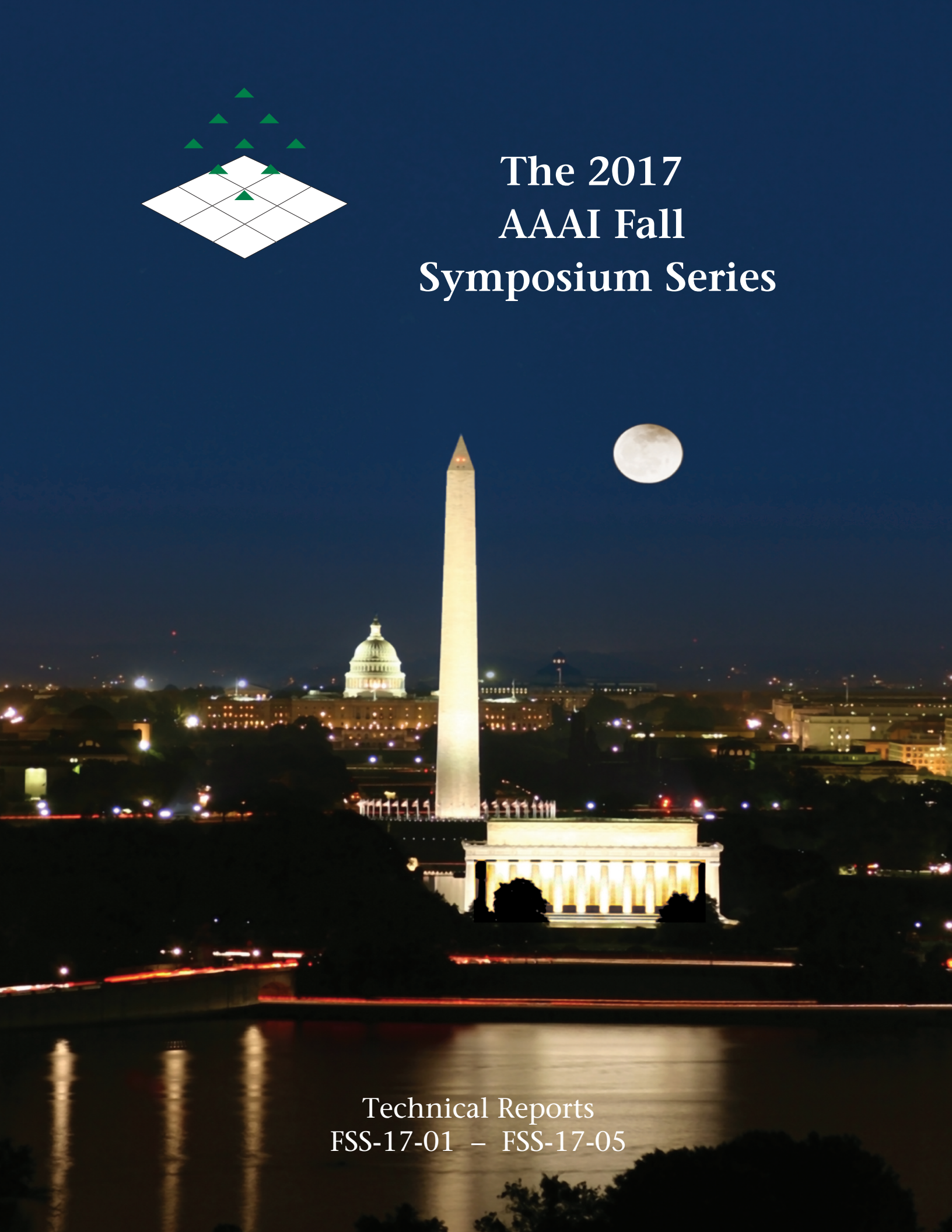

The 2017 AAAI Fall Symposium Series

Technical Reports
FSS-17-01 – FSS-17-05

Cognitive Assistance in Government and Public Sector Applications
AAAI Technical Report FS-17-02

Collaborative Cognitive Assistants for Advanced Persistent Threat Detection

Steven Meckl, Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, Ahmed Bin Zaman
Learning Agents Center, Volgenau School of Engineering, George Mason University, Fairfax, VA 22030, USA
smeckl@masonlive.gmu.edu, teuci@gmu.edu, dmarcu@gmu.edu, mboicu@gmu.edu, azaman6@masonlive.gmu.edu

Abstract

This paper presents current results on researching the automation of Cybersecurity Operations Centers (CSOC) with collaborative cognitive assistants that are able to capture and automatically apply the expertise employed by cybersecurity analysts when they investigate Advanced Persistent Threats (APT). An expert cyber analyst teaches a learning agent shell, through examples and explanations, how to generate and assess both APT intrusion and false positive hypotheses. The trained learning agent is then customized into specialized autonomous collaborative agents. This paper presents the operations of these agents in a CSOC.

Introduction

Modern cyber defense is currently done in a cybersecurity operations center (CSOC), which employs teams of network defense experts, analysts, system administrators, and forensics experts. CSOCs leverage a rich tool set including host-based and network-based intrusion detection systems (IDSs), data collections, analysis tools, and visualization tools. CSOCs receive incident information from high-value sources – law enforcement, user reporting, or threat intelligence from other CSOCs – and unconfirmed alerts from security infrastructure such as antivirus software, IDSs, heuristic alerts, or machine learning algorithms (Zimmerman 2014).

Among the most sophisticated threats are those known as Advanced Persistent Threats (APTs). An APT is an adversary that leverages superior resources, knowledge, and tactics to achieve its goals through computer network exploitation. APTs are characterized by their persistence in gaining and maintaining access to targeted networks and their ability to adapt to efforts of network defenders to identify and remediate their activity (Mandiant 2013).

Security research companies have been tracking APT groups for years, independently giving them unique names as specific tools, techniques, and procedures (TTPs) are attributed to a group. FireEye/Mandiant has published reports on 30 APT groups since 2013, naming them simply APT1 through APT30 (FireEye 2015).

APT1 is the name given by Mandiant (2013) to a group of APT actors, attributed to China's People's Liberation Army unit 61398, who have lead a campaign of cyber espionage since at least 2004. APT1 is known for a regimented approach to computer intrusion activity. An APT1 intrusion typically consists of the following phases: (1) gain access to a network by sending fraudulent, malicious email messages to specific users (spearphishing); (2) use multiple types of backdoor programs to maintain presence and provide remote connectivity to the target network; (3) use a collection of command-and-control (C2) servers to obfuscate the source of their attacks; (4) escalate privileges and acquire legitimate login credentials to access network resources; (5) move laterally within the target network using legitimate credentials to gain redundant points of presence and identify information of interest; and (6) exfiltrate targeted information through their C2 infrastructure (Mandiant 2013).

The responsibility of a CSOC's analysts is to monitor alerts and log information from all of the information sources, each having differing levels of credibility, and use them to make a determination about the presence or absence of intrusion activity (Zimmerman, 2014). However, because a single alert alone does not provide sufficient evidence that an intrusion event has occurred, and modern detection technologies are error-prone, each alert must be carefully examined and investigated by a human analyst (Zimmerman 2014). In a large enterprise, tens of thousands of alerts per day can be reported. Therefore, even sensors with a false positive rate of less than one percent can generate enough false positives to be unmanageable by even large CSOCs.

This paper presents current results on researching the automation of CSOCs with agile cognitive assistants that are able to capture and automatically apply the expertise employed by cybersecurity analysts when they investigate APTs. APTs adapt to efforts of network defenders to identify the malware. For example, Chinese Military's

APT1 evolved over time to include several malware families (Auriga, Bangat, Seasalt, Kurton, etc.). Therefore, another major goal of our research is to enable the developed agents detect not only the malware for which they have been trained, but also their variations, alerting and involving the cyber analysts in a mixed-initiative analysis process (Tecuci et al. 2005; Tecuci, Boicu, and Cox 2007a). Finally, because the reasoning of these agents is very explicit, we also develop them to be used in teaching new cyber analysts how to detect APTs (Tecuci et al. 2002).

We start with presenting an overview of our approach to teaching a learning agent shell how to detect APTs. The trained learning agent shell is then used to generate a collection of agents, each specialized for a certain phase of the analysis process. These specialized autonomous agents are those that are actually integrated into a CSOC. We then illustrate how they collaborate in detecting an Auriga intrusion. We conclude the paper with the main directions of our future work.

Agent Teaching and Learning

For many years we have researched a theory, methodology, and tools for the development of knowledge-based cognitive assistants that: (1) learn complex problem solving expertise directly from subject matter experts; (2) support experts and non-experts in problem solving; and (3) teach their problem solving expertise to students (Tecuci et al 2016a). The investigated approach relies on developing a powerful learning agent shell that can be taught by a subject matter expert (who does not have computer science or knowledge engineering experience) in ways that are similar to how the expert would teach a student or an apprentice, by explaining problem solving examples to it, and by supervising and correcting its problem-solving behavior. Because the resulting agent learns to replicate the problem-solving behavior of its human expert, we have called it a Disciple agent (Tecuci 1988; 1998; Boicu et al. 2000; Tecuci et al. 2005b; 2008; 2016a).

Building on the Disciple approach, we are researching the development of a learning agent shell that can be taught how to detect APT intrusions. The agent employs a learnable hybrid knowledge representation consisting of an APT ontology and various types of rules with ontology-based applicability conditions. The ontology language is an extension of RDFS (Allemang and Hendler 2011; W3C 2004) with additional features to facilitate learning and evidence representation (Tecuci et al. 2007b; 2016a).

The process of teaching the agent how to detect APT intrusions is abstractly illustrated in Figure 1. An expert analyst teaches the learning agent shell how to detect a specific APT1 intrusion (e.g., by the Auriga malware), by illustrating all the associated reasoning steps.

First the expert analyst specifies an event of interest that

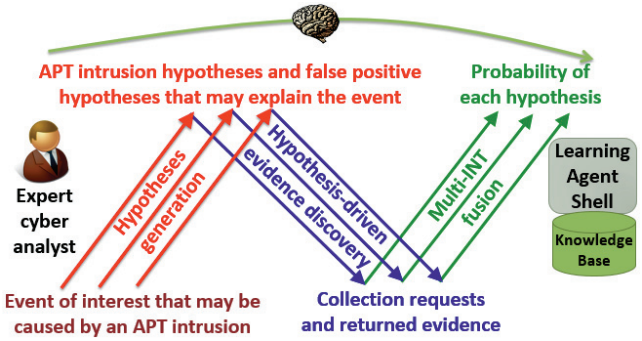

Figure 1: Teaching the learning agent shell to automatically detect Advanced Persistent Threat intrusions.

may be caused by such an APT1 intrusion. This is called ***trigger*** and is the kind of alert generated by a network IDS, such as BRO (2017) or SNORT (2017).

Then the expert defines alternative hypotheses that may explain the alert. Some of these hypotheses are APT1 intrusion hypotheses, but others are false positives. Each of them will have to be analyzed, based on evidence, to estimate its probability.

To analyze a hypothesis, it is successively decomposed into sub-hypotheses that more clearly point to the evidence that needs to be collected in order to assess them. Searches for evidence take place, and the found evidence is used to either further decompose the sub-hypotheses or to assess their probabilities. This process continues until the probabilities of all the generated hypotheses are determined.

From the various reasoning steps in this process the agent, with the help of the cyber expert, learns different types of rules, as follows:

- A ***trigger*** rule that represents the trigger (e.g., the BRO alert) into the agent's ontology and generates a basic hypothesis.
- An ***indicator*** rule that abductively generates an intrusion hypothesis from the basic hypothesis.
- Several ***question*** rules, each generating alternative hypotheses to analyze, as answers to the same intelligence question.
- Several ***hypothesis analysis*** rules that generate Wigmorean analysis fragments (Schum 2001; Tecuci et al. 2015; 2016a; 2016b). Each Wigmorean fragment decomposes a hypothesis into a tree of sub-hypotheses.
- Several ***evidence collection*** rules that invoke evidence collection agents for specific sub-hypotheses, and represent the found evidence in the ontology.

Each of the above rules is initially partially learned as an ontology-based generalization of one example and its explanation. They are then used in reasoning to discover additional positive and negative examples, and are further incrementally refined based on these new examples and their explanations. The approach is based on methods for

integrating machine learning with knowledge acquisition (Tecuci and Kodratoff 1995), and on multistrategy learning (Tecuci, 1988; 1993; Tecuci et al. 2016a).

The result of this agent teaching and learning process is a learning assistant that not only has reasoning modules for all the phases of the APT1 intrusion detection process from Figure 1, but also a knowledge base (KB) with a developed ontology and reasoning rules for all these phases.

Specialized Agents Integrated into a CSOC

From the trained learning agent shell several autonomous agents are generated, each specialized for a specific phase of APT intrusion detection, by including only the reasoning modules corresponding to that phase. These agents, using a shared knowledge base (shown in the left-hand side of Figure 2), are integrated into a specific CSOC and collaborate in APT intrusion detection, as explained below.

The ***Trigger Agent*** receives alerts from a variety of sources, such as the network IDSs BRO (2017), SNORT (2017), Suricata (2017), or Symantec Security Analytics (2017), from network anomaly detection, and from endpoint protection alerts (anti-virus, next-gen endpoint). In the current implementation, however, we are only processing BRO alerts. The trigger agent represents each alert into a different knowledge base (KB) that inherits knowledge from the shared KB. These KBs are organized into a *hypotheses generation queue* from which they are extracted by the Hypothesis Generation Agent.

The ***Hypotheses Generation Agent*** generates hypotheses corresponding to a trigger and places the KB into the *hypotheses analyses queue* from which the KBs are extracted by the Automatic Analysis Agents.

Each ***Automatic Analysis Agent*** decomposes the hypotheses from such a KB, as much as possible, down to the level of evidence collection requests. Then it places the KB into the *evidence collection queue* from where each KB is extracted by the Collection Manager.

The ***Collection Manager*** invokes specialized collection agents to search for evidence on the network. Then it represents the retrieved evidence into the corresponding

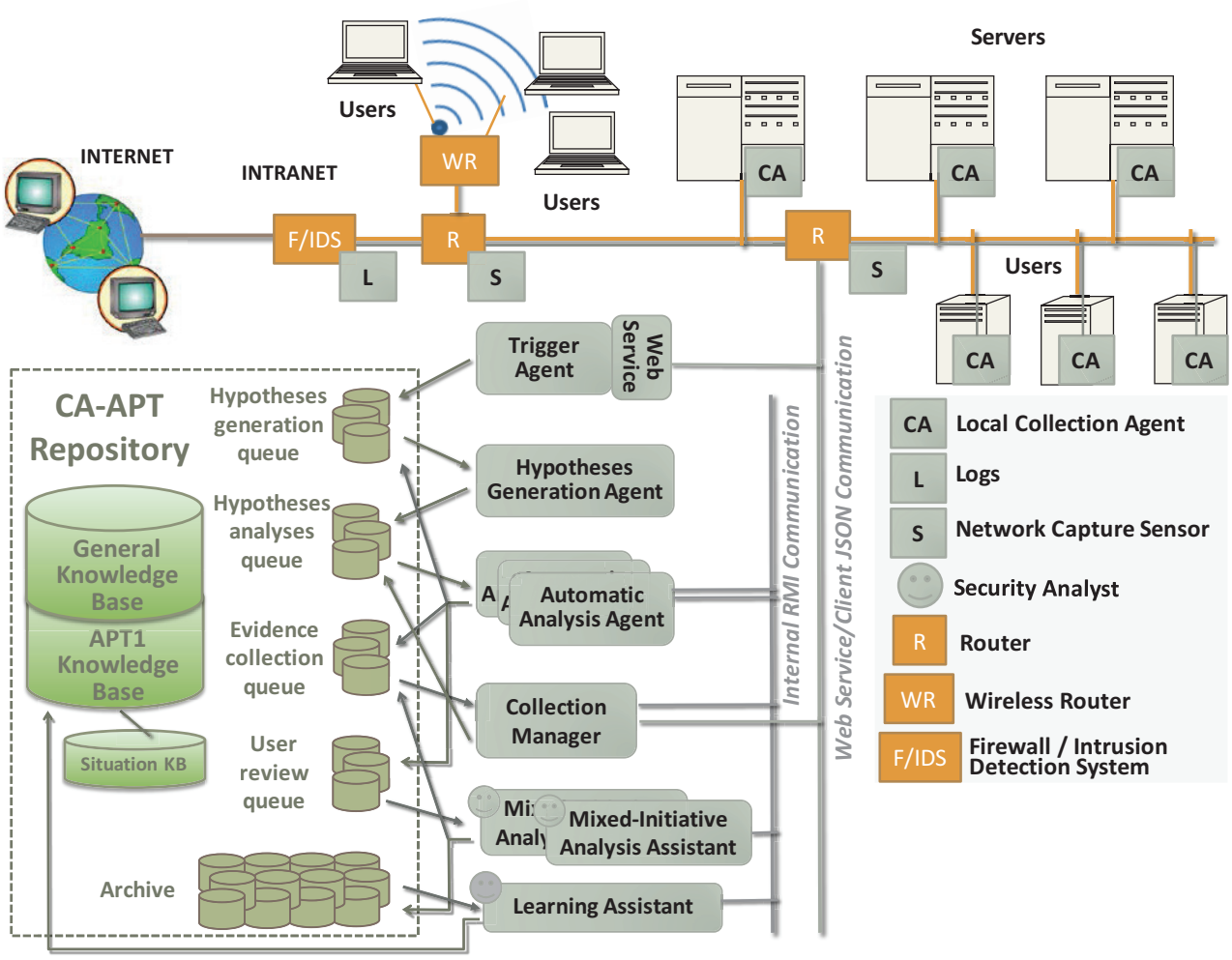

Figure 2. Specialized agents integrated into a cybersecurity operations center.

KBs and places these KBs in the *hypotheses analyses queue*.

When an automatic analysis agent has performed the most complete analysis possible of the alternative hypotheses corresponding to a trigger, it places the knowledge base into the *user review queue*, to be used by the Mixed-Initiative Analysis Assistant and the cyber analyst.

The ***Mixed-Initiative Analysis Assistant*** interacts with the cyber analyst, either by alerting the analyst of a detected intrusion, or by collaborating with them to finalize the analysis. After the analysis corresponding to a trigger is completed and necessary actions taken, the KB is placed into an *archive* by the mixed-initiative analysis assistant.

The KBs from the archive are used by the ***Learning Assistant*** and an expert cyber analyst to further refine the ontology and the rules shared by the specialized agents.

A novel feature of these agents, as compared to the Disciple agents developed in the past (Tecuci et al., 2016a), is that they are autonomous and they collaborate in addressing the complex APT intrusion detection problem.

The following sections illustrate the operation of the above agents in the context of detecting an Auriga intrusion.

Trigger Agent and Hypotheses Generation Agent

The bottom left of Figure 3 shows the JSON representation of a BRO alert received by the Trigger Agent through a Web service from the BRO IDS. The Trigger Agent employs a trigger rule to generate the ontological representation of this alert, as shown in the bottom right of Figure 3. It also generates the following basic hypothesis (shown above the JSON representation):

Suspicious connection1 from 10.10.1.11 (port 11234) to 8.8.8.8 (port 53) at 05/15/2017 16:23 GMT, using known APT1 domain a-jsm.infobusinessus.org

Both the ontological representation of the trigger and its representation as a basic hypothesis are generated in the trigger KB.

The Hypotheses Generation Agent employs an indicator rule to abductively generate the following hypothesis:

connection1 from 10.10.1.11 (port 11234) to 8.8.8.8 (port 53) at 05/15/2017 16:23 GMT, using known APT1 domain a-jsm.infobusinessus.org, is part of APT1 intrusion

Then it uses a question rule to generate the question which has the above hypothesis as a possible answer:

What has generated connection1 from 10.10.1.11 (port 11234) to 8.8.8.8 (port 53) at 05/15/2017 16:23 GMT, using known APT1 domain a-jsm.infobusinessus.org?

After that the agent uses two other question rules to generate the other possible answers of the above question:

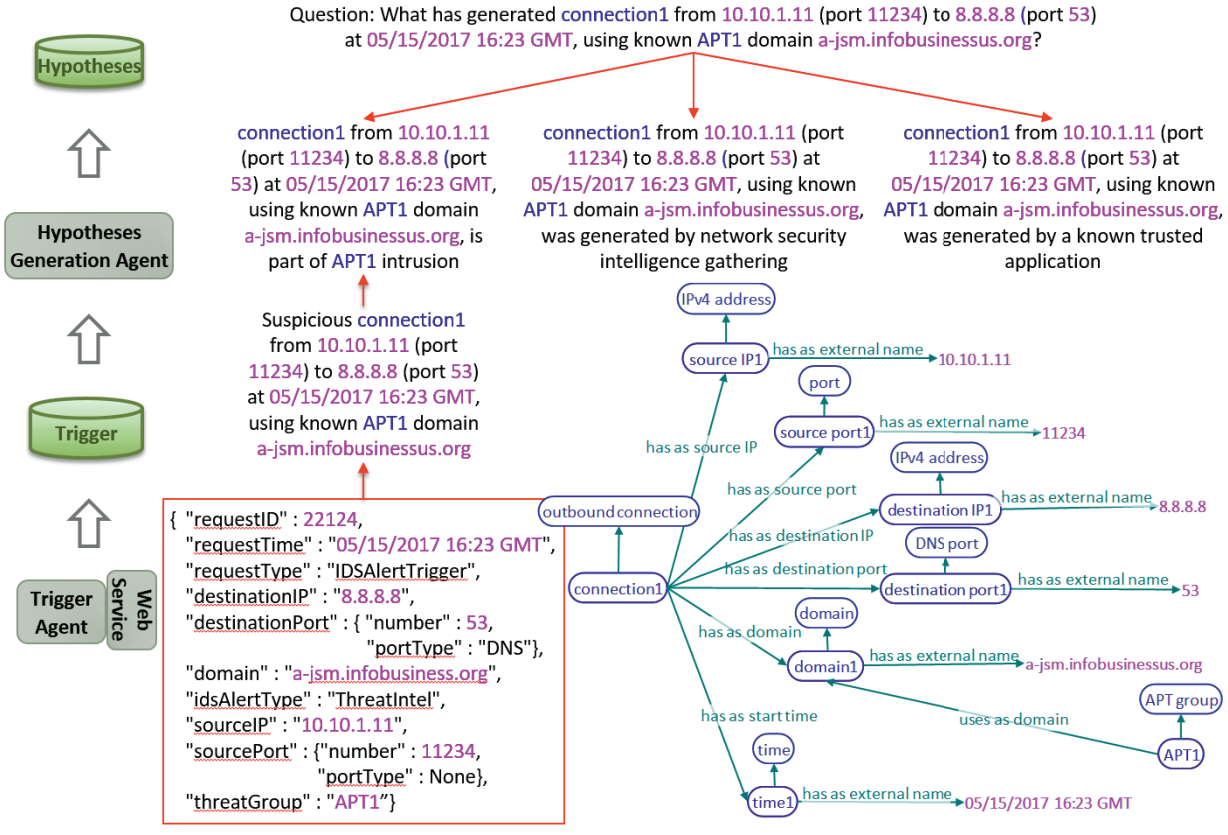

Figure 3. Automatic hypotheses generation from alerts.

connection1 from 10.10.1.11 (port 11234) to 8.8.8.8 (port 53) at 05/15/2017 16:23 GMT, using known APT1 domain a-jsm.infobusinessus.org, was generated by network security intelligence gathering

connection1 from 10.10.1.11 (port 11234) to 8.8.8.8 (port 53) at 05/15/2017 16:23 GMT, using known APT1 domain a-jsm.infobusinessus.org, was generated by a known trusted application

These are false positive hypotheses that also may explain the BRO alert.

Automatic Analysis Agent

The automatic analysis agent employs hypothesis analysis rules to build Wigmorean networks (Schum 2001; Tecuci at al. 2016a; 2016b) by decomposing all the three hypotheses from the top part of Figure 3, as much as possible, down to the level of specific evidence collection requests.

Figure 4 illustrates the decomposition of the first hypothesis which is the intrusion hypothesis. There are two main indicators of this hypothesis. The left branch investigates whether connection1 involves an APT1 command and control server. In order to make this determination, it needs more information about the APT1 domain a-jsm.infobusinessus.org. Its information needs are expressed through specific search requests to be processed by the Collection Manager, such as the following one:

Search for the IP address mapped to domain a-jsm.infobusinessus.org at time 05/15/2017 16:23 GMT

The right branch of the decomposition in Figure 4 investigates whether the program that made connection1 is an APT1 malware. To investigate this further, however, the agent needs to identify this program, and therefore it formulates another information request for the Collection Manager:

Search the computer 10.10.1.11 for the program that used port 11234 to communicate with 8.8.8.8 on port 53 at 05/15/2017 16:23 GMT

The Collection Manager, described in the next section, invokes specialized collection agents to search for this information on the network and on the host computer.

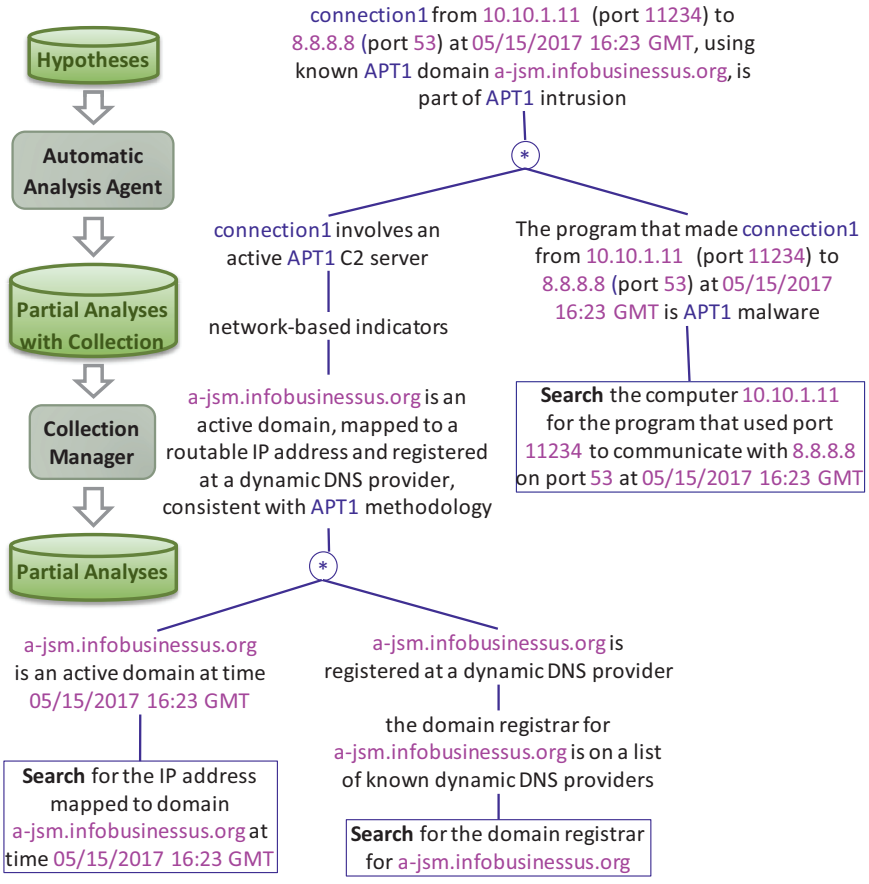

Figure 4. Automatic hypothesis decomposition and evidence search.

Collection Manager

The Collection Manager is the main integration point between the analysis agents and the CSOC infrastructure. The analysis agents know what information is needed to expand their analyses, but the search requests are in abstract form. They are not tied to specific data sources. The primary function of the Collection Manager is translating high-level (abstract) search requests into specific API calls to host and network agents, determining to which such agent to send the search request on behalf of the analysis agents, and wrapping calls to specific search agents with a JSON API. Results returned from a specific search agent to the Collection Manager are then converted into evidence and added to the KBs of the analysis agents.

Figure 5 is an overview of the Collection Manger process. When the analysis agents analyze competing hypotheses, many of the searches generated by the hypothesis-driven search process (such as the ones illustrated in Figure 4) may be the same. In order to increase performance of the system and minimize network and processing utilization, the Collection Manager performs caching of search results. Each search request is hashed and the hash value is used as a key to the cache table. A duplicate search will have a matching hash value and its result can be used instead of re-executing the search. When a search is conducted, results of each search are added to the cache with the appropriate

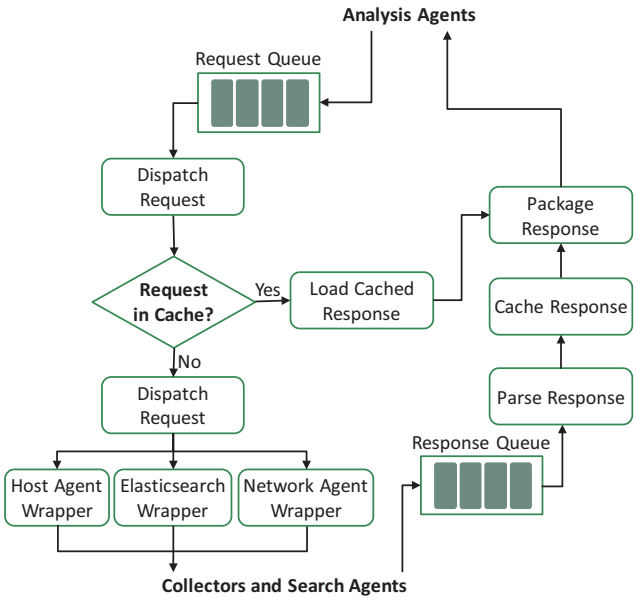

Figure 5. Collection Manager process overview.

key and a time to live (TTL) value. Once the TTL is expired, the search results are considered invalid and are purged from the cache. This step is required because the state of a computer network changes very rapidly and search results must be used near in time to the event being analyzed.

The abstract searches requested by analysis agents require evidence from multiple types of data sources available to CSOC security infrastructure. There are hundreds of security appliance, log source, and data store combinations in real-world networks. In order for the analysis agents to integrate with real networks, the Collection Manager uses a plugin architecture with search agent wrappers, allowing it to easily translate abstract search requests into requests for information from real data stores such as Elasticsearch and Splunk, on-demand search agents like Google Rapid Response (GRR 2017), Encase Endpoint Investigator (2017), and disk forensic collectors and memory forensics tools such as Volatility (2015) and Rekall (2017).

Depending on the amount of time required to collect the information, requests to an ad hoc search agent can be either synchronous or asynchronous.

To illustrate the Collection Manger's operation, consider the Search from the bottom left of Figure 4. This will lead to the invocation of the ***GetIPFromDomain*** search function from Table 1. The two results returned by this function are shown in Figure 6, an item of evidence with credibility certain (shown at upper right) and its ontological representation (to be included into the KB).

Once an analysis agent receives any result from a Search request, it attempts to further refine its analysis, as described in the next section.

Table 1. Search Function: GetIPFromDomain.

Input Parameters: ...	**Output Parameters: ...**

Search Process:
1. Connect to passive DNS provider via RESTful API
2. Request Domain/IP mapping history for *domain*
3. If the map exists:
   1. Search the map for a domain mapping in effect at time *timestamp*
   2. If the relevant record exists, add the *ipAddress* to the output parameter list
4. The Collection Manager is alerted that the search request has been completed.

Partially Completed Automatic Analysis

Figure 7 shows how the analysis in Figure 4 was refined after the automatic analysis agent has received the results of the three search requests from the bottom of Figure 4.

Our agents employs an intuitive system of Baconian probabilities (Cohen 1977; 1989) with Fuzzy qualifiers (Negoita and Ralescu 1975; Zadeh 1983) which are shown in Table 2. Notice that some of the probability intervals are associated with familiar names, such as likely (60-65%) or almost certain (95-99%).

Table 2. Probability scale.

L11	100%	certain
L10	95-99%	almost certain
L09	90-95%	
L08	85-90%	very likely
L07	80-85%	
L06	75-80%	
L05	70-75%	more than likely
L04	65-70%	
L03	60-65%	likely
L02	55-60%	
L01	50-55%	barely likely
L00	50%	lacking support

Let us first consider the left branch of the Wigmorean argumentation from Figure 4. Notice in Figure 7 that the search nodes were replaced by the corresponding evidence items returned. In this case, the credibility of each evidence item is certain. As a result, the analysis agent infers that the following two indicators of the sub-hypothesis "connection1 involves an active APT1 C2 server" are also certain:

a-jsm.infobusinessus.org is an active domain
at time 05/15/2017 16:23 GMT

a-jsm.infobusinessus.org is registered
at a dynamic DNS provider

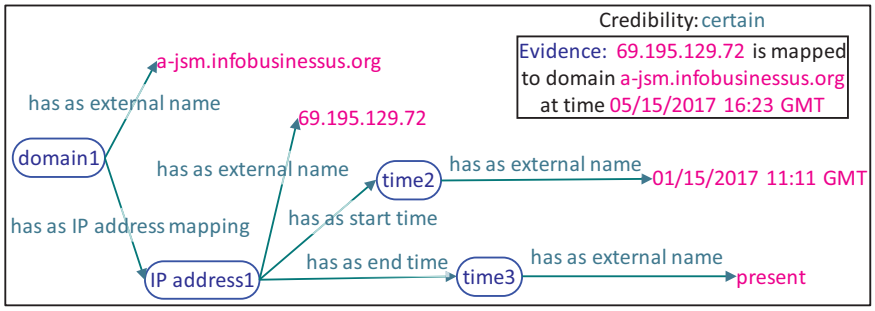

Figure 6. Results returned by the Search request from Table 1.

Because, as shown in Figure 7, the combined relevance of these two indicators is certain, the agent concludes that the sub-hypothesis "connection1 involves an active APT1 C2 server" is certain. Notice that, if only one of these two indicators would have been detected, the inferred probability would have been more than likely (70-75%). In general, the probability of a hypothesis that has n indicators, is computed based on the combined relevance of the detected indicators, and the probabilities of these indicators.

Let us now consider the right branch of the argumentation from Figure 4. In this case the result of the search was the identification of the program that has made the suspicious connection1 (i.e., svchost.exe 176). This enables the automatic analysis agent to further investigate whether this program is an APT1 malware by considering various possible indicators for specific APT1 malware, such as Auriga. This leads to additional evidence collection requests, such as that shown at the bottom right of Figure 7:

Check whether Auriga command shell is present on the host computer 10.10.1.11

However, at this point in time, the automatic analysis cannot infer the presence of the second indicator for the top level hypothesis in Figure 7, and it has to determine the probability of this top hypothesis based only on the left indicator which is certain. Because the relevance of this indicator alone is only likely, the automatic analysis agent infers that the top-level hypothesis is, at this point, likely.

After the results of the new evidence collection requests are returned by the Collection Manager, the automatic analysis agent will refine the analysis, and so on, until no further refinements are possible. At that point, the KB with the performed analysis is added to the user review queue. The cyber analyst will be alerted by the Mixed-Initiative Analysis Assistant, as discussed below.

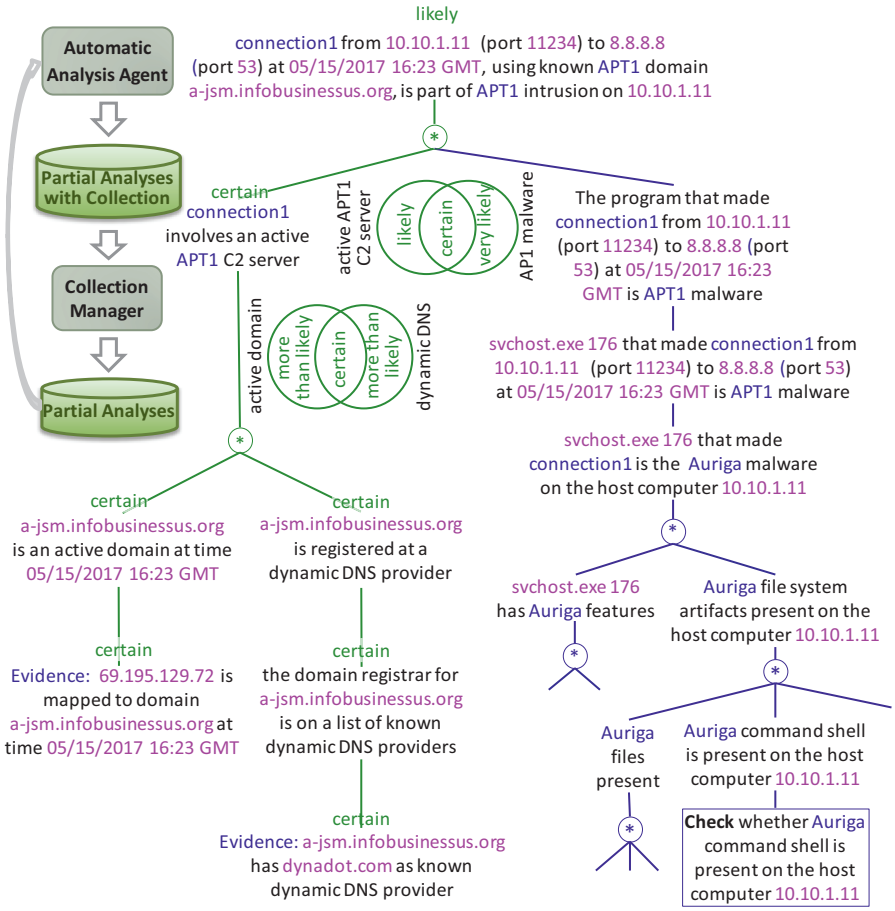

Figure 7. Automatic hypothesis analysis.

Mixed-Initiative Analysis Assistant

APTs adapt to efforts of network defenders to identify the malware. For example, Chinese Military's APT1 evolved over time to include several malware families: Auriga → Bangat → Seasalt → Kurton.

Let us consider the situation where the agents have been trained to recognize malware from the Auriga family, but they are now facing an intrusion attempt by a malware from the newer Bangat family. The collaborating agents will perform the reasoning described in the previous sections. In particular, they will develop the reasoning tree from Figure 7. The right branch of the reasoning tree checks whether the malware that made the suspicious connection1 is Auriga, but it will not be able to conclude this because the malware is Bangat. Nevertheless, the probability of the top-level hypothesis is likely (60-65%) because the left branch of the argumentation has shown that connection1 involves an active APT1 command and control server.

At this point the cyber analyst is alerted of probable suspicious activity requiring further investigation. The mixed-initiative analysis assistant is used to develop the modeling for the new malware (i.e., Bangat), by following and adapting the Auriga analysis. As a result, new reasoning rules will be learned, some existing rules that apply to both Auriga and Bangat will be refined, and the ontology will be extended with information specific to Bangat.

Status and Future Research

A first prototype of the presented system is currently under development and will be completed by the end of 2017. Future research will focus on improving agent

teaching, experimentation within a simulated CSOC, and experimental integration into a real CSOC. We expect that these agents will significantly increase the probability of detecting intrusion activity while drastically reducing the workload of the operators.

Acknowledgements

This research was performed in the Learning Agents Center and was sponsored by the Air Force Research Laboratory (AFRL) under contract number FA8750-17-C-0002, and by George Mason University. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

Allemang D. and Hendler J. 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and Owl*, Morgan Kaufmann Publishers.

Boicu, M., Tecuci, G., Marcu, D., Bowman, M., Shyr, P., Ciucu, F., and Levcovici, C. 2000. Disciple-COA: From Agent Programming to Agent Teaching, *Proc. 27th Int. Conf. on Machine Learning* (ICML), Stanford, California, Morgan Kaufman, http://lac.gmu.edu/publications/data/2000/2000_il-final.pdf

Bro, 2017. *BRO*, http://www.bro-ids.org

Cohen L. J. 1977. *The Probable and the Provable,* Clarendon Press, Oxford.

Cohen L. J. 1989. *An Introduction to the Philosophy of Induction and Probability*, Clarendon Press, Oxford.

Encase Endpoint Investigator. 2017. *EnCase Endpoint Investigator - Remote Digital Investigation Solution.* https://www.guidancesoftware.com/encase-endpoint-investigator

FireEye. 2015. APT30 and the Mechanics of a Long-running Cyber Espionage Operation, *FireEye,* April.

GRR. 2017. *Google Rapid Response: Remote Live Forensics for Incident Response*. Python. 2013. Reprint, Google, 2017. https://github.com/google/grr

Mandiant Intelligence. 2013. APT1: Exposing one of China's cyber espionage units, *Mandiant.com*.

Negoita, C. V., and Ralescu, D. A. 1975. *Applications of Fuzzy Sets to Systems Analysis*, Wiley, New York.

Rekall. 2017. *Rekall Memory Forensic Framework.* Accessed July 20, 2017. http://www.rekall-forensic.com/

Schum, D. A. 2001. *The Evidential Foundations of Probabilistic Reasoning.* Northwestern University Press.

Snort, 2017, *SNORT*, https://www.snort.org/

Suricata. 2017. *Suricata*. https://suricata-ids.org/

Symantec Security Analytics. 2017. *Network Forensics & Security Analytics*, Symantec. https://www.symantec.com/products/web-and-cloud-security/network-forensics-security-analytics

Tecuci G. 1993. Plausible Justification Trees: a Framework for the Deep and Dynamic Integration of Learning Strategies, *Machine Learning Journal,* vol.11, pp. 237-261.

Tecuci G. 1988. *Disciple: A Theory, Methodology and System for Learning Expert Knowledge*, Thése de Docteur en Science, University of Paris South.

Tecuci G. and Kodratoff Y. (eds.) 1995. *Machine Learning and Knowledge Acquisition: Integrated Approaches,* Academic Press.

Tecuci G. 1998. Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, San Diego: Academic Press.

Tecuci G., Boicu M., Marcu D., Stanescu B., Boicu C., Comello J., Lopez A., Donlon J., Cleckner W. 2002. Development and Deployment of a Disciple Agent for Center of Gravity Analysis, in *Proceedings of the Eighteenth National Conference of Artificial Intelligence and the Fourteenth Conference on Innovative Applications of Artificial Intelligence*, AAAI-02/IAAI-02, pp. 853-860, Edmonton, Alberta, Canada, AAAI Press/The MIT Press. http://lac.gmu.edu/publications/data/2002/dddacga.pdf

Tecuci, G., Boicu, M., Ayers, C., and Cammons D. 2005a. Personal Cognitive Assistants for Military Intelligence Analysis: Mixed-Initiative Learning, Tutoring, and Problem Solving. In *Proc. 1st Int. Conf. on Intelligence Analysis*, McLean, VA. http://lac.gmu.edu/publications/data/2005/Tecuci-Disciple-LTA.pdf

Tecuci G., Boicu M., Boicu C., Marcu D., Stanescu B., and Barbulescu M. 2005b. The Disciple-RKF Learning and Reasoning Agent, *Computational Intelligence*, Vol.21, No.4, pp. 462-479.

Tecuci, G., Boicu, M., Cox, M. T. 2007a. Seven Aspects of Mixed-Initiative Reasoning: An Introduction to the Special Issue on Mixed-Initiative Assistants, *AI Magazine*, 28(2):11-18, Summer.

Tecuci, G., Boicu, M., Marcu, D., Boicu, C., Barbulescu, M., Ayers, C., and Cammons D. 2007b. Cognitive Assistants for Analysts, *Journal of Intelligence Community Research and Development* (JICRD). Also in Auger J. and Wimbish W. eds. *Proteus Futures Digest*, 303-329, Joint publication of National Intelligence Univ., Office of the Director of National Intelligence, and US Army War College Center for Strategic Leadership.

Tecuci G., Boicu M., Marcu D., Boicu C., and Barbulescu M. 2008. Disciple-LTA: Learning, Tutoring and Analytic Assistance, *Journal of Intelligence Community Research and Development.* http://lac.gmu.edu/publications/2008/Disciple-LTA08.pdf

Tecuci, G., Marcu, D., Boicu, M., Schum, D. A. 2015. COGENT: Cognitive Agent for Cogent Analysis, in *Proceedings of the 2015 AAAI Fall Symposium "Cognitive Assistance in Government and Public Sector Applications,"* 58-65, Arlington, VA, Technical Report FS-15-02, AAAI Press, Palo Alto, CA. http://lac.gmu.edu/publications/2015/Cogent-overview.pdf

Tecuci G., Marcu D., Boicu M., Schum D.A. 2016a. *Knowledge Engineering: Building Cognitive Assistants for Evidence-based Reasoning*, Cambridge University Press.

Tecuci, G., Schum, D. A., Marcu, D., and Boicu, M. 2016b. *Intelligence Analysis as Discovery of Evidence, Hypotheses, and Arguments: Connecting the Dots,* Cambridge University Press.

Volatility. 2017. *Volatilityfoundation/Volatility. GitHub*. Accessed May 15, 2015. https://github.com/volatilityfoundation/volatility

W3C. 2004. http://www.w3.org/TR/rdf-schema/

Zadeh, L. 1983. The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, *Fuzzy Sets and Systems*, Vol. 11, pp. 199 - 227.

Zimmerman C. 2014. *Ten Strategies of a World-Class Cybersecurity Operations Center.* MITRE Corporation.