CONF-890736--8

The Outsider Interruption Algorithm

Mark Snell
Sandia National Laboratories,* Albuquerque, NM 87185

Bryan Bingham
Science & Engineering Associates, Inc., Albuquerque, NM 87110

SAND--89-1604C

DE89 014799

ABSTRACT

The Outsider Analysis (Outsider) module is part of the Analytic System and Software for Evaluation of Safeguards and Security (ASSESS). Outsider and the ASSESS Facility Descriptor (Facility) module together supersede the Systematic Analysis of Vulnerability to Intrusion (SAVI) PC software package. Outsider calculates P(I), the probability that outsiders are interrupted by security forces at a facility during an attack on the facility, and P(W), the probability of security system win. SAVI exhaustively examines every possible path to find the ten most vulnerable paths. Exhaustive search is adequate if the number of paths to examine is small, but moderately complex facilities can have millions of paths, making exhaustive search too slow for practical purposes. Outsider has two new algorithms that generate paths in order of vulnerability, finishing in a fraction of the time required by SAVI. The new Outsider algorithms make containment analysis easier for analysts than ever before. We describe the new algorithms and show how much better they perform than the SAVI exhaustive search algorithm.

INTRODUCTION

The Outsider Analysis (Outsider) module is part of the Analytic System and Software for Evaluation of Safeguards and Security (ASSESS)[1,2]. Outsider and the ASSESS Facility Descriptor (Facility) module together supersede the Systematic Analysis of Vulnerability to Intrusion (SAVI) PC software package[3,4]. Using new, fast algorithms that generate paths in order of vulnerability, Outsider calculates the vulnerability of a facility against intrusion by outside adversaries. We present a review of the timely detection model that defines vulnerability in terms of the probability of interruption, P(I), and discuss how the new algorithms work and how their performance compares to the SAVI exhaustive search algorithm.

*This work was supported by the United States Department of Energy under Contract DE-AC04-76DP00789.

THE MODEL

To calculate the vulnerability of a physical security system against intrusion by outsiders, Outsider takes a representation of the security system, called an Adversary Sequence Diagram (ASD), created by Facility, and uses the model of timely detection to determine the probability that the security system can detect intrusion with enough time to deploy a response force to interrupt the adversaries before they complete their mission[5]. P(I) is the primary measure of vulnerability.

An ASD created by Facility consists of up to ten physical areas separated from one another by protection layers made up of 0 to 15 path elements (PEs). The Target at the bottom of the ASD is protected by target location elements, which are also path elements. Figure 1 shows a small ASD with three areas and five PEs representing the facility diagrammed in Figure 2. The Target, a can of plutonium, sits on an open shelf inside the Material Access Area and is therefore modeled as an Open Location Target Location Element (OPN) in the ASD.

Areas represent zones with no security safeguards; intruders are free to move anywhere in the area with no chance of detection. To cross between areas, intruders must pass through PEs. PEs contain all safeguards: detectors, barriers, security inspectors, searches, and ID checks. Crossing a PE subjects intruders to the safeguards installed and active in the direction the intruder is traveling. The Target itself (the plutonium can in Figure 1) provides no additional security, but might have characteristics such as radioactivity that trigger detectors in PEs crossed later. The same Target is assumed to exist in each target location element.

A PE can connect two areas not adjacent in an ASD. The Tunnel PE in Figure 1 is called a jump element because an intruder can go from Offsite directly to the Material Access Area. A protection layer can have zero PEs, meaning there is no security system preventing movement from one area to the next, or between the last area and the Target. A special PE called a bypass is displayed in that case; bypasses behave just like PEs with no safeguards installed.

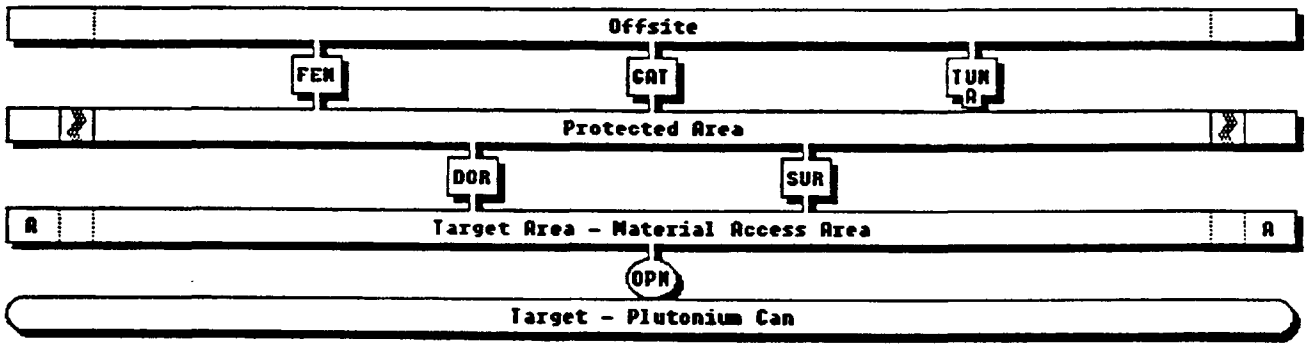


Figure 1. Example ASD

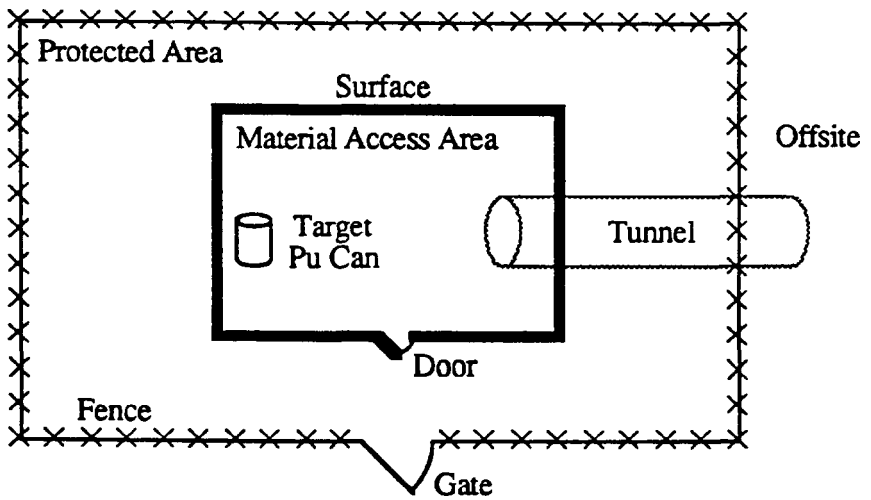


Figure 2. Example Physical Protection System

Adversaries can use two methods to defeat the safeguards at a PE: force/stealth and deceit. Force/stealth means the adversaries attempt to disable or destroy the safeguards they encounter using tools, explosives, or weapons. Deceit means adversaries use falsified credentials and smuggle contraband equipment past security inspectors and contraband detectors. Some PEs such as surfaces cannot be crossed using deceit; all PEs can be crossed using force/stealth.

The analyst can choose one of two response strategies to protect the Target: denial or containment. Denial means the response force must interrupt adversaries before they reach the Target—successful denial prevents sabotage. Containment means the response force must interrupt adversaries sometime before they leave the facility —successful containment prevents adversaries from stealing the Target. A security system that successfully prevents sabotage also prevents theft, but is usually more expensive.

A path is the series of areas and PEs crossed by adversaries trying to defeat the response strategy, starting with Offsite, the first area in every ASD. If the response strategy is denial, adversaries must penetrate down to the Target, generating an entry path. To steal the Target, adversaries must first get the Target, then exit the facility, crossing PEs in the opposite direction, generating an entry/exit path. For example, a denial path for Figure 1 is {Offsite, FEN, Protected Area, DOR, Target Area, OPN}, and a containment path is {Offsite, TUN, Target Area, OPN, Target Area, TUN, Offsite}.

The term "path scenario" means a path with the adversary tactic, force/stealth or deceit, used to cross each PE identified. For example, a denial path scenario for Figure 1 is {Offsite, FEN-FORCE, Protected Area, DOR-DECEIT, Target Area, OPN-FORCE}. A path can have many different path scenarios, but only the most vulnerable path scenario is interesting. Therefore, the vulnerability of a path is defined to be the vulnerability of its most vulnerable path scenario.

The vulnerability of a path scenario is measured by P(I), the probability the intruders are detected with enough time left for the response force to interrupt them before they complete their mission, based on the response strategy. The Response Force Time (RFT) is how long it takes for the response force to deploy and interrupt intruders after correctly assessing an alarm. Detection must occur at or before the Critical Detection Point (CDP) in the path, or else the intruders can complete

their mission because the response force cannot deploy fast enough to interrupt them. The accumulated detection probabilities in each PE before and at the CDP determine P(I).

Figure 3 shows how the RFT, CDP, and P(I) are related. RFT is measured back from the end of the path by summing the delays ($t_6+t_5+t_4$...) provided by barrier safeguards and transit times. The point in the path where the adversaries require greater than RFT seconds to complete is the CDP. Detection safeguards (p_1, p_2, etc.) at CDP and back up the path to Offsite provide useful detection; their accumulated probability of detecting the intruders is P(I). The difference between the actual time to finish the mission starting at CDP and the RFT is called the Time Remaining after Interruption (TRI).

It is possible for paths to have insufficient delay in them to place the CDP inside the facility. Because the P(I) for these paths is zero, they are always listed as the most vulnerable paths for the given RFT.

SAVI and Outsider find and report the one to ten most vulnerable paths for a range of one to ten RFTs. The most vulnerable path for a given RFT is the path with the lowest P(I). If two or more paths have the same P(I), the Detection Potential (DP) is used to break the tie in Outsider. DP is the count of places inside the PEs before CDP where detection can be installed. A PE can have up to four detection points. If the DPs are equal, the path with the smallest TRI is considered the most vulnerable. DPs are not used in SAVI; only TRI is used to break ties.

SAVI uses the same algorithm for denial and containment analysis. It is simple to state: generate every possible path scenario; find the CDP and calculate the P(I) and TRI; record the path scenario if it is one of ten most vulnerable path scenarios found so far. SAVI runtime performance is proportional to the number of path scenarios in the facility. For denial analysis, the number of paths can be relatively small, partly because SAVI limits the number of protection layers in a facility to five, with no more than 11 PEs in each layer. For containment analysis, however, the number of entry/exit path scenarios is proportional to the square of the number of denial scenarios. A moderately sized ASD can have millions of entry/exit path scenarios; containment analysis for such an ASD can take hours or days. SAVI containment analysis is not as useful as it should be because it often takes too long to determine initial system vulnerability and then test upgrades.

Our goals for Outsider were to invent new algorithms that reduce containment run-time, support bigger ASDs, and incorporate more sophisticated modeling, including poorly equipped threats, contraband smuggling, and less than perfect alarm assessment.

THE OUTSIDER ALGORITHMS

Outsider uses a modified shortest-path algorithm for denial analysis and a custom-designed greedy algorithm incorporating the denial algorithm for containment analysis. Both algorithms generate paths in order of vulnerability and stop when all desired paths are found.

Before generating any paths, Outsider (and SAVI too) first calculates the minimum delay across each PE and area, and the minimum probability of detecting intruders at each PE in both directions for both force/stealth and deceit penetration. This calculation is done by using a database of intrusion equations that define all ways intruders might try to defeat the safeguards conceivably present at each kind of PE. An intrusion equation is a list of safeguards that, if present and active in a given direction, intruders must defeat. Outsider performs more sophisticated intrusion equation processing than SAVI.

The performance of a safeguard depends on its type and what threat equipment intruders have. Some safeguards perform the same no matter what kind of threat equipment intruders possess. Other safeguards, such as contraband detectors, provide no detection if the intruders do not possess the right sort of contraband. Other detectors might perform better in the presence of threat equipment like vehicles, or worse because the intruders use the equipment to tamper with or disable the safeguard. Safeguard performance values can be user-defined in Facility; this means Outsider uses those values instead of the safeguard performance database values.

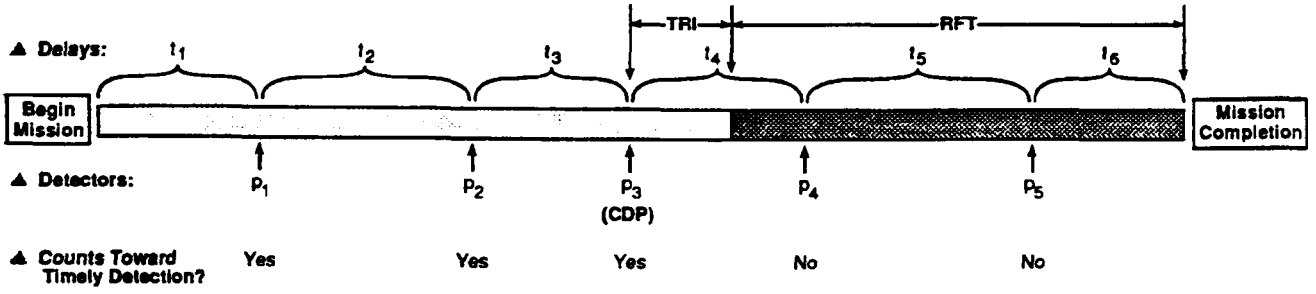


Figure 3. Timely Detection Timeline

It is possible that a safeguard such as a concrete wall can provide effectively infinite delay against intruders who do not possess any equipment capable of penetrating or disabling the safeguard. If all intrusion equations at a given point in a PE contain an infinite delay safeguard, Outsider considers the element impassable by the intruders in that direction using force. It is possible to model facilities that cannot be forcibly penetrated by poorly equipped intruders, providing perfect security.

Outsider lets intruders attempting deceit exploit any gaps in search and contraband detection policies at a PE. Contraband can be smuggled through a PE by hiding it on the intruder's person, in various kinds of packages, or in a vehicle, if intruders are equipped with one. Outsider considers all possible ways to hide all the contraband intruders might possess and chooses the way that minimizes detection for a given deceit equation. For intruders with no equipment, deceit is usually the only viable method of penetration; if there is even one locked door in a path, then penetration is not possible.

A dependent PE is a PE crossed on entry that the intruders re-cross on exit. Dependent PEs are force-dependent if the intruders used force/stealth on entry, or deceit-dependent if the intruders used deceit on entry. Outsider models force-dependent PEs as providing no security on exit; intruders using force/stealth on entry disable or destroy all safeguards they encounter. Intruders using deceit to minimize detection on entry are assumed able to totally defeat the badge checks and searches in the same PE on exit because they beat them before on entry. Future versions of Outsider might modify this conservative assumption.

The denial algorithm is a modified version of the Dreyfus K-best shortest-path algorithm[6]. "Shortest path" in Outsider means "most vulnerable path," and most of the modifications are related to the measurement of vulnerability, which in Outsider is a combined measurement of delay in seconds and detection as a probability. After intrusion equation processing, Outsider constructs what we call a Dreyfus network from the ASD. In the network for the example facility (Figure 4), each PE is an arc, each area is a node, and a path is a sequence starting at Offsite and proceeding to the Target.

Given the Dreyfus network, Outsider generates the most vulnerable path. Starting at the Target, Outsider examines each PE leading into it. Each PE and its source area are labeled with the delay necessary to cross the area and PE to get to the Target. If the delay is greater than RFT seconds, a CDP falls on that PE, and it and the area are labeled with a P(I), DP, and TRI. If two or more PEs connect an area to the Target, the PE offering the most vulnerable approach to the Target is recorded in the area as the PE to choose. After all PEs directly connected to the Target are labeled, all PEs and areas that connect to the next higher area are examined in the same way, and so on, working backwards from the Target to Offsite, until every PE and area has been labeled. After this is done, the area labels starting at Offsite can be read to find the most vulnerable path.

Because the Dreyfus K-best path algorithm stores more information in each label to mark which areas and PEs are used each time the algorithm is called, it can produce the K shortest paths in order of vulnerability by calling it K times.

Containment analysis is more difficult than denial analysis. Shortest-path algorithms like Dreyfus's do not work for entry/exit paths because of dependent PEs. An entry scenario, the sequence of PEs from Offsite to the Target and their defeat methods, must be known in order to determine which exit PEs are dependent. Given an entry scenario, the denial algorithm can be used to find the K most vulnerable exit scenarios. Figure 5 shows a Dreyfus network with an entry scenario and exit scenarios, with dependent PEs labeled. Therefore, a possible containment algorithm can be stated as: generate each entry scenario, construct a Dreyfus network from the entry scenario and all exit scenarios, run Dreyfus to get the K most vulnerable paths based on that entry scenario, and keep a list of the most vulnerable paths found.

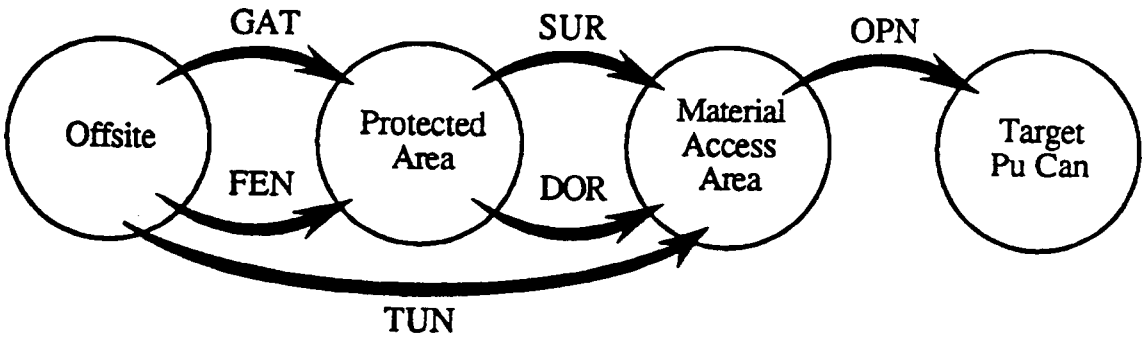


Figure 4. Dreyfus Denial Network

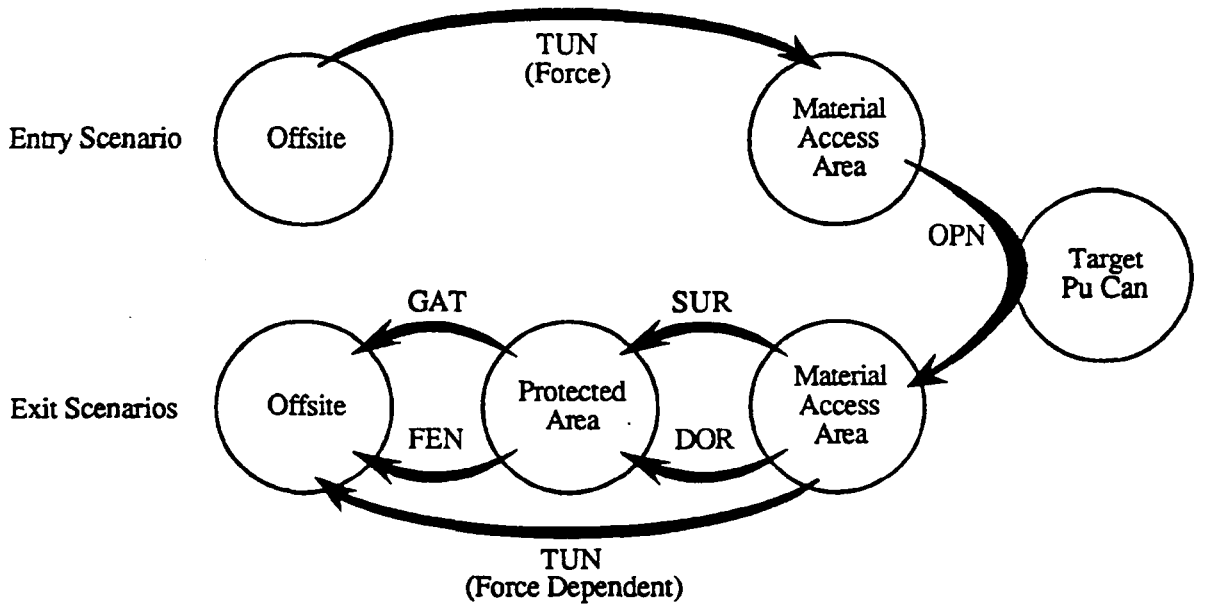


Figure 5. Dreyfus Containment Network

Such an algorithm runs in time proportional to the number of entry legs, similar to the SAVI denial analysis. This is a big improvement over SAVI containment analysis, but there is an even faster approach. Outsider generates entry path scenarios in order of vulnerability, halting after the K most vulnerable paths have been found instead of exhaustively generating every entry scenario.

Outsider generates partial entry scenarios starting from Offsite, extending the most vulnerable one found so far downward toward the Target by examining each PE leading out of the last area in the partial entry scenario, looking for a CDP. Because it always chooses to work on the most vulnerable partial entry scenario found so far, the algorithm is called greedy. When a CDP is found, the complete entry scenario is constructed and stored; otherwise the extended partial entry scenario is stored for later extension. When no partial entry scenarios exist that are more vulnerable than the most vulnerable complete entry scenario, that complete entry scenario is part of the most vulnerable path and is passed to the Dreyfus algorithm to calculate the K-best exit scenarios based on it.

Given a partial entry scenario from Offsite to an intermediate area, Outsider knows which PEs are dependent on exit from the intermediate area to Offsite. The extension PE is added to the scenario and checked to see if CDP falls on it on entry or anywhere on exit.

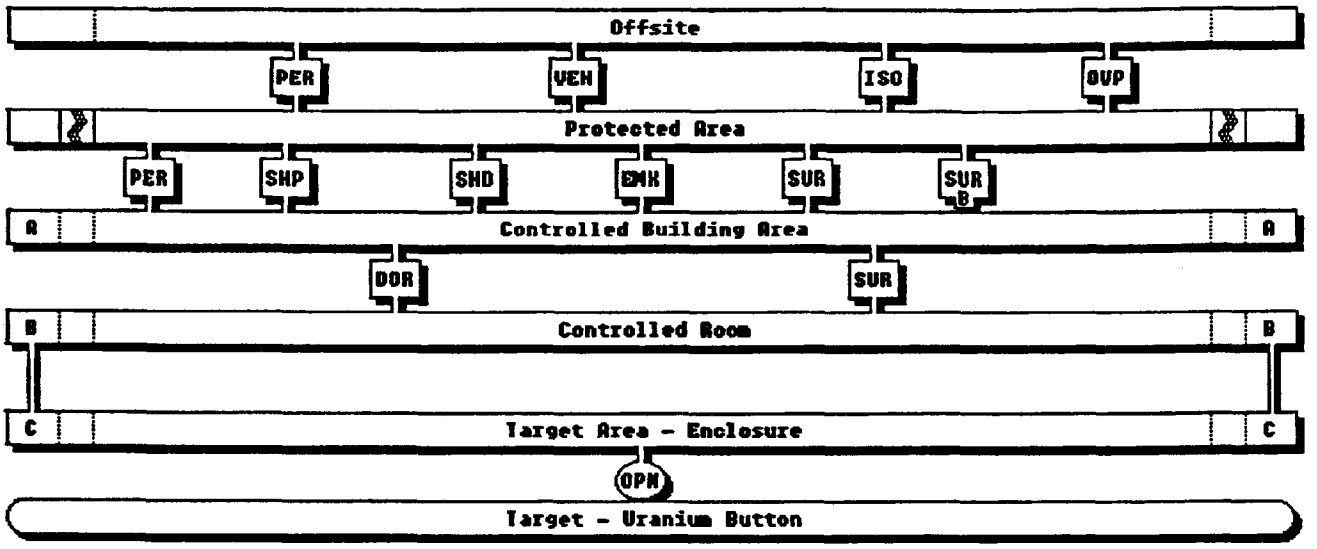
The containment algorithm terminates when enough complete paths have been put into the most vulnerable path list or when there are not enough complete paths in the ASD to fulfill the request. Because the greedy algorithm homes in on the most vulnerable paths quickly, Outsider containment analysis is often faster than SAVI denial analysis of a similar ASD.

PERFORMANCE

All the timing tests were performed on the same computer, a 12.5-MHz 80286 with expanded memory. Figure 6 shows the two ASDs. The Button Fabrication and Reprocessing Facility (BFR) ASD is used in the SAVI portion of the Threat Vulnerability Assessment Training Program (TVATP) course sponsored by the Department of Energy. Large is an ASD with more path scenarios tuned to demonstrate the worst-case behavior of the Outsider containment algorithm.

Both Outsider and SAVI spend time at the beginning of analysis processing intrusion equations and determining the detection and delay performance at each PE. SAVI generates and stores 24 numbers per PE, whereas Outsider generates and stores 101 numbers per PE.

BFR ASD



Large ASD

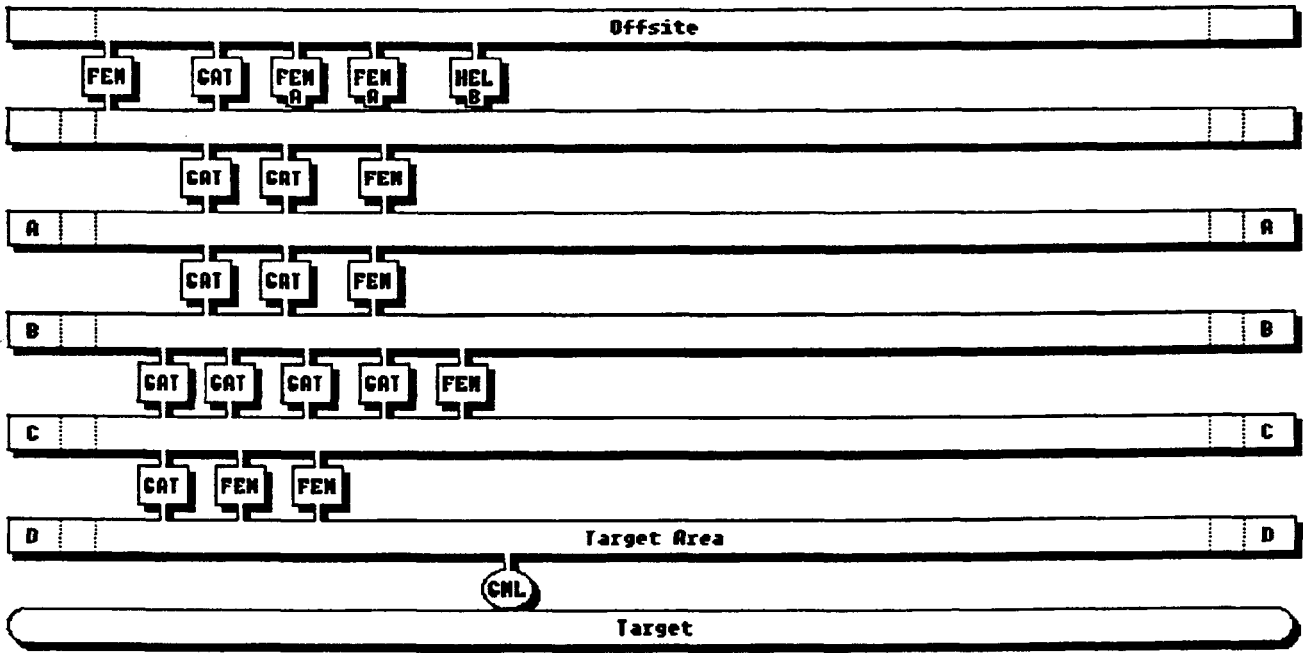


Figure 6. BFR and Large ASDs

Table 1 shows the run-times for the SAVI and Outsider denial algorithms. BFR is a small ASD, with only 60 denial path scenarios; thus, SAVI is actually a bit faster than Outsider because Outsider does more work at the beginning. But the Outsider denial algorithm is almost as fast for the Large ASD as for the BFR ASD. The run-time performance of Outsider denial analysis is dominated by the intrusion equation processing; the Dreyfus algorithm run-time is a small fraction of the overall time.

Table 2 shows the containment analysis run-times. Here the differences are dramatic. BFR is so small that SAVI runs in a reasonable length of time; Outsider runs about seven times faster. The Large ASD has about 6.7 million path scenarios, and SAVI examines each one, taking longer than one day to find the ten most vulnerable paths for ten RFTs. Outsider completes the same job in about four minutes, or 373 times faster. For one RFT, Outsider is 1400 times faster.

Table 1. Denial Run-Times

	BFR		Large	
	1 RFT (s)	10 RFTs (s)	1 RFT (s)	10 RFTs (s)
SAVI	1	2	12	66
Outsider	2	3	2	3

Table 2. Containment Run-Times

	BFR		Large	
	1 RFT (s)	10 RFTs (s)	1 RFT	10 RFTs
SAVI	20	57	10.5 h	25.5 h
Outsider	3	8	27 s	246 s

Outsider containment performance depends on two factors: how close the CDP of the most vulnerable path is to the Target, and how balanced the protection is across each protection layer. Worst-case performance is obtained when there are no jump elements, every PE has the same performance, and the first CDP falls in a target location element. Then the greedy algorithm can never find any good paths and simply generates all possible partial entry scenarios until the first CDP is found. The Large ASD meets these conditions, but few realistic ASDs will. Average run-time is impossible to determine because there is no way to define an average ASD. However, every Outsider containment analysis we have run on realistic ASDs takes under 30 seconds for ten RFTs.

SUMMARY

Vulnerability analysis using exhaustive search algorithms is impractical for all but small ASDs. For the ASSESS Outsider Analysis module we have developed two new algorithms that generate paths in the order of vulnerability and have improved the timely detection model. A version of the Dreyfus K-best shortest-path algorithm performs denial analysis. A custom-designed greedy algorithm that incorporates the Dreyfus algorithm performs containment analysis. Our tests show that, compared to SAVI, the new containment algorithm is anywhere from seven times faster for small ASDs to over 1000 times faster for large ASDs. The denial algorithm run-time depends on the amount of intrusion equation processing necessary; the actual path determination takes only a few seconds.

The Outsider vulnerability model is based on the SAVI model of timely detection, with many improvements and new features, including better threat and deceit modeling, support for larger ASDs, less than perfect alarm assessment, and an expanded set of safeguards and safeguard performance values. Timely detection is a mature model that effectively determines facility vulnerability against a wide range of threats with a minimum amount of subjectivity. The new algorithms make vulnerability analysis faster, easier, and more accurate than ever before.

REFERENCES

1. "An Overview of ASSESS – Analytic System and Software for Evaluating Safeguards and Security," T. D. Cousins, R.A. Al-Ayat, and J.C. Matter, INMM 30th Annual Meeting Proceedings, 1989.

2. "The ASSESS Outsider Analysis Module," A. E. Winblad, M. Snell, S. E. Jordan, B. Key, B. Bingham, and S. Walker, INMM 30th Annual Meeting Proceedings, 1989.

3. "The ASSESS Facility Descriptor Module," S. E. Jordan, A. E. Winblad, B. Key, S. Walker, T. Renis, and R. Saleh, INMM 30th Annual Meeting Proceedings, 1989.

4. "The SAVI Vulnerability Analysis Software Package," R.J. McAniff, W.K. Paulus, B. Key, and B. Simpkins, INMM 28th Annual Meeting Proceedings, Vol XVI, July 1987, 295-298.

5. "The SAVI Vulnerability Assessment Model," A. E. Winblad, INMM 28th Annual Meeting Proceedings, Vol XVI, July 1987, 24-28.

6. "An Appraisal of Some Shortest-Path Algorithms," S. E. Dreyfus, *Operations Res* 17, 1969, 395-412.