PROPOSTA DE UMA CLASSIFICAÇÃO DE AGRUPAMENTO DINÂMICA BASEADA EM ALERTAS DE UM IDS SNORT

Carlos Alexandre Carvalho Tojeiro[1]; Eduardo Alves Moraes[2]; Thiago José Lucas[3]

Resumo
As Redes de computadores estão incorporando cada vez mais recursos, onde administradores necessitam verificar milhares de alertas gerados por um IDS. Desta forma, surge a necessidade de ferramentas que possam classificar e agrupar alertas de forma dinâmica para que apoiem os administradores a identificar ataques futuros. Frente às necessidades surge a proposta de criar uma classificação-modelo dinâmica que possa representar o conhecimento de segurança de redes de computadores dos sistemas de detecção de intrusão. Neste sentido o trabalho tem como objetivo, analisar a estrutura de dados dos alertas, modelar o conhecimento do IDS em forma de um modelo de classificação, utilizando-se da ferramenta Protégé para mapeamento dinâmico entre o MySQL e SPARQL.
Palavras-chave: IDS, Agrupamentos de Alertas, Clustering, Ameaças, Vulnerabilidades.

Abstratc
Computer networks are incorporating more and more resources, where admins need to ceck th ousands of alerts generated by na IDS. In this way, there is a need for tools that can dynamically classify and group alerts to support administrators in identifying future attacks. In view of the needs, the proposal proposes to create a dynamic model classification that can represent the knowledge of security of computer networks of intrusion detection systems. In this sense, the objective of this work is to analyze the data structure of the alerts, to model the knowledge of the IDS in the form of a classification model, using the Protégé tool for dynamic mapping between MySQL and SPARQL.
Keywords: IDS, Alert Groups, Clustering, Threats, Vulnerabilities.

Introdução

As redes de computadores, dissipam cada vez mais recursos e serviços. Devido a este aumento, necessita-se de aplicações dinâmicas que possam auxiliar os administradores em ações mais eficazes em relações a ataques de intrusão. Os diversos dispositivos de redes existentes, como switches, roteadores, etc., contêm novos aplicativos que demandam milhares de alertas. E dentro deste quadro, os ataques podem ser identificados a partir da classificação de grupos de alertas.

O trabalho de Silva e Rafael (2017), destaca a importância de se criar uma classificação de agrupamento de alertas que abranja todos os aspectos da área de segurança de redes. Deve-

[1] Especialista em Segurança de Redes de Computadores pela Faculdade de Tecnologia de Ourinhos -FATEC. E-mail: carlos.tojeiro@fatecourinhos.edu.br.
[2] Mestre em Ciências da Computação pela Universidade Estadual de Londrina-UEL; professor do curso de Segurança da Informação da Faculdade de Tecnologias de Ourinhos– FATEC. E-mail: eduardo.moraes@fatecourinhos.edu.br.
[3] Mestrando em Ciência da Computação (Computação Aplicada/Inteligência Computacional) na Universidade Estadual Paulista Júlio de Mesquita Filho-Unesp Bauru; professor do curso de Segurança da Informação da Faculdade de Tecnologias de Ourinhos– FATEC. E-mail: thiago.lucas@fatecourinhos.edu.br

se observar, entretanto, se a base de alertas e o conhecimento humano prévio permitem a criação de um agrupamento de alertas que envolva todos os aspectos.

Gyrard *et al.* (2013), apresentou uma proposta de agrupamento de alertas com foco em segurança de aplicações Web, qual descreve para cada possível ataque uma sequência de alertas.

De acordo com Xu *et al.* (2009), existe a necessidade de uma nova solução para a segurança de redes de computadores. Os autores afirmam que classificações de agrupamentos de alertas podem fornecer uma abordagem prática e eficiente para a representação formal do conhecimento humano na correta mitigação de eventuais ataques.

Neste mesmo pensamento Undercoffer *et al.* (2003), apresentam em seu trabalho um modelo que além de criar um agrupamento de alertas, permite que diversos IDS compartilhem conhecimento sobre como agir em relação a determinados ataques.

Portanto, uma vez realizada a classificação, agentes inteligentes que representam sistemas de segurança podem se comunicar sobre qual seria a atitude ideal frente a um ataque desconhecido.

O problema observado da modelagem deste conhecimento (humano), com base nos alertas de uma rede em tempo real e sem a necessidade de remodelação do Sistema de Detecção de Intrusão, pode servir de base para a geração futura de agentes inteligentes, capazes de detectar intrusões correlacionando classes a partir de um agrupamento de alertas?

O objetivo geral deste trabalho foi criar uma classificação-modelo dinâmica que refletisse o conhecimento de segurança de redes de computadores dos sistemas de detecção de intrusão da Faculdade de Tecnologia de Ourinhos.

Já os objetivos específicos baseiam-se em prover base teórica fundamental acerca do tema proposto, analisar a estrutura de dados dos alertas, modelar o conhecimento do IDS em forma de um modelo de classificação utilizando-se da ferramenta Protégé para mapeamento dinâmico entre o MySQL e SPARQL.

1 Métodos de Classificação em Trabalhos Correlatos

Ataques a redes de computadores são modelados como padrões de eventos específicos e ao momento que ocorrem, sendo assim, padrões são observados para que se possa identificar ataques baseados em um conjunto de classes que apontam uma sequência que podem ser baseados em vários métodos.

Um importante levantamento foi feito por Silva e Rafael (2017), para que fosse possível conhecer os principais trabalhos no desenvolvimento de classificações de agrupamentos de

alertas que envolve Segurança de Redes de Computadores. Os autores categorizaram as pesquisas de acordo com o método principal empregado, os quais podem ser:

1. Foco em Ameaças;
2. Foco em Sistemas de Detecção de Intrusão;
3. Foco em Alertas;
4. Foco em Ataques;
5. Foco em Vulnerabilidades;
6. Foco em Contramedidas;
7. Foco em Políticas de Segurança;
8. Foco em Gerenciamento de Redes.

Utilizando simulação de ataques em redes de computadores, Undercoffer *et al*. (2003) propuseram um método de compartilhamento de conhecimento de ataques entre Sistemas de Detecção de Intrusão.

Já o método de classificação aplicado por Pinkston *et al.* (2004), baseou-se de quatro mil classes de ataques relacionando o atributo de cada uma.

O Snort IDS gera milhares de alertas em um tempo pequeno de monitoramento, tornando-se difícil saber quais alertas podem conduzir a graves ataques no futuro porém, o agrupamento de alertas e dados juntamente com ferramentas de mineração torna-se possível encontrar estágios que podem ser considerados como chave para uma descoberta de novos ataques (YANG, 2010).

Outro importante resultado foi alcançado pela implementação de Razzaq *et al.* (2009), que propuseram um sistema de detecção inteligente de ataques. A proposta dos autores foi capaz de detectar ataques de forma semântica, gerando uma relevante taxa de detecções positivas.

Também Simmons (2014), teve como proposta uma classificação que relaciona diversos conceitos das áreas de Ataques e Contramedidas, de forma a sugerir as soluções mais adequadas para cada ataque detectado.

Já a análise de Gao (2013), classificou de cada detecção em cinco dimensões: (1) alvo, (2) vulnerabilidade, (3) assinatura, (4) impacto e (5) contramedida. O alvo define o host vítima do ataque, geralmente caracterizado pelo seu endereço IP; A vulnerabilidade está relacionada ao CVE (*Common Vulnerabilities and Exposures*), um código padrão para informar uma brecha que pode ser explorada por um atacante, seja a nível de software ou de hardware); A assinatura define detalhes do ataque e serve de base para a categorização no Sistema de Detecção de Intrusão. O impacto é categorizado de forma a documentar o prejuízo que determinado ataque

pode causar e em contramedida define qual a melhor estratégia para se defender de um ataque em específico.

Symmon (2014), também propôs um modelo genérico de classificação que pode ser usado como base para a criação de Sistemas de Detecção de Intrusão contra ataques cibernéticos.

Da mesma forma, Razzaq (2014), com foco em segurança para aplicações Web, demonstraram que sistemas de segurança baseados em classificação de alertas (seja no projeto ou na avaliação) entregam uma importante contribuição na proteção dos perímetros de rede com estas características.

O trabalho de Karande (2015), documenta um modelo de Sistema de Detecção de Intrusão baseado em classificação de alertas com foco na detecção do método de ataque, mais especificamente para detecção de scripts maliciosos.

Wang e Guo (2009), utilizam-se de um cruzamento entre os dados de vulnerabilidades e os dados de impactos conhecidos (o que cria o conceito de risco) e propuseram uma classificação para segurança de redes de computadores baseada na definição dos riscos (têm-se que risco = vulnerabilidade x impacto).

Com foco específico, o trabalho de Elahi (2009) concentrou esforços na análise das vulnerabilidades. Como objetivo, aplicou-se o conhecimento acerca das vulnerabilidades existentes em uma rede de computadores como base na criação da classificação dos alertas.

Igualmente Wang (2010), direcionou o foco de seu trabalho também para vulnerabilidades, porém durante a fase de desenvolvimento de software. A classificação proposta aplica lógica semântica como base para o desenvolvimento seguro de sistemas.

Já com o objetivo de realizar uma auditoria e extrair o estado atual de segurança de uma rede de computadores, Bhandari (2014) propuseram uma classificação que relaciona vulnerabilidades a ataques, com base na extração de vulnerabilidades de uma rede de computadores, a abordagem permite criar um modelo de ataques que podem ser realizados naquele perímetro auditado.

A abordagem proposta por Do Amaral (2006) tem foco menos técnico. Com base em processamento de linguagem natural, o método de classificação de agrupamento consiste em avaliação das políticas de segurança da informação de uma determinada instituição para que seja possível modelar o conhecimento acerca daquele perímetro, não com base em eventos reais, mas baseado em um texto de uma política, que normalmente é escrito de forma personalizada.

Li e Tiam (2010) desenvolveram um sistema que correlaciona dados obtidos de diferentes Sistemas de Detecção de Intrusão. Esta proposta, bem como a maioria das citadas, modela o conhecimento na ferramenta Protégé.

A Figura 1 ilustra a relação de métodos usados como base na criação das classificações de agrupamento de alertas em relação ao levantamento realizado por Silva (2017).

Figura 1 - Relação de métodos usados como base na criação das classificações de alertas

Fonte: Elaborado pelos autores (2020).

Desta forma observa-se que existem vários métodos de classificações de agrupamentos de alertas providos de ataques detectados por IDS para prever futuros ataques em uma rede. Porem nenhuma com a proposta de agrupamento de alertas de forma dinâmica baseada em um IDS Snort aplicada a uma base real foi encontrada.

2 Materiais, Métodos e Resultados

Neste capítulo pretende-se abordar de forma objetiva, os materiais e métodos utilizados, seja no projeto, na implementação e nos testes. Os resultados obtidos durante a implementação também podem ser observados neste tópico.

2.1 Snort

Este tópico pretende descrever de forma simples e objetiva as definições de Snort mais importantes para este trabalho.

O Snort é um software livre e de código-fonte aberto que implementa um Sistema de Detecção de Intrusão completo. Como o objetivo deste relatório é descrever o processo de implementação do modelo e os testes, optou-se por não se aprofundar em alguns detalhes.

O IDS Snort em execução na rede da Faculdade de Tecnologia de Ourinhos fora, quando da época de detecção dos ataques, implementado pela equipe de Tecnologia da Informação da instituição, da qual faziam parte os autores desde artigo.

É importante salientar que um pré-processamento fora realizado na base de dados de alertas do Snort de forma com que fosse possível eliminar registros falso-positivos (ataques que não eram reais). Nesta ocasião, usou-se de um algoritmo de classificação de métrica kNN que foi treinado para que pudesse apontar quais alarmes provavelmente poderiam ser excluídos da base de dados sem prejuízo à integridade da realidade.

2.2 MySQL

Este tópico pretende descrever de forma simples e objetiva o relacionamento do SGBD MySQL no âmbito deste trabalho.

O Snort nativamente armazena seus registros em tabelas de um SGDB MySQL. Para maior riqueza de detalhes, alguns campos foram adicionados pelos autores de forma com que maiores detalhes pudessem ser observados no momento de classificação. Para maior organização, os registros foram divididos em quatro tabelas (seguindo relativamente os nomes das classes):

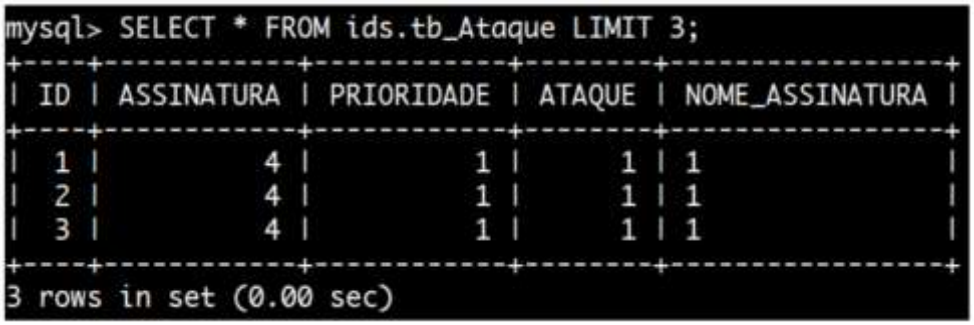
1. tb_Ataque
2. tb_Calendario
3. tb_Localidade
4. tb_Rede

2.2.1 Detalhes da tabela tb_Ataque

A tabela tb_Ataque possui informações como: (1) ID do ataque; (2) Número da assinatura geral; (3) Prioridade do ataque; (4) Ataque; e (5) Nome da Assinatura.

Em (1) observa-se a chave primária da tabela; em (2) o número da assinatura geral que descreve de forma genérica o tipo de ataque; em (3) a prioridade do ataque, onde o valor 1 representa alta prioridade e 2 baixa prioridade; em (4) um valor booleano que descreve se o ataque é real ou falso positivo, assumindo-se que o valor 1 indica ataques reais de acordo com o algoritmo kNN e em (5) um detalhe da assinatura genérica que descreve o tipo específico de ataque detectado.

Figura 2 - Exemplo de dados na tabela tb_Ataque (com LIMIT 3)



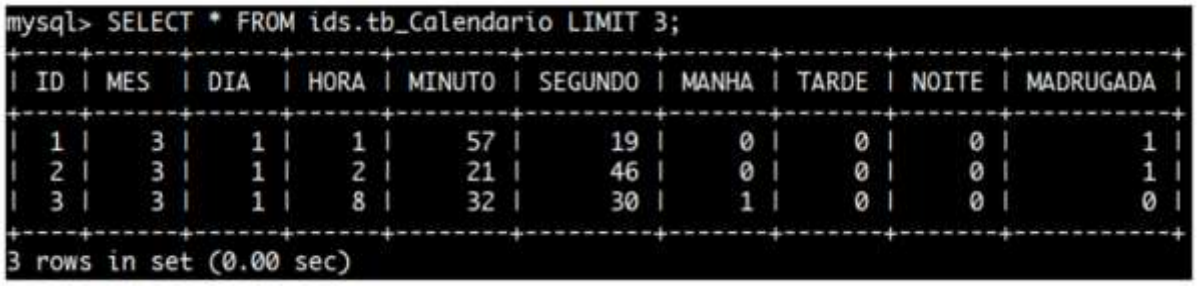
```
mysql> SELECT * FROM ids.tb_Ataque LIMIT 3;
+----+------------+------------+--------+-----------------+
| ID | ASSINATURA | PRIORIDADE | ATAQUE | NOME_ASSINATURA |
+----+------------+------------+--------+-----------------+
|  1 |          4 |          1 |      1 | 1               |
|  2 |          4 |          1 |      1 | 1               |
|  3 |          4 |          1 |      1 | 1               |
+----+------------+------------+--------+-----------------+
3 rows in set (0.00 sec)
```

Fonte: Elaborado pelos autores (2020).

2.2.2 Detalhes da tabela tb_Calendario

A tabela tb_Calendario possui nos nomes de suas tuplas as devidas explicações das existências e dos valores de seus respectivos campos:

Figura 3 - Exemplo de dados na tabela tb_Ataque (com LIMIT 3)



```
mysql> SELECT * FROM ids.tb_Calendario LIMIT 3;
+----+------+------+------+--------+---------+-------+-------+-------+-----------+
| ID | MES  | DIA  | HORA | MINUTO | SEGUNDO | MANHA | TARDE | NOITE | MADRUGADA |
+----+------+------+------+--------+---------+-------+-------+-------+-----------+
|  1 |    3 |    1 |    1 |     57 |      19 |     0 |     0 |     0 |         1 |
|  2 |    3 |    1 |    2 |     21 |      46 |     0 |     0 |     0 |         1 |
|  3 |    3 |    1 |    8 |     32 |      30 |     1 |     0 |     0 |         0 |
+----+------+------+------+--------+---------+-------+-------+-------+-----------+
3 rows in set (0.00 sec)
```

Fonte: Elaborado pelos autores (2020).

2.2.3 Detalhes da tabela tb_Rede

A tabela tb_Rede pode ser analisada da seguinte forma: em (1) tem-se o ID que é a chave primária da tabela; em (2) a porta de origem da camada de transporte do modelo OSI utilizada no ataque, geralmente TCP ou UDP; em (3) a porta de destino da camada de transporte do modelo OSI que permite observar a aplicação objeto do ataque (na Figura 4 observam-se portas 443 que remetem a um servidor web funcionando sob HTTPs); em (4) observa-se o primeiro octeto do endereço IPv4 do atacante; em (5) observa-se o segundo octeto do endereço IPv4 do atacante; em (6) observa-se o terceiro octeto do endereço IPv4 do atacante; em (7) observa-se o primeiro octeto do endereço IPv4 do alvo; em (8) observa-se o segundo octeto do endereço IPv4 do alvo; e em (9) o terceiro octeto do endereço IPv4 do alvo. Esses octetos, tanto do atacante quanto do alvo, permitem a localização geográfica do host.

Figura 4 - Exemplo de dados na tabela tb_Rede (com LIMIT 3)

```
mysql> SELECT * FROM ids.tb_Rede LIMIT 3;
+----+-------+-------+-----------+-----------+-----------+-----------+-----------+-----------+
| ID | P_SRC | P_DST | PART1_SRC | PART2_SRC | PART3_SRC | PART1_DST | PART2_DST | PART3_DST |
+----+-------+-------+-----------+-----------+-----------+-----------+-----------+-----------+
|  1 | 50643 |   443 |       177 |        45 |       192 |       201 |        55 |         1 |
|  2 | 61129 |   443 |       177 |        45 |       192 |       201 |        55 |         1 |
|  3 | 30865 |   443 |       186 |       219 |        96 |       201 |        55 |         1 |
+----+-------+-------+-----------+-----------+-----------+-----------+-----------+-----------+
3 rows in set (0.00 sec)
```

Fonte: Elaborado pelos autores (2020).

2.2.4 Detalhes dos totais de alertas

A Figura 5 ilustra a documentação da quantidade de alertas obtidos durante três meses de coletas na rede de computadores da Faculdade de Tecnologia de Ourinhos:

Figura 5 - Quantidade de dados existentes obtidos pelo IDS

```
mysql> SELECT COUNT(*) FROM ids.tb_Total WHERE ATAQUE = 1;
+----------+
| COUNT(*) |
+----------+
|    74586 |
+----------+
1 row in set (0.10 sec)
```

Fonte: Elaborado pelos autores (2020).

2.3 Protégé

Este tópico pretende descrever de forma simples e objetiva a definição de Protégé e a classificação de agrupamento modelo criada.

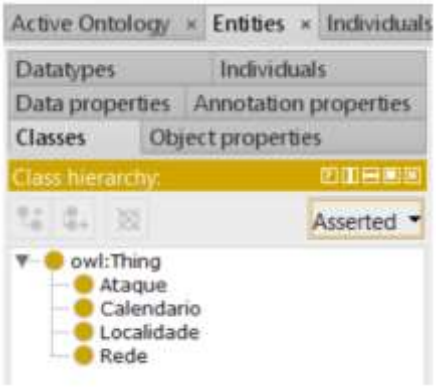
Como ferramenta utilizada na modelagem do conhecimento humano (modelo de classificação agrupamento) foi utilizado o software Protégé criado pelo Centro de Informática Biomédica da Universidade de Stanford.

De acordo com Musen (2015), o Protégé tornou-se o software mais utilizado para a criação e manutenção de classificação de agrupamento de alertas, embora não seja o único.

2.3.1 Classes

De acordo com a organização das tabelas do MySQL, criaram-se quatro classes no Protégé e para cada classe foram definidas propriedades que refletem tuplas no banco de dados. A Figura 6 ilustra a estrutura de classes criada:

Figura 6 - Relação de classes criadas na ferramenta Protégé



Fonte: Elaborado pelos autores (2020).

2.3.2 Data properties

Na Figura 7 é possível observar as propriedades que refletem as tuplas:

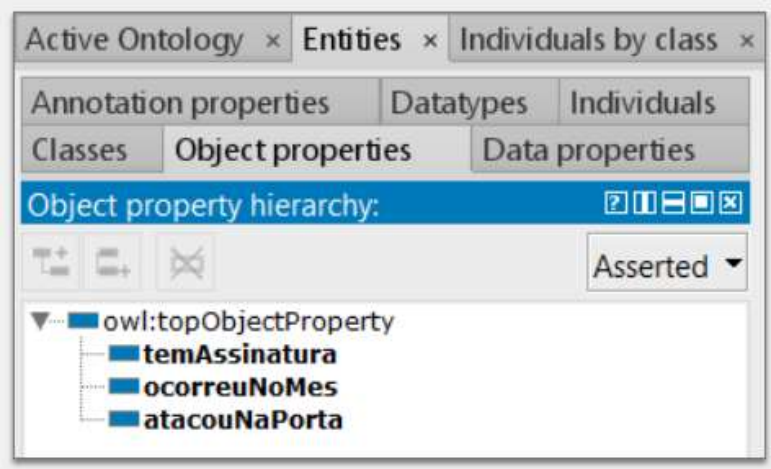
Figura 7 - Relação de data properties

Fonte: Elaborado pelos autores (2020).

2.3.2 Object Properties

Na Figura 8 é possível observar as propriedades dos objetos que serão utilizadas para criação de expressões que relacionam Classes a Data Properties:

Figura 8 - Relação de Object Properties



Fonte: Elaborado pelos autores (2020).

As relações foram criadas de forma a relacionar:

1. O tipo de ataque (classe) ao seu número de assinatura (data property ataqueAssinatura);
2. O mês de ocorrência do ataque (classe) ao seu número correspondente (data property calendarioMes);
3. A aplicação ou o protocolo de aplicação atacado (classe) ao seu número de porta correspondente (data property redePortaDestino);

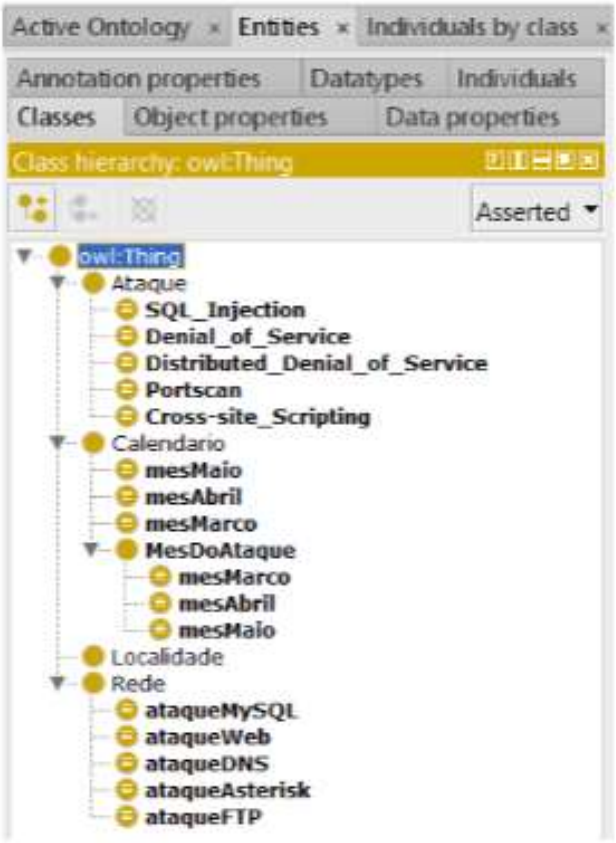
A Figura 9 ilustra alguns relacionamentos (ataqueWeb, mesAbril e SQL_Injection):

Figura 9 - Relação de alguns descriptions criadas pela ferramenta Protégé

Fonte: Elaborado pelos autores (2020).

Após a especificação de algumas relações entre Classes, Object Properties e Data Properties, o ambiente de trabalho ficou como ilustrado na Figura 10:

Figura 10 - Classes após especificação das relações



Fonte: Elaborado pelos autores (2020).

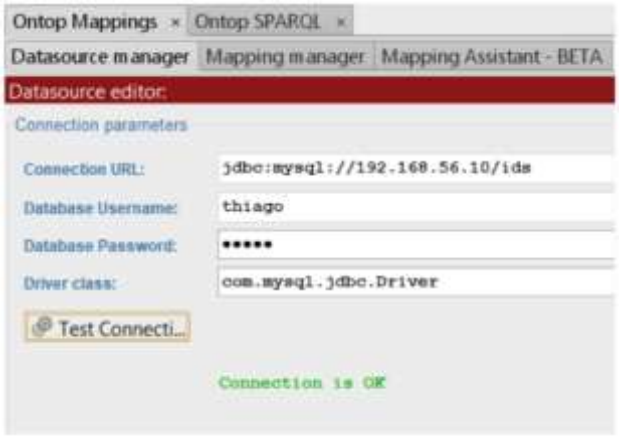
2.4 OBDA

Este tópico pretende descrever de forma simples e objetiva a definição de OBDA.

Por definição, OBDA consiste em acessar informações por meio de três camadas: classificação de agrupamento, mapeamento e fonte de dados. Nesta implementação têm-se como classificação de agrupamento o modelo criado no Protégé, como mapeamento o Ontop (plugin do Protégé) e como fonte de dados o MySQL. Em resumo o usuário formula consultas SPARQL (abordado adiante) que são traduzidas para SQL.

Figura 11 ilustra a configuração para que o Protégé se conecte à uma base de dados externa (192.168.56.10) utilizando driver específico JDBC.

Figura 11 - Configuração para busca em SGBD externo à rede



Fonte: Elaborado pelos autores (2020).

Adiante serão apresentados dados detalhes de como o mapeamento foi realizado (SPARQL/SQL).

2.5 RDF e SPARQL

Este tópico pretende descrever de forma simples e objetiva as definições de RDF e SPARQL.

Resource Description Framework (RDF) pode ser considerado um protocolo para representação de dados. Trata-se de um método para a modelagem da informação para recursos Web com o objetivo de prover de forma simplificada acesso a dados com uma semântica formal, usando vocabulários baseados em URIs e sintaxes baseadas em XML.

SPARQL Protocol and RDF Query Language (SPARQL) é uma linguagem de recuperação de dados em bases RDF. Como os dados coletados pelo Snort estão armazenados em banco de dados relacional, essa linguagem não conseguiria extrair os dados de forma correta, pois a tipagem adequada para este cenário é a linguagem SQL. Neste contexto surge a necessidade de um ODBA (Ontop) para realizar o mapeamento entre SPARQL e SQL.

De forma a automatizar os processos de consulta na base de dados para que a classificação de agrupamento pudesse ser atualizada em tempo real, os autores deste relatório desenvolvolveram uma rotina em bash (linguagem escolhida por se tratar de um Sistema de Detecção de Intrusão que está em execução em uma distribuição Linux) capaz de fornecer os seguintes dados para o mapeamento e as consultas RDF:

1. Target -- sintaxe de como as triplas serão organizadas (recurso, propriedade e valor);
2. Source -- sintaxe em SQL para busca no banco de dados MySQL;
3. SPARQL -- sintaxe em SPARQL para busca em RDF.

A rotina necessita de um arquivo de configuração no mesmo diretório do script em bash com o nome snortMySQL2snortSPARQL.conf. Esse arquivo de configuração deve possuir a sintaxe ilustrada na Figura 12:

Figura 12 - SnortMySQL2snortSPARQL.conf

```
# Arquivo de configuracao usado pelo snortMySQL2snortSPARQL.sh
#
# Organizacao:
# Coluna 1 - atributo criado na ontologia
# Coluna 2 - Coluna de valor no MySQL
# Coluna 3 - Tabela no MySQL
# Coluna 4 - Variavel a ser exibida no Protege
# Caractere separador: ":"

# Exemplo:
redePortaDestino:P_DST:tb_Rede:porta_dst
```

Fonte: Elaborado pelos autores (2020).

Quando a rotina é executada a saída padrão é prover para o usuário as informações para que o mapeamento seja criado no Protégé, conforme ilustra a Figura 13.

Figura 13 - Execução do script snortMySQL2snortSPARQL.sh

```
****** CLASSE :Ataque & ATRIBUTO :redePortaDestino ******
[TARGET]
:ataque/{ID} a :Ataque ; :redePortaDestino {P_DST} .

[SOURCE]
SELECT ID, P_DST FROM tb_Rede;

[SPARQL]
PREFIX : <http://www.semanticweb.org/thiag/ontologies/2018/10/trabfinal#>

SELECT * WHERE {
  ?ataque a :Ataque .
  ?ataque   :redePortaDestino ?porta_dst
}

/!\ NAO SE ESQUECA DE REINICAR O REASONER NO PROTEGE! /!\
```

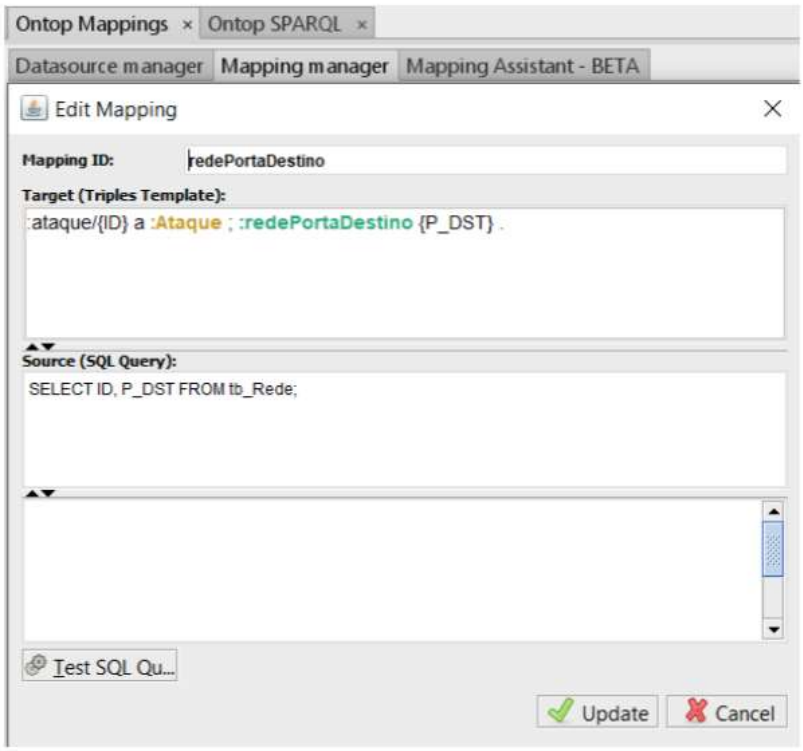
Fonte: Elaborado pelos autores (2020).

A saída da seção [TARGET] pode ser interpretada da seguinte maneira:

1. Ataque/{ID} – Recurso
2. Ataque – Propriedade
3. RedePortaDestino {P_DST} – Valor

A partir deste ponto, basta adicionar as informações providas pelo script na \textit{Ontop Mappings} do Protégé, conforme ilustra a Figura 14.

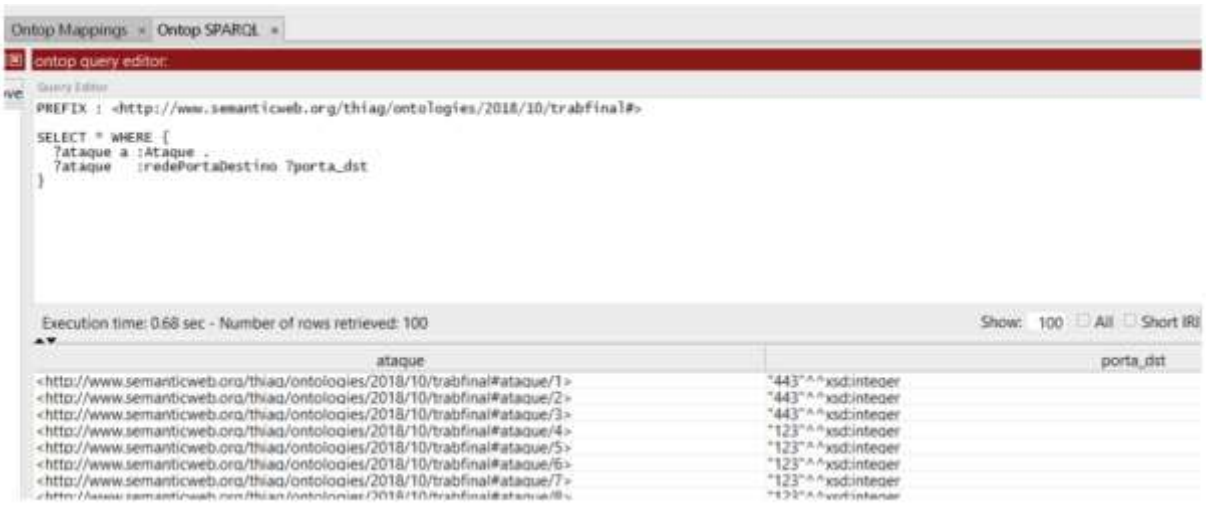
Figura 14 - Informações do script adicionadas ao Protégé



Fonte: Elaborado pelos autores (2020).

O script snortMySQL2snortSPARQL.sh também fornece a consulta SPARQL adequada para recuperar os dados de exemplo. Esta consulta pode ser realizada na seção Ontop SPARQL do Protégé, conforme ilustra a Figura 15.

Figura 15 - Consulta SPARQL realizada no Protégé



Fonte: Elaborado pelos autores (2020).

3 Conclusões e Trabalhos Futuros

A utilização do Protégé como ferramenta de modelação do conhecimento humano, conforme já observado anteriormente por Musen (2015) é um importante passo na construção de modelo classificações de agrupamento de alertas. Este trabalho procurou criar uma base de referência no Protégé para cenários de Sistemas de Detecção de Intrusão usando Snort de forma dinâmica, permitindo que dados adicionados recentemente sejam mapeados por meio do plugin Ontop sendo alimentado com informações geradas pela rotina snortMySQL2snortSPARQL.sh, de autoria própria. Como sugestão para futuros trabalhos, recomenda-se implementar mapeamentos para todos os dados contidos nas tabelas do Snort de forma a gerar uma base RDF completa e que reflita o cenário de segurança de uma rede de computadores como um todo, além de permitir criar maiores relacionamentos entre as classes e as propriedades.

Referências

AN WANG, J., GUO, M. M., CAMARGO, J. **An ontological approach to computer system security**. Information Security Journal: A Global Perspective, 19(2):61–73, 2010.
BHANDARI, P., GUJARAL, M. S. **Ontology based approach for perception of network security state**. In Engineering and Computational Sciences (RAECS), Recent Advances in, pages 1–6. IEEE, 2014.

DO AMARAL, F. N., BAZILIO, C., DA SILVA, G. M. H., RADEMARKER, A., HAEUSLER, E. H.,. **An ontology-based approach to the formalization of information security policies.** In EDOC Workshops, page 1, 2006.

ELAHI, G., YU, E., ZANNONE, N. **A modeling ontology for integrating vulnerabilities into security requirements conceptual foundations.** In International Conference on Conceptual Modeling, pages 99–114. Springer, 2009.

GAO, J. B., ZHANG, B. W., CHEN, X. H., LUO, Z. **Ontology-based model of network and computer attacks for security assessment.** Journal of Shanghai Jiaotong University (Science), 18(5):554–562, 2013.

GYARD, A., BONNET, C., BOUDAOUD, K. **The stac (security toolbox: attacks & countermeasures) ontology.** In Proceedings of the 22nd International Conference on World Wide Web, pages 165–166. ACM, 2013.

IANNACONE, M., BOHN, S., NAKAMURA, G., GERTH, J., HUFFER, K., BRIDGES, R., FERRAGUT, E., GOODALL, J. (**Developing an ontology for cyber security knowledge graphs.** In Proceedings of the 10th Annual Cyber and Information Security Research Conference, page 12. ACM, 2015.
KARANDE, H. A., GUPTA, S. S. (2015). **Ontology based intrusion detection system for web application security.** In Communication Networks (ICCN), International Conference on, pages 228–232. IEEE, 2015.

KHAIRKAR, A. D., KSHIRSAGAR, D. D., KUMAR, S. **Ontology for detection of web attacks.** In Communication Systems and Network Technologies (CSNT), 2013 International Conference on, pages 612–615. IEEE, 2013.

L. YANG, W. GASIOR, R. KATIPALLY, X. CUI, **"Alerts Analysis and Visualization in Network-based Intrusion Detection Systems"**, *IEEE Second International Conference on Social Computing*, Minneapolis, MN,2010, pp.785-790.doi: 10.1109/SocialCom.2010.120, *2010.*

LI, W., TIAN, S. **An ontology-based intrusion alerts correlation system.** Expert Systems with Applications, 37(10):7138–7146, 2010.
MUSEN, M. A. **The proteg´ e project: a look back and a look forward.** ´ AI matters, 1(4):4–12, 2015.

PINKISTON, J., UNDERCOFFER, J., JOSHI, A., FININ, T. **A target-centric ontology for intrusion detection.** In proceeding of the IJCAI-03 Workshop on Ontologies and Distributed Systems. Acapulco, August 9 th. Citeseer, 2004.

RAZZAQ, A., AHMED, H. F., HUR, A., HAIDER, N. **Ontology based application level intrusion detection system by using bayesian filter.** In Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on, pages 1–6. IEEE, 2009.

RAZZAQ, A., ANWAR, Z., AHMED, H. F., LATIF, K., MUNIR, F. **Ontology for attack detection: An intelligent approach to web application security.** computers & security, 45:124–146, 2014.

SI, C., ZHANG, H., WANG, Y., Liu, J. **Network security situation elements fusion method based on ontology.** In Computational Intelligence and Design (ISCID), 2014 Seventh International Symposium on, volume 2, pages 272–275. IEEE, 2014.

SILVA, D. V., RAFAEL, G. R. **Ontologies for network security and future challenges.** In International Conference on Cyber Warfare and Security, page 541. Academic Conferences International Limited, 2014.

SIMMONS, C. B., SHIVA, S. G., SIMMOS, L. L. **A qualitative analysis of an ontology based issue resolution system for cyber attack management.** In Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on, pages 323–329. IEEE, 2014.

UNDERCOFFER, J., JOSHI, A., PINKSTON, J. **Modeling computer attacks: An ontology for intrusion detection.** In International Workshop on Recent Advances in Intrusion Detection, pages 113–135. Springer, 2013.

WANG, J. A. GUO, M. **Security data mining in an ontology for vulnerability management.** In 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, pages 597–603. IEEE, 2009.
XU, H., XIA, X., XIAO, D., LIU, X. **Towards automation for pervasive network security management using an integration of ontology-based and policy-based approaches.** In Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on, pages 87–87. IEEE, 2008.

XU, H., XIAO, D., WU, Z. **Application of security ontology to contextaware alert analysis.** In Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on, pages 171–176. IEEE, 2009.

YE, D., BAI, Q., ZHANG, M. **Ontology-based knowledge representation for ap2p multi-agent distributed intrusion detection system.** In Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on, pages 111–118. IEEE, 2008.