Ontology-based Adaptive Systems of Cyber Defense

Noam Ben-Asher*‡, Alessandro Oltramari†, Robert F. Erbacher*, Cleotilde Gonzalez†
*U.S. Army Research Laboratory Adelphi, MD, USA
nbenash@us.ibm.com, robert.f.erbacher.civ@mail.mil
†Carnegie Mellon University Pittsburgh, PA, USA
aoltrama@andrew.cmu.edu, coty@cmu.edu
‡IBM T.J.Watson Research Center, Yorktown Heights, NY

***Abstract*—In this paper we outline a holistic approach for understanding and simulating human decision making in knowledge-intensive tasks. To this purpose, we integrate semantic and cognitive models in a hybrid computational architecture. The contribution of the paper is twofold: first we describe a packet-centric ontology to represent network traffic. We show how the ontology is used to describe real-world network traffic and also serve as a basis for higher level ontologies of cyber operation, threat and risk. Second, we demonstrate how the combination of the packet-centric ontology with an adaptive cognitive agent with learning capabilities, can be used to understand the human defender reasoning processes when monitoring network traffic. Through simulation experiments we evaluated the proposed hybrid computational architecture and demonstrate its ability to successfully detect malicious port scanning within legitimate network traffic. We discuss the implications of these findings for improving our understanding of the cognitive processes and knowledge requirements of the cyber defender, as well as the possible use of the hybrid architecture as a cognitively inspired decision support tool.**

I. Introduction

Disruption of computers and the loss of sensitive information through cyber-attacks are becoming a widespread threat and a critical concern for citizens, organizations, and governments. Even with recent advances in information and network security and the development of new monitoring and threat detection tools, many of the tasks performed by cyber-defenders (i.e., security analysts) remain challenging, resulting in weak and uncertain cyber-defense. The analytical capabilities of the human decision maker are needed and indispensable for the process of cyber-defense [1]. Security analysts transform network traffic data into cyber situation awareness, a high level of processing that is difficult to automate [2]. This process may be seen as analogous to the Data-Information-Knowledge-Wisdom (DIKW) hierarchical model that is central for information and knowledge management [3]. Within this context, cognition serves as the driver that governs the transitions between the different levels of information representation [4]. While there is a large body of research on technologies that detect port scanning [5], there is a limited understanding of the cognitive processes cyber security analysts use to detect port scanning and specifically how these cognitive abilities interact with and information representation. In this regard, the contribution of this paper is twofold: first we describe a packet-level ontology that represents network traffic. Second, we demonstrate how the integration of this ontology with a computational cognitive agent can be used to understand the human analyst reasoning process, which may then serve as guide to develop decision support technology for the analyst.

II. Knowledge Model

From a cyber security standpoint, variations in network traffic are the primary prompts of analyst's behavioral responses; nevertheless, full situational awareness can emerge only from a projection of observations and decisions into a more comprehensive context that includes knowledge about threat and attack types, executable defensive maneuvers, system vulnerabilities, risk mitigation and time constraints, among others. In this regard, building a rigorous model of this complex context is a key requirement for the study of human decision making in cyber security. Computational ontologies are the knowledge component in this *holistic* approach, as they can provide a machine-readable semantic representation of cyber scenarios. In virtue of their logical properties and schematic structure, ontologies can be used by automatic reasoners in dynamic tasks: in particular, in our work we apply ontology-based reasoning to a detection task, where an agent simulates a human analyst's cognitive capabilities, including the capability of using domain knowledge and temporal information to reason about perceived events [6]. To this purpose, we engineered a packet-centric ontology of network traffic, a module of a larger ontology framework called `CRATELO` [7], the suite of modular ontologies under development in the U.S. Army Research Laboratory Cyber Security Collaborative Alliance. `CRATELO` is constituted of several domain ontologies (collectively indicated as `OSCO`), integrated on the basis of `DOLCE` top level [8] extended with a security-related middle layer. These top, middle and domain level ontologies currently add up to 330 classes, connected by 162 relationships (132 object properties and 30 datatype properties) and encoded in OWL-DL. The packet-centric ontology presented in this paper, henceforth abbreviated to `PACO`, is a partition of `OSCO`[1].

Our reseach efforts in developing `CRATELO` are inspired by Obrst and colleagues's proposal of a wide-ranging ontology framework of cyber security [9], that spans from top-level, system-oriented ontologies and human factors ontologies. In

[1]`CRATELO` stands for 'Three Levels Ontology for the ARL Collaborative Research Alliance'. `OSCO` stands for ontology of cyber operations. For more details about the program see also: http://www.arl.army.mil/www/default.cfm?page=1417

this long-term endeavour, we have been working with ARL domain experts and cyber analysts to distill the necessary knowledge of the cyber domain. As the state of the art shows, a preliminary step in understanding any new domain is to produce accessible definitions and classifications of entities [10]: discussions on cyber security often begin with the difficulties created by misused terminology (such as characterizing cyber espionage as an attack). In this regard, the Joint Chiefs of Staff created a list of cyber term definitions (allegedly extended and refined for a classified version). None of these definitions, however, were formulated as an ontology. Likewise, various agencies and corporations (NIST, MITRE, Verizon) have formulated enumerations of types of malware, vulnerabilities, and exploitations. In particular MITRE, which has been very active in the field, maintains two dictionaries, CVE (Common Vulnerabilities and Exposure) and CWE (Common Weakness Enumeration), a classification of attack patterns (CAPEC - Common Attack Pattern Enumeration and Classification), and an XML-structured language to represent cyber threat information (STIX - Structure Threat Information Expression).

Despite of the important role played by these and further initiatives, the lack of a shared formal semantics make terminologies hard to define, sustain, and port into a machine-processable format: here we try to overcome these problems, embracing a holistic approach to model cyber security factors. In fact, if the ontology outlined in this paper is tailored to a packet-centric model of network traffic, it can be framed at a higher level of conceptualization by means of the integration with `CRATELO`: for instance, when modeling the behavior of a cyber analysts during an attack, packets can be seen as parts of the evidence collection process, and specific attributes of packets (e.g. internal or external IP addresses, low or high packet rate, etc.) may hint to specific intentions of the adversary (also called *anti-goals*). As mentioned at the beginning of the section, ontologies can serve as knowledge bases to agents: conversely, the dynamics of the agent's decision process and learning from experience are captured by an Instance-based Learning (IBL) cognitive model [11], which is a computational representation of the processes that guide human behavior. Next section reviews what cognitive models are, and how they can be used to study human decision making.

III. Cognitive Model

In a dynamic decision making setting, cognitive architectures, such as ACT-R [12], SOAR [13] and others, have been commonly used to provide an integrated representation of human cognition. Cognitive models, constructed using these architectures, allow for a careful examination of various cognitive processes that drive human decision making [11]. Cognitive models based on IBL theory (IBLT) focus on decision making and learning from experience in dynamic settings [11]. IBLT emerging from ACT-R, proposes a generic decision-making process that recognizes decision situations, generates instances through the interaction with the decision task, and finishes with reinforcement of the instance leading to desired outcomes. According to IBLT, the decision maker represents decision making situations as instances stored in memory. An instance is composed of three parts: (1) *situation* (S) a set of attributes representing a situation; (2) *decision* (D) that is made in the particular situation; and (3) *utility* (U) that is the experienced outcome from a decision. The IBLT decision cycle includes several stages: recognition, judgment, choice, and execution. In the *Recognition* stage, a decision maker identifies relevant attributes for a specific decision situation. *Judgment* stage determines the relevancy of past experiences (instances) in current decision making situation. The activation of instances in memory is a representation of relevancy. Activation is influenced by the recency and frequency an instance occurred in the past and the similarity between the current decision situation and the situation stored in the instance. This activation mechanism is a simplification of the mechanism originally proposed in the ACT-R architecture. Memory activation determines the probability that an instance will be retrieved from memory and participate in the next phase. In the absence of previous experiences that may be relevant to the current situation, pre-defined heuristics are triggered for decision making. In the *Choice*, the retrieved instances and their retrieval probability are used to calculate the expected utility for each of the decision options, and the option with the highest expected utility is chosen. Finally, in the *Execution*, feedback regarding the last decision is provided to the decision maker [11]. In this work, we chose IBL to model the decision making as it captures the adaptive human decision making and learning processes in dynamic environment as well as the transition between exploration and maximization.

Agents based on IBL models successfully account for human decision making and behavior in a variety of tasks. Lejarraga *et al.* [14] demonstrate that a single IBL model constructed for a specific repeated binary choice task can be generalized to different variants of repeated tasks requiring a binary decision as well as to probability learning tasks. More specifically, IBL models can reflect human behavior in simple *stimulus-response* practice and skill acquisition tasks and training. Furthermore, the experience-based learning process of an IBL model was successfully extended to include descriptive information and biases as risk aversion [15]. A pair of IBL models successfully consider the dynamics of cooperation in iterated Prisoner's Dilemma as well as reciprocity and other complex social interactions [16], [17].

IV. A Packet-Centric Network Ontology

In this section we describe the structure of `PACO`, and how it can be used to instantiate thousands of packets generated by capturing actual network traffic. As Fig. 1 shows, the class 'PacketTransmission' is considered the atomic element of a 'NetworkSession'. Intuitively, this means that without an actual exchange of packets between a source and a destination node, no network session can be deemed as properly complete. In fact, there are additional features of network sessions: for instance, when considering TCP connections, a complete handshake with SYN, SYN+ACK and ACK packets transmis-

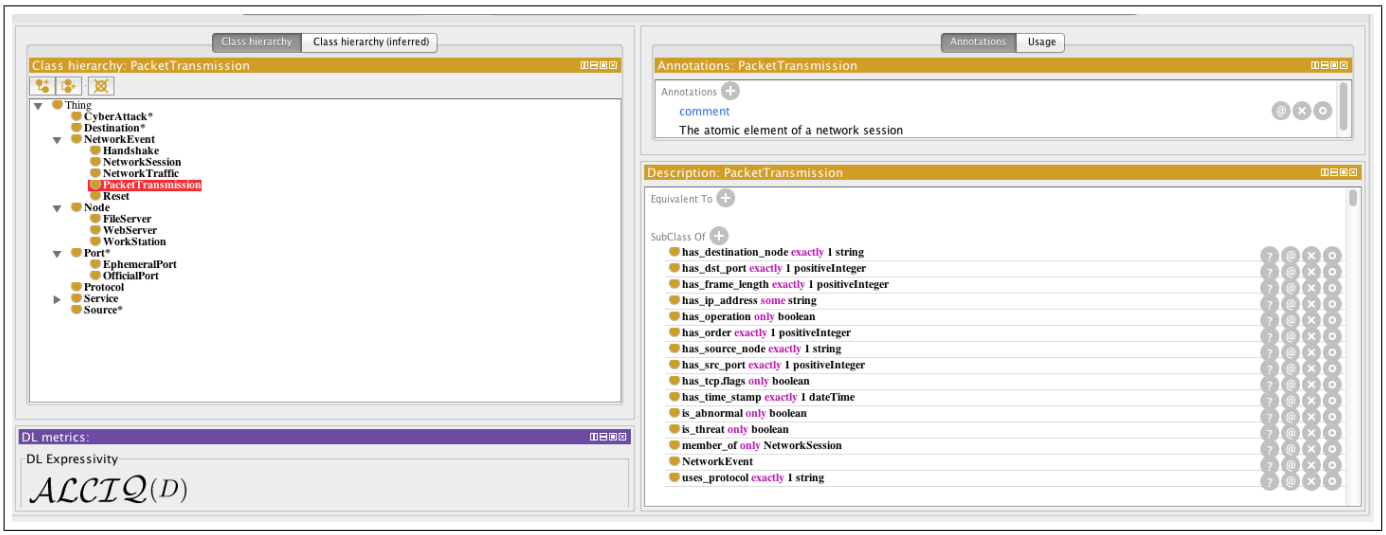

Fig. 1. A Protégé visualization of PACO. From the bottom-left corner (clockwise): 1) The DL expressivity derived by the HermiT 1.3.8 reasoner; 2) the backbone taxonomy of classes; 3) an informal definition of packet transmission as value of annotation property; 4) property restrictions.

sion is necessary to enable a packet transmission between two nodes, though this is not the case for communication protocols like UDP, where handshake dialogues are not supported. Following the actual packet transmission between the two network nodes and after the data are exchanged, a session is usually resetted (although this final stage is not essential to qualify it as complete - and session can also end due to a timeout). In summary, when a communication between a source and a destination node is established, a complete network session consists of the transmission of a unit of data from source A to destination B, and of the transmission of a unit of data from source B to destination A. From the ontological standpoint, this constraint is represented by the cardinality restriction 'min 2' on the object property 'has_member' holding between 'NetworkSession' and 'PacketTransmission' classes, respectively the domain and the range of 'has_member'.

Apart from network-specific information associated to source and destination nodes, like IP and port numbers, communication protocols, packet size, etc., we have introduced a data property 'has_time_stamp' that assigns a specific time stamp to each network event and a data property 'has_order' that binds each individual network event to its relative position in a given sequence (the first event, the second event, etc.). This twofold modeling choice provides us with a flexible model of temporal knowledge: 1) it pinpoints the discrete temporal coordinates of each event according to a universal time format (based on the XML schema specifications[2]); 2) it allows for representing and reasoning over qualitative temporal relations like 'before', 'after', and 'overlap', as defined by Allen's temporal axioms [18]). Figure 2 shows a situation where the ordinal scale of the packet is captured (i.e., the 1024th packet) but the time stamp is not represented: the reason is that the former is more appropriate than the latter for the simulation experiment reported in the next section, since the dataset was collected with a rate of about 83 packets per second. In other words, in our specific cyber scenario knowing the sequence of events is more meaningful than knowing the real time stamps from the defender's perspective, although - to be general enough - the ontology has to support both representational formats. As depicted in Fig. 2, the role of a packet in the handshake sequence can be captured by three booleans data properties, respectively 'has_tcp.flags.syn', 'has_tcp.flags.ack' and 'has_tcp.flags.reset'. In the 'PacketTrasmission1024' case, however it is unclear whether this packet represents the first stage of a handshake or is part of a port scanning [19]. This can be resolved by evaluating the properties of the proceeding packet exchange (i.e., session) between the two nodes. As the next section will show, we conducted an experiment to elicit relevant information from instantiated ontology, and make the resulting knowledge chunks available to the cognitive model of a cyber defender. This process of knowledge elicitation from PACO is driven by a set of SPARQL queries[3], properly designed to extract and present relevant information that an agent can use to decide whether a specific event is a threat or not. For instance, the query in Fig. 3 is designed to collect all the pairs of distinct source and destination ports in the dataset of network events: on the basis of the retrieved information, an analyst can gauge the volume of network traffic on a per unique port basis; moreover, Fig. 4 represents a query built to assess how many times a given source has sent a packet to a closed port. In the latter case, the returned result, around one thousand times, can be used as a clue of the maliciousness of the source: so many attempts of communication with closed ports may, in fact, suggest a port scanning attack. Note that

[2]http://www.w3.org/TR/xmlschema11-2/

[3]http://www.w3.org/TR/rdf-sparql-query/

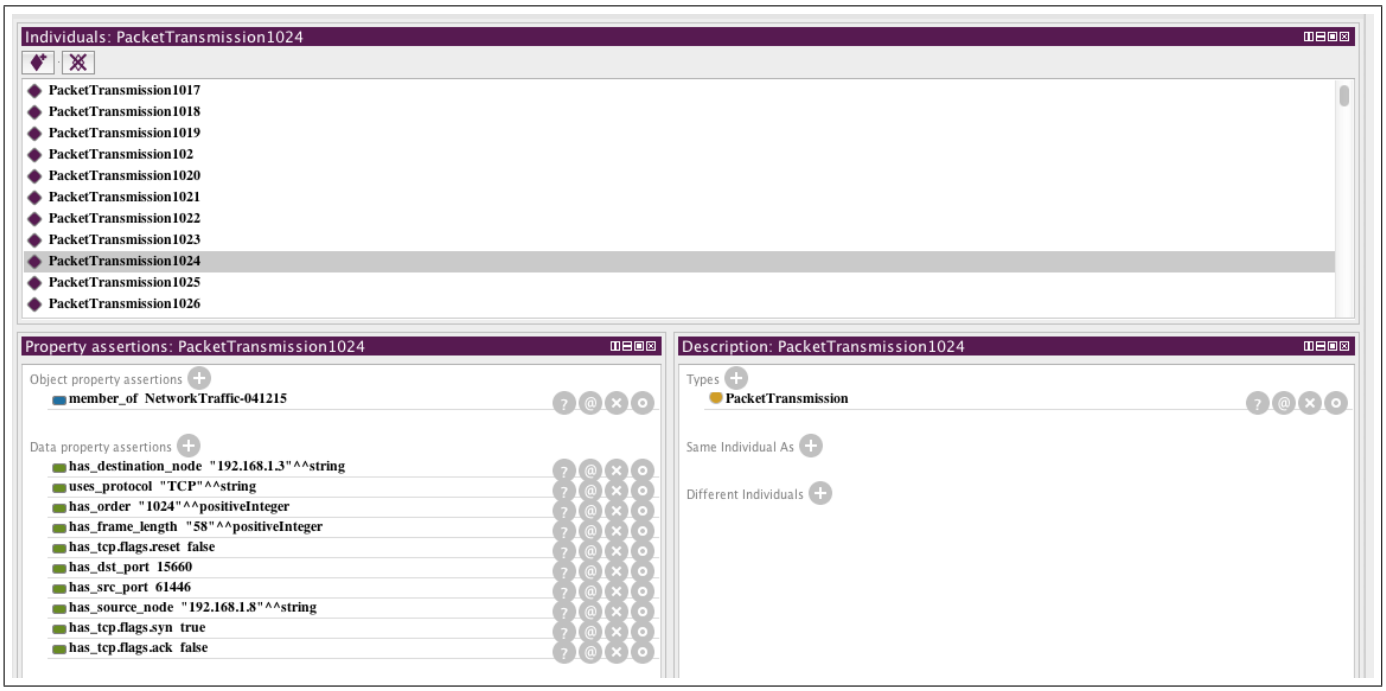

Fig. 2. A Protégé visualization of a specific instance of the 'PacketTransmission' class.

both queries have been used dynamically in the experiment described in the next section, where the goal is to replicate the analyst's incremental understanding of the considered cyber scenario.

Following a basic modeling strategy, in PACO we directly assign specific data sizes to each network event through the data property 'has_frame_length': an alternative option would have been to introduce the class 'Packet' (a subclass of 'information object' in DOLCE), and use the object property 'participation' to link 'Packet' and 'PacketTransmission', switching the domain of the data property 'has_frame_length' from PacketTransmission' to 'Packet'. At the current stage of development, representing the data contents of packet transmissions doesn't add any fundamental benefit to our modeling framework, although we don't exclude this option in the future.

Additional semantic structures of PACO concern network topology and services: for instance, every network node runs a set of services, and each service uses an official communication port and a specific protocol to establish a network session with another node. It follows that when a port is open, a service is running on a node, and if a port is closed, no services are currently running for that particular node. Thanks to the interoperability between PACO and CRATELO, services can be modeled in the context of user's actions: for instance, a system administrator can decide to start or stop an HTTP service, or access to the event log service on a server. By and large, the originality of our approach relies on the flexibility in the granularity of the representation: PACO is only a module of a more comprehensive framework that sees the detection as a socio-technical task, where packet-centric information can

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX IBLOd: <http://www.cra.psu.edu/IBLOd#>
 SELECT DISTINCT ?srcport ?dstport
  WHERE {{?event IBLOd:member IBLOd:NetworkTraffic-041215;
         IBLOd:has_src_port ?srcport;
         IBLOd:has_dst_port ?dstport;
         IBLOd:has_source_node ?s;
         IBLOd:has_destination_node ?d;
         IBLOd:has_order ?order.
          FILTER(?order >= "1"^^xsd:positiveInteger &&
                 ?order<= "4735"^^xsd:positiveInteger).}}
```

Fig. 3. A SPARQL query that returns all the distinct combinations of source and destination ports for a packets exchange sequence between two nodes.

be used by the decision maker at the cyber operation level. In principle, using CRATELO we can also model beliefs, goals and emotions of defenders and attackers, although it's beyond the scope of the current work to address these dimensions.

V. Using Hybrid Models in Cyber Defense

Next, we examine the interplay between knowledge and cognition in cyber defense by integrating the packet-centric ontology with cognitive agents who make decisions regarding the state of a network into a hybrid computational architecture. For the packet-centric knowledge-base we use PACO and the agents are computational models of the IBL theory.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX IBLOd: <http://www.cra.psu.edu/IBLOd#>
 SELECT (COUNT(?order) AS ?numberOfACKResponses)
  WHERE  {?event IBLOd:member IBLOd:NetworkTraffic-041215;
         IBLOd:has_source_node ?sn;
         IBLOd:has_destination_node ?dn;
         IBLOd:has_tcp.flags.syn false;
         IBLOd:has_tcp.flags.ack true;
         IBLOd:has_tcp.flags.reset true;
         IBLOd:has_order ?order.
          FILTER (?order >= "1"^^xsd:positiveInteger &&
                ?order <= "4735"^^xsd:positiveInteger).}
```

Fig. 4. A SPARQL query that returns the number of times a source node sent packets to a closed port on the destination node.

A. Port Scanning Scenario

Port scanning is designed to probe network nodes for open ports. The existence of an open port can provide some indication on the availability of services. This type of information gathering can be part of a defensive or offensive operation. From the attacker's perspective, a port scan is useful for gathering relevant information for launching a successful attack and indeed most attacks are preceded by some form of scanning activity (reconnaissance), particularly vulnerability scanning [20]. Therefore, the defender will try to detect external scans while the attacker interest is to perform a scan without being detected [21].

In this work, we assume first that the attacker uses external resources to identify the attack IP address (i.e., the target). Following, the attacker identifies port ranges to scan on the specific target. These are the ports for services for which the attacker has sophisticated attacks available. We also assume, that the target is using standard ports and not randomized ports. Thus, knowing that a port is open provides an accurate indication that a service is running on the target.

B. Cognitive Models for Port Scanning Detection

To better understand the interplay between cognition and knowledge and how semantic information supports the ongoing work of the cyber defender, we developed two cognitive models for cyber defender agents. Both agents observe a situation, make decisions whether there is a scan or not, and learn from feedback and past experiences. However, the one agent operates without the knowledge based provided by PACO, while the other is querying PACO to acquire temporal information and situational awareness.

1) Experience Only Agent: To examine the interplay between information, cognition and knowledge, we initially constructed an agent using an IBL model which classifies network events based on their attributes and learns from experience only. The decision making process of this IBL agent depends on the low level network traffic information, and the agent could learn only from its own experiences without the ability to acquire knowledge by querying the ontology. The situation as observed by the agent in this condition is given by

$$S_i = \{p, sIP, dIP, SYN, ACK, RST\} \quad (1)$$

TABLE I
PAYOFF MATRIX WHICH DETERMINING THE FEEDBACK FOR AN AGENT MAKING A DECISION IN A GIVEN SITUATION

		Agent's Decision	
		Scan	No Scan
Packet Type	Scan	Hit: 10	Miss: -10
	No Scan	False alarm: -5	Correct Rejection: 5

Where p is the protocol type (e.g., TCP, HTTP) of the packet, sIP and dIP are the source and destination IP addresses of the packet. SYN, ACK and RST are 1-bit boolean flags that indicate on the state of a connection.

The agent observed a situation S_i and made a decision which corresponds to classifying a packet as being part of a scan or not. This decision process involves retrieving relevant instances (i.e., past experiences) from the agent's memory, computing retrieval probability for each of the instances and, choosing the decision option that yields the highest expected utility, based on the previous decisions recorded in the instances. The process of choosing the option with the highest expected utility is influenced by the recency and frequency of past experiences, memory decay (d) and a noise parameter for capturing the variability in memory activation (σ) [11].

After making a decision, the agent received a utility feedback, representing the outcome of the decision in a given situation. The experienced utility (i.e., payoff) is determined based on the payoff matrix illustrated in Table I. The payoff that an agent receives following a decision, is determined by the accuracy of the decision, based on the ground truth, detailed in section V-C. The payoffs in the matrix emphasize the positive and negative utilities from hits and misses over correct rejections and false alarms.

2) Semantic Information and Experience Agent: In contrast to the previous agent model, this agent can send SPARQL queries to the PACO ontology, that provides specific knowledge of the scenario, temporal information and augmented situational awareness. As such, this model observes the same situation as the *Experience Only* agent: however, instead of using this information to make a decision, the agent uses the information to generate queries (which, in turn, provides richer information). Using PACO, the agent can generalize from and reason about the characteristics of a sequence of packets transferred from one network node to the other. Therefore, the situation observed by the agent consist of the outputs from multiple queries regarding the *conversation* between two specific IP addresses, where one is the source and the other is the destination. The situation for any packet, transmitted between a source and a destination IP addresses, is given by

$$S_i = \{p, sPorts, dPorts, avgSYN, avgACK\text{-}RST\} \quad (2)$$

Where the attributes of the situation represent properties of a communication between source and destination IPs, using protocol p. The communication consists of a sequence of packets exchanged between the two network nodes up to the current packet. Thus, the agent can examine each packet within

the context of a sequence. Given the source IP of the current package, attribute $sPorts$ indicates on the average number of ports in the source node that sent packets to the destination node. Similarly, attribute $dPorts$ indicates how many ports in the destination node received packets from the source. The attribute $avgSYN$ describes the average ratio between SYN packets and normal traffic recived from the source of the packet. Attribute $avgACK\text{-}RST$ provides complementary information, the average ratio of between ACK-RST packets and normal traffic the destination sent back to the source. This type of answer indicates that the packet was sent to a closed port (i.e., a port that is not used by any service on the target node).

Based on the set of attributes described above, the *Semantic information and Experience* agent classified packets. The *Semantic information and Experience* agent received feedback for these decisions using the same payoff matrix as the *Experience Only* agent.

C. Simulation Experiment

We evaluated the differences between the two agent models through simulation experiment. In the experiment, agents classified the packets captured from the traffic in a small network with 16 nodes (i.e., unique IP addresses). The captured communication between the network nodes included 4735 packets. The nodes used several types of protocols to exchange packets, for example SMB and SSL. However, the majority of the traffic (99.56%) used the TCP protocol. Within this network, the adversary was located in a node with the IP address of 192.168.1.8. The adversary used a specific port to scan the 1000 common ports of the target node (192.168.1.3) using Nmap defaults [22]. This information was not provided to the agents and served as the ground truth for evaluating the detection performance of the agents and providing them with feedback. The captured network traffic was converted into an XML data structure that was used to populate `PACO` and the *Semantic Information and Experience* agent could then query using SPARQL. The output of the SPARQL queries served as the attributes of a situation as described in Eq. 2.

The values of the free parameters across the two agents were kept the same, with $d = 1.5$ for memory decay and $\sigma = .25$ for noise. These values are considered to be the ACT-R defaults and are commonly used for IBL models as well [23]. Each agent classified the 4735 packets and received feedback following each decision, and this was repeated for 20 iterations.

To compare the performance of the *Experience Only* agent with the *Semantic Information and Experience* agent we used the following metrics:

1) **Correct packet classification** indicates on the proportion of packets classified correctly as being a *Scan* or *No Scan* packet.
2) **Correct detection of scanning sequence** indicates on the proportion of conversations between two IPs that were correctly classified as scans.

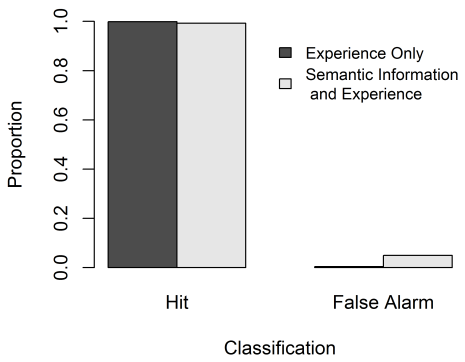

Fig. 5. Proportions of hits and false alarms for the two agents.

3) **Learned classification rule** indicates on the decision rule the agents constructed from the repeated experiences.

VI. Results

In this section, we show our experimental results and analyze the observed trends based on the performance comparison of the two modeling approaches.

Correct packet classification When analyzing the ability of the agents to classify correctly a scan packet, and as seen in Fig. 5, we find that the *Experience Only* agent (mean=.999, SD=0) and the *Semantic Information and Experience* agent (mean=.992, SD=.002) performed similarly with a minor advantage to the *Experience Only* agent, $t(38)=.387$, p=ns. However, the *Semantic Information and Experience* agent (mean=.050, SD=.077) generated a significantly higher number of false alerts compared to the *Experience Only* agent (mean=.004, SD=0), $t(38)=2.661$, $p=.011$.

Correct detection of scanning sequence utilizes the classification of a packet as belonging to a scan or to normal traffic between two network nodes. This high level decision aims to answer the question whether network node A is scanning network node B. With respect to this question, if the network traffic from node A to node B includes one or more packets that are classified as scan packets, then node A is scanning node B. When analyzing the ability of the two agents to answer the question whether node A is scanning node B, both agents detected that the adversary was scanning a specific network node (i.e., 192.168.1.8 $\overset{SYNscan}{\rightarrow}$ 192.168.1.3). However, the *Experience Only* agent detected on average additional 2.3 out of 22 sequences between network nodes as scans (i.e., 10% false scans), while the decisions of the *Semantic Information and Experience* agent yielded 0 false classification of packet sequences. Despite the higher false classification rate of individual packets the *Semantic Information and Experience* agent had, all these false classified packets belonged to the responses of the scanned node (ACK packets) to the adversary scan (i.e., 192.168.1.3 $\overset{ACKresponse}{\rightarrow}$ 192.168.1.8).

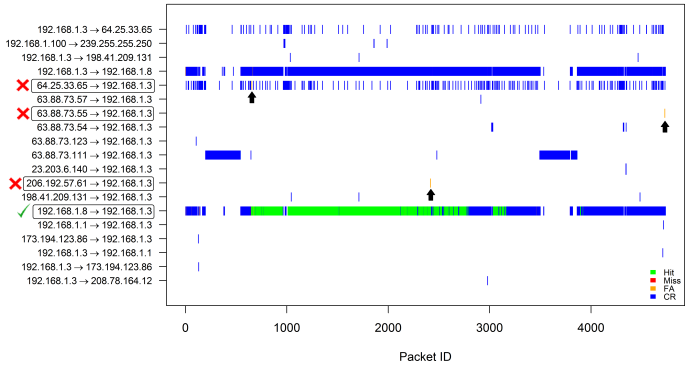

Fig. 6. Detection outcomes of the *Experience Only* agent during a single iteration with black arrows highlighting false classification of packets, red cross marks indicating on sequences of packets that were incorrectly classified as scans and green check mark for correct classification.

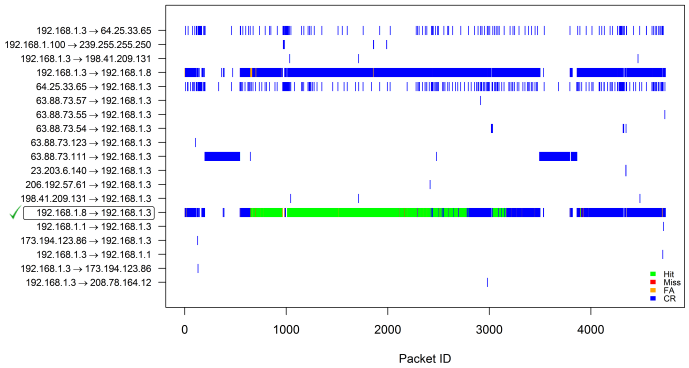

Fig. 7. Detection outcomes of the *Semantic Information and Experience* agent during a single iteration with green check mark indicating on the correct classification of a packet sequence.

Figures 6 and 7 illustrate the interplay between packet classification and sequence classification. In both figures, we present the same network sequences and how the packets were classified by each agent. As seen in Fig. 6, the *Experience Only* agent generated a very low number of false alerts (highlighted by arrows). However, these packets corresponding to these alarms were distributed across multiple sequences. As a result, the entire sequence was classified as a scan. On the other hand, and as seen in Fig. 7, the false alerts generated by the *Semantic Information and Experience* agent were all part of the communication between the scanned node to the source of the scan. Note that both agents were able to separate between legitimate traffic between 192.168.1.8 and 192.168.1.3 that was not part of the scan and used UDP and other protocols.

The **Learned classification rule** used by each agent can be formalized by examining the instances stored in the memory of each agent, and their activation. The combination of attributes and decision in highly activated instances represent beliefs regarding a relationship between a situation and the appropriate decision. The decision rule formulated by the *Experience only* agent was that any TCP packet with a SYN flag is part of an ongoing scan between the source of the packet and its destination. This rule yields high accuracy in detecting scan packets as all the scan packets had a SYN flag. However, packets with SYN flag are also part of legitimate handshake between network node and for that reason the *Experience Only* agent detected a higher proportion of packets sequences as scans. In contrast, the *Semantic Information and Experience* agent observed the temporal properties of a packet sequence. The decision rule formulated by this agent suggests that a scan packet uses TCP protocol and is part of a sequence of packets in which the source node is using a low number of ports to send packets to a high number of destination ports and the average number of SYN packets sent to a port is very close to 1. In addition, the rule constructed by the *Semantic Information and Experience* agent indicates the based on experience, the target node of the packet is very likely to respond to the current packet with a ACK-RST packet, indicating that the destination of the packets coming from the source node tends to be a closed port.

VII. Discussion and Conclusion

Analytical capabilities of the human decision maker are needed and are indispensable when ensuring the security of any cyber infrastructure. It is the human abilities to integrate information, to reason, to learn and to quickly adjust to changes that make such significant contribution to cyber security. The understanding of these processes relies on our integration of knowledge from human cognitive theories and knowledge-based technologies. In this study we propose an architecture to combine cognitive models and ontologies in the domain of cyber defense.

We developed a packet-centric ontology PACO which allows us to represent and capture the atomic elements of network communication, i.e., packets and sequences of packets. PACO serves as the basis for more holistic semantic representations of cyber operation, cyber assets, threats and risks, available through CRATELO. We also developed an IBL cognitive model capable of accessing the information in PACO and using it when detecting adversarial port scan. When making decisions, the ability of the IBL agent to access PACO and retrieve information improved its performance, compared to the same IBL agent that did not utilize PACO. We show that when answering the questions 'Is IP A scanning IP B?', an agent with access to a packet-centric ontology delivers a much lower false alerts rate and by that show superior performance. Overall, the access to semantic information allowed the agent to acquire better situation awareness by incorporating summarized information into the decision making process. PACO extended the agents ability to inspect temporal relationship between a packet sent from a specific source and previous replays of packet's destination to communication coming from that source. Such reasoning requires a representation of a source and a destination, as well as the ability to switch between these roles in order to observe the response patterns.

The agents explored rules in the form of *IF* a situation *THEN* a decision, and learned which rule maximizes their

payoff. While the attributes of the situation part are influenced by the availability of information, the cutoff values of the attributes were learned from experience. Furthermore, the decision rule that the agent with access to a packet-centric ontology learned from experience is valid and useful beyond the limited scope of the network scenario we used in the study.

However, the existence of knowledge is a precondition rather than a guarantee for improvement: correctly querying the information is the key for the major improvement. In the process of modeling, we used domain experts to construct the queries that aggregate and retrieve information. By using cognitive agent we were able to test different queries and combinations of attributes, to identify representations that facilitate the decision making process of a network defender.

While PACO has the potential of representing packet level information for complex and diverse network communication, the current cognitive model was developed to accommodate a simplistic network scenario. Port scanning can take many forms (vertical and horizontal scans), can use different protocols and can be highly distributed over time (i.e., low-and-slow scan). Therefore, although we used a high fidelity network traffic, future research should scale up the volume of traffic as well as the complexity of the network scan. Such additions will likely challenge the cognitive agent. However, providing the agent access to the middle and high levels of CRATELO might be the key component for the agent's success in more complex and challenging tasks. The benefit of pairing cognitive agents and ontologies goes beyond the ability to gauge into the decision making process of the human analyst. Such combination can serve as an initial step towards the development of cognitively inspired decision aid tool for automating some tasks that are currently performed by human analyst.

ACKNOWLEDGMENT

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] C. Gonzalez, N. Ben-Asher, A. Oltramari, and C. Lebiere, "Cognition and technology," in *Cyber Defense and Situational Awareness*, ser. Advances in Information Security, A. Kott, C. Wang, and R. F. Erbacher, Eds. Springer International Publishing, 2014, vol. 62, pp. 93–117.

[2] A. DAmico and K. Whitley, "The real work of computer network defense analysts," in *VizSEC 2007*. Springer, 2008, pp. 19–37.

[3] J. E. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, 2007.

[4] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Computers in Human Behavior*, vol. 48, pp. 51–61, 2015.

[5] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks," *Technical report, Univeristy of California, Department of Computer Science and Engineering*, 2003.

[6] A. Oltramari, N. Ben-Asher, L. Cranor, L. Bauer, and N. Christin, "General requirements of a hybrid-modeling framework for cyber security," in *Military Communications Conference (MILCOM)*. IEEE, 2014, pp. 129–135.

[7] A. Oltramari, L. F. Cranor, R. J. Walls, and P. McDaniel, "Building an ontology of cyber security," in *9th International Conference on Semantic Technologies for Defense, Intelligence and Security (STIDS)*, 2014, pp. 54–61.

[8] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, R. Oltramari, L. Schneider, L. P. Istc-cnr, and I. Horrocks, "Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology," 2002.

[9] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain." in *7th International Conference on Semantic Technologies for Defense, Intelligence and Security (STIDS)*, 2012, pp. 49–56.

[10] D. A. Mundie and D. M. McIntire, "The mal: A malware analysis lexicon," DTIC Document, Tech. Rep., 2013.

[11] C. Gonzalez, J. F. Lerch, and C. Lebiere, "Instance-based learning in dynamic decision making," *Cognitive Science*, vol. 27, no. 4, pp. 591–635, 2003.

[12] J. R. Anderson and C. Lebiere, *The atomic components of thought*. Lawrence Erlbaum Associates Publishers, 1998.

[13] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[14] T. Lejarraga, V. Dutt, and C. Gonzalez, "Instance-based learning: A general model of repeated binary choice," *Journal of Behavioral Decision Making*, vol. 25, no. 2, pp. 143–153, 2012.

[15] N. Ben-Asher, V. Dutt, and C. Gonzalez, "Accounting for the integration of descriptive and experiential information in a repeated prisoner's dilemma using an instance-based learning model," in *22th Behavior Representation in Modeling & Simulation (BRiMS) Conference*, 2013, pp. 11–14.

[16] C. Gonzalez, N. Ben-Asher, J. M. Martin, and V. Dutt, "A cognitive model of dynamic cooperation with varied interdependency information," *Cognitive science*, 2014.

[17] C. Gonzalez and N. Ben-Asher, "Learning to cooperate in the prisoners dilemma: Robustness of predictions of an instance-based learning model," in *35th annual meeting of the Cognitive Science Society (CogSci 2014)*, 2014, pp. 2287–2292.

[18] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[19] M. De Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, "A review of port scanning techniques," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 2, pp. 41–48, 1999.

[20] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, p. 80, 2011.

[21] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *The Computer Journal*, vol. 10, pp. 1565–1581, 2011.

[22] Nmap network mapper. [Online]. Available: https://nmap.org/

[23] N. Ben-Asher, J.-H. Cho, and S. Adalı, "Cognitive leadership framework using instance-based learning," in *24th Conference on Behavior Representation in Modeling and Simulation (BRiMS)*, March 2015.