Comparing Strategic and Tactical Responses to Cyber Threats

Jordan T. Thayer*, Mark Burstein*, Robert P. Goldman*, Ugur Kuter*,
Paul Robertson†, Robert Laddaga†,

*SIFT Inc., Minneapolis, MN, Email: burstein@sift.net
†DOLL Inc., Lexington, MA, Email: rladdaga@dollabs.com

Abstract

The STRATUS project is building a capability for resilience against cyber threats to distributed systems. STRATUS is designed to anticipate, diagnose and respond proactively to threats. It uses a reactive technique to respond to the latest events quickly and a more 'strategic' techniques that involves early recognition and response to attack plans. We focus here on a set of simulation experiments where we approximate the behavior to be produced by STRATUS in order to evaluate its responses given a variety of missions and attacks on those missions. We show the relative merits of responding to threats using local, reactive responses versus strategically ones and present evidence that justifies combining the two.

Keywords—cyber-security, plan-recognition, plan-generation, intrusion-detection, multi-hypothesis-tracking, model-based-diagnosis, middleware.

I. Introduction

Defending cloud computing resources against cyber attacks once exterior defenses have been breached can be extremely difficult given, among other things, the uniformity of the underlying systems environment. We have been developing STRATUS as part of a larger program that is looking at how cloud-based distributed system resilience can be enhanced through new hardware and software approaches. STRATUS focuses on improving resilience by quietly managing distributed process commications channels and introducing redundant processes when systems are threatened and removing or fishbowling them they are compromised. It uses both a tactical approach and a strategic approach to identifying threats and selecting responses. The tactical approach models threats as contagious and distrusts neighboring processes accordingly. The strategic approach anticipates threats to mission related processes and recognizes attack plan as they unfold in order to anticpate how to deploy defenses. In this paper we briefly describe STRATUS' approach to tactical and strategic defenses and discuss a series of experiments we are conducting using a simulation of STRATUS' expected behavior against generated attacks to evaluate and refine our approach.

II. STRATUS Description

STRATUS (see [1] for full details of the functioning of the STRATUS system). is designed to work in an anticipated future environment in which hardware and OS or VM improvements in cloud clusters provide process isolation within a given host. Despite those improvements, attacks can still succeed if the attacker knows how to utilize application-specific distributed communications pathways to insert exploitive payloads. Such attacks would require knowledge of the application systems involved, and specific plans to target particular resources.

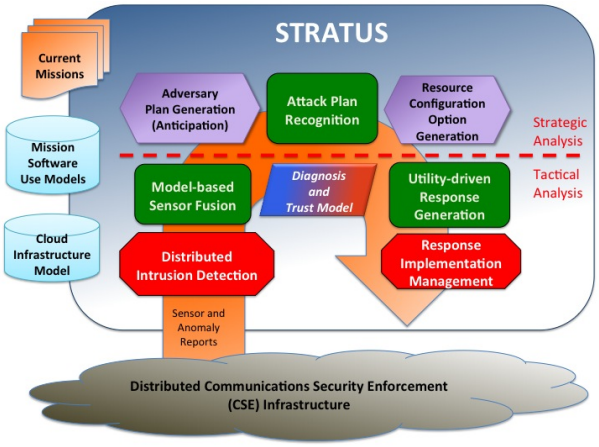

Fig. 1: STRATUS information flow architecture

STRATUS tries to provide mission-level resilience despite such attacks by using modest overhead in computational resources to diagnose attacks, switch rapidly to computed backup contingencies, and predict downstream events in order to make mission critical functions resilient to those attacks. Figure 1 provides an overview of STRATUS' functional organization. STRATUS assumes a suite of anomaly detection sensors is deployed in the mission-level interprocess communications network managed by its distributed object-oriented communications infrastructure, CSE (Communications Security Enforcement). Sensor reports of process failure or disruption, anomalous communications, etc. are analyzed and fused into hypotheses about the most likely attack events (such as an attack from one compromised component on another) that would produce

those reports. CSE is responsible for implementing changes specified by STRATUS to the communications network channels between components, and starting and stopping processes. It also will, if STRATUS determines components have been compromised, isolate those components in 'fishbowls' where all of their communications are redirected to special fishbowl manager components where their messages can be monitored while simultaneously denying them access to the active system elements.

STRATUS' tactical analysis uses the hypothesized attack events to determine the most likely attack sources and next targets. Possible targets assume adjacency over communication channels or co-residence on a host as means for contagion spread. Once an attack is considered to have occurred at a node, attack sources may be determined by a diagnosis strategy that can use weak evidence of anomalous behaviors on paths from the network edge to the attacked components. Both attack sources and possible nearby targets become less trustworthy, as they may respectively be or become compromised and need to be replaced. The main methods of responding to such events are to shut off communications from compromised components and to start backup processes that can then be used to replace quickly components when evidence that they are compromised increases.

STRATUS' strategic threat analysis is based on the identification of the most likely attack plans from a wide set of possible attacks that were generated by the system in advance. STRATUS' attack plan recognition system identifies the most likely attack plans based on the event hypotheses derived from observation evidence and, when it can do so before the expected target is reached, those attacks can be thwarted using the same kinds of responses: channel reconfiguration, the replacement of processes by backups, and the shutdown or fishbowling of compromised components.

The difference between the two defenses is roughly that the tactical model tries to back up as many neighbors of corrupted components as it can given available resources, while the strategic defense tries to *anticipate where the attack plan is going* in order to get ahead of the attack and uses spare resourses to defend those elements. The tradeoffs between these approaches arise from the partial observability of attacks and the resulting uncertainty in the conclusions derived, especially in large, densely connected networks. Our goal in this study is to explore tradeoffs between the approaches given limited reserve resources, so that we use each where it is most effective.

III. Simulator Description

The simulator serves as an efficient way of approximating performance results for the STRATUS system. The simulator is implemented as an abstract model of the entire STRATUS system; it simulates the mission components and their communications pathways in the CSE channel network, the mission tasks being performed, and the attack plans and attack actions unfolding over time. STRATUS defensive responses to attacks are interleaved with the attacks themselves. The simulator first generates a random mission network model. It builds attack plans against that mission network, and when it executes, interleaving attack actions and then defensive responses, the attack actions are successful if they haven't been blocked by prior defensive actions, and sometimes cycles occur. STRATUS is given 'observations' of the attack events with a some probability and these observations are used to develop STRATUS' belief state.

While components in the real STRATUS network can have a wide variety of states, there are only five main ones: live, shutdown, compromised, backup, and fishbowled. A live component performs its tasks and may be compromised. Compromised components can broadcast infectious data through the network along the CSE channels that connect it to other functionally related components. A component in the backup state (started but not transmitting data) may be transitioned to live, but this process takes time (currently one time step). Finally, a fishbowled component, (which would be placed in that state because STRATUS believes it is compromised) is disconnected from the other live components but its communications (to a special receiver) continue and are monitored. It has no further impact on the network. In the current simulation, fishbowled components and shutdown components are equivalent, so we distinguish four states.

Mission Model represents the task to be achieved by the network. For simplicity, each component only does one task, which may involve subtasks which require it to request and receive results from subtask components. The task is modeled as an AND-OR tree [2, pg. 435]. Tasks at AND nodes can only be achieved if all child-tasks of the AND node are accomplished. OR-nodes only require that some child succeed to succeed themselves. Tasks in the network are never really permanently achieved, but must instead be sustained for the duration of the mission, and so re-achieved if they die. The relationship between tasks causes failure in one part of the network to cascade to other areas, even when the attacker does not directly target those other nodes, modeling that the output from one objective typically serves as input to another task down the line.

Attack Model describes various forms of attack against the mission and its underlying network. We consider both point-to-point attacks, which attempt to compromise a single node in the network by infecting a shortest path through other nodes in order to reach it, as well as flooding attacks which attempt to bring down every node in the network by means of contagion.

Defense Model Builds tactical, strategic, and combinations of tactical and strategic defense plans for the

network from beliefs about the current state of the network and mission. We discuss each of these in further detail below.

A. Modeling Belief States

To mount a reasonable defense of the modeled network and mission, the simulator must maintain an internal model of the possible states of network nodes and ongoing attacks. We refer to this belief about the state of the network and attack as the belief state of the system. In STRATUS itself, this is called the trust model, and is a qualitative uncertainty model. For simplicity, the simulator's partially correct beliefs are formed by introducing weighted random noise over the actual simulated world state and independently randomly making only some events observable.

Constructing a belief state for STRATUS from simulated "ground truth" requires that we introduce noise over two aspects of the belief: the state of nodes in the network and the identity of the attack that is occurring. The latter only affects the strategic defense. The belief state for the network consists of a belief about whether each node is compromised or not (true or false). For every node in the network, we throw a weighted coin to decide whether to report truth or to lie about that node. Lying means reporting that a node is fine when it is compromised, and vice versa.

The contagion model used in STRATUS infers additional uncertain belief about compromised nodes from an observed infection event. When it sees that a given node has been infected, it infers with lower certainty that it's neighbors are also infected. In the current simulator, we do not model this belief explicitly, but the tactical defense acts to protect the neighbors of infected nodes, up to its resource budget, by giving the backups, which is much the same as the practical effect of STRATUS' contagion model of belief.

The simulator also throws a weighted coin to decide which attack is occurring, so that its approximation of the strategic defense can operate. Again, if it is heads, it is given a belief that the actual attack is occurring, and if tails, then it believes some other (randomly chosen) attack is occurring. No attack is one possibility.

B. Defense Models

The defense models are different approaches to responding to beliefs about the current state of the simulated mission, either by planning a response to expected attack actions several steps into the future (the strategic defense model) or by computing a response directly to corruption in the current believed state of the network, with minimal consideration of future states (the tactical defense model). The combined defense, as the name implies, combines these two approaches to network defense into a single coherent defense.

Tactical Defense: The tactical or reactive defense model constructs a set of actions that will bring damaged elements of the network into a state where the current compute mission could be completed (the components needed for all unfinished task elements are operating normally). STRATUS' defensive actions are: restarting crashed components, fishbowling or shutting down and perhaps restarting corrupted components, and stationing backup components around the network so if the components that are backed up are subsequently compromised, they can more quickly be returned to a normal operating state. The reactive defense only addresses nodes that it believes are infected, not those that actually are, which it cannot know. For each node believed to be infected, it determines if there are sufficient resources available to fishbowl it. If so, it fishbowls the node and leaves it running. If not, it activates an existing backup or just kills and restarts it. In addition, if enough resources are available, it creates backups for the neighbors of those nodes so that they can be restarted more quickly, approximating the STRATUS notion of contagion of distrust, where, if enough evidence has accumulated that neighboring nodes might be corrupted because its neighbors are, it treats them as also corrupted.

Strategic Defense: In STRATUS, the strategic defense mechanisms only apply when it has a strong belief that a particular attack is correctly identified, in which case it uses its defenses to ensure that the path to the node that is the attacker's goal is well backed up and channels from known infections on that path are disconnected from those goals. The current simulator emulates the strategic defense somewhat less accurately. With probability equal to the experimental variable **model accuracy** it believes the attack is the one that is actually occuring. Otherwise, it believes it is another randomly chosen attack. [1]

Its defense against whatever attack it believes is occurring is constructed by seaching for the plan that maximizes the utility of the given mission, given that the remaining steps in the attack occur in the future. It does this using anytime tree search [3]. From an initial network configuration, the strategic defense planner simulates the network (including an attack) going forward to some fixed time cutoff, or until the mission is completed. The utility of each resulting plan is calculated and the one with highest utility is the new strategic defense plan. These plans typically consist of restarts and fishbowling of nodes along the attack plan pathway, as opposed to the reactive model, which

[1] *maybe move to conclusions?* – At present, the simulator's belief about what attack plan is occurring when it "believes" in the wrong plan can be inconsistent with the nodes it also believes are corrupted. Furthermore, in STRATUS, it's estimate of the certainty in the possible plan(s) it thinks might be occurring goes up with the number of observations, while in the simulator, it believes it is one plan and with only a certain constant probability it is the correct plan. We are in the process of correcting these shortcomings for the next iteration of the model.

focuses on backing up nodes adjacent to the observed corruption.

Combined Defense: The combined defense is, as the name implies, a combination of the strategic and tactical defense models. At a high level, the combined defense computes both the tactical and strategic defenses, taking the actions from both where those actions don't conflict with one another. In the case of a conflict, the strategic actions are preferred over the tactical actions, given available resources, so we might think of the combined defense as a strategic defense supplemented by further actions suggested by the tactical defense. [2]

Combining the two approaches to defense can be quite beneficial, as we will see in the experimental evaluation. One of the purposes of the current set of experiments is to determine which model is better when, and determine the right way to combine the two.

IV. Experiments

Having discussed STRATUS and its simulator we will now investigate the performance of parts of the simulated STRATUS system. In particular, we investigate the differences between the tactical, strategic, and combined defenses of the network. After presenting the results of these experiments, we summarize them and briefly discuss what implications these studies have for future work in the STRATUS system.

A. Defense Model

We consider two forms of defense within the simulator: a planning based strategic defense model, a more rule oriented tactical defense model, and the combination of these. We compare these approaches along two axis: the impact over-provisioning has on each defense as well as the impact of belief accuracy on the effectiveness of the defenses. We additionally investigate the impact the contagion model has on defending the network. We will see that the reactive approach tends to have better performance when our beliefs are uncertain, that the strategic approach to defense can make more efficient use of resources so long as the underlying model of the network and attack are accurate, and we find that explicitly leading an attack has better performance than relying on the contagion model alone.

1) Performance vs. Provisioned Factor: We begin by examining the performance of the defenses as a function of the number of resources assigned to the network, with results shown in Figure 2. In the figure, the x-axis represents amount of resources allocated to the network as a factor of the number of nodes in the network. So a network that has one resource for each node in the network has a provisioned factor of 1, while a network with a provisioned factor of 2 could backup every node in the network.

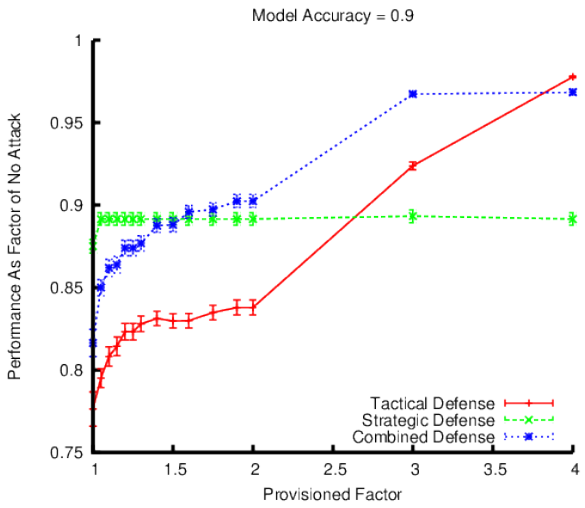

Fig. 2: Planning provides more benefit with fewer resources.

The y-axis shows the performance of the defenses relative to the performance of the mission when no attack is conducted. We do this for two reasons: it provides an upper bound on the performance of the system and it makes the performance of defenses comparable across instances. Recall that the score achieved by a simulation is dependent, in part, on the number of nodes in a network. By normalizing scores to a theoretical maximum score for the instance, we make comparison across instances of differing sizes possible. The lines show the mean performance of one hundred random samples of the given defense technique, while the error bars show 95% confidence intervals about the mean.

In Figure 2, the strategic defense model is represented by a green dashed line, the tactical model is shown using a solid red line, and the combined approach is shown using a dotted blue line. Unfortunately, we see that there is no one approach to mission defense that dominates the other two. For the least provisioned networks, the strategic defense has the best performance. Once we pass a provisioned factor of about 1.5, the combined defense begins to have better performance than either the strategic or tactical defense alone. Finally, for extremely well provisioned networks, we see that the tactical defense model has the best performance.

First, we consider the fact that the strategic defense alone performs better than the combined defense for

[2] More specifically, we can think of the strategic defense as a queue consisting of a set of steps, ranging from s_0 to s_n. At each step in the simulation, we pull the set of actions to execute for the defense from the queue, and whenever the queue is empty, we construct a new plan, thus refilling the queue. For the combined defense, every time we pull a step in the strategic defense from this queue, we also compute a tactical response, t. We then combine s_i and t by assuming that each element in the network can take only one action per round, and giving the actions in s_i preference in the case of a conflict.

networks with few resources (provisioned factors between 1 and 1.5). While we do take some care not to hinder the actions taken by the strategic defense with the added tactical actions in the combined defense, we only consider interactions at a single time. If the added tactical defense suggest a action taking more than a single step, say re-imaging a network element, it could potentially hinder the strategic defense further down the line, resulting in diminished performance seen in the plot.

One might naturally expect that the combined defense would provide the better of strategic and tactical defense performance, but as we see in Figure 2, the combined defense is often better than either of its components alone. This is likely because of the differing goals of the strategic and tactical defense. The strategic defense is interested in defeating predicted attacks, while the tactical defense is concerned with protecting mission performance in the immediate sense. Thus, the tactical defense will consider actions which the strategic won't, namely having multiple live versions of resources when provisions allow. Thus, the combined defense will attempt to drive attacks out of the network while trying to keep as much of the mission going as possible, while either defense alone will only focuse on one aspect of the defense.

The tactical defense model does eventually dominate the other defense models it simply takes a large number of resources. One might suspect that a provisioning factor of 2i s sufficient for perfect performance (one live, one backup), however because of delays in reimaging a corrupted machine and standing up backup nodes, more resources are required for perfect reliability. Again, the imperfect combination of defense hinders the performance of the combined defense model in an extreme case.

2) Performance v. Observation Probability: Figure 3 considers the performance of the two defensive approaches along our other axis of control: belief accuracy. A model accuracy of 1 means that we always report the true state of the network to the defense, whereas a model accuracy of 0.5 is just noise. As before, the y-axis shows how effective the defense was, relative to the same network not under attack.

Figure 3 shows that, as previously noted, the strategic defense model requires an accurate representation of the state of the world in order for its planned defense to be at all effective. We see that it is frequently worse than the tactical defense. That the strategic model performs well when the beliefs are very accurate makes good sense: planning is an effective technique when you can effectively reason about future states of the network from the current state. The combined defense has better performance than the strategic alone, but it is still not competitive with the tactical defense for situations with poor information. While it may use portions of the tactical defense, recall that it gives preferential treatment to

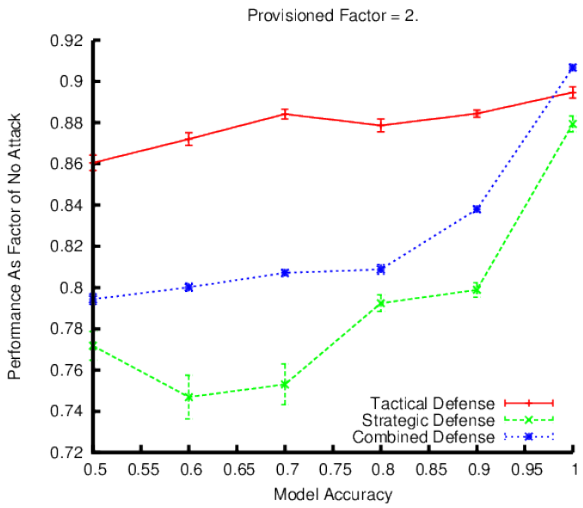

Fig. 3: Strategic defenses require accurate world models.

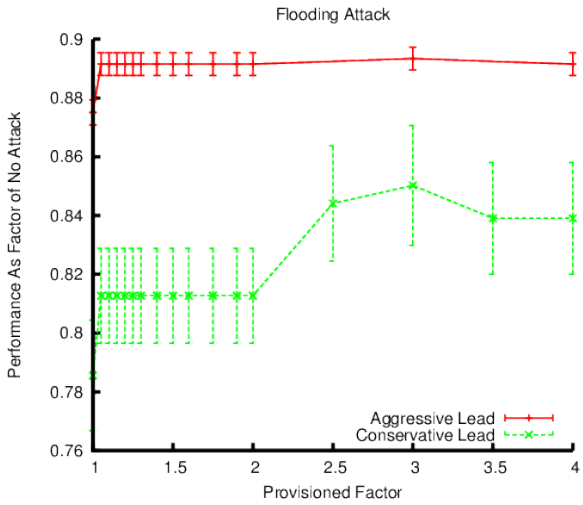

Fig. 4: Impact of Attack Prediction on Defense Under Flooding

the actions from the strategic defense model, explaining its poor behavior.

3) Leading the Attack: The strategic and combined defense models require that we attempt to predict the next several steps of an ongoing attack, which we refer to as leading an attack. Leading the attack is trying to predict, given the currently compromised nodes (and optionally an estimated attack) which nodes are likely to be compromised next.

Figures 4 and 5 illustrate that the way in which we lead the attack needs to align with the attack itself. If we have a model of the ongoing attack leading the attack is easy. We simply map the existing network state to a point in the assumed attack, and then look at the next

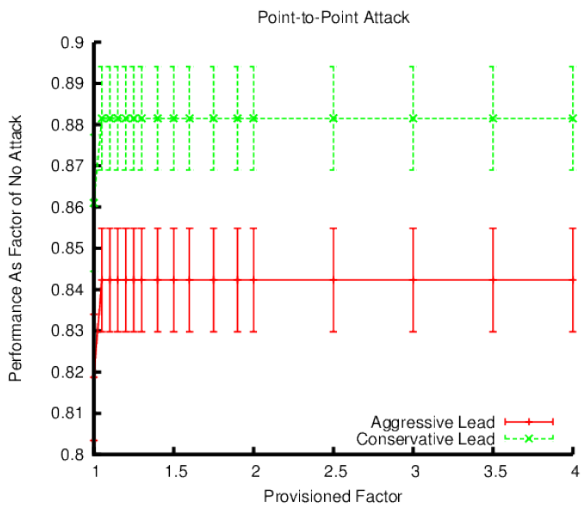

Fig. 5: Impact of Attack Prediction on Defense Under Point-to-Point

steps in the attack in order to lead it.

In the event that we have no such model, or alternatively that we do not trust the model that we have, we have to lead the attack solely from the state of the network. We consider two methods of leading the attack, either one node, selected at random from uncorrupted nodes adjacent to the most recently corrupted node is going to be attacked next, or all nodes adjacent to the most recently corrupted node are going to be attacked next. We refer to these as the conservative and aggressive leads respectively.

We examine the tactical defense model using these two kinds of leads under flooding attacks, and point-to-point attacks. Note that, for the flooding attack, the aggressive lead has performance that dominates the conservative lead, while for the point-to-point attack, we observe exactly the opposite. So long as the lead matches the actual attack, performance is improved.

B. Summary

To summarize, we examined the performance of the simulation of STRATUS under a variety of configurations. We saw that the amount of additional provisions, accuracy of the networks model of itself, and even the type of ongoing attack all play critical roles in determining the effectiveness of any defense of the network. Specifically, we saw that the accuracy of the underlying belief was the determining factor in the effectiveness of strategic defense; an inaccurate world model cannot be used to reliably plan into the future. Provisioning also plays a key role in defending the mission, and given the amount of time it takes to clean out and bring an infected machine back on line, we actually need far more resources for a completely reliable system than we might first expect. Finally, while it is ideal to know exactly what attack is going on in the network, measurable benefits can be had from accurately identifying the nature of the attack (e.g. flooding versus targeting a single machine).

V. Conclusion

We investigated the relative merits of two techniques for responding to a cyber threat against a distributed system, which is the goal of the STRATUS project. We found that reactive plans for network repair and defense as well as longer lived strategic plans to defeat recognized attacks both fail to dominate one another in a distributed defense setting. Strategic plans are able to provide a better defense with fewer resources, but rely on very accurate models of the state of the network which may be difficult to obtain. While reactive defenses are not as reliant on model accuracy, they require far more resources to provide the same level of resilience as a strategic defense. The experimental results inform a decision procedure for when to use which of the two defenses investigated and suggest a number of future directions to explore in improving the STRATUS system.

Acknowledgment

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 author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

[1] M. Burstein, R. Goldman, P. Robertson, R. Laddaga, R. Balzer, N. Goldman, C. Geib, U. Kuter, D. McDonald, J. Marist, P. Keller, and D. Wile, "Stratus: Strategic and tactical resiliency against threats to ubiquitous systems," in *Proceedings of the Adaptive Host and Network Security Workshop*, 2012.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2003.

[3] A. Caprara and M. Fischetti, *Annotated Bibliographies in Combinatorial Optimization*. John Wiley and Sons, 1997, ch. Branch and Cut Algorithms, pp. 45–63.