An Expert System for Mitigation Actions

Ilkka Karanta, Mika Rautila
VTT Technical Research Centre of Finland
Espoo, Finland
{Ilkka.Karanta, Mika.Rautila}@vtt.fi

***Abstract*—This paper describes an approach, based on ontologies and expert system technology, for assisting the mitigation of advanced persistent threat (APT) attacks against critical infrastructures. We describe the approach, and a prototype expert system based on it. We delineate a case study, involving an APT against a financial information infrastructure. Finally, we outline some conclusions and recommendations for future work.**

I. Introduction

Advanced persistent threats (APT) are a relatively new information security threat, but they have become one of the most notorious security threats nevertheless, perhaps due to some well-publicized cases. APT's are exceptionally dangerous, but especially dangerous when targeting critical infrastructures, due to the exceptional value and propensity to damage of the latter.

The defenders of the security of the critical infrastructure - typically, the personnel of a security operating center (SOC) - need assistance and advice in their task: they might lack experience in the particular threat, they might be undermanned, and busy. It is unrealistic to expect that competent experts from beyond the SOC would be available at the time of the attack, or in recovery after it. Existing incident management systems generally only collect all incident-related information into one place and handle it as a whole, but do not provide active assistance in mitigation. If mitigation-related information is only available in a passive form, for example as a part of a help system, the user is unlikely to find the needed information or even search for it in a meaningful way. Therefore, the idea of an automated system providing assistance is tempting.

Such an automated system should contain a description of the general features of the threat, the system to be defended, and ways to mitigate the threat over the attack lifecycle. Furthermore, it should be able, one way or another, to gain information on the particular situation, and make inferences from all the information available to it, providing the SOC personnel with solid, relevant, consistent and unambiguous advice, based on up-to-date knowledge on the domain. This sets exceptional demands on the automated system.

We describe a prototype of such an advanced automated advisory system. The central tools and methods we use, expert systems and ontologies, meet the demands by together providing a versatile platform for knowledge representation and reasoning. To demonstrate our point, we have built a prototype expert system for APT mitigation in critical infrastructures. We demonstrate its use with a case study of an APT against a critical infrastructure in the financial sector.

The rest of the paper is organized as follows. We first consider APT's and their implications to security. Then we describe how our expert system based approach may help in solving some important issues in mitigation. This is followed by a description of the small prototype we constructed to illustrate this. Finally, we consider what role expert systems could have in future mitigation management.

II. Advanced persistent threats against critical infrastructures

A. Advanced persistent threats

Advanced persistent treats [4], [14] are cyberattacks aiming at gaining unauthorized access to the targeted system for a long period of time without being detected.

The attacks are precisely targeted, and often contain customized attack steps in contrast to more traditional attacks that exhibit the same behavior against all targets. An attack may involve an email formulated so that the recipient feels that her or his obligation as an employee is to follow the http link, which of course is malicious.

The phases of an APT attack may be divided as follows:

- Initial reconnaissance phase, which aims at finding potential ways to establish beachhead at the target system. This includes studying the target systems, their processes, people, partners and vendors, and this may take several months.
- In the incursion phase, customized attacks are created to intrude to the target system and establish a beachhead. The attacker must ensure permanent access to the target system.
- Once inside, in the internal reconnaissance phase the attacker collects information on the internal network. It is essential that the attacker remains undetected.
- The attacker expands presence in the internal network towards the goal of the mission and ensures continued control of the access channel.
- The final step is to complete the mission.

Each phase may require a significant amount of resources and may take several months to complete. If the mission is to collect data from the target organization, the final step may take years.

ISSN 2305-7254

Being undetected is crucial for the attacker. If the attack is detected, a substantial amount of work may be lost - even worse, the security of the target system may be strengthened which may make future attacks more difficult. Therefore, only small amounts of information will leak from the attack over possibly a lengthy period of time, and each piece of information is insignificant in itself. Only when assembled as a part of larger picture do these pieces of information constitute a pattern that indicates an ongoing attack, and gives out some of its features.

This relative scarcity and vagueness of information points to the need for a method to analyze the vast amounts of data produced by any system, form a meaningful summary of it, and detect hidden patterns in them. Although these kinds of systems have received a lot of research attention, it seems that relatively little has been done to augment such systems to provide mitigation action recommendations.

As APT attacks require a lot of resources and high capability, it is believed that APT groups are supported by some nation-states.

B. *APT against critical infrastructures*

By critical infrastructure, we mean assets that are essential for the functioning of a society and economy [9] such as

- utilities (electricity, gas, water),
- communication,
- food production and distribution,
- transportation,
- financial services

These systems are nowadays highly automated, controlled by computers, and connected to the Internet.

A special feature of many of the infrastructure systems is that their life cycle is very long. This means that the systems contain old components that are not fully supported anymore. For instance, the automation systems may contain components that use old software for which security updates are not available anymore. Even worse, the system may contain so old software that security threats were not considered relevant when the software was developed. Most likely, those components contain security vulnerabilities that can be exploited.

If the attacker is interested in causing damage, then the automation systems of critical infrastructure are potential targets. If the attacker can control the control system, then the attacker can usually damage the system. For example, it is speculated that the Stuxnet malware caused damage to the uranium enrichment facility at Natanz, Iran.

There are interdependencies between the critical infrastructures. Functioning of electricity network depends on the communication infrastructure, and the communication infrastructure depends on the electricity network. Many of the critical infrastructures depend on power and communication infrastructure. These interdependencies make the critical infrastructures even more critical.

C. *The defense lifecycle*

In any attack, including APT's, a distinct set of phases can be distinguished. This ordered set of phases is called the attack or incident lifecycle. Incident lifecycles have been specified by various parties, for example ENISA [10].

Here we use a simple lifecycle model presented in Fig. 1. We developed it, as we found the existing lifecycle models to be too complex for our purposes. Furthermore, existing incident lifecycle models usually take the attack point of view rather than the defense point of view that is more natural from the mitigation viewpoint.

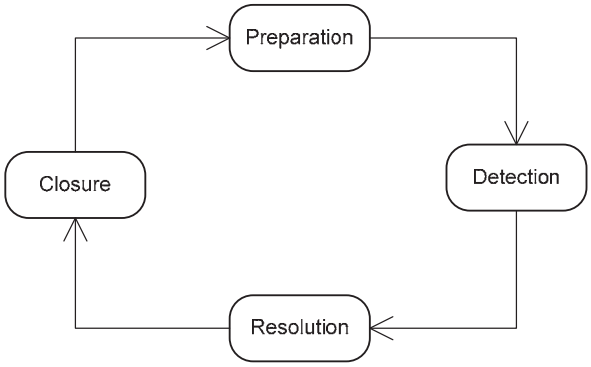

Fig. 1. Incident defense lifecycle

The lifecycle consists of the following phases:

- Preparation. No attack has been detected, and the system is used for its intended purpose. Mitigation might consist e.g. of user education, operator training, and purchase of incident detection and management software.
- Detection. It is recognized that something exceptional is occurring. Attack mode and possibly its source are identified, and the decision to take action is made. Mitigation consists of various diagnostic measures, and for example installation of honeypots.
- Resolution. Based on the information available, an applicable mitigation strategy is selected and its actions are put into use. Mitigation actions might consist for example of eradication of viruses from executable files, network use restrictions and redirection of Internet traffic.
- Closure. The system is brought back to normal operation. Mitigation consists of taking the lessons learned from the incident, and putting their implied conclusions into action.

III. THE SOLUTION APPROACH

A. *Methodological basis*

Ontologies [12] are a way of organizing the central concepts of a problem domain for knowledge representation. Although a relatively recent development, the field is relatively mature with widely used development methods [6] and tools such as the Protégé editor [11]. An ontology is a conceptual model where

concepts are organized in a hierarchical or network manner, and the relevant relationships between the concepts are represented explicitly.

Ontologies have been applied widely in the information security domain. For example, [1] describes a security incident ontology, [13] an ontology for intrusion detection, and [1] an IT asset ontology. A relatively recent review of information security ontologies is provided in [2].

Ontologies provide several benefits in the representation of complex domains:

- They provide a meaningful and easy to understand way of representing domain concepts and relations between them,
- They provide a systematic backbone for representing domain knowledge,
- They are universal, easily portable, and adaptable to various needs.

Expert systems [5], [7] are a way of representing and reasoning with knowledge. In an expert system, domain knowledge is encoded in a symbolic manner to a unified presentation of domain entities and inference rules. The presentation usually consists of facts and rules or of a network.

Expert systems provide several advantages in knowledge representation and reasoning:

- Knowledge is explicitly and easily available to domain experts for e.g. assessment,
- Expert systems can explain their reasoning, and the way an expert system arrived at a conclusion can be easily tracked,
- Expert systems can adapt to new or changed information by using machine learning techniques.

Expert systems have been applied in a wide variety of domains, including

- system configuration (assembling proper components of a system in a proper way),
- diagnosis (infer underlying problems based on observed evidence) [8],
- interpretation (explain observed data),
- monitoring (compare observed data to expected data to judge performance),
- planning (devise actions to yield a desired outcome),
- prognosis (predict the outcome of a given situation), and
- remedy (prescribe treatment for a problem).

B. Attack and mitigation ontology

We designed a small security management ontology to serve as the backbone of knowledge representation in the expert system. The role of the ontology is to represent general knowledge about the system and the attacks that might target it.

The main elements of the ontology are represented in Fig. 2.

An asset is a part of the information system, for example, a server, program, a database or such. A threat is directed at some specific assets in the system. It utilizes vulnerabilities in the system to achieve its goals. The purpose of a countermeasure is to mitigate some threats. In doing so, the countermeasure realizes a security goal.

Each main concept is general, and may consist of several concepts that are special cases of the main concept. These special case concepts and their relationship to the main concept are represented by inheritance hierarchies, An example is given in Fig. 3.

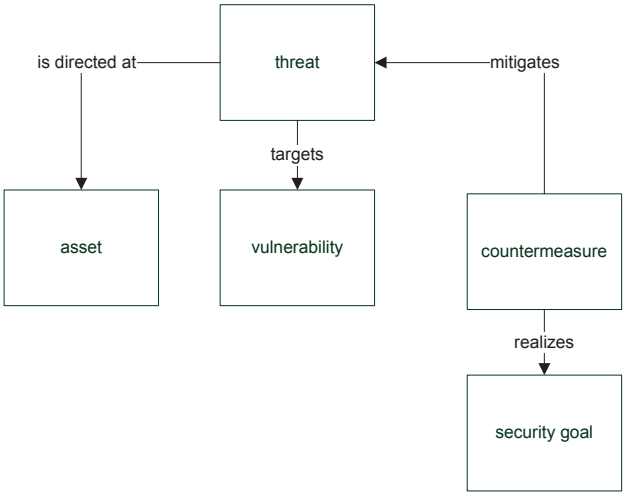

Fig. 2. The main concepts of the security ontology and their principal relationships

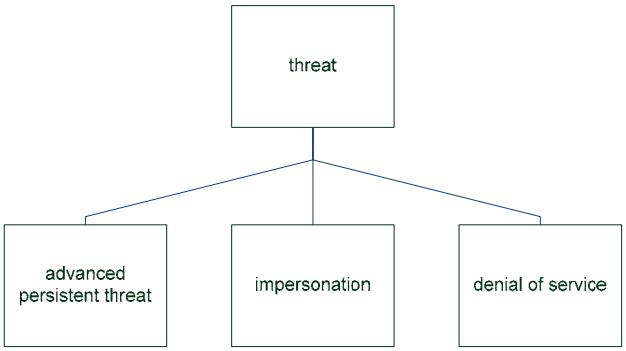

Fig. 3. A part of the inheritance hierarchy: the threat concept and some of its special cases

Each concept and relationship plays also a role in the expert system. For example, the user may specify security goals, and the ES then selects countermeasures that satisfy those goals in mitigating given threats.

C. Expert system

We deemed that a rule-based expert system would provide the functionality we needed, and a natural way of representing the knowledge and inference needed in mitigation assistance.

A rule-based expert system consists of facts and rules. Facts represent what is considered true about the problem domain. In practice, the ontology is represented as facts. The general knowledge about the problem domain contained in the ontology is complemented by facts describing knowledge about the situation; for example, the current phase of the attack life cycle is represented by a fact.

Rules describe what can be concluded from the facts. Each rule contains a premise part and an action part. The premise part tells under what conditions the rule will be applied. If the premises are true, then the actions listed in the action part are taken.

An example rule, which is an actual rule from our expert system, is presented in Fig. 4.

```
(defrule install-honeypots
  (problem (phase preparation) (threat spoofing))
  (target-system (countermeasures $?countermeasures))
  (not (member$ honeypot $?countermeasures))
  =>
  (printout t $?countermeasures crlf)
  (printout t "RECOMMENDATION: install honeypots" crlf))
```

Fig. 4. An example rule represented in the CLIPS language

The rule states that if we are in the preparation phase of the incident lifecycle, considering the mitigation of spoofing, and we have not installed honeypots yet in our system, then the expert system will recommend us to install honeypots.

IV. Implementation

A. The CLIPS language and environment

We used CLIPS (C Language Integrated Production System) in the implementation of the expert system. CLIPS consists of an eponymous rule-based language and an expert system development and runtime environment. Initially developed by NASA's Johnson Space Center from 1985 to 1996, it has been used in commercial and non-commercial settings for more than 30 years, and is free and open-source.

The CLIPS programming language provides facilities for rule-based programming in the language core, and for object-oriented programming in the CLIPS Object-Oriented Language (COOL) extension. The rule-based subsystem uses frames as the structured knowledge representation formalism, but contains also features for more conventional data types and data structures familiar from mainstream programming languages. The language core provides versatile support for conventional imperative programming in the spirit of Pascal, C and Java. It also supports functional programming, and the CLIPS programming language syntax resembles the syntax of the LISP language. Thus, software can be developed using the CLIPS programming language within the imperative, rule-based, functional and object-oriented paradigms, or any combination of these, which makes the CLIPS language exceptionally versatile.

The expert system we developed uses purely the rule-based parts of the language, using frames in knowledge representation. We made this choice because only one of us had previous experience of the CLIPS language, and even that was extremely minor and outdated; therefore, familiarization with the whole rich feature set of the language would have taken a disproportionate amount of time considering the total time available.

During development, it became obvious that the CLIPS language and system provide excellent support for these kind of endeavors. The whole development process, including familiarization with the CLIPS language and system, software design and implementation, took approximately one person-month of development time. We found the answers to all questions but one concerning the language and system from CLIPS manuals. Even the one time we needed external help, we got an answer within two days from posing the question to the CLIPS development group in Google Groups. The question concerned a CLIPS language feature we did not find in the manuals; it turned out that the feature has not been implemented in the CLIPS language, but we figured a way to circumvent this in the same evening we received the answer.

B. Development of the expert system

The starting point of the development of the expert system was that eventually, the system would support the whole incident lifecycle, and arbitrary target systems against arbitrary security threats. This notion provided the core objectives for both the program's design and its implementation.

The design of the expert system (ES) proceeded on two fronts. On the knowledge-modeling front, an ontology was first sketched to serve as the conceptual backbone model of the ES. The Protégé ontology editor and framework for constructing intelligent systems [11] was used in a minor way, its role being to serve as a graphical editor of the ontology model. This use naturally utilized only a tiny fraction of the features of this powerful, advanced and versatile ontology development environment.

The ontology development did not follow any development process, but rather was based on superficial examination of existing information security ontologies, our pre-existing knowledge on the information security domain, and the modest experience of the former author on ontology development. The central objectives of the ontology were, besides the one of providing the conceptual backbone of the knowledge model, to keep the result as small and simple as possible, in the spirit of prototyping. The development proceeded so that first, some central concepts of the information security domain were chosen, and then the most important relations occurring naturally between those concepts were specified. This approach served the development of this small ontology well, and only slight changes were made to the ontology in the ES implementation phase. A simplified visual representation of the result is to be found earlier in this paper as Fig. 2.

The other front of the development was the construction of the ES (program) itself. Here, too, the main guiding principle of design and implementation was simplicity in all respects. For example, it was decided that no effort to modularize the ES would be made, although CLIPS provides ample support for it.

Knowledge needed in the expert system was acquired in four interview sessions, each lasting two hours. The former author was in the role of the interviewer, and the latter author was the domain expert. The relatively small amount of worktime spent on this crucial task is explained by the facts that both of us had a clear view what knowledge and information was needed, and the latter author had a clear vision of how mitigation of an APT should be conducted, and what dependences there are between various mitigation actions.

C. The expert system

At the present, the ES advises the user on what mitigation actions to take against APT in the resolution phase. The system consists of 24 rules (and the frames needed in them), and thus can be considered to be a relative small expert system.

A short summary of the identified mitigation actions and their mutual precedence relations is presented in Table I.

TABLE I. THE MAIN MITIGATION ACTIONS AGAINST AN APT, AND THEIR PREDECESSORS

#	Description	predecessors
1	Disconnect unnecessary Internet connections in the network physically	
2	Update firewall rules to allow only necessary traffic from the outside world from known sources	
3	If a subsystem is not protected by a firewall, disconnect all the workstations in it from the network	
4	Find out about the current network configuration	1, 2, 3
5	Find out which computers in the network are contaminated	1, 2, 3
6	Find out what configurations of the currently used version of software used are safe.	1, 2, 3
7	Find out which computers in the system have been contaminated	1, 2, 3
8	Disconnect the contaminated computers from the system (if not already done), and connect them to a single network that is isolated from the rest of the system if possible	7
9	You can open the network connections of uncontaminated computers	8
10	Reduce user privileges to a minimum required to enable their work, take away administrative privileges from anyone who does not need them	9

V. A CASE STUDY

A. ECOSSIAN project

We developed the ES prototype as a part of ECOSSIAN project. ECOSSIAN is an EU-funded project that started in 2015 and will end in 2017. The mission of ECOSSIAN is to improve the detection and management of highly sophisticated cyber security incidents and attacks against critical infrastructures by implementing a pan-European early warning and situational awareness framework with command and control facilities. One goal is a prototype that facilitates preventive functions like threat monitoring, early indicator and real threat detection, alerting, support of threat mitigation and disaster management. The main deliverable of the project is a prototype system with which various solutions to pertinent security problems and technical issues may be demonstrated.

B. Case description

In ECOSSIAN Work Package 5, three cases were specified for demonstration purposes [3]. The first concerns an attack against the SCADA system of an Irish gas operator, targeting the gas pipeline system. The second concerns an APT against a financial infrastructure in Italy. The third concerns an attack against the Portuguese railway system. Here we consider only the second demonstration, because we constructed the ES to complement it.

A financial infrastructure contains plenty of sensitive information, of potential interest to intruders. For example, the uncontrolled disclosure of solvency of a company might have severe impacts, potentially leading to the complete disruption of its business; this could result from e.g. loss of lines of credit, inability to insure operations, increased loan repayments, increased premiums, or supply chain operators not wishing to continue to do business.

The demonstration scenario concerns and APT against a financial infrastructure. The demonstration infrastructure consisted of a network protected by a firewall, and three subnetworks. The first contained three work stations, and it did not have a separate firewall. The second contained two servers, protected by a dedicated firewall. The third subnetwork was the network of a security operating center (O-SOC) guarding the infrastructure, containing three work stations, and protected by a firewall. The structure and assets of the infrastructure were represented with the concepts of the ontology, using frames in the ES.

In the demonstration, the attacker first gathers information about the target system, and tailors an e-mail attack against a detected user. After the user clicks a link in the mail, a malware is installed to the user's computer, enabling persistent control of it. The malware collects information about the surrounding network and computers. Little by little, the attacker finds ultimate targets, and detects their associated vulnerabilities.

Some of these actions produce security events that are processed by a sensor. The sensor detects an abnormal pattern (traffic between nodes that have not exchanged information previously), and triggers an alarm to the O-SOC console. At this point, the malware signature is unidentified. The O-SOC analysts decide to investigate the event and generate an incident report. The report, together with associated information, is sent to N-SOC where an analyst identifies the incident as a potential APT attack.

At this point, the analyst takes the expert system into use. It interactively provides assistance to the analyst on what mitigation actions to take at which phase. The analyst forwards the mitigation recommendations to the O-SOC, where the security engineers get the attack under control.

VII. Conclusions

We have described an expert system that gives advice on mitigation actions against an APT, in the resolution phase of the attack lifecycle. The approach of combining expert system technology with ontologies proved to be good, providing the needed functionality with small amount of work effort. The resulting system is conceptually simple and clear, and thus it is easy to maintain and enhance it.

A natural way to add functionality to the ES is to enhance it to cover all phases of the attack lifecycle, and all plausible threats against the system.

There are many ways to enhance the functionality of the ES. For example, methods can be constructed to automatically collect and retrieve information from the affected computers and network. Also diagnostic facilities of the ES could be extended so that it may be used in the automatic detection of attacks, and in clarifying the root causes of observed system behavior.

From the information security point of view, the role of expert systems in the mitigation of attacks (for example APT) might be as analyzers and interpreters of attack-related information. An APT, by default, lasts a long time and generates lots of information - logs, measurements etc. - that is tedious and difficult to analyze by hand, and would require great expertise in both computer security and data analysis. The expert system could codify knowledge - statistical, security-related, data mining and so on - into a meaningful whole, and conduct statistical and other analyses automatically to provide human operators with meaningful analysis results and recommendations that could greatly facilitate mitigation. In this way, big data and machine learning could be harnessed to serve information security management.

Eventually, mitigation actions might be implemented as programs, and the role of the expert system could be to automatically conduct mitigation after a threat has been recognized.

Acknowledgment

This work was funded by the European Union FP7 project ECOSSIAN (607577). We would like to thank Pia Olli for helpful comments on the manuscript.

References

[1] H. Birkholtz, I. Sieverdingbeck, K. Sohr, C. Bormann, "IO: An interconnected asset ontology in support of risk management processes", *in Seventh International Conference on Availability, Reliability and Security (ARES), 2012*. IEEE, 2012.

[2] C. Blackwell, "A security ontology for incident analysis", in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research,* ACM, 2010.

[3] C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, M. Piattini, "A systematic review and comparison of security ontologies", *in Third International Conference on Availability, Reliability and Security, ARES 2008*. IEEE, 2008.

[4] R. Brewer, "Advanced persistent threats: minimising the damage", *Network Security*, Volume 2014, Issue 4, April 2014, pp. 5-9.

[5] G. Brost, M. Gall (editors). Demonstration scenarios definition. ECOSSIAN project deliverable D5.3, deliverable reference number SEC-607577 / D5.3/ 1.0, May 23, 2016, 35 pages, unpublished.

[6] J.C. Giarratano, G.D. Riley, *Expert Systems - Principles and Programming*, Fourth Edition. Boston: Course Technology, 2005.

[7] A. Gómez-Pérez, M. Fernández-López, O. Corcho. *Ontological Engineering - with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. London: Springer, 2004.

[8] P. Jackson, *Introduction to Expert Systems*, Third Edition. Harlow: Addison-Wesley, 1999.

[9] E. Keravnou, L. Johnson, *Competent Expert Systems - a Case Study in Fault Diagnosis*. London: Kogan Page, 1986.

[10] T. Macaulay, *Critical Infrastructure - Understanding Its Component Parts, Vulnerabilities, Operating Risks, and Independencies*. Boca Raton, FL: CRC Press, 2009.

[11] M. Maj, R. Reijers, D. Stikvoort, *Good Practice Guide for Incident Management*, European Network and Information Security Agency (ENISA), (2010), 110 pages.

[12] M. Musen, "The Protégé Project - a look back and a look forward", *AI Matters*, vol. 1, June 2015, pp. 4-12.

[13] S. Staab, R. Studer (eds.), *Handbook on Ontologies*, Second Edition. Berlin, Heidelberg: Springer, 2009.

[14] C. Tankard, "Persistent threats and how to monitor and deter them", *Network Security*, Volume 2011, Issue 8, August 2011, pp. 16-19.

[15] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling Computer Attacks: An Ontology for Intrusion Detection", in *The Sixth International Symposium on Recent Advances in Intrusion Detection*. 2003: Springer.