Multi-agent System for APT Detection

Wim Mees, Thibault Debatty

Multi-Agent System for APT Detection

Wim Mees
Royal Military Academy
Brussels, Belgium
Email: wim.mees@rma.ac.be

Thibault Debatty
Royal Military Academy
Brussels, Belgium
Email: thibault.debatty@rma.ac.be

Abstract

Advanced Persistent Threats (APTs) are targeted cyber attacks committed over a long period of time by highly skilled attackers. The ever increasing number of successful attacks indicates that classical network protection solutions (firewalls, Intrusion Detections Systems, proxies etc.) are no longer sufficient. Therefore, in this paper we propose a new system that combines multiples approaches using advanced aggregation techniques to achieve a better detection performance. We also test the system on real data from a small corporate network, and show that our system is able to attain a high probability of detection to probability of false alarm ratio.

1. Introduction

Advanced Persistent Threats are targeted cyber attacks committed over a long period of time by highly skilled attackers.

An example of an APT is the Miniduke attack that targeted the governments of at least 20 countries in 2013. The malware targeted government computers in the Czech Republic, Ireland, Portugal and Romania along with think tanks, research institutes and healthcare providers in the United States. The malware used Twitter and Google to get instructions and updates. It allegedly infected PCs when victims opened a cleverly disguised Adobe PDF attachment to an email, which was specifically tailored to the target. The attachment referred to highly relevant subjects like foreign policy, a human rights seminar, or NATO membership plans.

Such attacks are becoming evermore sophisticated and manage to bypass the state of the art commercial-off-the-shelf solutions that are currently in place. The attackers regularly succeed in remotely controlling hosts in our networks long enough to locate the information they are after, gain access to it and finally exfiltrate sensitive data. APT attacks have therefore become a major concern for network security professionals around the world.

Therefore, in this paper we propose a new system that combines multiples approaches using advanced aggregation techniques to achieve a better detection performance.

The rest of the paper is organized as follows. In Section 2 we present current APT detection techniques along with aggregation operators. In Section 3 we present the detection agents we implemented, and in Section 4 we explain how the evidences produced by these agents are aggregated to produce a single suspiciousness score. In Section 5 we test the system on real data from a small corporate network. Finally, in Section 6 we present our conclusion and directions for future work.

2. Related work

2.1.APT Detection

APTs usually use zero-day vulnerabilities that cannot be detected by antivirus or other security softwares. Furthermore they all use different and novel attack patterns, which makes them very difficult to detect.

Nevertheless most APTs have in common the fact that they establish a command and control (CnC) channel with the outside world to receive new instructions and to exfiltrate data. This channel is necessarily tunneled through one of the protocols that is allowed at the level of the choke points that interconnect the corporate network with the outside world.

Protection against APTs currently mainly relies on classical tools like firewalls, Intrusion Detections Systems (IDS), proxies etc. The ever increasing number of successful APT attacks indicates that these commonly used solutions are no longer sufficient.

Some companies started proposing dedicated tools, like Verint CYBERVISION Advanced Detection System, FireEye Threat Prevention Platform and ISC8 Cyber adAPT system. But these tools are largely derived from the classical above mentioned tools. Moreover, their closed-source structure makes it very hard to objectively assess and compare their effectiveness.

Therefore, in this paper we propose a new system that combines multiple approaches to achieve a better detection performance. Our system has the additional advantages that it is open and extensible, which make it possible to integrate future detection tools and approaches. This also make it possible to easily adapt the system to specific needs and infrastructure, and to objectively measure its effectiveness. Our approach also takes the human analyst into account, which will remain an important part of the APT detection system.

To achieve this goal, the system uses multiple analysis agents and aggregates the suspiciousness scores attributed by the agents using advanced aggregation operators.

2.2Aggregation Techniques

We will here consider aggregation techniques on the real interval $I = [0, 1]$, representing the degrees of suspiciousness as they are produced by the APT detection agents.

In general, an n-ary aggregation function can be any non-decreasing function $A^{(n)} : I^n \to I$ that fulfils the boundary conditions $\inf_{x \in I^n} A^{(n)}(x) = \inf I$ and $\sup_{x \in I^n} A^{(n)}(x) = \sup I$. Moreover, when dealing with a variable number of inputs, for instance when the number of agents that produce evidence can vary from one client host to another, an extended aggregation function is required. This can be any function $A : \bigcup_{n \in \mathbb{N}} I^n \to I$ such that $\forall n > 1 : A^{(n)} = A_{|I^n}$ is an n-ary aggregation function and for $n = 1 : A^{(1)}$ is the identity on I.

Dubois and Prade [1] distinguish four categories of extended aggregation functions: those generalizing the notion of conjunction, those generalizing the notion of disjunction, averaging aggregation functions, and finally mixed aggregation functions which do not belong to any of the other three categories.

For our application, conjunctive aggregation functions are too pessimistic. Indeed, if several agents report a high degree of suspiciousness, but one agent is doubtful and reports a low degree of suspiciousness, the aggregated result would be determined by this last agent. The convincing evidence of the other agents would thus be ignored, which is not the behaviour we want. Moreover, if the undecided agent would have hesitated even more and had not produced an output at all, the aggregated suspiciousness level would have been a lot higher, which is counter-intuitive.

In the same way, when a single agent reports a high score, the fact whether or not a large number of other agents confirm this observation is largely ignored by the maximum-type operators that are used for a disjunctive aggregation. This is again not what we are looking for. A client host that is reported by a number of agents with a high suspiciousness degree, should be assigned a significantly higher aggregated suspiciousness level than one that is strongly supported by just one agent and received a low score from the others.

Therefore we will base the multi-agent evidence aggregation on an averaging aggregation function. Some well-know options are the arithmetic mean $M = \frac{1}{n} \sum_{i=1}^{n} x_i$ and the quasi-arithmetic mean $M_f = f^{-1}(M(f(x_1), \ldots, f(x_n))$ with $f : I \to [-\infty, \infty]$ a continuous strictly monotonic function. The choice of the aggregation function and of the function f is a way of bringing domain knowledge into the aggregation process.

If we want to bring in even more domain knowledge, we can do so in the form of a normal weighting vector $w = (w_1, \ldots, w_n)$ with $\forall i : w_i \geq 0$ and $\sum_{i=1}^{n} w_i = 1$. Some typical examples of weighted aggregation functions are the weighted arithmetic mean $M_w = \sum_{i=1}^{n} w_i x_i$ and the weighted quasi-arithmetic mean $M_{fw} = f^{-1}\left(\sum_{i=1}^{n} w_i f(x_i)\right)$.

A major problem with the operators we have discussed until now, however, is that the addition of supporting evidence can result in a reduction of the aggregated suspiciousness level. Indeed, consider a situation where three agents report evidence for a client host, for instance x_1 = 0.8, x_2 = 0.7, and x_3 = 0.3. The basic arithmetic mean function $M(x_1, x_2, x_3)$ will produce an aggregated evidence level of 0.6. Consider now that a fourth agent joins in and reports a suspiciousness level x_4. The resulting arithmetic mean

value $M(x_1, x_2, x_3, x_4)$ for values of x_4 ranging from 0 to 1 is shown in figure 1.

A solution to ensure that additional agents don't lower the aggregated output, is the use of ordered operators [4], such as the ordered weighted average operator (OWA) $M'_w = \sum_{i=1}^{n} w_i x'_i$ and the ordered weighted quasi-arithmetic operator $M'_{fw} = f^{-1}\left(\sum_{i=1}^{n} w_i f(x'_i)\right)$ with x'_i the i-th order statistics from the sample $(x_1, \ldots, x_n)$.

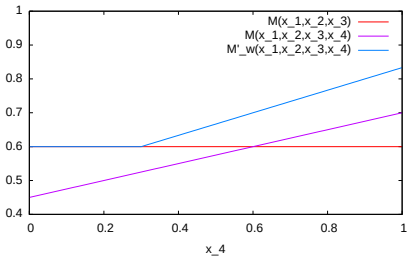

Figure 1: arithmetic mean versus OWA

Indeed, when we use for instance an OWA operator with weights $w = (1/3, 1/3, 1/3, 0)$, the resulting $M'_w(x_1, x_2, x_3, x_4)$ in figure 1 clearly shows that the additional evidence x_4 does not influence the final result as long as its level remains below $\min(x_1, x_2, x_3)$, yet pushes the aggregated result higher when it surpasses this level.

3. Multi-Agent System

The challenge when trying to detect the CnC channel of a previously unseen APT, is to distinguish it from the background noise of user initiated connections, for instance while surfing the web, as well as from other software initiated connections, for instance originating from operating system and client software automated updates, VoIP clients or cloud services that tunnel through HTTP, etc.

Since we want to detect previously unseen malwares, we cannot simply look for activity that matches hand-crafted signatures or precise behavioural descriptions, resulting from the reverse engineering of known malware samples. We will rather have to implement a number of detection algorithms that look for generic behavioral patterns that we have observed over a wide variety of malware instances and aggregate evidence produced by a wide range of agents in order to reduce

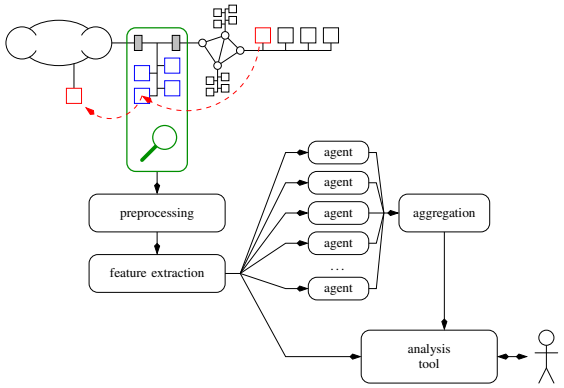

Figure 2: detection system architecture

the inevitably high false alarm rate of each individual pattern.

Modern corporate network environments typically contain a wide variety of platforms. They may even host legacy systems that are operating on old operating system and application software versions, for which security updates are no longer available. Deploying the APT detection algorithms as an endpoint security solution would require supporting a wide range of operating systems and configurations. Therefore we chose to perform the detection at the level of the choke point with the Internet, as is shown in figure 2.

The detectors are implemented as agents, that each independently verify a different characteristic of malware behavior. Since the aggregation is performed over all available sources of evidence, one can over time easily retire agents when they are no longer relevant or add new agents that verify new patterns as they are discovered.

Some examples of detectors that have already been implemented for outgoing HTTP requests include:

- *Frequency analysis:* Malwares will often wait for a fixed period of time between outgoing connections to poll their CnC server. The detector looks for this periodicity based on the timestamps of the requests.
- *Time-domain impulse detector:* Human-initiated connections are characterized by periods of bursty activity, with a number of outgoing connections spread out over a certain period of time. CnC connections on the other hand are often isolated in time,

especially at times when the human user is not active.

- *Flow direction evaluator:* When a compromised PC is exfiltrating information, it will use different HTTP methods than when it is retrieving information from the outside. The detector looks at the amount of information that is going out versus the amount of information that is coming in and tries to detect any imbalance, compared with the typical behavior of other hosts in the network.

- *Geographic outlier detector:* Users tend to visit websites that are regionally grouped. This detector will therefore report hosts that are visiting one specific website that is geographically located far away from all the other sites that are normally visited by the same host.

- *High fan-in, fan-out detector:* Malwares sometimes use pseudo-randomly generated domain names for locating their CnC server or use a battery of infected hosts as alternating contact points hidden behind a common hostname. The detector therefore looks for abnormal domainnames.

The indicators of suspiciousness, produced by the individual agents, will inevitably contain a large amount of false alarms. Therefore the outputs are aggregated into a global suspiciousness rating. The system will however not automatically classify hosts as clean or infected. Indeed, because of the characteristics of an APT, an approach based on the automatic thresholding of the aggregated suspiciousness score at a predefined level that ensures an acceptable probability of detection, would inevitably result in a high probability of false alert. Therefore the system is conceived as a semi-automatic data exploration system, with a human expert in the loop. The system just draws the attention of the human expert to the most important potentially suspicious events and provides him with the necessary analysis tools to explore these further.

In order to detect previously unseen malwares, the detectors must inevitably be targeted at generic rather than very specific behavioral descriptions. Therefore the inclusion of domain knowledge, on the one hand in the form of algorithms and parameters and on the other hand as a human expert in the loop, are needed to separate the very weak CnC signal from the background noise of regular Internet traffic.

4. Evidence Aggregation

Every connection that passes through the choke point is made available to all the agents for inspection. Some of the agents may judge that the traffic associated with a given connection resembles typical CnC traffic, at least for those characteristics of such traffic that a specific agent is looking for, while other agents may not find anything suspicious and therefore not react.

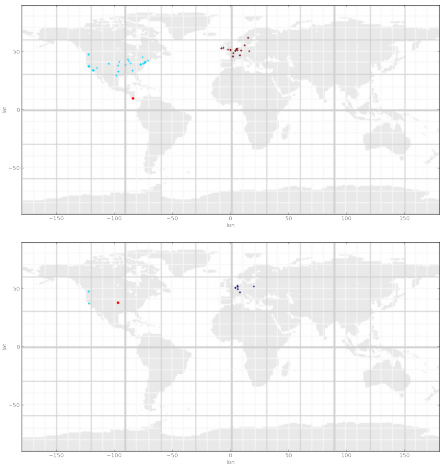

Figure 3: geographic outlier detection for two clients

The output of an agent is not a binary result but rather the degree to which the agent estimates that the connection belongs to the set of connections that show a CnC-like behavior with regard to the specific characteristic this agent is looking at.

Consider for instance the “geo-outlier” agent that searches for connections from a client towards a server in an area where the same client does not visit any other server. In figure 3 these isolated servers are indicated by a red dot. In the case of client 1, a large cluster of servers is found in the US and another large cluster in western Europe but there is one single server that is visited in central America. The agent will therefore produce a suspiciousness degree value for this client-server combination. In the case of client 2, again three clusters have been identified, one of which contains a single server and therefore this server can be considered a geographic outlier as well.

However, a human analyst will be more convinced by the first example than by the second since it is based on a much larger number of samples. This domain knowledge of the human expert, used for evaluating

the result of the clustering operation, is captured in the form of fuzzy rules in the following way [2]:

if cluster contains single server
and dataset covers large number of connections
and dataset covers sufficiently long timespan
then
connections to server suspicious
else
connections to server not suspicious
end if

The natural language expressions "large number of connections" and "sufficiently long timespan" are defined by fuzzy set membership functions that allow the human expert to integrate part of his domain knowledge into the system.

The geo-outlier agent will thus first geographically cluster the servers visited by each client, and subsequently evaluate the degree of suspiciousness for all one-element clusters that it finds. For this purpose, it first computes the crisp input values representing the total number of connections in the dataset initiated by the given client and the period of time covered by these connections. Then the antecedent fuzzy sets are evaluated, and finally the firing degree of the rule is computed.

In a similar way, each of the other agents will look for certain characteristics and produce fuzzy evidence against a number of client-server combinations. In order to fuse for a given client-server combination the outputs of a variable number of agents, an extended aggregation function is needed [3]. For the reasons explained in section 2.2, our multi-agent system uses an OWA aggregation operator.

The weights for the OWA operator allow the human expert to select any aggregation behavior between a maximum $(w = 1, 0, 0, \ldots, 0)$, and a minimum $(w = 0, 0, 0, \ldots, 1)$ operation. In our experience the human experts tend to prefer a weight combination that suppresses single agent alarms and encourages mutual reinforcement, by choosing a low first weight and assigning the main part of the weight mass over the second and third position of the weight vector.

There are actually two aggregation levels involved. The first level consists in combining the opinions produced by the different agents for a given server or connection. The second level is the aggregation over different connections or contacted servers for a given client. This last aggregation results in a single score per client. Here we also use an OWA operator but with weights that are chosen in such a way that already a single suspicious repetitive connection or contacted server can lead to a relatively high aggregated suspiciousness degree for the client.

5. Experimental Results

The multi-agent system has first been validated using simulated data, in order to develop and tune the detectors as well as the aggregation operator. Subsequently real traffic traces from a small subnet with about 100 client PCs were used as background traffic and simulated APT activity was added.

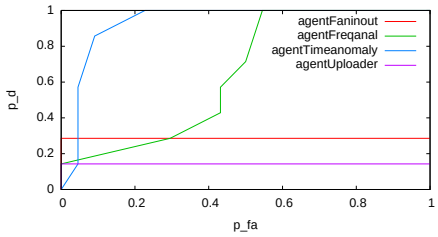

Figure 4: ROC for the individual agents

Figure 4 shows the individual ROC curves for four agents, applied to a simulated network of about 100 clients. The clients perform a random number of website visits each day in a simulated working day cycle, where each visit consists of a variable number of pages obtained from a range of different servers. The distributions used for generating the characteristic parameters for these visits were based on logfiles from a real network.

The APT network activity simulation is based on a state space with transition probabilities, that can depend on the activity of the underlying infected host. In total 8 different APT models were used for this test, all of them based on existing malwares.

Some agents only detect very specific types of CnC communications and therefore are unable to detect all the APTs that were simulated; they generate few false alarms however. Others detect a wide range of malware traces yet produce higher numbers of false alarms.

The goal of the aggregation is to reduce the false alarm rate while preserving successful detections. Figure 5 shows the results for different aggregation operators, applied to the detection agents for which the performance was depicted in figure 4. The maximum operator is fooled by high valued false alarms from single agents, and as a result suffers from a higher

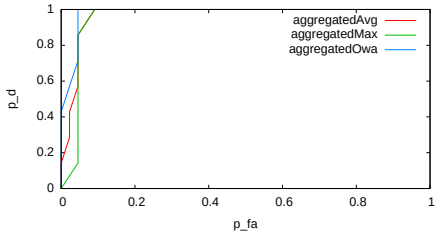

Figure 5: ROC for different aggregation operators

false alarm rate itself as well. The averaging operator on the other hand takes into account the outputs from all agents and as a result performs better than the maximum operator in this simulation. However, the tests clearly show that the OWA operator, even with a basic set of weights (in this test the weights $[0.2, 0.5, 0.3, 0.0]$ were arbitrarily chosen by the operator), performs better than both other operators on this dataset.

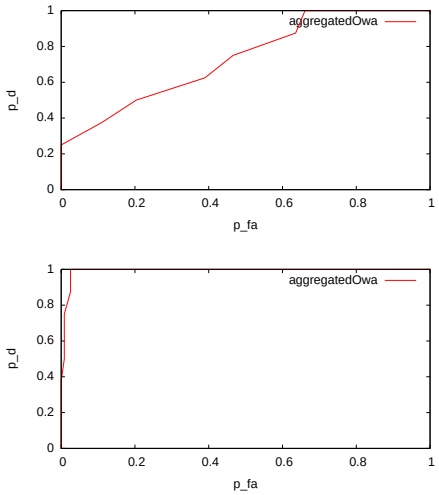

Figure 6: ROC for OWA aggregation on real data, without and with whitelisting

Figure 6 shows the OWA aggregated ROC curve for a test on real data. The network consists of about 100 client hosts with a wide range of user profiles and with the same 8 APT models attached to randomly selected client hosts. This results in a large number of false alarms, corresponding with updating background agents of all sorts, cloud storage services, etc.

However, when whitelisting is applied based on previously seen false alarms, the resulting detection performance becomes usable in the sense that it would be possible for a human analyst to systematically analyze the newly reported alerts and whitelist the limited number of new false alarms that are produced. This whitelisting is applied in a client-specific, agent-specific and where possible even a value range-specific way in order to reduce the risk of malware evidence being inadvertently erased by the whitelisting process.

6. Conclusion and Future Work

In this paper we proposed a new system that combines multiples approaches using advanced aggregation techniques to detect APT attacks. We tested the system on simulated data and on real data from a small corporate network. We showed that our system is able to attain a high probability of detection to probability of false alarm ratio.

In the future, we plan to implement additional agents, that will look for other hints of an APT attack going on, for example using other sources of information (firewall logs, IDS logs ...). We also plan to test the system more extensively, on data from other real networks, and to compare our aggregation techniques with state-of-the-art classifiers. We will also use machine learning algorithms to automatically optimize agent and aggregation parameters, and compare them with parameters suggested by human experts. Finally, we plan to study the temporal behavior of the system, to determine how many events, and how much work from a human expert, are required to achieve a good performance of detection.

References

[1] Didier Dubois and Henri Prade. On the use of aggregation operations in information fusion processes. *Fuzzy Sets and Systems*, 142(1):143 – 161, 2004. Aggregation Techniques.

[2] George J. Klir. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ (USA), 1995.

[3] Radko Mesiar, Anna Kolesárová, Tomasa Calvo, and Magda Komorníková. A review of aggregation functions. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, pages 121–144. 2008.

[4] Ronald Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on System, Man and Cybernetics*, 18(1):183–190, 1988.