorado e 12,5% graduação.

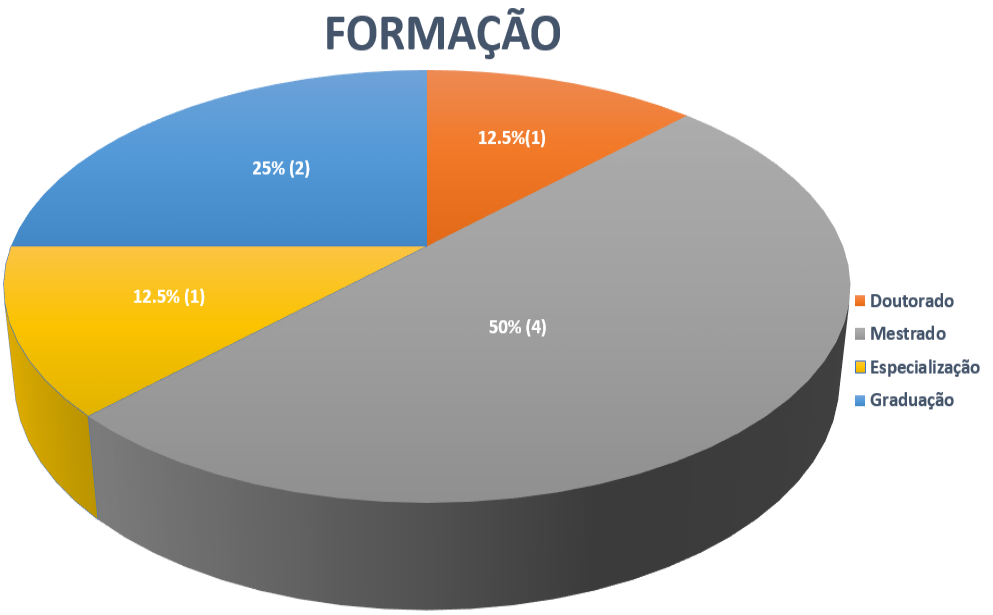


Figura 6.4: Grau de Instrução dos Participantes.

A Figura 6.5a apresenta a área de atuação dos avaliadores, enquanto, a Figura 6.5b apresenta a experiência com segurança da informação. Com relação a área de atuação, 50% atuam em segurança da informação, enquanto 50% não atuam na área. Considerando o tempo de atuação em segurança da informação, temos que 37,5% com mais de 4 anos, 0% com 3 a 4 anos,12,5% 1 a 2 anos e 50% com 0 anos (*i.e.*, os usuários que não atuam na área).

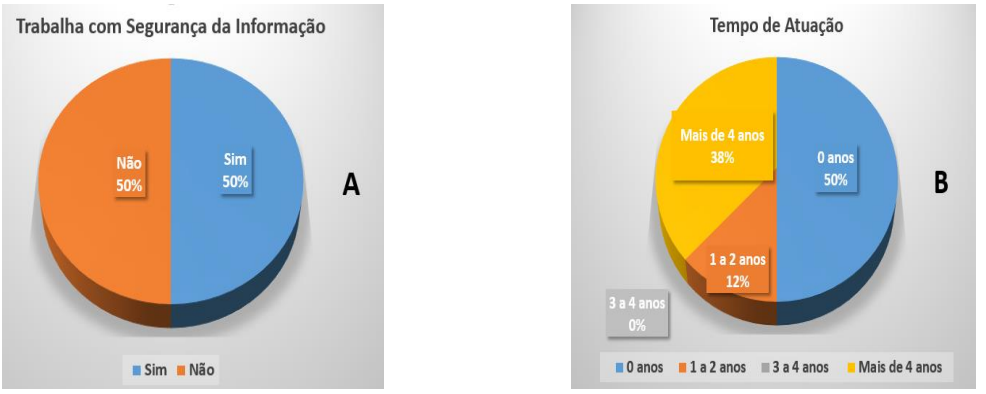


Figura 6.5 Área de atuação e experiência com segurança da informação.

As próximas perguntas respondidas pelos usuários são sobre as funcionalidades do sistema, isto é, envolvem *crowdsourcing*, *malware* e ontologias. O foco é analisar funcionamento geral do sistema CrOMa, visando facilidade e entendimento por meio da plataforma desenvolvida.

A Figura 6.6 contém uma pergunta que questiona se com a utilização do sistema ele/ela poderia entender melhor o problema do *malware*, em específico seus comportamentos. Os resultados demonstram que 72% dos avaliadores concordam plenamente com esta pergunta, enquanto 14% concorda parcialmente e 14% é neutro. Isto demonstra que a maior parte dos avaliadores considerou o sistema útil para este propósito.

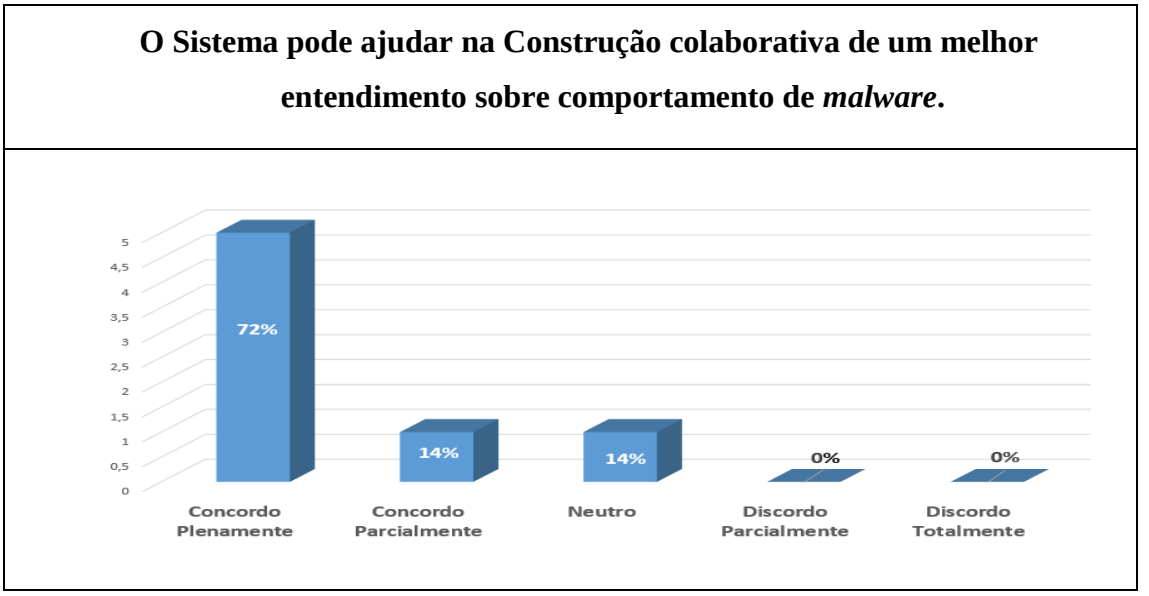


Figura 6.6: Respostas para a pergunta 3.1 do formulário.

Figura 6.7 refere-se sobre a opinião dos usuários se a proposta de solução, isto é, o uso de ontologias é uma estratégia promissora. Os resultados demonstram que 71% dos avaliadores concordaram plenamente.

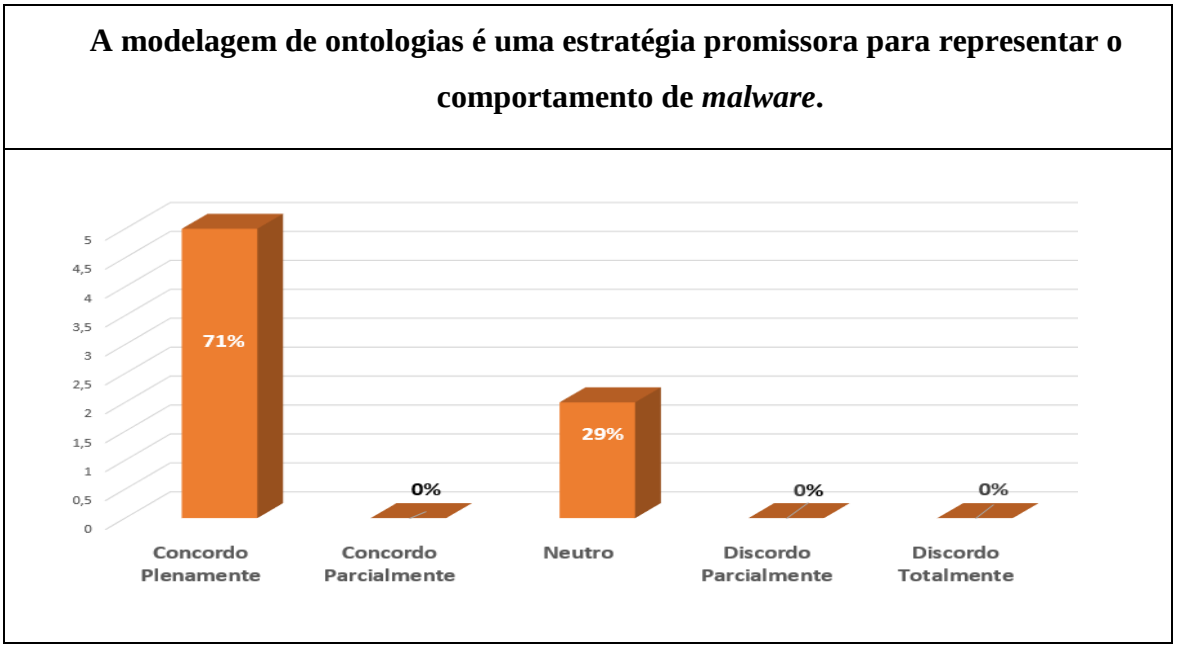


Figura 6.7: Respostas para a pergunta 3.2 do formulário.

A Figura 6.8 apresenta a opinião sobre a estratégia de *c rowdsourcing* para o compartilhamento do conhecimento sobre comportamento de malware. Os resultados demonstram que 86% das pessoas concordaram plenamente.

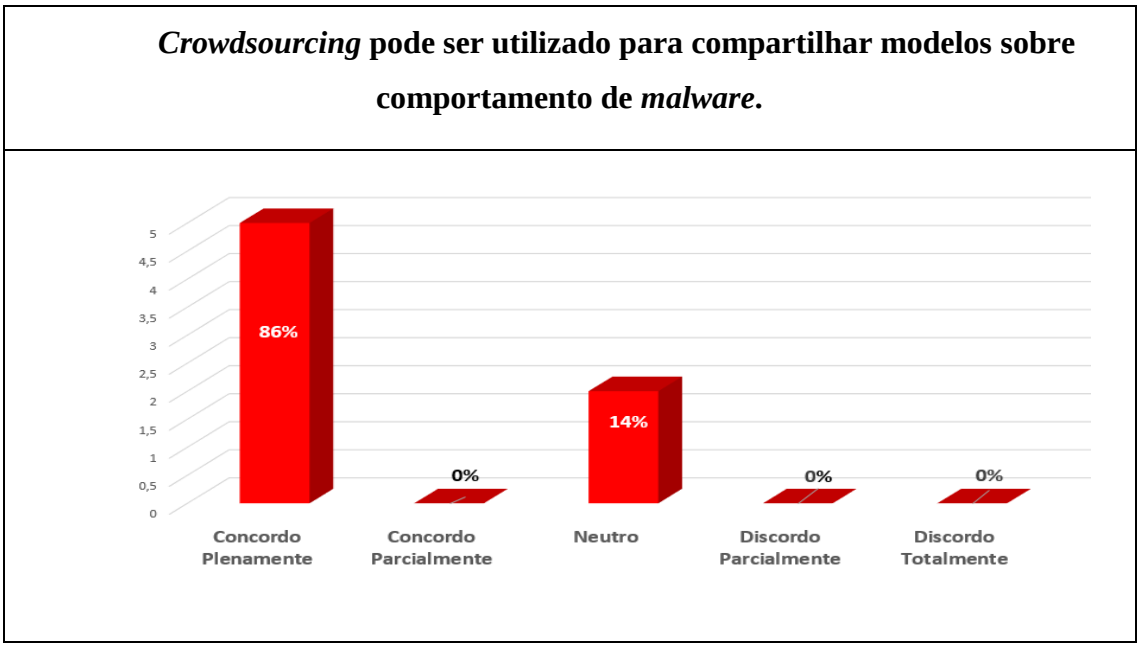


Figura 6.8: Respostas para a pergunta 3.3 do formulário.

A Figura 6.9 apresenta a opinião sobre até que ponto o conhecimento dos comportamentos de *malware* podem ajudar a aperfeiçoar os sistemas de informação, para termos assim sistemas computacionais mais seguros. Os resultados demonstram que 83% das pessoas concordaram plenamente.

O conhecimento modelado pode aprimorar a segurança em sistema de informação, por exemplo, no planejamento de sistemas mais seguros ou na detecção de *malware*.

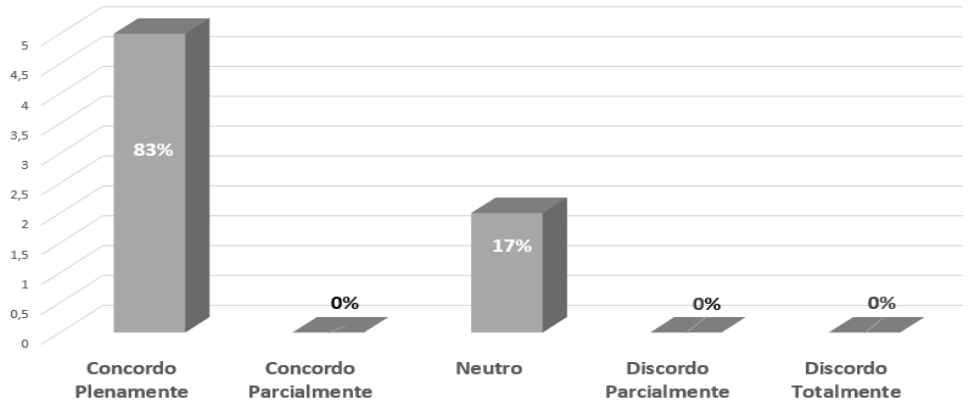


Figura 6.9: Respostas para a pergunta 3.4 do formulário.

A Figura 6.10 apresenta a última pergunta deste bloco, com ela desejava-se saber se o profissional utilizaria esta proposta de solução no dia a dia em seu trabalho. Os resultados demonstram que 72% das pessoas concordaram plenamente, enquanto 14% concordaram parcialmente e 14% foram neutros.

Uma vez solucionados problemas, em uma versão estável do sistema, utilizaria em minhas atividades profissionais?

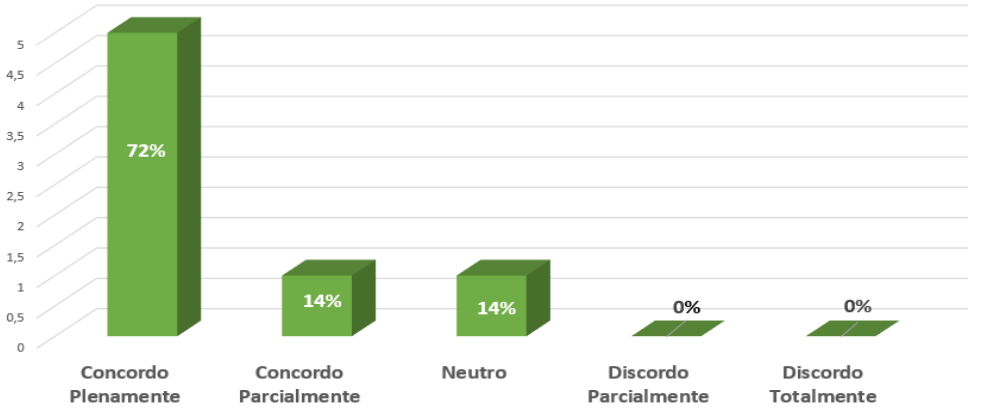


Figura 6.10: Respostas para a pergunta 3.5 do formulário.

Tabela 6.1 apresenta a média da porcentagem das respostas de todos os usuários para as perguntas sobre as funcionalidades do sistema e separadas por tipo de resposta da escala *likert*, sendo que: 76,8% dos usuários concordaram plenamente, 9% concordaram parcialmente e 14,2% foram neutros em relação ao apresentado.

Tabela 6.1: Média das percentagens das 5 perguntas sobre as funcionalidades do sistema.

Média das 5 perguntas sobre Funcionalidades do Sistema				
Concordo Totalmente	Concordo Parcialmente	Neutro	Discordo Parcialmente	Discordo Totalmente
76.8%	5.6%	17.6%	0%	0%

No próximo bloco as perguntas são mais relacionadas com interface e usabilidade do sistema. Elas foram elaboradas para medir a satisfação do usuário com a utilização do sistema.

A Figura 6.11 apresenta a primeira pergunta deste bloco, com ela desejava-se saber sobre a facilidade de utilização do sistema. Os resultados demonstram que 25% das pessoas acharam o sistema ótimo neste aspecto, enquanto 37% bom e 38% foram neutros.

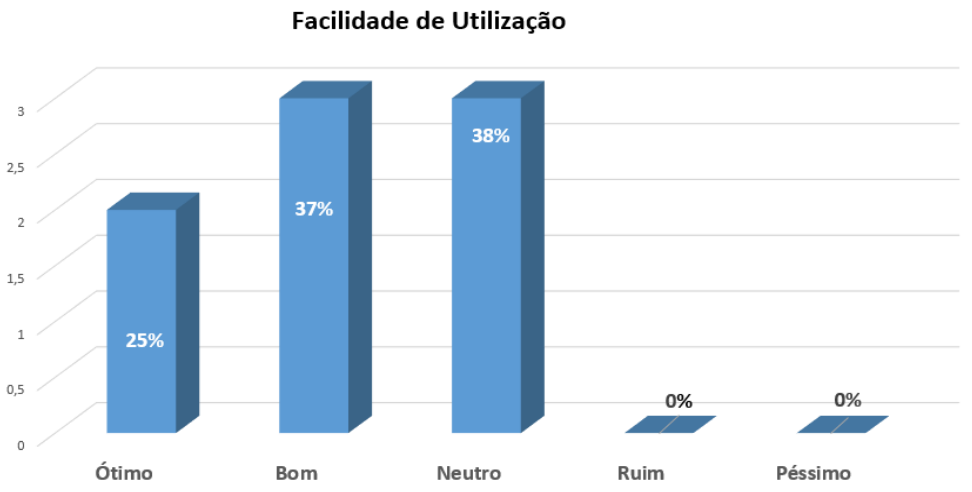


Figura 6.11: Respostas para a pergunta 4.1 do formulário.

A Figura 6.12 apresenta a segunda pergunta deste bloco, com ela desejava-se saber sobre a organização das informações no sistema. Os resultados demonstram que 25% das pessoas acharam o sistema ótimo neste aspecto, enquanto 62% bom e 13% ruim.

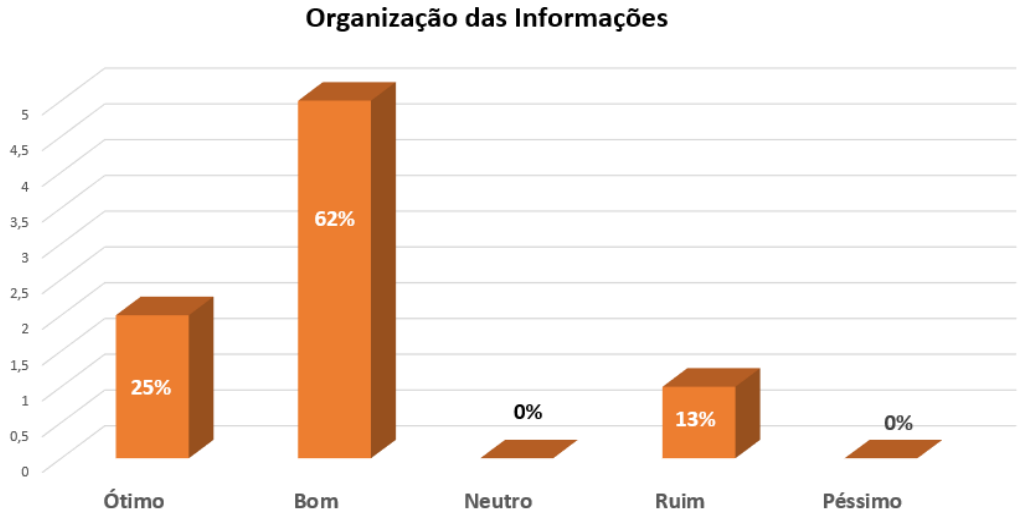


Figura 6.12: Respostas para a pergunta 4.2 do formulário.

A Figura 6.13 apresenta a terceira pergunta deste bloco, com ela desejava-se saber sobre a facilidade de envio de contribuições no sistema. Os resultados demonstram que 29% das pessoas acharam o sistema ótimo neste aspecto, enquanto 57% bom e 14% ficaram neutros.

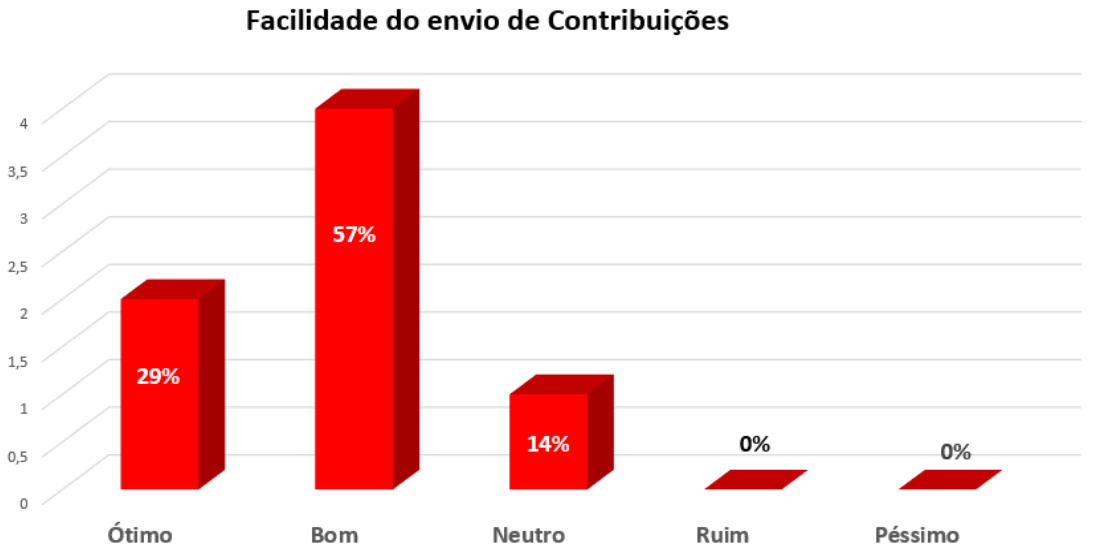


Figura 6.13: Respostas para a pergunta 4.3 do formulário.

A Figura 6.14 apresenta a quarta pergunta deste bloco, com ela desejava-se saber sobre o uso do fórum do sistema. Os resultados demonstram que 37% das pessoas acharam o sistema ótimo neste aspecto, enquanto 38% bom e 25% acharam ruim.

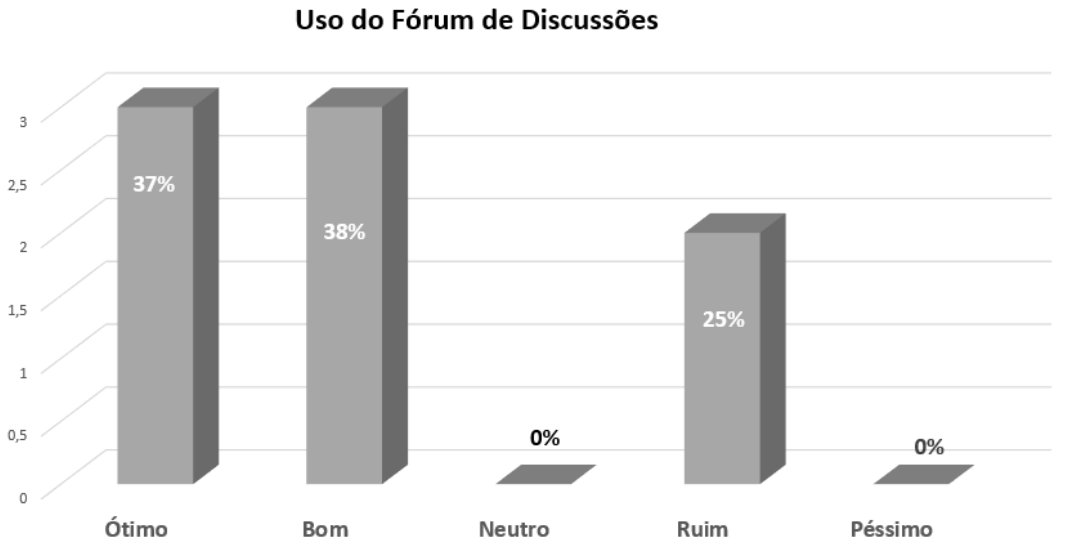


Figura 6.14: Respostas para a pergunta 4.4 do formulário.

A Figura 6.15 apresenta a última pergunta deste bloco, com ela desejava-se saber sobre os níveis de acesso do sistema. Os resultados demonstram que 43% das pessoas acharam o sistema ótimo neste aspecto, enquanto 57% acharam bom.

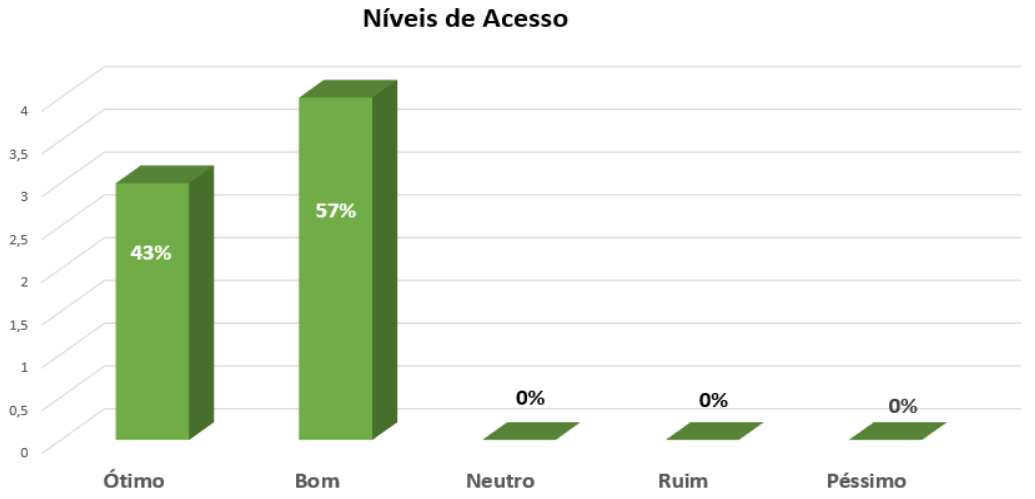


Figura 6.15: Respostas para a pergunta 4.5 do formulário.

Tabela 6.2 apresenta a média da porcentagem das respostas de todos os usuários para as perguntas sobre a usabilidade do sistema e separadas por tipo de resposta, sendo que: com 31,8% acharam ótimo, 50,2% bom, 10,4% neutro e 7,6% ruim. Ou seja, foram priores do que as perguntas do primeiro bloco.

Tabela 6.2: Média das percentagens das 5 perguntas sobre a usabilidade do sistema.

Média das 5 perguntas sobre Usabilidades do Sistema				
Ótimo	Bom	Neutro	Ruim	Péssimo
31.8%	50.2%	10.4%	7.6%	0%

A pergunta 3.6 visava apontar sugestões ou críticas sobre as funcionalidades do sistema. As respostas destacam a dificuldade de entendimento do controle de acesso do sistema, necessidade de aprimoramento da segurança e praticidade, e contém elogios sobre a possibilidade de compartilhar conhecimento, por exemplo: "*O sistema é um meio muito bom para compartilhar conhecimento*".

Já a pergunta 4.6 visava apontar sugestões ou críticas sobre a usabilidade/interface do sistema. As respostas destacam sugestões de melhorias na janela de postagem do fórum, alguns problemas com as janelas e guias, e sugestão para: "*melhorar autonomia do usuário*".

A pergunta 5.1 do questionário foi perguntado por que eles recomendariam o sistema, sendo que 87% responderam que sim. As respostas dos usuários focaram em destacar o caráter inovador da proposta e o potencial de compartilhamento de informações, por exemplo, um usuário declarou "*Proposta inovadora e importante nos dias atuais*. ", enquanto outro declarou *"Permite a captura e validação das informações sobre malware junto à comunidade. "*

For fim, a pergunta 5.2 visava verificar se eles achavam que o programa tinha atingido seus objetivos, ou não. Todos os avaliadores responderam positivamente a esta pergunta, entre os principais pontos a serem evoluídos segundo os usuários estão os relacionados a pequenos *bugs*, compatibilidade de navegadores e melhorias na interface. Um usuário, por exemplo, destacou: "*Sim, apesar dos problemas no meu navegador. Isso é uma questão de evolução do sistema*".

A Tabela 6.3 sumariza as respostas dos usuários para as questões discursivas divididas pela área de atuação e formação do usuário, e suas respectivas respostas por cada pergunta classificadas como “satisfeito”, “insatisfeito” e “não respondeu”. Podemos salientar, portanto, que a incidência de "não respondeu" é maior entre usuários que não atuam com segurança da informação.

Tabela 6.3: Avaliador e tipo de respostas.

AVALIADOR	PERGUNTA 3.6	PERGUNTA 4.6	PERGUNTA 5.1	PERGUNTA 5.2	TRAB. SEG. INF.	FORMAÇÃO
Usuário 1	satisfeito	satisfeito	satisfeito	satisfeito	Sim	Mestrado
Usuário 2	insatisfeito	insatisfeito	satisfeito	satisfeito	Sim	Pós-Doc
Usuário 3	não respondeu	não respondeu	satisfeito	não respondeu	Não	Mestrado
Usuário 4	satisfeito	satisfeito	insatisfeito	satisfeito	Não	Especialização
Usuário 5	não respondeu	não respondeu	satisfeito	insatisfeito	Não	Graduação
Usuário 6	insatisfeito	insatisfeito	satisfeito	satisfeito	Sim	Especialização
Usuário 7	satisfeito	insatisfeito	satisfeito	satisfeito	Não	Mestrado
Usuário 8	não respondeu	satisfeito	satisfeito	satisfeito	Sim	Mestrado

A Figura 6.16 mostra o resultado da pergunta: “Sugestões ou críticas sobre as funcionalidades o sistema”, mostrando o resultado onde 25% não respondeu, 37% colocaram sugestões e 38% colocaram críticas.

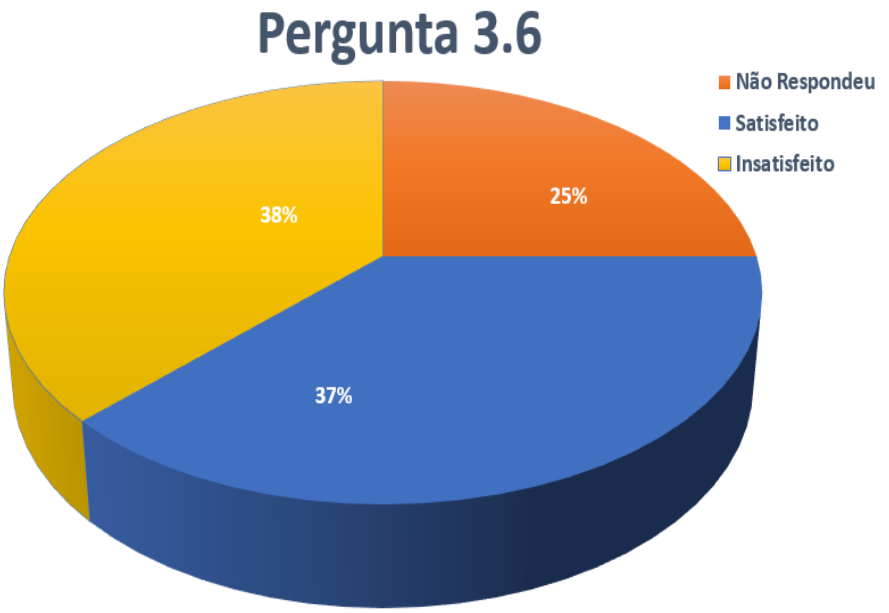


Figura 6.16: Respostas para a pergunta 3.6 do formulário

A Figura 6.17 mostra os resultados da pergunta: “Descreva sugestões ou críticas sobre usuabilidade do sistema”, onde 38% não respondeu, 37% colocaram sugestões e 38% colcocaram críticas.

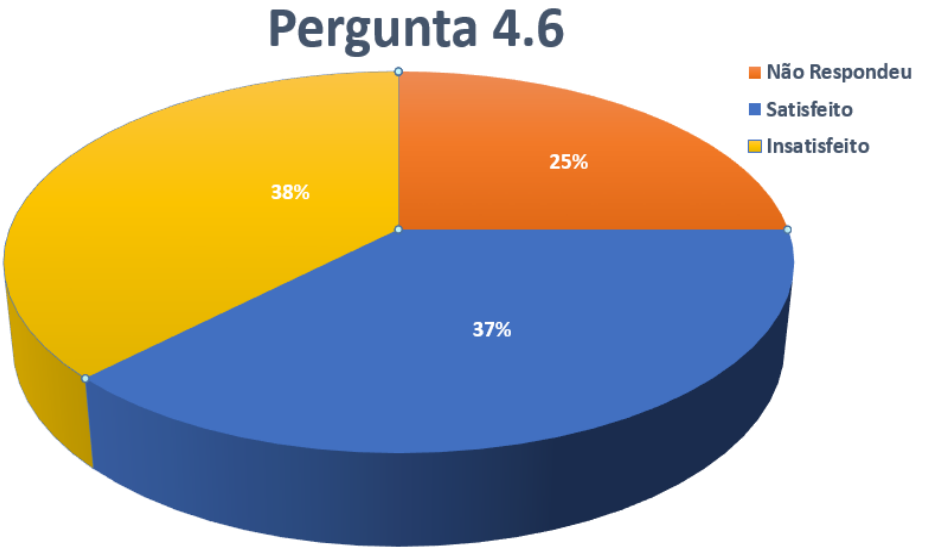


Figura 6.17: Resposta da pergunta 4.6.

A Figura 6.18 apresenta os resultados da pergunta: “Se indicaria o sistema para outros usuários”, onde todos responderam com 13% não indicaria e 83% indicaria o sistema para outros usuários.

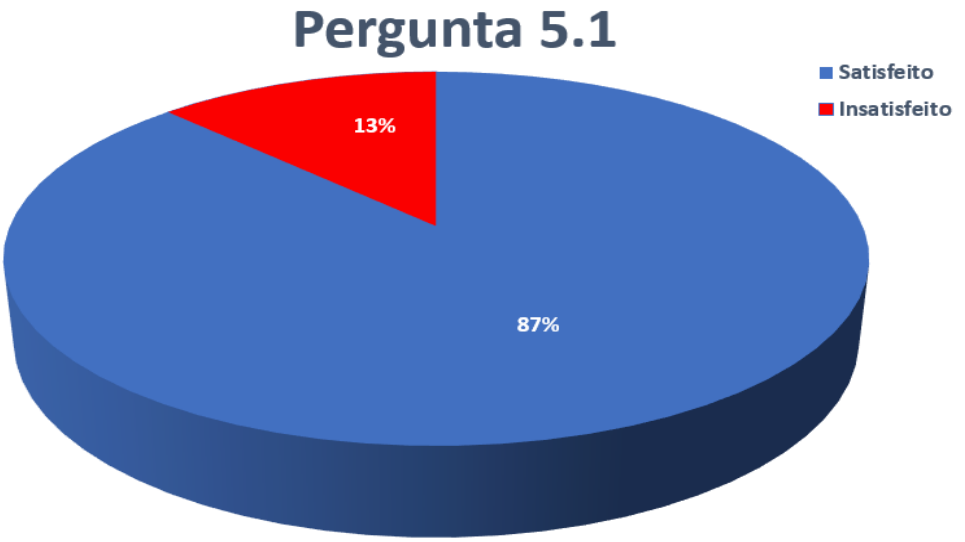


Figura 6.18: Resposta da pergunta 5.1.

A Figura 6.19 mostra o resultado da pergunta: “Se o sistema atingiu seu objetivo”, onde tivemos 12.5% não respondeu, 75% satisfeito e 12.5% insatisfeito.

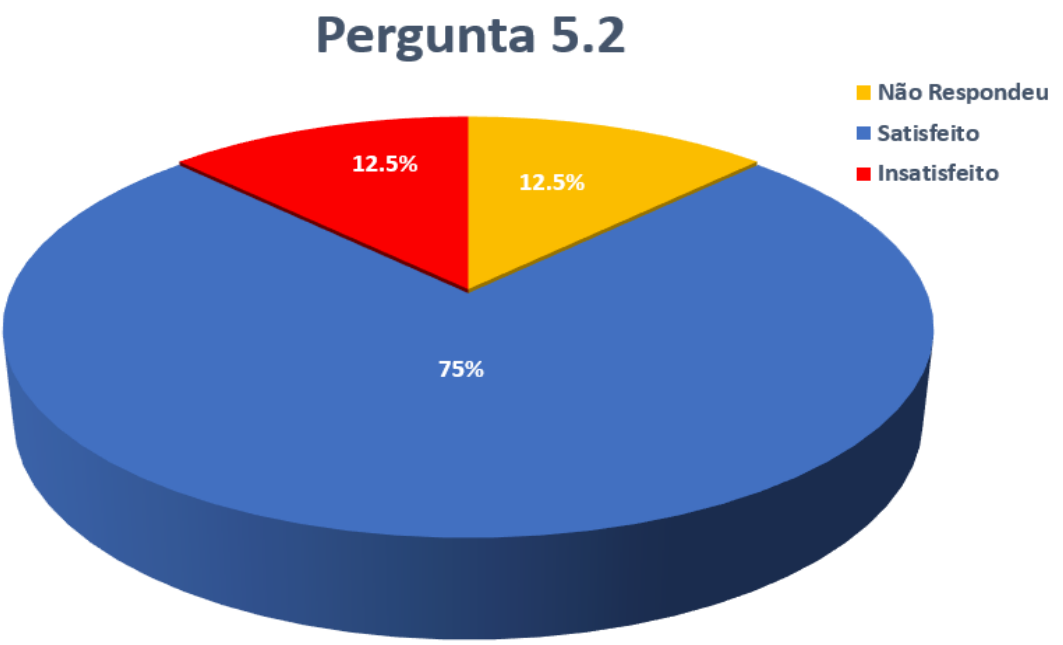


Figura 6.19: Classificação das respostas da pergunta 5.2.

6.3. Discussão

O principal objetivo do estudo era desenvolver diferentes formas de contribuições, e consequentemente com o uso de ações da inteligência colaborativa poder auxiliar na evolução de ontologias de *malware*. Para a realização do experimento foi desenvolvido um protótipo descrito anteriormente que se encontra disponível em: projectweb-solutions.myscriptcase.com/cm/.

Para analisar a aplicabilidade do protótipo, nesta Seção são discutidos os resultados obtidos por meio da análise das respostas apresentada na seção anterior, onde se buscou analisar a funcionalidade do sistema, aspectos de usabilidade e coletar opiniões do usuário.

Os resultados das perguntas relacionadas com a funcionalidade em sua maioria foram positivos, uma vez que a maioria dos avaliadores observou a praticidade e boa funcionalidade do sistema, além da possibilidade de envio de contribuições. Como relatado, alguns avaliadores declararam dificuldades no entendimento dos níveis de acesso

e uso das funcionalidades. Faz-se necessário chamar a atenção que a avaliação inclui usuários que não atuam com segurança da informação, assim, alguns avaliadores deixaram em branco algumas respostas, possivelmente em função da pouca familiaridade com o assunto.

Os resultados de usabilidade também mostram que o sistema teve uma boa aceitação entre os avaliadores, porém ficou evidente que ele ainda necessita de algumas melhoras, por meio de uma melhor estrutura de informação, correção de *bugs* de interface, simplificação de tarefas e uma linguagem mais direta. Mas, de forma geral podemos afirmar que ele atingiu seu objetivo, uma vez que é o primeiro protótipo colocado para testes com usuários, e que os problemas apontados não impediram que os usuários explorassem o sistema. Os especialistas em segurança de informação tiveram uma avaliação mais crítica neste bloco de perguntas. No futuro, uma visão mais precisa poderá ser obtida pelo método de avaliação heurística em conjunto com os especialistas.

Conclui-se, portanto, pelas respostas das questões de múltipla escolha, que o resultado geral do experimento foi positivo. Salienta-se, entretanto, que os resultados sobre as funcionalidades foram ligeiramente superiores que usabilidade/interface, destacando que os maiores problemas foram constatados nas perguntas sobre *Organização das Informações* e *Uso do Fórum*.

As questões discursivas foram desenvolvidas visando uma análise qualitativa do sistema, buscamos por meio destas respostas uma avaliação da satisfação dos usuários, bem como identificar problemas do sistema. Merecem destaque que as respostas sobre se *Indicaria o Sistema para outro usuário*, e se *O Sistema atingiu seu objetivo* foram muito bem avaliadas. Pode-se concluir que de forma geral as respostas foram positivas, mas com algumas críticas e alguns *bugs* relatados.

Quanto às limitações do formulário de avaliação, podemos destacar a constatação, a partir dos resultados, da necessidade de diversificar as perguntas para os avaliadores, bem como promover uma avaliação em longo prazo. Não conseguimos avaliar o uso das funcionalidades de versionamento da ontologia e nem das ferramentas compartilhadas. Só com testes a longo prazo poderíamos avaliar estas funcionalidades, bem como se o sistema efetivamente contribuiu para a evolução da ontologia. Outra limitação do experimento é a quantidade de avaliadores, para uma plataforma de *crowdsourcing* funcionar na prática,

são necessárias centenas ou milhares de pessoas compartilhando conhecimento e contribuindo colaborativamente.

Alguns problemas de uso já foram analisados para uma nova versão, tais como alterar o menu, deixando ele com menos itens e mais subitens onde o usuário tenha acesso as funcionalidades, bem como o uso de itens com identificações mais representativas. Outra estratégia, foi disponibilizar vídeos com tutoriais de utilização do sistema CrOMa, bem como vídeos de utilização das ferramentas compartilhadas e da ontologia (a página da Figura 5.3 foi desenvolvida após a avaliação).

Outro problema a ser destacado é a compatibilidade entre navegadores, o problema maior foi com o *Safari*. Como este é o navegar padrão para dispositivos da *Apple*, pretende-se fazer um estudo sobre o que pode ter ocasionado o problema. Uma das hipóteses é o uso da biblioteca que possibilita a construção de páginas responsivas ou o *Zend Optmizer*.

Problemas menores, como *bugs* relacionados ao tamanho de campos e visualização já foram resolvidos. Entretanto, salientamos que para o uso em longo prazo se faz necessário a correção das limitações destacadas acima e aprimoramento da confiabilidade do sistema.

Nesse sentido, uma implicação prática é a dificuldade de conciliar uma boa segurança com uma interface agradável, o fato de ter vários níveis de acesso (necessários para a segurança do sistema) resulta em problemas para o entendimento do funcionamento do sistema e em uma interface mais complexa. Por exemplo, a solução dada para implementar os níveis de acesso exigiu a criação de menus com muitos subitens para poder separar na camada das regras de negócios os tipos de acesso. Uma maior autonomia do usuário poderia facilitar o acesso às diversas funcionalidades do sistema, mas ao mesmo tempo poderia trazer riscos à segurança e consistência do modelo compartilhado. Mecanismos de inteligência artificial que facilitem a contribuição indireta e detecção de problemas no modelo poderiam contribuir para melhores resultados, e para tentar conciliar o uso fácil do sistema com a confiabilidade e segurança requerida.

Uma das possíveis expansões do sistema é capturar informações automaticamente junto às discussões no fórum, entretanto isto requer o uso de técnicas de processamento de linguagem natural e/ou de algoritmos de classificação e aprendizagem de máquina.

Outra funcionalidade desejável é poder editar diretamente a ontologia no sistema, pode-se destacar ainda o fato que as regras poderiam ser adicionadas diretamente na ontologia após a aprovação dos especialistas de domínio. Na versão atual é necessário que eles abram um editor, como o Protégé, para realizar esta tarefa. A *API* da *OWL* utilizada no *Protégé*, por exemplo, poderia ser utilizada para fazer tarefas simples como a inclusão de classes e regras na ontologia, mas temos que ter em mente que não é prudente, do ponto de vista de foco da pesquisa e esforço necessário, reconstruir uma ferramenta complexa como o *Protégé*.

Como lições aprendidas, o autor pode afirmar as dificuldades em conciliar a parte teórica e conceitual sobre ontologias com conceitos de segurança de informação. Também houve dificuldades com conseguir usuários adequados para os testes. Este foi o principal problema enfrentado com o experimento, assim foi utilizado apenas alunos, pesquisadores e professores. A falta de tempo dos profissionais e disponibilidade é sempre uma questão primordial na condução de experimentos deste tipo. Ainda, existe a dificuldade de encontrar pessoas que conhecessem as tecnologias ontologias e *malware*. Outra dificuldade foi relacionada ao entendimento da linguagem *SWRL/OWL*, a qual foi usada na concepção da ontologia que foi utilizada no sistema.

O objetivo principal da dissertação é a elaboração do modelo de *crowdsourcing*, mas, o fato de tentar conciliar dois campos distintos a análise de comportamento de *malware* e ontologias trouxeram uma estrutura diferenciada para o sistema. Esta estrutura impõe dificuldades e levanta desafios futuros a serem adotados por pesquisadores de várias áreas da computação.

Por exemplo, a pesquisa por uma plataforma de *crowdsourcing*, onde especialistas em segurança de informação que não conhecem ontologias possam contribuir em linguagem natural e o sistema fizesse a tradução automaticamente para linguagem *OWL* é muito desejada nesse contexto. Isto poderia ser acompanhada da verificação automática da consistência da ontologia. O uso e reuso de integrado de outras ontologias (e modelos) existentes também poderia trazer benefícios neste contexto.

Neste Capítulo, foram apresentados os resultados obtidos por meio de questionário de avaliação e uma breve discussão dos resultados, bem como limitações, dificuldades e problemas encontrados. No próximo Capítulo, é realizada a conclusão dessa dissertação.

7- Conclusão

Neste trabalho foi proposto uma plataforma de *crowdsourcing* e foi implementado um protótipo, para auxiliar na evolução de ontologias para modelagem de comportamentos de *malware*. O estudo do comportamento dos códigos maliciosos se faz necessária em vistas a novos *malware* que utilizam mecanismos complexos que burlam a análise por assinatura de código, bem como inutilizam as classificações já existentes que tentam atribuir uma única característica comportamental aos *malware*, tais como *vírus* e cavalo de *tróia*. Portanto, a evolução dos códigos dos *malware*, utilizadas para burlar os sistemas de defesa, é um problema crônico a ser enfrentado, que pode ser minimizado com a análise dos comportamentos dos mesmos. Além disso, a construção de uma linguagem comum para a utilização entre os profissionais de segurança se faz necessário. Neste contexto, se destaca o uso de ontologias como alternativa viável. Porém a manutenção e evolução de ontologias complexas é um problema de difícil solução frequentemente relatado na área da *web semântica*. Portanto, nesta dissertação foi apresentada uma alternativa para esse problema, que é o uso de *crowdsourcing*. Nessa alternativa é possível utilizar o esforço coletivo de trabalhadores em tarefas que não podem ser facilmente automatizadas. O desafio central explorado nesta dissertação foi conciliar, de forma inédita, requisitos das áreas de segurança de informação e *web semântica* em um modelo de *crowdsourcing*. Este modelo foi implementado em um sistema e avaliado, de forma preliminar, por 8 usuários. Na Seção 7.1 são apresentadas as contribuições desta pesquisa, a Seção 7.2 apresenta propostas de trabalhos futuros e, por fim, a Seção 7.3 faz as considerações finais.

7.1 Contribuições

Retomando a questão de pesquisa apresentada no capítulo 1 “Como utilizar mecanismos de *crowdsourcing* para possibilitar o compartilhamento de informações e evolução colaborativa de ontologias e regras para análise comportamental de *malware*? ”, nesta dissertação, ao tentar respondê-la foram geradas diferentes contribuições científicas e tecnológicas. Essas contribuições vão desde a análise do estado da arte sobre o tema, formulação de hipóteses para solução do problema, proposta de um modelo, construção de um software e avaliação dos resultados.

De maneira sucinta, destacam-se as seguintes contribuições principais:

- Desenvolvimento inédito de modelo de *crowdsourcing* que leva em consideração requisitos de ambas as áreas, segurança de informações e ontologias. Este modelo propicia meios para evoluir a ontologia por usuários com pouco conhecimento sobre a sintaxe da *OWL* (ou outra linguagem de descrição de ontologias), ao propiciar contribuições via fórum de discussão e linguagem natural. Ao mesmo tempo o modelo leva em consideração aspectos específicos de segurança para garantir a evolução controlada do modelo e evitar o uso inadequado da plataforma. Este modelo contribui para o avanço do conhecimento de como as áreas *Web Semântica* e ontologias podem se beneficiar do uso de *crowdsourcing*;
- Construção de um sistema computacional que utiliza este modelo. Este sistema computacional implementa os principais requisitos descritos pelo modelo e encontra-se funcional e disponível na *Web*. Com o sistema é possível postar ideias e contribuições, as quais são analisadas por especialistas de ontologias e transformadas em regras, classes ou outros elementos da ontologia. Embora ainda em estágio inicial, este sistema serve de ponto de partida, do ponto de vista técnico, para o uso em larga escala em uma plataforma de *crowdsourcing* pela comunidade de segurança de informação;
- A avaliação preliminar do sistema possibilitou verificar a viabilidade da proposta, bem como apontar suas limitações e desafios futuros a serem enfrentados. Tais desafios apontam para pesquisas em áreas como inteligência artificial, processamento de linguagem natural, ontologias, segurança de informação e interação humano-computador.
- A dissertação resultou em um artigo publicado no *journal Applied Ontology*, sobre ontologia e *malware* com o título "*An Ontology of Suspicious Software Behavior* "onde foi desenvolvido um sistema para a inserção de regras com os plug-ins *OWL*, diretamente na ontologia, para tanto foi desenvolvida uma ferramenta inédita de carregar logs na linguagem *Java* pelo autor, utilizada para testes e comprovada a eficiência com a publicação do artigo.

- Um artigo publicado na *WETICE*, 2017 Polônia sob o título "*Enhancing the Creation of Detection Rules for Malicious Software through Ontologies and Crowdsourcing*", que mostra como melhorar a criação de regras para análise de *malware* de uma ontologia utilizando *crowdsourcing*, um trabalho inédito envolvendo as tecnologias comportamento de *malware*, ontologias e c*rowdsourcing*.

7.2- Trabalhos Futuros

Esta pesquisa foi concluída com uma solução para evolução da ontologia por meio de contribuições colaborativas, entretanto se faz evidente os próximos passos para a pesquisa, conforme descritos a seguir:

- O desenvolvimento de uma plataforma onde a linguagem natural seja transformada automaticamente em regras *SWRL*, facilitando assim o trabalho dos especialistas de domínio. Porém deve existir o controle na inserção dessas regras na ontologia, para evitar erros e manter a consistência na nova versão da ontologia;
- Validação automática das alterações com o desenvolvimento de um mecanismo para verificar inconsistência nas regras;
- A mineração dos dados do fórum para filtrar as mensagens/contribuições que possam ser transformadas em regras, classes ou outros elementos da ontologia. Para tanto, é possível combinar programas preexistentes para fazer análise dos textos em fórum de discussão, mas são necessárias pesquisas aprofundadas sobre a aplicação no contexto em questão (segurança de informação);
- Uma integração do sistema CrOMa com *APIs* para inserção e manipulação da ontologia diretamente da interfase do sistema, sem a necessidade de ferramentas externas;
- Estudo em larga escala com múltiplos usuários, com vários especialistas de domínio. Isto possibilitará aprimorar o sistema para seu uso na prática.

Desta forma, com estas propostas pode-se melhorar a criação de regras sobre comportamento de *malware*, melhorar a interface do sistema e sua escalabilidade. Uma avaliação da evolução de uma ontologia em termos de sua representatividade e qualidade baseadas na interação entre os participantes também é um objetivo a ser atingido em longo prazo.

7.3- Considerações Finais

Conforme demonstrado durante o trabalho, o uso de *crowdsourcing* para evoluir ontologias para análise de comportamento de *malware* se mostrou uma alternativa confiável, que pode dar subsídios para atividades futuras, verificar os comportamentos de programas suspeitos e lidar com a complexidade dos novos *malware*. O *crowdsourcing*, um conceito acessível e com sistemas de fácil utilização, pode agregar conhecimento compartilhado, entretanto o protótipo do sistema CrOMa deve sofrer muitas mudanças e evoluir, para que seja de fato utilizado pela comunidade de segurança de informação. Isso esbarra em desafios de pesquisas que foram trazidos a partir da experiência obtida nesta dissertação.

Por fim, destaca-se a contribuição desta dissertação como peça única até então, e inovadora ao buscar soluções integrando temas distintos como *crowdsourcing*, ontologias e análise de comportamento de *malware*. Espera-se ter contribuído para o avanço do conhecimento nesses assuntos e levantadas novas questões de pesquisas.

Referências

Afonso, V. M. 2011. Um sistema para análise e detecção de ataques ao navegador *Web*. *Dissertação de Mestrado*, FEEC/UNICAMP.

Al-Bataineh, A., & White, G. 2012. Analysis and detection of malicious data exfiltration in web traffic. IEEE In *Malicious and Unwanted Software (MALWARE) 7th International Conference on*. pp 26-31.

Armoun, S. E., S. Hashemi, 2012. General Paradigm for Normalizing Metamorphic Malwares. *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, Islamabad, n. 13272078, pp. 348 – 353.

Ask, K. 2006. Automatic malware signature generation. *Tese de Doutorado, Master'sthesis, KTH Royal*.

Barakat, O. L., Ramli, A. R., Hashim, F., Samsudin, K., Al-baltah, I. A., & Al-Habshi, M. M. 2013. SCARECROW: Scalable Malware Reporting, Detection and Analysis. *Journal of Convergence Information Technology*, *8* (14), 1.

Bastos, A. B. 2013. Uma abordagem ontológica baseada em informações de contexto para representação de conhecimento de monitoramento de sinais vitais humanos. *Dissertação de Mestrado*, UFG Universidade Federal de Goiás.

Bencsáth, B., Pék, G., Buttyán, L., & Félegyházi, M. 2012. Duqu: Analysis, detection, and lessons learned. In *ACM European Workshop on System Security (EuroSec)* (Vol. 2012).

Berners-Lee, T., Hendler, J., & Lassila, O. 2001. The semantic web. *Scientific american*, *284*(5), pp. 28-37.

Blount, J. J., Tauritz, D. R., & Mulder, S. 2011. Adaptive rule-based malware detection employing learning classifier systems: A proof of concept. *In Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*. pp.110-115.

Braun, S., Schmidt, A. P., Walter, A., Nagypal, G., & Zacharias, V. 2007. Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering. *Ckc*, *273*.

Butinhão, R. D., & Soares, L. T. 2014. Modelo de Contribuições Coletivas com Base em Crowdsourcing. *SEGeT 2014, Simpósio de Excelência em Gestão e Tecnologia*. Rio de Janeiro, Brasil.

Ceron, J. M., Granville, L. Z., & Tarouco, L. M. R. 2009. Taxonomia de Malwares: Uma Avaliação dos Malwares Automaticamente Propagados na Rede. *IX Simpósio Brasileiro Segurança da Informação e Sistemas Computacionais*. pp.43-56, Campinas, Brasil.

Chi, E. H., & Bernstein, M. S. 2012. Leveraging online populations for crowdsourcing**.** *IEEE Internet Computing, 16*(5), pp 10-12.

Cisco 2014, Disponível: http://www.cisco.com/c/dam/r/pt/br/internet-of-everything-ioe/assets/pdfs/whitepaper_c11-733367_PTBR.pdf. [Acessado 16 Agosto 2015].

Cocate, F. M., Arbex, R. M., & Resende, V. L. 2014. Crowdsourcing: análise do fenômeno sob a ótica do poder das multidões. *Intercom Sociedade Brasileira de Estudos Interdisciplinares da Comunicação,* Foz do Iguaçu, Brasil

Coelho, K. C., & Almeida, M. B. 2012. Aquisição de conhecimento para construção de ontologias: uma proposta de roteiro metodológico aplicado ao contexto da hematologia. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, *17*(35), 47-74.

Costa, J. S., Cruz, M. A. P., & Silva, J. 2012. Segurança de Redes de Computadores na Internet. Revista Inovação, Teresina/PI, v.1, n.2, p.6. Disponível: http://www4.fsanet.com.br/revista/index.php/inovaacao/article/view/480/pdf. [Acesso 25 Março 2015].

Christoforidis, C., Vlachos, V., & Androulidakis, I. 2014. A crowdsourcing approach to protect against novel malware threats. *IEEE In Telecommunications Forum Telfor (TELFOR),* pp 1063-1066.

Davis, J. G. 2011. From crowdsourcing to crowdservicing. *IEEE Internet Computing, 15*(3), 92-94.

Dias, T. D., & Santos, N. 2001. *Web* Semântica: Conceitos Básicos e Tecnologias Associadas (Tutorial). 14f. *Monografia (Bacharelado em Ciência da Computação)* - Instituto de Matemática e Estatística - UERJ, Rio de Janeiro, 2001.

Di Lorio, A., Musetti, A., Peroni, S., & Vitali, F. 2010. Crowdsourcing semantic content: a model and two applications. *IEEE In Human System Interactions (HSI), 2010 3rd Conference on*, pp 563-570.

Pinheiro, J. M. S.2007. Ameaças e Ataques a Sistema de Informação: Prevenir e Antecipar. 5° ed, p.12.

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. 2012. Towards an integrated crowdsourcing definition. *Journal of Information science*, *38*(2), pp.189-200.

Falliere, N., Murchu, L., & Chien, E. 2014. W32. Stuxnet Dossier. Symantec Security Response, disponível: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. [Acesso16 Julho de 2015].

FEBRABAN, Pronunciamentos. Disponível: https://www.febraban.org.br/Noticias1.asp?id_texto=2364. [Acesso 14 Julho de 2015].

Filho, D. S. F., Afonso, V. M., Martins, V. F., Grégio, A. R. A., de Geus, P. L., Jino, M., & dos Santos, R. D. C**.,** 2009. Técnicas para Análise Dinâmica de Malware. *Simpósio Brasileiro em Segurança da Informação e Sistemas Computacionais*.

Filho, D. S. F., Grégio, A. R., Afonso, V. M., Santos, M. J., & de Geus, P. L**.,** 2010**.** Análise Comportamental de Código Malicioso Através da Monitoração de Chamadas de Sistema e Tráfego de Rede.

Filho, D. S. F., & de Geus, P. L. 2010. 15 Análise Comportamental de Código Malicioso Através da Monitoraçao de Chamadas de Sistema e Tráfego de Rede. *dernas 10 5 Security Data Parametrization and Visualization 12 6 Multiscale Parameter Search (MSPS) 14 7 Um Oráculo para Teste de Robustez Baseado em Algoritmos de Alinhamento de Sequências 16*, 33.

Gavrilut, D., Cimpoeşu, M., Anton, D., & Ciortuz, L. 2009**.** Malware detection using machine learning. In *Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on*, pp 735-741.

GData, 2014. New Computer Malware Every 8.6 Seconds. Disponível: https://www.gdatasoftware.com/newsroom/news/article/g-data-new-computer-malware-every-86-seconds. [Acesso 26 Março 2015].

GizModo Brasil, O worm Stuxnet também tinha um irmão gêmeo maligno, 2013, disponível em: http://gizmodo.uol.com.br/stuxnet-irmao-gemeo/. [Acesso 14 Julho de 2015].

Grégio, A., Bonacin, R., de Marchi, A. C., Nabuco, O. F., & de Geus, P. L. (2016). Na ontology of suspicious software behavior. *Applied Ontology*, pp 29-49.

Grecio, A., Bonacin, R., Nabuco, O., Monte Afonso, V., Licio De Geus, P., & Jino, M. 2014. Ontology for Malware Behavior: A Core Model Proposal. *IEEE In WETICE Conference (WETICE) 23rd International*, pp. 453-458.

Golbreich, C. 2004. Combining rule and ontology reasoners for the semantic web. In *Rules and rule markup languages for the semantic Web*. Springer Berlin Heidelberg, pp. 6-22.

Guarino, N. 1998. Formal ontology in information systems: *Proceedings of the first international conference (FOIS'98), (pp-6-8)*, (Vol. 46). IOS press, Trento, Italy.

Horridge, M., Knublauch, H., Rector, A., Stevens, R., & Wroe, C. 2004. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. *University of Manchester*.

Horrocks, I. 2008. Ontologies and the semantic web. *Communications of the ACM*, *51*(12), 58-67.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. 2004. SWRL: A semantic web rule language combining OWL and Rule ML. *W3C Member submission*, pp.*21*-79.

Howe, J. 2009. Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business. *Journal of Consumer Marketing*, v. 26, n. 4, p. 305-306.

Howe, J. 2006. The Rise of Crowdsourcing. WIRED Magazine. Disponível: http://www.wired.com/wired/archive/14.06/crowds.html.[Acesso 25 Agosto 2015].

Huang, H. D., Chuang, T. Y., Tsai, Y. L., & Lee, C. S. 2010. Ontology-based intelligent system for malware behavioral analysis. *IEEE In Fuzzy Systems (FUZZ), 2, International Conference on* (pp. 1-6). pp 1-6.

Huang, H. D., Acampora, G., Loia, V., Lee, C. S., Hagras, H., Wang, M. H., & Chang, J. G. 2013. Fuzzy markup language for malware behavioral analysis. In *On the Power of Fuzzy Markup Language*. Springer Berlin Heidelberg, pp 113-132.

Hu, J., Gao, H., Li, Z., & Chen, Y. 2011. Poster: CUD: crowdsourcing for URL spam detection. In *Proceedings of the 18th ACM conference on Computer and communications security*, pp 785-788.

Hwang, H. S., Cui, Y., Kim, C. S., & Suh, Y. C. 2008. The Inference Search Process for Managing Control Points Using OWL and SWRL. In *Convergence and Hybrid Information Technology, 2008. ICCIT'08. IEEE Third International Conference on*, Vol. 2, pp. 802-807.

Jasiul, B., Sliwa, J., Gleba, K., & Szpyrka, M. 2014. Identification of malware activities with rules. In *Computer Science and Information Systems (FedCSIS), IEEE 2014 Federated Conference on*, pp 101-110.

Kanich, C., Checkoway, S., & Mowery, K. 2011. Putting Out a HIT: Crowdsourcing Malware Installs. In *WOOT* , pp. 71-80.

Kargl, F., Maier, J., & Weber, M. 2001. Protecting web servers from distributed denial of service attacks. *ACM In Proceedings of the 10th international conferenceon World Wide Web* (pp. 514-524).

Kiesling, E., Ekelhart, A., Grill, B., Stummer, C., & Strauss, C, 2014. Evolving secure information systems through attack simulation. In *System Sciences (HICSS), IEEE 2014 47th Hawaii International Conference on* (pp. 4868-4877).

Kropiwiec, D. D., & de Geus, P. L. (2004, May). Paradigmas de segurança em sistemas operacionais. *In Anais do IV Workshop de Segurança em Sistemas Computacionais*.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lin, H., & Davis, J. 2010. Computational and crowdsourcing methods for extracting ontological structure from folksonomy. In *The Semantic Web: Research and Applications*, Springer Berlin Heidelberg, pp 472-477.

Lin, H., Davis, J., & Zhou, Y. 2010. Ontological services using crowdsourcing. In *21st Australasian Conference on Information Systems*.

Lopes, M. 2012. Evolução das Redes Sociais: conheça melhor o crowdfunding, crowdsourcing e crowdturfing. Disponivel: http://www.marcelolopes.jor.br/noticia/detalhe/10192/evolucao-das-redes-sociais-conheca-melhor-o-crowdfunding-crowdsourcing-e-crowdturfing. [Acesso 15 Fevereiro 2017].

Lukyanenko, R., & Parsons, J. 2012. Conceptual modeling principles for crowdsourcing. *ACM In Proceedings of the 1st international workshop on Multimodal crowd sensing*, pp 3-6.

Maia, M. A. 2013. O que é Segurança da Informação. Disponível: http://segurancadainformacao.modulo.com.br/seguranca-da-informacao. [Acesso: 26 Março 2015].

Maizero, C. A. 2014. Sistemas Operacionais: Conceitos e Mecanismos. 1°. ed. Curitiba 360p.

Marciano, J. L. P. 2006. Segurança da Informação - uma abordagem social. 2006. 2012f. *Tese de Doutorado* - Universidade de Brasília, Brasília, Brasil.

Martins, G. B., Souto, E., Freitas, R., & Feitosa, E. 2014. Estruturas Virtuais e Diferenciação de Vértices em Grafos de Dependência para Detecção de Malware Metamórfico. In *XIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg14), Belo Horizonte, MG, Brasil, pp.* 260-273.

Masood, R., & Anwar, Z., 2011.SWAM: Stuxnet Worm Analysis in Metasploit. *IEEE In Frontiers of Information Technology (FIT)*, pp 142-147.

Michelin, R. A. 2015. Mitigação de Ataques de Negação de Serviços em Redes Autenticáveis na Nuvem. 89f. *Dissertação de Mestrado* - Pontifícia Universidade Católica do Rio Grande do Sul, Brasil.

Mitre, 2010. Science of Cyber-Security**,** JASON *The MITRE Corporation*.

Morris, C. W. (1938). Foundations of the Theory of Signs, International Encyclopedia of Unified Science. *University of Chicago Press*.

Mortensen, J. M., Musen, M. A., & Noy, N. F. 2013.Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, p 1020.

Mortensen, J. M., Musen, M. A., & Noy, N. F. 2013. Ontology quality assurance with the crowd. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Mundie, D., & Mcintire, D. M. 2013. An Ontology for Malware Analysis. In *Availability, Reliability and Security (ARES), IEEE 2013 Eighth International Conference on*, pp 556-558.

Noor, M., & Abbas, H. 2013. Anticipating Dormant Functionality in Malware: A Semantics Based Approach. In *Biometrics and Security Technologies (ISBAST), IEEE 2013 International Symposium on*, pp 20-23.

Pereira, P. J. F. 2005. Segurança da Informação Digital. Cadernos BAD, São Paulo, n.1, p.2. Disponível: http://eprints.rclis.org/10305/1/CadBAD105_Pereira.pdf. [Acesso26 Março 2015].

Pinheiro, J. M. S. 2004. Redes de Perímetro. Projeto de Redes,. Disponível: http://www.projetoderedes.com.br/artigos/artigo_redes_de_perimetro.php. [Acesso 25 Março 2015].

Porto, M. 2012. O que é Crowdturfing, a nova forma de spam nas redes sociais. Disponivel: http://www.techtudo.com.br/artigos/noticia/2012/05/o-que-e-crowdturfing-nova-forma-de-spam-nas-redes-sociais.html. [Acesso 15 Fevereiro 2017].

Saxe, J., Turner, R., & Blokhin, K. 2014. CrowdSource: Automated inference of high level malware functionality from low-level symbols using a crowd trained machine learning model. In *Malicious and Unwanted Software: The Americas (MALWARE), IEEE 2014 9th International Conference on*, pp 68-75.

Security-faqs, Malware That Changed the World – The Melissa Virus, 2010. Disponível: http://www.security-faqs.com/malware-that-changed-the-world-the-melissa-virus.html#comments. [Acesso14 Julho 2015].

Simião, R. S. 2009. Segurança da Informação e Comunicações: conceito. 81f. *Monografia (Especialização em Ciência da Computação)* - Universidade de Brasília, Brasilia, Brasil.

Shoaib, M., & Farooq, M. 2015. USpam--A User Centric Ontology Driven Spam Detection System. In *System Sciences (HICSS), IEEE 2015 48th Hawaii International Conference on*, pp 3661-3669.

Tecmundo, 2010. Stuxnet: o vírus da pesada, disponível: https://www.tecmundo.com.br/virus/5878-stuxnet-o-virus-da-pesada.htm. [Acesso: 15 Julho de 2015].

Tanenbaum, A. S. 2010. Sistemas Operacionais Modernos. 3°. ed. São Paulo: Pearson, 2010. pp 653.

The Agency, 2000, Information Security and Privacy Training for Information System Security Officers. Disponível: http://www.iwar.org.uk/comsec/resources/fasp/ISSO-participant-book.doc. [Acesso 14 de Março 2014].

Thuan, N. H., Antunes, P., Johnstone, D., & Truong, M. N. Q. 2014. An Architecture Utilizing the Crowd for Building an Anti-virus Knowledge Base. In *Future Data and Security Engineering*, Springer International Publishing, pp 164-176.

W3c, 2012, OWL 2 *Web* Ontology Language, Document Overview (Second Edition), W3C Recommendation 11 December 2012. Disponível: http://www.w3.org/TR/owl2-overview/. [Acesso 12Agosto 2015].

Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., & Zhao, B. Y. 2012. Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856*.

Wang, J., & He, K. 2006. Towards representing FCA-based ontologies in semantic web rule language. *IEEE In Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on*, pp. 41-41.

Wu, L., Ping, R., Ke, L., & Hai-xin, D. 2011. Behavior-based malware analysis and detection. In *Complexity and Data Mining (IWCDM), IEEE 2011 First International Workshop on*, pp 39-42.

Yao, J., Wei, M. 2014.A New Bionic Architecture of Information System Security Based on Data Envelopment Analysis. *International Conference on Management of e-Commerce and e-Government*, Shanghai, n.14934192, pp. 93 – 97.

Yuen, M. C., King, I., & Leung, K. S. 2011.A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 766-773.

You, I., & Yim, K. 2010. Obfuscation Techniques: A Brief Survey. *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, Fukuoka, n.11642071, pp.297-3.

Apêndice I – Artigo Publicado

Grégio, A., Bonacin, R., de Marchi, A. C., Nabuco, O. F., & de Geus, P. L. (2016). An ontology of suspicious software behavior. *Applied Ontology*, pp 29-49. (Qualis B1, Ciência da Computação, 2015).

Abstract – Malicious programs have been the main actors in complex, sophisticated attacks against nations, governments, diplomatic agencies, private institutions and people. Knowledge about malicious program behavior forms the basis for constructing more secure information systems. In this article, we introduce MBO, a Malicious Behavior Ontology that represents complex behaviors of suspicious executions, and through inference rules calculates their associated threat level for analytical proposals. We evaluate MBO using over two thousand unique known malware and 385 unique known benign software. Results highlight the representativeness of the MBO for expressing typical malicious activities.

Keywords: Security ontology, Malware behavior, Threat analysis

Apêndice II – Artigo Aceito.

(Previsão de Publicação Junho de 2017).

de Marchi, Grégio, A., Bonacin, R. (2017). Enhancing the Creation of Detection Rules for Malicious Software through Ontologies and Crowdsourcing. (WETICE 2017), *Polônia.* (Qualis B1, Ciência da Computação, 2012).

Abstract – The analysis of malicious software (malware) is one of the hardest open problems in computer security, since there is a huge and varied number of samples produced daily. In addition, modern malicious programs have automatic mutation capabilities. Through behavior analysis of existing malware, we are able to understand new variants and develop new protection methods. Ontologies can be used to model those behaviors, enabling experts to define classes and rules that represent complex behaviors. In this paper, we used an ontology and architecture built during our previous studies as a starting point to inspire the development of a crowdsource-based framework and platform. The objective of this work is to explore crowdsourcing mechanisms to collaboratively evolve ontologies, in which users can propose new classes and rules that increasingly identify potential malicious programs. With a user-friendly platform, we expect to leverage a model that could be used in other malware analysis systems, as well as to quickly respond to new malware variants. Eight domain experts evaluated this platform with the goal of validating and identifying the platforms potentials and limitations.

Keywords - Crowdsourcing, Malware; Ontology; Computer Security