The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Artificial Intelligence for Cyber Security: Technical Report WS-16-03

Active Perception for Cyber Intrusion Detection and Defense

J. Benton, Robert P. Goldman, Mark Burstein, Joseph Mueller
SIFT, LLC, 319 N. First Ave, Minneapolis, MN 55401 USA
{jbenton,rpgoldman,mburstein,jmueller} at sift.net

Paul Robertson, Dan Cerys, Andreas Hoffman
DOLL Labs, 114 Waltham Street, Lexington, MA 02421 USA
{paulr,dan,andreas} atdollabs.com

Rusty Bobrow
Bobrow Computational Intelligence, LLC
rjbobrow at gmail.com

Abstract

Most modern network-based intrusion detection systems (IDSs) passively monitor network traffic to identify possible attacks through known vectors. Though useful, this approach has widely known high false positive rates, often causing administrators to suffer from a "cry wolf effect," where they ignore all warnings because so many have been false. In this paper, we focus on a method to reduce this effect using an idea borrowed from computer vision and neuroscience called *active perception*. Our approach is informed by theoretical ideas from decision theory and recent research results in neuroscience. The active perception agent allocates computational and sensing resources to (approximately) optimize its *Value of Information*. To do this, it draws on models to direct sensors towards phenomena of greatest interest to inform decisions about cyber defense actions. By identifying critical network assets, the organization's mission measures self-interest (and value of information). This model enables the system to follow leads from inexpensive, inaccurate alerts with targeted use of expensive, accurate sensors. This allows the deployment of sensors to build structured interpretations of situations. From these, an organization can meet mission-centered decision-making requirements with calibrated responses proportional to the likelihood of true detection and degree of threat.

1 Introduction

Present day cyber defense systems rely on fixed sets of sensors, or *Intrusion Detection Systems* (IDSes), of a limited set of types, that are active all the time. Often these IDSes employ inexpensive, broad spectrum detectors, which typically have extremely high false positive rates. These sensors are typically either signature-based – in which case they cannot detect so-called "zero day" attacks, exploits that have never been seen before – or based on anomaly detection, in which case entirely normal deviations from a statistical model are misclassified as attacks. Other detection tools, typically applied only after an attack, have a low false positive rate, but consume so many computational, storage, and attentional resources that they can only be used very sparingly. The IDSes are also typically "context unaware," unable to incorporate information about the network in which they are installed, its intended purpose, and known threats, except through labor-intensive and obscure tuning processes. Lack of contextual information contributes to the false positive problem, as IDSes misinterpret known benign behaviors (*e.g.*, periodic network backup jobs) as malicious attacks (*e.g.*, exfiltration). Finally, since these systems are not context aware, sensing and information presentation is not directed to provide the information needed to direct actions. For these reasons, users often turn off, ignore, or don't install sensors, so cyber-attacks go undetected.

This paper describes an approach to cyber defense based on *active perception*. As the name suggests, active perception involves the active control of sensing. Sensors are controlled in order to (approximately) optimize the information they provide. That optimization is characterized in terms of expected improvement to cyber defense decision making which, in turn, is defined in terms of performance of the mission of the defended network.

Active perception is a *model-driven* process. Sensor control must be driven by top-down information, as well as bottom-up sensor inputs, and that top-down information is contained in models of sensors, the environment (network, threats, etc.), and the mission of the defended network. Active perception uses models for many purposes:

- to inform context-dependent enabling, disabling, and tuning of sensors,
- to direct sensors towards phenomena of greatest interest,
- to follow up initial alerts from cheap, inaccurate sensors with targeted use of expensive, accurate sensors,
- and to intelligently combine results from sensors with context information.

Our ideas about active perception have both theoretical and empirical background. Our theoretical framework comes from decision theory, and its notion of the *value of*

information. Key ideas about implementation, and precedent for the integration of high-level models with low-level sensory processing come from recent developments in neuroscience.

We have prototyped two elements of our active perception concepts. The first is a *sensor placement* component, which uses information about the defended network, the computational tasks it is intended to perform, and a threat profile to locate sensors. The second prototyped element actively manages sensors in order to resolve uncertain hypotheses. We demonstrate this component in the context of the STRATUS system for autonomous cyber defense. The STRATUS system has an IDS fusion subsystem, MIFD, which combines the results of multiple IDS sensors into a set of event hypotheses, and weighs the evidence for and against these hypotheses using qualitative probability. We are working to extend MIFD to seek out new information to resolve uncertainty about key hypotheses, by finding sensors that provide relevant information, then activating those sensors. We have developed a Prolog-based proof-of-concept for this new capability, and will soon be integrating it into our STRATUS system for autonomous cyber defense.

In this paper, we review our decision-theoretic and neuroscience inspiration for active perception, then outline how our active perception approach works. We then describe an example scenario, inspired by the Stuxnet attack, and explain how it would be handled by an active perception system. Using the scenario as a running example, we then describe our two active perception subsystems. Finally, we conclude with some remarks about future work.

2 Inspiration

Our work on active perception has been inspired by developments both in decision theory and in neuroscience. Decision theory provides a normative framework that describes how sensing resources should be allocated in ways that will optimize the outcome of *decisions* that need to be made. Sensing resources should be allocated to optimize return in terms of the expected outcome of decisions influenced by observations, discounted by the costs of making and processing those observations. However, decision theory has little to say about how perceptual problems should be structured and modeled, and how relevant contextual information should be brought to bear. For answers about these questions, we have been guided by neuroscience, and particularly the neuroscience of visual perception. Recent findings in visual perception have revealed a pervasive influence of *top-down* contextual information, and a mechanism, "gisting" that suggests how that top-down information can be activated and brought to bear on sensory interpretation problems.

Decision Theory

The decision-theoretic notion of *value of information* (VOI) provides a general theoretical framework for sensor management. In theory, one should simply choose the application of sensors that maximizes the value of information, or, equivalently, act according to the optimal policy for a partially observable Markov Decision Process. In practice, these models are difficult to build, and solution algorithms scale poorly in space and time. While we cannot simply naively apply decision theoretic solutions to our cyber defense problems, the decision theoretic framework provides a gold standard against which our techniques can be compared.

The value of information for a sensor configuration/observation ω, with respect to a decision D measures the additional expected value for D gained by taking the observation ω versus not taking it: $\mathrm{EU}(D|\omega) - \mathrm{EU}(D)$ (Shachter 1986; Pearl 1988, Chapter 6). For example, in Raiffa's famous oil wildcatter problem (Raiffa 1968), a question to be answered is whether it is worth performing a seismographic test before drilling an exploratory well. For small, high stakes problems, it can be worthwhile posing and solving VOI problems. However, as problem size grows, VOI computations rapidly become infeasible, since they require computing all outcomes of all combinations of observations, for each possible state of the world. Often a *myopic approximation* is used, and one assesses whether a single observation provides value, rather than explicitly considering combinations.

Another challenge for active perception is that work in decision analysis has focused on choosing which tests to run, rather than on finding the set of relevant tests, an important aspect of our work. Typically, in decision analysis, the set of available tests is treated as given, as part of the framing of the problem. Some decision analysis texts (*e.g.* (Keeney 1996; Hammond, Keeney, and Raiffa 1998) discuss how to frame problems, but as a human process, rather than an automated one.

Ahmad and Yu (2013) propose a POMDP-based approach to active perception in the context of visual perception, but their test examples feature very small decision spaces (*e.g.*, three-location visual search). Eidenberger, *et al.* (2009) also propose a POMDP-based approach to active vision, this time embedded in a robot, where they tradeoff information gain against control action costs, but their work aims at continuous action spaces, rather than the discrete decisions we address.

Neuroscience

Neuroscientific inspiration for our active perception approach comes from recent research results that show that top-down (contextual) information flow plays a critical role in the operation of the visual object-recognition system. Early anatomical and functional models of the object recognition pathway in vision (starting with Hubel and Wiesel (1962)) were essentially hierarchical. Many purely bottom up models for visual processing (Serre, Oliva, and Poggio 2007) were developed based on this hierarchical structure. These models attempted to explain the ability of the human visual system to rapidly (within 100 ms) identify objects and categorize complete scenes. However, recent detailed anatomical and physiological evidence paints a much more complex picture: the object recognition system is now (Kravitz et al. 2013) known to be organized as a series of a) overlapping b) bidirectionally coupled recurrent networks with c) long range interconnections that skip over intermediate levels. This anatomical structure provides the basis for bottom up hierarchical processing to be modulated

and controlled by top-down sources of information.

Lee and Mumford (2003) have shown that the local feedback in the anatomy is just what is needed to implement a hierarchical Bayesian inference mechanism. In this scheme, top-down estimates of the likelihood of various object features bias the interpretation of sensory data in a recursive, hierarchical fashion.

This contextual Bayesian inference is characterized by Ganis and Kosslyn (2007) as a primary example of one of two major modes of top down influence on perception supported by both neural and psychological data. This is "reflexive top down processing," a process which modulates the interpretation of bottom up data, by changing both 1) the sensitivity of individual sensors and 2) the amount of sensor data needed to support various detection decisions. The modulation is based on top-down, contextual estimates of the likelihood of various causes of sensor data. These processes are not consciously accessible, and they operate through the bidirectionally coupled recurrent networks characterized by Kravitz in the "ventral visual stream."

Ganis and Kosslyn also describe, and provide experimental evidence for, an often conscious second class of top-down process: "strategic top-down processing."

> Strategic top-down processing relies on "executive control mechanisms" (which provide input ... to direct a sequence of operations in other brain regions, such as is used to engage voluntary attention or to retrieve stored information voluntarily).

It can involve "covert attention," a mechanism involving allocation of perceptual resources to part of the visual field outside the fovea ("looking out of the corner of one's eye"). This "strategic" process, as described by Kosslyn, identifies partially obscured objects, and handles degraded sensory data. In this case, initial sensor data, plus top-down biases from the "reflexive" system is used to surface one or more plausible objects or events in long-term memory to be treated as a "hypothesis." These hypotheses are then used by the executive system to control a set of specific neural and muscular components which Ganis and Kosslyn call the "information shunting subsystem," This subsystem actively and sequentially directs attention to sensory data that would discriminate between the alternative hypotheses.

For many years it was a mystery how the right expectations could be activated at the right time. Over the past decade neuroscientific and computational modeling evidence research addressed this gap with evidence of a "gisting" process. Experimental work supported this evidence, revealing that people can identify the category of a scene presented for as little as 100 ms. The gisting process is supported anatomically and physiologically by the previously cited existence of neural connections that jump multiple levels, and which lead directly to associative memory elements such as the para-hippocampal cortex and the retro-splenial cortex. Some of these links are particularly high-speed projections. Experimental evidence shows that such projections can form the basis for the initial scene level gist in the brain (Kveraga, Boshyan, and Bar 2007), allowing the context of an image to help identify its content.

3 Example Scenario

To showcase the application of active perception to cyber defense, we produced a demonstration of an example scenario, loosely based on Stuxnet. The demonstration illustrates how active perception can augment an intrusion detection system and subvert a complex attack on a network. To this end, our web-based demonstration steps through an attack scenario, sensor placement, and the steps performed by active perception techniques to help detect and counter the threat.

Attack plan The layout for the scenario is in Figure 1. In the scenario, an attacker uses a well-designed, careful attack on a base's command-and-control server, running on server S4. The attack relies on a combination of user-fallibility and a series of exploits. The attacker uses a phishing email to trick the user into clicking on a link to a website. The website causes a "drive-by download" of malware. The malware then uses a privilege-escalation exploit to gain administrative privilege on the user's workstation, W8.

With administrative abilities, the malware can begin its attack on the command-and-control server. It begins reconnaissance to find a particular SOAP service that marks the command-and-control server. To do this, it probes all servers with SOAP services and finds the server. Though the probe, the malware learns that W8 lacks permission to interact with the command-and-control server. This forces the malware to find workstations that have access to the server. It scans the plan cell of workstations near the server, and transmits its payload through a PowerPoint document delivered to those workstations. As the final step, the malware generates PDF documents crafted to exploit a flaw in the command-and-control server PDF processing. It then submits those documents, which establishes a backdoor that allows the attacker to read confidential information.

Initial Sensor Placement In the scenario, each network node sensor may be enabled or disabled. We seek to optimize coverage of our sensors when 100% coverage is impossible due to resource limitations. We use a mixed integer program (MIP), discussed in Section 4, that we configured based on resource availability, the types of threats to consider, and the expected level of internal and external threats. The initial sensor placement, shown in Figure 1, dictates that sensors be enabled in a perimeter around the most critical resources.

Active Perception We discuss how the attack plan may be successfully detected with active perception and the sensor placement methods we discussed. Recall that the scenario starts by the attacker gaining control of a user's workstation. Our sensor placement techniques did not enable sensors there, so that step cannot be detected. During the reconnaissance step of the attack, the malware installed on the user's workstation looks for SOAP services. Defense activates with the reconnaissance traffic coming from the workstation. It considers the possibility that there is a local attacker conducting recon. However, the sensor reports might instead be evidence of a misconfigured SOAP client, looking for its server. shown in Figure 2. There is not enough evidence to tell the two hypotheses apart, so defense must perform an active investigation.

The influence diagram in Figure 3 illustrates the decision

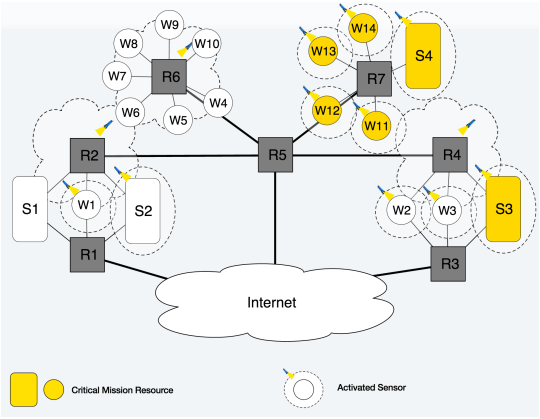

Figure 1: Initial sensor placement.

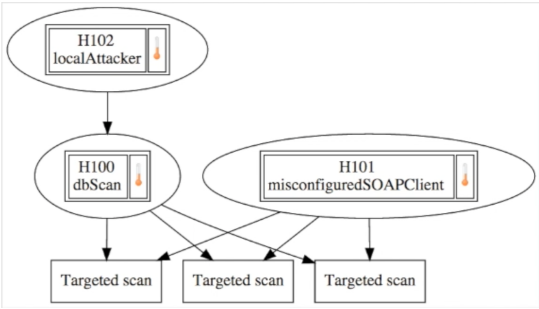

Figure 2: Hypotheses prior to investigation.

making performed. In this situation, there are two possible sensing actions. First, defense could examine past alerts for other evidence of a local attacker, or we could conduct a forensic probe of the workstation. It is possible to do both actions. The question at hand is whether to quarantine the user's workstation. If defense does quarantine it, and there's a local attacker, it will limit mission disruption. However, if defense quarantines the workstation and there is no attacker, its no longer possible to use the workstation for the mission, which will negatively impact performance. Defense examines past alerts in attempt to dismiss the possibility of an attack. If its unable to do so, a more expensive probe is carried out.

The probe focuses attention on evidence on the user's workstation. It discovers sensor reports relative to the phishing attack, and updates its beliefs. An attack cannot be ruled out, and defense conducts an active probe of the user workstation, finding an unauthorized root-privileged process, which is a component of the malware. The system now believes there exists a local attack, and conducts further defensive measures. It looks at the reconnaissance targets chosen by the malware, and hypothesizes two *gists* about the attack's intent. The attacker may be aiming at either the web server, on S1, or the command-and-control server. Note that, in the event of an attack on the command-and-control server, defense can expect island-hopping to occur, where the attacker tries to infiltrate a client that has access to the

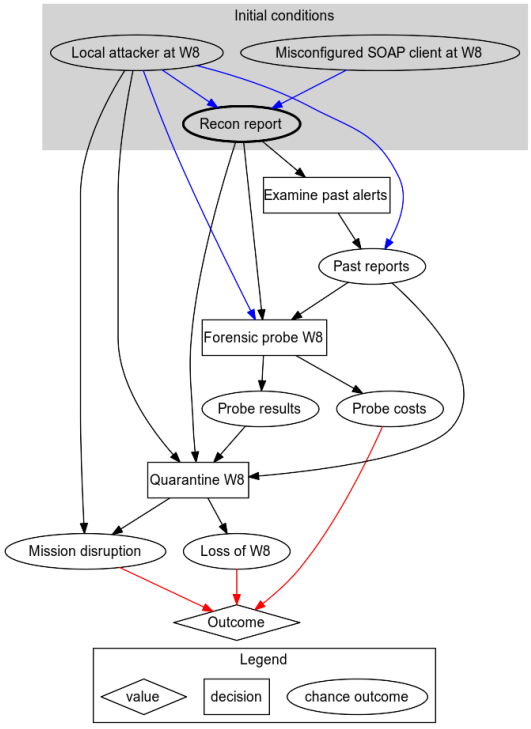

Figure 3: The shaded area of the influence diagram shows the initial conditions and the two competing hypotheses.

command-and-control server. Defense may now choose to isolate the workstation from the network, in attempt to halt the spread of the attack, or it could redirect it to a honeynet to gain more information.

The malware attempts to break into workstations. These operations link with the hypothesis that the attacker plans to break into the command-and-control server, confirming one of the two gists. At this point, defense should quarantine the user workstation, and carefully monitor workstations in the plan cell. Using available methods for hardening, such as binary diversity, would also be a reasonable precaution.

4 Sensor Placement

The sensor placement system attempts to find a sensor positioning that will optimize situation awareness about the *most likely* and *most important* attacks, for a fixed budget of sensing resources. The sensor placement system draws on a model of the importance of defended resources to the mission of the organization, which can change depending on the situation. It has a sensor cost model that is based on estimates of how many sensor reports a given sensor is likely to emit, since these sensor reports consume the attention of scarce administrative and security personnel. Finally, the system can draw upon threat information to focus on targets that are likely to be attacked.

The optimal way to solve the sensor placement problem would be to optimize the *value of information* (see Section 2). Unfortunately, value of information computations are

badly intractable, since they require enumerating possible situations and then evaluating an exponential number of sensor placements against them. We have adopted a mathematical programming approximation that optimizes *weighted attack coverage*, subject to resource constraints. Here we describe the model, and analyze its behavior.

Problem Formulation Our general problem formulation is summarized as follows. Given a set of nodes on a network, including workstations, servers and routers, we may activate one or more sensors at each node to inspect network traffic. While activating a sensor provides value in the sense that it increases coverage, it also incurs some cost related to the inspection of sensor reports, and it consumes finite resources such as network bandwidth and processor usage. The optimization problem we pose is to maximize a weighted metric of sensor coverage over the network against hypothetical attacks. This maximization is subject to to individual resource constraints at each node as well as constraints on shared resources for the entire network.

Considering one sensor and two resources at each node, plus two shared resources for the network, the problem is formulated as the following mixed integer program (MIP):

$$\max_{x_k \in \{0,1\}} \quad \sum_{k \in \mathcal{S}} \left[\sum_{h \in \mathcal{H}} W(h)E(k,h) - C(k) \right] x_k \quad (1)$$

$$R_1(k)x_k \leq \bar{R}_1 \qquad \forall k \in \mathcal{S} \quad (2)$$

$$R_2(k)x_k \leq \bar{R}_2 \qquad \forall k \in \mathcal{S} \quad (3)$$

$$\sum_{k \in \mathcal{S}} R_1(k)x_k \leq \overline{SR}_1 \quad (4)$$

$$\sum_{k \in \mathcal{S}} R_2(k)x_k \leq \overline{SR}_2 \quad (5)$$

In the objective function (1), the binary variable x_k represents the decision to activate sensor k, set $\mathcal{S} = [1, \ldots, N_S]$ represents the set of sensors (one at each node), and $\mathcal{H} = [1, \ldots, N_H]$ is our candidate set of hypotheses. $E(k,h) \geq 0$ gives the expected value of using sensor k to investigate hypothesis h, and $W(h) > 0$ represents the importance of hypothesis h relative to others. The product of $W(h)E(k,h)$ therefore provides a weighted value of using sensor k to investigate hypothesis h. Our coefficient on x_k is the weighed net value of the sensor, which is reduced from the weighted value by subtracting the cost of using the sensor, $C(k)$.

Equations (2) and (3) are the constraints applied at each node for two separate resources, while equations (4) and (5) are the constraints on the two resources shared by the entire network.

The formulation described above includes N_S variables and $2N_S + 2$ constraints. It can be used to find a sensor laydown that maximizes the total value of information, assuming that the expected value data, $E(k,h)$ is available and accurate. If simple coverage is the priority, then an alternate formulation may be developed by adding hypothesis coverage constraints with slack variables, and incorporating satisfaction of those constraints into the objective function. The alternate problem formulation is:

$$\max_{x_k, x_h \in \{0,1\}} \quad \sum_{k \in \mathcal{S}} \left[\sum_{h \in \mathcal{H}} W(h)E(k,h) - C(k) \right] x_k + \sum_{h \in \mathcal{H}} W_C(h)x_h \quad (6)$$

$$\vdots \text{ Constraints (2) through (5)}$$

$$\sum_{k \in \mathcal{S}} C_H(h,k)x_k \geq x_h, \forall h \in \mathcal{H} \quad (7)$$

Here we introduce N_H binary variables x_h and N_H constraints, where each constraints allows $x_h = 1$ only if that hypothesis is covered by the sensor activation in x_k. We make use of the hypothesis coverage matrix C_H, which has $C_H(h,k) = 1$ iff sensor k provides evidence for hypothesis h. The added term in the objective function effectively provides a reward proportional to $W_C(h)$ for activating sensors that provide coverage of hypothesis h. This formulation includes both value of information and hypothesis coverage, and we can prioritize one over the other in our definition of the weights. For example, by setting $W_C = 100W$, we first prioritize coverage and then prefer solutions that would improve value of information for the same coverage.

Network Model In the examples presented here, we focus on detecting attacks by monitoring network traffic on our notional network of 4 servers, 14 workstations, and 7 routers (see Figure 1). We assume that there exists one attack hypothesis for each peer-to-peer connection between network nodes (servers and workstations only), plus one for each connection to the internet. Let N_P be the number of peers to consider, where $N_P = 4 + 14 + 1 = 19$ including the servers, workstations, and internet. The number of hypotheses is therefore $N_H = N_P(N_P - 1)/2 = 171$.

We define the sensor cost, resource usage, and hypothesis values to be consistent with an actual network structure. Starting with a 26x26 node connectivity matrix for our notional network, we use the Floyd-Warshall algorithm to compute shortest paths between each network node. We then assign an integer traffic level to each node equal to the number of paths that include it. Cost and resource usage values for each sensor are set equal to some base value plus a value that is proportional to the traffic level at that node. The coverage matrix is defined by setting $C_H(k,h) = 1$ if sensor k is included in the connection path for hypothesis h. Assuming that all sensors provide equal value, the expected value matrix is equivalent to the coverage, $E_V = C_H$. Finally, we define each hypothesis weight $W(h)$ as the product of the severity and likelihood of an attack at the two end nodes of the hypothesis. This approach allows us to define hypothesis weights indirectly by instead prescribing severity and likelihood numbers for a suspected attack scenario.

Sample Results We present four different examples for the sake of illustration and comparison. The results are summarized in Table 1. In Case 1, we attempt to maximize coverage with the fewest sensors. We achieve this by forcing

Table 1: Summary of sensor placement results for four example cases. Case 3 is used for the demonstration.

	Case 1	Case 2	Case 3	Case 4
Cost $\propto$	Sensors	Traffic	Traffic	Traffic
Max Coverage?	Yes	Yes	Yes	Yes
Max VOI?	No	Yes	Yes	Yes
Avail. Resources	High	High	Med.	Low
# Sensors	6	14	13	11
# Hypoth.	171	171	170	159
% Coverage	100	100	99.4	93.0
Total VOI	1192	1680	1657	1263

$E_V(k,h) = 0$, $C(k) = 1$ and $W_C(h) = 1$ for all k, h, and by keeping the resource limits sufficiently high. The solution is intuitive, activating sensors only at the routers, which see the most traffic. In Case 2, we again seek to maximize coverage, but we also try to maximize the VOI as a secondary objective. Resources are abundant in each of the first two cases, but are progressively reduced in Cases 3 and 4, leading to solutions with fewer sensors and less than full coverage. The solution for Case 3 is used in the demonstration. Even with limited resources, we still find a sensor placement solution that achieves 99% coverage and attains nearly the same VOI as in Case 2, where resources were abundant.

5 Active Perception via Sensor Goals

Sensor Fusion Our experiments on active perception have been done in the context of the STRATUS system, a multi-agent cognitive architecture for cyber defense (Thayer et al. 2013). A key component of STRATUS is Model-based Intrusion Fusion and Detection (MIFD), the place where sensor information enters STRATUS and is fused together. The MIFD system builds on a qualitative probabilistic approach developed for a predecessor system, Scyllarus (Goldman and Harp 2009).

MIFD, like its predecessor Scyllarus, views sensor fusion as an *abductive*, or *diagnostic* process that reasons to the best explanation. MIFD fuses reports from *Intrusion Detection Systems* (IDSes). When it receives *sensor reports* (IDS reports), MIFD forms *event hypotheses* to explain those sensor reports. MIFD fuses multiple sensors, so multiple sensor reports can provide support for a single event hypothesis. *Ambiguous* sensors may have multiple alternative event hypotheses that would explain a single sensor report. Finally, events may be components of complex event hypotheses. For example, a single denial of service attack hypothesis might explain multiple flooding attacks on a set of web servers. Note that the set of event hypotheses need not be exhaustive: some sensor reports are simple *false positives*.

The sensor reports and the event hypotheses that could explain them constitute a *Bayes network*. We refer to the process of constructing this Bayes network as *clustering*. For example, Figure 2 shows the Bayes net constructed in response to three "Targeted scan" IDS reports. One hypothesis (H101) is that these reports are the result of a SOAP client misconfiguration, and another (H100) is that there is a database scan going on. In turn, that database scan might be evidence of a local attacker (H102).

After clustering, a separate *assessment* process occurs, where MIFD uses the evidence in the Bayes networks to assign a qualitative likelihood ranking to each of the event hypotheses. The thermometer icons in Figure 2 show the results of assessment: initially MIFD cannot determine which of the two hypotheses is most likely, and they are both assessed as "plausible." We will not discuss assessment further in this paper; for more details see (Goldman and Harp 2009).

As described above, the two key data items in the cluster preprocessor are sensor reports and event hypotheses. Various sensor programs may issue sensor reports. Each report contains a *report type*, which specifies the condition the sensor claims to have detected. IDSes must generally infer the existence of a security-related event from data (e.g., packet headers) that provides only very indirect and noisy indications. Such sensors often have a high false positive rate, and detect conditions that their designers had not anticipated. Sensor reports also contain information about the location of the detection.

The clustering process generates event hypotheses to explain or interpret the sensor reports. The event hypotheses similarly have event types. These event types disambiguate the sensor reports. For instance, when generating a "gist" in our example scenario, the malware's probe for a SOAP client generated two alternative event hypotheses: (1) That a misconfigured SOAP client exists on the network, or (2) we have a local attacker.

Background information mediates the process of forming event hypotheses, populating them, and linking them to sensor reports and to each other in *hypothesis matchers*. An individual hypothesis matcher, M pairs a *phenomenon*, $P(M)$, a sensor report type or an event type, with an *explanation*, $E(M)$ event type. Hypothesis matchers perform a rule-like function. To a first approximation, a hypothesis matcher records the following inference pattern (Charniak and Goldman 1988): $\forall x : P(M)(x) \rightarrow E(M)(x)$. For example, one hypothesis matcher we use represents the following inference: "If there is a sensor report of type `unexpected component restart`, hypothesize an event of type `compromised component`."

Hypothesis matchers can *fuse* information from multiple sensors by linking a phenomenon to an explanation event hypothesis: it can either cause a new event to be hypothesized, *or* it can match an existing event hypothesis, and cause the new phenomenon to be linked in as additional evidential support. To control information fusion, hypothesis matchers contain *data checks*. For example, in order for an additional sensor report to support (be explained by) the `compromised component` hypothesis in our previous example, the data check would require the `destination component` of that sensor report to be the same as the `target component` of the `compromised component` hypothesis.

Sensing Goals As our work on STRATUS progressed, we became convinced that dynamic, autonomous cyber defense critically requires context-sensitive sensor control. MIFD's fusion of multiple heterogeneous IDSes mitigates their high false positive rate. Even so, we still require follow-up "forensic" investigation. The required, in-depth investigation performed by a security analyst precludes us from deploying "always on" sensing actions that a security analyst does for this in-depth event investigation. As in our scenario, a potential attack may prompt one to check all past reports and alerts. Sensing action cost may also outweigh the benefits of routine sensing, or might produce too much information for constant processing. We might reserve these sensors for only the most high value targets. In conventional network defense, one would identify such targets *a priori* and statically. STRATUS uses mission models to identify high value assets dynamically, in order to support defense of cloud style networks in which computational resources are fungible, and computational tasks can be moved around the network more or less at will.

For these reasons, we determined that STRATUS should be able to dynamically control its sensing. To do so, it issues sensor requests. It will be the job of the Mission-Oriented Threat Hypothesis Evaluation and Response (MOTHER) component to evaluate the importance of these requests, and determine whether to act upon them.

STRATUS will form two kinds of sensing goals: forensic sensing goals, which attempt to find more evidence to reason about existing event hypotheses, and proactive sensing goals, that seek to place sensors to monitor expected threats. We describe the reasoning processes and the supporting knowledge representation.

The formation of forensic knowledge goals may be triggered when MIFD finds an event hypothesis whose uncertainty it cannot resolve: it neither thinks it likely nor unlikely. In this circumstance, MIFD will examine the model of the event hypothesis to determine how critical the event is: i.e., how bad it would be if the hypothesis was true. STRATUS event type models contain *impact specifications* which indicate what security goals would be compromised when the event occurs. In the event of a high-criticality unresolved event hypothesis, MIFD will move to form sensor goals.

MIFD will invoke the sensor selection module (described in the following section) to identify sensors that could provide additional information to resolve the uncertainty about the event in question. If it finds such sensors, MIFD will publish a sensor goal, requesting this additional sensing. The STRATUS MOTHER module, responsible for resource management, will receive the sensor request, assess the criticality to the mission of the resource(s) to be examined, and consider the available sensing and sensor processing resources. Based on this information, MOTHER will decide whether or not to grant the sensing request. If MOTHER grants the sensing request, it adds additional information needed to realize the goal and publish it to CSE. The CSE infrastructure will carry out the necessary actions to implement the requested sensing.

Forming proactive sensing goals is a more open-ended process, open to arbitrary STRATUS subsystems. Proactive sensing will begin when a STRATUS subsystem, S identifies an event (an attack on a particular component or from a particular component) that it deems likely. After S identifies events and locations that it considers of interest, it will build representations for that information.This information will be used to query the sensor selection module, and find appropriate sensors and placements. From here the processing proceeds as per forensic sensing requests.

KR to Support Sensing Goal Formation We have developed a knowledge representation (KR) scheme to support formulating sensor requests assuming that STRATUS components will know *what* they want to see, and *where* they are interested in looking for it.

The KR facilitates requests for identifying candidate sensor prototypes. Each request contains an event report, which includes the event type and the components involved in the event. In order to test our ideas, we have developed a proof-of-concept implementation of sensor identification. In this Prolog-based system, we provide a set of rules that associates report data with particular hypotheses and information gathering sensors. This allows us to quickly identify the sensors that an various types of events should trigger. In the demo scenario, a search for a SOAP service, detected by network-accessible machines surrounding router R7, triggers a sensing event report through this mechanism. The hypothesis structure itself, encoding in our Prolog system, directs the sensing goals, automatically indicating how we can provide more evidence for a hypothesis we wish to prove (or disprove). In other words, the report begins the process of starting a *forensic* probe, with the report information being used as parameters a process that identifies sensing goals.

The forensic process can continue in a chain. In our scenario, for example, the examination of past alerts discovers a phishing attack. The sensor that gives the "phishing attack" alert triggers the generation of a sensor report. In turn, the new sensor report triggers the generation of a new sensing goal to search for active processes. This then uncovers an unauthorized process with root privileges, until finally we have enough evidence to definitively say that an attack is underway. This active perception process allows for dynamic evidence gathering, positioning STRATUS to re-focus sensors appropriately.

6 Conclusions

In this paper, we have described how to use *active perception* to actively manage sensing for cyber defense, enabling directed, automated investigation into threats. By using inexpensive, inaccurate sensors already used by IDSes in an initial phase, then following up with more expensive investigation, it provides a methodology for active threat identification and response.

We have developed a partial, preliminary implementation our approach, which includes evaluation of sensor placement, hypothesis generation, and sensor identification. Our experiments on abstract models of active perception show its value in situations with the observation characteristics of cyber defense. We will be incorporating the active perception techniques we have developed in our STRATUS system

for autonomous cyber defense. We also plan to further investigate active perception, and potentially apply it to other domains.

Acknowledgments

Thanks to the entire STRATUS project and our funders. This work was supported by Contract FA8650-11-C-7191 with the US Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Ahmad, S., and Yu, A. J. 2013. Active sensing as bayes-optimal sequential decision making. In *UAI*. AUAI Press, Corvallis, Oregon.

Charniak, E., and Goldman, R. P. 1988. A Logic for semantic interpretation. In *Proceedings of the Annual Meeting of the ACL*, 87–94.

Eidenberger, R.; Grundmann, T.; and Zoellner, R. 2009. Probabilistic action planning for active scene modeling in continuous high-dimensional domains. In *ICRA*, 2412–2417. IEEE.

Ganis, G., and Kosslyn, S. 2007. Multiple mechanisms of top-down processing in vision. In Funahashi, S., ed., *Representation and Brain*. Springer Japan. 21–45.

Goldman, R. P., and Harp, S. A. 2009. Model-based intrusion assessment in common lisp. In *Proc. Int'l Lisp Conference*.

Hammond, J.; Keeney, R.; and Raiffa, H. 1998. *Smart Choices: A Practical Guide to Making Better Decisions*. Harvard Business Review Press.

Hubel, D., and Wiesel, T. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol* 106–154.

Keeney, R. 1996. *Value-Focused Thinking*. Harvard University Press.

Kravitz, D. J.; Saleem, K. S.; Baker, C.; Ungerleider, L. G.; and Mishkin, M. 2013. The ventral visual pathway: An expanded neural framework for the processing of object quality. *Trends in Cognitive Sciences* 17(1):26–49.

Kveraga, K.; Boshyan, J.; and Bar, M. 2007. Magnocellular projections as the trigger of top-down facilitation in recognition. *J. Neuroscience* 27(48):13232–40.

Lee, T., and Mumford, D. 2003. Hierarchical Bayesian inference in the visual cortex. *J Opt Soc Am A Opt Image Sci Vis* 20(7):1434–48.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Raiffa, H. 1968. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. New York: Random House.

Serre, T.; Oliva, A.; and Poggio, T. 2007. A feedforward architecture accounts for rapid categorization. *Proc Natl Acad Sci U S A* 104(15):6424–9.

Shachter, R. D. 1986. Evaluating influence diagrams. *Operations Research* 34(6):871–882. Reprinted in (Shafer and Pearl 1990).

Shafer, G., and Pearl, J., eds. 1990. *Readings in Uncertain Reasoning*. Los Altos, CA: Morgan Kaufmann.

Thayer, J.; Burstein, M.; Goldman, R. P.; Kuter, U.; Robertson, P.; and Laddaga, R. 2013. Comparing strategic and tactical responses to cyber threats. In *SASO Workshop on Adaptive Host and Network Security*.