

An Initial Foray into Understanding Adversary Planning and Courses of Action

John Lowry

BBN Technologies, LLC
jlowry@bbn.com

Abstract

The term Information Warfare encourages a mapping between traditional or ‘kinetic’ warfare and the use of information technology. To prevail in modern warfare, you need a high degree of knowledge of implicit and explicit behaviors exhibited by defenders and attackers. These behaviors revolve around logistics, tactics, strategy, planning, and intelligence gathering and preparation. Little work has been done to examine these behaviors and activities in the domain of information warfare. The activity of planning and executing an attack, with fallback positions and alternatives is called attacker course of action. This paper describes an experiment that was designed to look explicitly for the broad characteristics of a sophisticated adversary during the planning and execution phases of an attack.

1. Introduction

This paper describes an experiment that was conducted as part of the Defense Advanced Research Projects Agency (DARPA) Information Assurance (IA) program. The experiment was designed to investigate the processes, methods, and behavior of a sophisticated adversary during the planning and execution phases of a computer network attack on a model DoD network simulating an operational configuration.

1.1. Information Assurance Program

The DARPA IA program is a multi-year effort to apply research results, new technologies, and improved understanding to the task of information system security assurance. The program uses a variety of DARPA-developed technology solutions and commercial and open-source technology. Many security solutions have been devised for information systems, but the IA program is unique in its research focus and the use of experimentation to test solutions.

1.2. Experimentation Approach

The IA program adopted an experimentation approach because there is no well-developed systems engineering methodology for information assurance that meets the requirements of real-world users. In the real world, there is wide use of fundamentally insecure commercial technologies, woefully inadequate configuration management capabilities, and a lack of expert human resources to design, deploy, and maintain systems securely.

Given the realities outlined above, the old model of hardened systems with strong configuration management, design, deployment, and maintenance had to be abandoned. Therefore, the IA program is based on “Grand Hypotheses” that were incorporated from modern military principles and that guide the research. The overall program hypothesis is: *Trustworthy systems can be built from less trusted components.* This is a basic tenet of the fault-tolerance system community.

Two sub hypotheses refine that stated above. The first, derived from military thought on defense in depth, is: *suitably layered defenses improve a system’s information assurance posture.* The second, derived from the principle that mobility is generally superior to static defense, is: *Dynamic defense increases a system’s information assurance and survivability.*

1.3. Model Adversaries

The IA program uses experimental methods ranging from “table-top exercises” or “white-board investigations” to engagement with a sophisticated model adversary. The latter are called “red team experiments.” This range of methods enables experiments of varying complexity to be conducted. In the model adversary method, the red team members are full partners in proposing, designing, executing, and analyzing the experiment. Their motivation is the successful exploration of the hypothesis.

A model adversary should have the characteristics of a true adversary. Unfortunately, not much is known about true adversaries, but gross characteristics and behaviors

can be adopted and examined based on analogy to types of other real-world adversaries and on analysis of current, real-world threats and attacks. The IA program chose a range of adversary classes to model. The least threatening adversaries do not receive any attention on this program. Sometimes called *crackers*, *script-kiddies*, or *ankle-biters*, this class is very visible and already receives a lot of industry attention. More significant threats are seen as coming from *organized criminals*, *cyber-terrorists*, *nation-state military*, and *foreign intelligence*.

The *cyber-terrorist* is the best-developed model in the IA program. The assumed characteristics of cyber-terrorists are as follows:

- ?? They typically work as a team or as a group.
- ?? They are well educated, well motivated, and generally well supplied with money, equipment, and other necessary resources.
- ?? They have a well-defined target and probably a particular time-window to attack the target.
- ?? They endure *risks* that can occur during planning and execution of an attack. Risk sensitivity is believed to change at various points but most significantly, is expected to sharply decrease after the goal is achieved. In other words, they often claim attribution.
- ?? They incur *costs* that appear as limitations in time, money, expertise, and other resources.

1.4. 'Dark Spaces'

In addition to the grand hypotheses, the IA program is investigating additional hypotheses with an aim to informing a broad research agenda in system construction and operations for cyber defense. These efforts have been referred to as investigations into "dark spaces" because they involve problem areas in which no research has been done before. The main areas are as follows:

1. **Course of Action (CoA).** Investigations into defender CoA development and evaluation of attacker CoA projection. These are roughly coupled through presumed dependencies and the assumption that the ability to predict one will help determine the other.
2. **Denial of Service (DoS).** DOS (and distributed denial of service, DDOS) attacks are fundamentally hard to prevent or counter. The IA program made an initial foray into DDOS, and this area is now a focused research area within DARPA independent of the IA program.

3. **Sensor Placement and Choice.** This area addresses the problem that sensor resources are limited in capability and number and that there is no theory or method to determine optimal placement.
4. **Insider Attack.** Insiders pose a significantly greater threat. There is currently no well-defined model of how inside attacks and attacker behavior differ from external attacks and attackers.
5. **Deception.** Research in this area seeks to understand the role and value of deception in preventing and defeating attacks.
6. **Information Condition Changes.** U.S. Department of Defense (DoD) policy for responding to information system attacks mandates fixed levels of response based on severity of attack threat or effect. There is little or no understanding of the effectiveness of these response levels or their effect on the ability of U.S. forces to carry out their missions.
7. **Identification and Authentication.** Many techniques exist for identification and authentication of humans, agents, and systems. However, there is little or no understanding of the costs and benefits of one approach versus another, or of how these techniques might be combined or integrated to achieve significant improvements in assurance posture.
8. **Coalition Policy.** To achieve their goals, both the U.S. DoD and, increasingly, commercial organizations must operate in coalitions or partnerships. Security policy and mechanisms to facilitate coalition formation and operation are not well understood. This is now a focused research area within DARPA.

2. Experiment Construction

The experiment described in this paper was designed with the primary goal of investigating adversary CoA. Previous experiments suggested that enough was now known about adversary planning and execution to make measurements and analyze adversary behavior. A secondary goal was to continue investigation into layering, i.e., defense in depth. Several previous experiments had looked at layering, and this experiment was intended to expand the scale of the target system and look at fine-grained policy. This experiment was designated Red Team experiment for year 2000, number 01, or RT0001. This designation further indicates a large-scale experiment with complex hypotheses and high execution costs. The number of people involved in planning and execution ranged from

ten to fifteen. Planning began in January 2000 and continued at varying levels of effort until execution in June 2000. Execution of the experiment lasted for five days.

2.1. Insights from Previous Experiments

Previous experiments into static layering showed that adversary work factor and risk could be increased by increases in the number of complementary defensive layers and by maintaining conscientious and complete configuration control over defensive assets. However, previous experiments also showed that the adversary could generally defeat the defender, either by taking advantage of incomplete configuration management and control or by attacking at points where the defender had allowed abstraction. Here, “abstraction” refers to the point at which the defender begins to make implicit assumptions about the configuration, capability, or features of the system.

The previous experiments generally followed best practice for deployment of layers, e.g., enclaves with firewalls, VPNs between enclaves, and *good hygiene* in terms of password management and availability of network protocols and services. That kind of configuration is often derisively referred to as *hard and crunchy on the outside, soft and chewy on the inside*. That is because there usually is only one significant defensive barrier – that at the enclave boundary. Once the enclave is penetrated, attackers can achieve their goals without significant additional technical challenge. However, the recent commercial availability of individual host-based firewalls (i.e., personal firewalls) and host-based IPsec VPNs, suggested that we should examine how hardening the soft inside of the enclave would affect attacker behavior, costs, and risks.

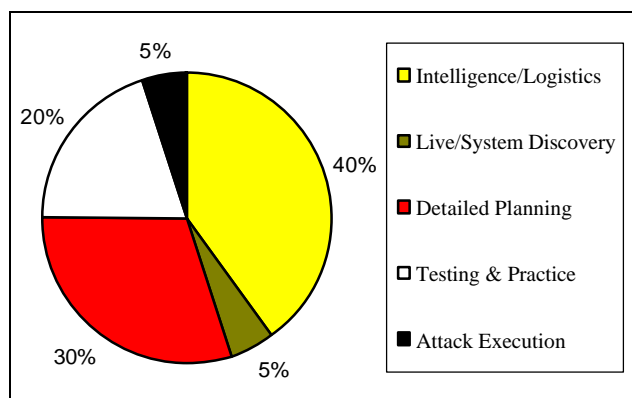


Figure 1. Adversary time expenditures. (%)

Figure 1 shows resource expenditures in labor hours for the model adversary. Previous experiments had the result that the adversary always achieved his goals when

allowed to reach the attack execution phase without hindrance. One experiment series looked at the effect of increasing the adversary’s difficulty in performing intelligence and discovery. That series showed that complicating the adversary’s intelligence and discovery work could significantly increase his risk and ability to carry out an attack.

The IA team had previously noted that the model adversary expends significant energy in detailed planning. This had been observed during two years of prior experimentation with a model adversary exploring other hypotheses. Further, it was noted that execution did not always go like clockwork – there were always surprises for the adversary that had to be overcome. These surprises are analogous to what the military calls *the fog of war*. These observations suggested that observing and analyzing adversary planning and CoA during attack execution could reveal interesting new approaches for the defender. Therefore, one of the purposes of this experiment was to observe how an adversary prepares and what actually goes on during execution.

2.2. Hypothesis Generation

The first group of hypotheses for this experiment involves mechanism layering within an enclave. While adversary CoA is the primary focus, a suitable environment is necessary to observe the adversary and we took the opportunity to extend our previous work in layered defenses.

Layering within the enclave implies that the security policy within the enclave differs from that of the external boundary and, further, that the internal policy will differentiate between systems within the enclave. The experimenters decided that internal policy should be driven by relative differences in the value of assets inside the enclave. In this experiment, value would be determined by importance to a particular mission or function.

The relative contribution of internal and boundary protection mechanisms would be determined by examining each in combination with the other and with a focus on whether the attack originated from within the enclave or outside the enclave. The layering hypotheses are as follows:

Hypothesis – Distributed protection mechanisms with value-driven policies can be composed to prevent or deter unauthorized access.

Sub-hypothesis – Combining enclave boundary protection mechanisms with intra-enclave protection mechanisms can prevent or deter unauthorized access from outside the enclave.

Sub-hypothesis – Deploying intra-enclave protection mechanisms that enforce value-based policies can prevent or deter unauthorized access from insiders.

Sub-hypothesis – Deploying intra-enclave protection mechanisms that enforce value-based policies without deploying protection at the enclave boundary can prevent or deter unauthorized access from outside the enclave.

Figure 2. Layering hypotheses.

A second group of hypotheses was created to examine adversary planning and CoA. As noted earlier, the model adversary has a process and a set of techniques for planning an attack. The process is called the Information Assurance Design and Red Teaming (IDART) Methodology. One of the planning methods is analogous to tree-based analyses used in the fault tolerance community. The fault trees generally take the form of weighted directed graphs. Each node on the graph represents a result or desired system state. Nodes are traversed through detailed understanding of system vulnerabilities and attacker exploits (i.e. attacker acts) that exist or that can be created to achieve the desired state. A sequence of exploits that leads to a desired end state (or goal) is called an attack path or simply just an attack.

Hypothesis - Adversary CoA can be determined by generation and evaluation of a weighted directed graph.

Sub-hypothesis – The choice of next immediate node is a function of perceived risk, cost to develop the exploit, and likelihood of failure.

Sub-hypothesis – The choice of one overall path versus another is a function of the number of nodes, perceived risk, cost to develop the exploit, and likelihood of success.

Figure 3. CoA and planning hypotheses.

The hypotheses in Figure 3 raise several additional questions. The first is whether the adversary really follows the graphs defined during planning. If so, then perhaps the defender can duplicate the process and learn to identify vulnerabilities before they are exploited. Further, an attack graph represents many possible paths to a goal. The model adversary had never developed a formal approach to choosing a particular *next-node* or overall attack path. We surmised that the following four variables are significant determinants in the adversary's choice function:

?? **Perceived risk.** All sophisticated adversaries are assumed to be risk averse during all phases leading up to an attack. This is because detection prior to attack execution may cause the defender to respond. The degree of aversion depends on the class of adversary. Some adversaries are assumed to be risk averse during and after attack execution, e.g., foreign intelligence agents, who are assumed to want to exploit the results of penetration.

?? **Cost to develop an exploit.** Many exploits can be found on the Internet but search and evaluation has a cost. Further, reaching many nodes require exploits that have not been developed previously or are generally believed to be hard. For example, gaining access to a VPN connection can be as easy as stealing keys from an unprotected host or as hard as factoring a public key.

?? **Likelihood of failure.** Reaching each node generally has a perceived likelihood of failure. This is extremely hard to quantify but is assumed to affect the human choice of which node to attack next. It was recognized that this variable may be a composite or of little deterministic value but, if determining, would point to an area to be investigated further.

?? **Total number of nodes traversed.** It was assumed that, in general, a higher number of nodes to be traversed represent greater risk and cost to the adversary. This is not always true as complexity and risk can vary greatly between two nodes.

2.3. Challenge Problem and Environment

To ensure that its experiments are relevant to the DoD mission, the IA program sets them in a real-world context. This is true not only for the overall hypotheses but also for the experiment platforms and environment.

The challenge problem for this experiment required multiple enclaves operating in a wide-area network environment. Critical services were assumed to be provided within various enclaves whose policies would normally be more restrictive than those enforced at the boundary. A scenario was chosen where the U.S. is supporting deployed coalition partners by providing logistics order management in the field.

Four phases were developed to facilitate the exploration of the experimental hypothesis.

Phase 1: Enclave-to-enclave VPNs, running IPsec in tunnel mode, were used to force all enclave-to-enclave

communications to be encrypted. Application protocols were proxied at the firewalls.

Phase 2: Besides the enclave-to-enclave VPNs and proxying, personal firewalls were installed on all Windows 2000 hosts. These hosts provided the clients and servers that were critical to the scenario mission. These firewalls were configured to allow “least privilege per host,” thus adding a prevent layer within the enclaves.

Phase 3: Personal firewalls were not used during this phase. However, besides the enclave-to-enclave VPNs and proxying, host-to-host VPNs were installed, using the IPsec feature of Windows 2000. Host-to-host IPsec connections were established between machines where normal traffic relevant to the mission takes place. Traffic from these hosts to other, non-mission relevant machines was not IPsec protected.

Phase 4: This combined all the protections of the previous phases and used enclave-to-enclave VPNs, application proxying at the firewalls, personal firewalls, and host-to-host VPNs.

2.4. Execution Rules

Three desired *end states* (or *goals* or *flags*) were identified for the model adversary, and these goals remained the same for all four phases. The overall goal was to violate the trust in logistics order traffic by being able to *modify, delete, or add* messages. The Red Team was challenged to achieve these goals (capture the flags) in each of the four phases.

Prior to attack execution, the Red Team generated attack trees with weightings at each node for perceived risk, cost to develop, and likelihood of failure. The Red Team along with commentary and review from the Blue Team and White Team chose and ordered the paths to be executed preferentially. During attack execution, the Red Team attacked according to an ordered list of preferred attacks. Alternate paths could be chosen during execution once preferred paths had been attempted. For planning purposes, attack execution time was assumed to be limited to two hours including time for skipped steps. However, actual experiment execution relaxed this requirement significantly.

The defender was assumed to establish a static set of defenses for each phase. The defender was assumed not to respond to attacks, but the attacker was to maintain a stealthy behavior that acknowledged the defender’s ability to respond if the attacker was detected. In the IA experiment methodology, this is a *semi-automated* experiment.

The target networks were filled with simulated traffic that mimicked real-world traffic. The simulated traffic included most of the popular network protocols including

web and email. Also, the defender provided a continual flow of target network traffic for the attacker to process, analyze, and attack.

To minimize experiment cost and to focus on the experiment hypotheses, the attacker was given all of the network and host configuration data prior to planning and execution. This simulates a state where the attacker has completed the intelligence gathering and system discovery phase, and permits the experiment to focus on attack planning and attack execution.

2.5. Data Gathering and Analysis

Several methods were used to gather experimental data. The most basic were network sniffer logs, host logs, and firewall logs. Added to these were DARPA-developed technologies, including EMERALD and NetRadar intrusion detection sensors. The intruder detection and isolation protocol (IDIP) was used to facilitate reporting of events between systems, including the personal firewalls.

Data gathering for planning and CoA was done procedurally. The model adversary developed an attack tree for each of the four phases. The trees were created in Visio and produced and analyzed as large-format plots. Each node in the tree was assigned ordinal values *low, medium, and high* for the variables *risk, cost, and likelihood of failure*. Further, the model adversary identified three preferred attack paths to reach each end state. Consequently, each attack phase had nine preferred attack paths, resulting in a total of 36 attack paths to be analyzed.

Adversary behavior during attack execution was captured through observation and updating of the printed attack trees. Where CoA dictated creation of additional nodes, those nodes were penciled-in for later capture in Visio.

3. Experiment Preparation

Experiment preparation consisted of defining a network topology with the desired layering of IA technologies and defining the four attack trees with weightings and preferred paths.

3.1. Layering Preparation

A network topology was created to match the scenario and technical requirements. Three principal enclaves were created where the target applications ran. (An additional enclave was created to capture instrumentation and observe the performance of prototype technologies, but this enclave was declared out-of-bonds for attackers.)

The first enclave was notionally named the “104th Peacekeeping Brigade” and nicknamed the “Preserves”; it played the role of a forward logistics support element for the scenario. The second enclave was notionally named the “DLA Processing Center” and received orders for supplies (general-purpose electrical storage batteries) from the “Preserves”. DLA then sent orders to the “Battery Corporation” over the Web.

Each enclave was similarly equipped with an IPsec-capable proxy firewall (Network Associate’s Gauntlet), Windows 2000 hosts running IPsec, and personal firewalls. Figure 4 shows the network topology and the various IA layers.

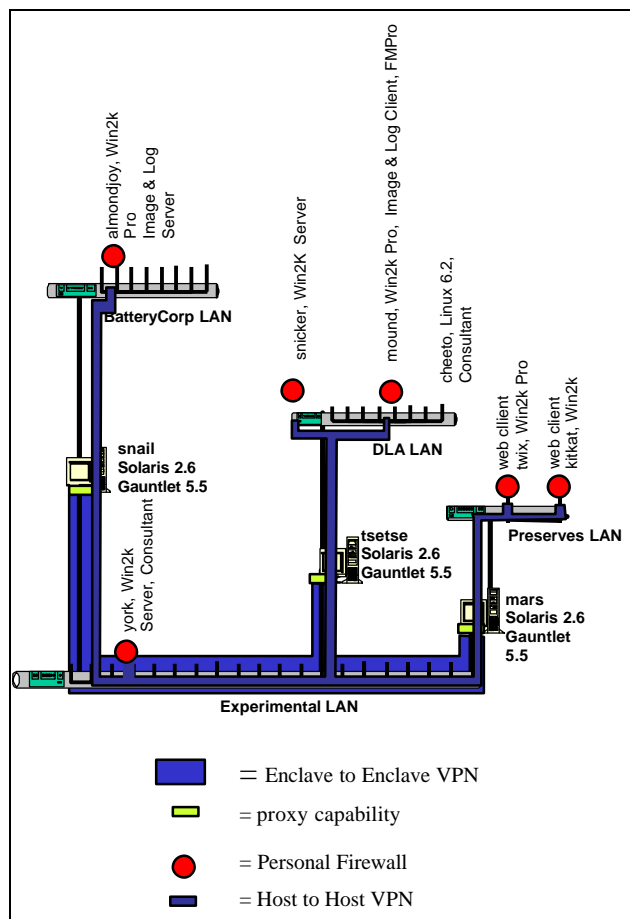


Figure 4. Layering detail, topology view.

The “DLA Processing Center” included a “Consultant” function that was designed to add realism and to enrich the opportunities for the model adversary. The job of the “Consultant” was to help manage and tune the order processing system. The “Consultant” could access the order processing system but only from a special machine on the DLA LAN. Thus, access by the consultant to the order processing system required a two-stage login, first from outside the enclave to the special machine within the enclave, and then from the special machine inside the

enclave to the order processing system. “Consultant” traffic crossing the enclave boundary was TELNET traffic carried over SSH or Win2K IPSEC.

3.2. Adversary Planning

The adversary starts planning attacks once sufficient information is known about the target system. In this experiment, the adversary was given complete knowledge of system design and configuration and the opportunity to observe traffic flows prior to experiment execution.

Building an attack tree is a group effort by the adversary red team. Planning generally starts from two initial fixed points, the *starting access point* and the *end-state or flags*. For this experiment, the starting access point was set outside the enclaves on the experimental LAN. The end-state did not have a physical location since the effect of adding, modifying, or deleting orders could be accomplished at several locations. An often-observed approach is for the adversary to start at the ends and work toward the middle, i.e., to develop the paths simultaneously from access point and end-states.

Some red team members work better with graphic representations of a tree and others work better with a textual flow tree. Either way, the goal is to produce both a graphic tree diagram *and* a textual description of the nodes, exploits, and attack path. Figure 5 below shows the actual text outline for the second most preferred attack (Attack B) for Phase 1 as planned.

Attack B – Database Change	
This attack was executed in RT0001 as an option to Attack A. See Attack A Detailed Steps- Phase 1.	
This attack was designed to:	
<ul style="list-style-type: none"> ⚡ exploit the york to cheeto ssh connection to get onto cheeto ⚡ sniff traffic for the telnet and dba passwords ⚡ telnet to snicker ⚡ modify the database (add, modify, or delete orders) 	
Description:	Exploit the york to cheeto ssh connection to get onto cheeto and sniff traffic for the telnet and dba passwords. Telnet to snicker and add/modify/delete orders.
Preconditions:	Scan york and tsetse for vulnerabilities Development of ssh exploits Development of Win2K exploits (option) Development of hostile e-mail exploit to give administrator access to york (option)
Detailed Steps:	<ol style="list-style-type: none"> 1. Buffer overflow attack on york to cheeto ssh connection- to get remote shell on cheeto as sshd user 2. Linux exploit to get root on cheeto (necessary for sniffing) (PAMslam , etc.) 3. Find sniffer to use on cheeto or ftp sniffer to cheeto 2. Sniff traffic for the telnet and dba passwords 3. Telnet to snicker 4. Access sqlplus, using sniffed passwords sqlplus USER/PASSWORD 5. For modify: See the Database Modify SQL Script (alters the description of an unprocessed order) For add: See the Database Add SQL Script (adds new orders)
Verification:	To verify success of data change, track the confirmation number of altered order(s). See the Database Modify SQL Script or Database Add SQL Script.

Figure 5. Attack detail.

The attack tree for each phase of this experiment had over one hundred nodes. Each node was assigned values for risk, cost, and likelihood of failure. Attack tree nodes are ordered by dependency on previous nodes. Once ordered, the read team identifies selected paths through the tree.

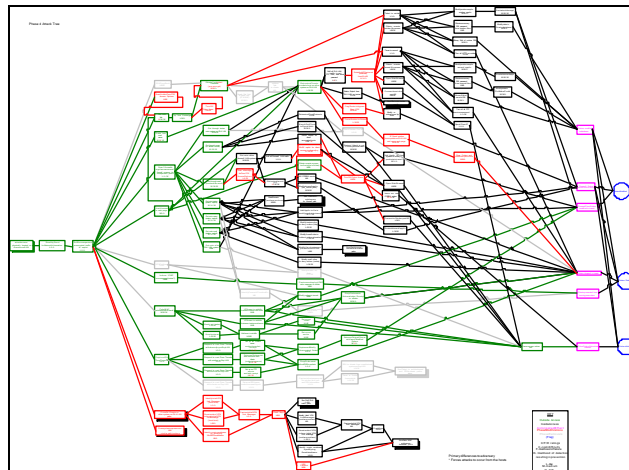


Figure 6. Phase 4 attack tree.

Figure 6, which is the actual tree developed for phase 4 of the experiment, is presented here to highlight some features of an attack tree. (Unfortunately, a full attack tree is best read when printed on 48-inch plotter paper.) The single node on the left is the *access point* to the network. The three nodes on the right represent the *end-states or flags* of add, modify, and delete. The various colored connecting lines (and node colors) identify attack paths – in this case, three for each end-state. For this experiment, the three most-preferred paths traversed the top of the graphic. The reason for that is explained in the results section of this paper.

4. Experiment Results

The experiment was executed during five days, from June 12 to June 16, 2000. Eight attack sessions were executed with each session lasting at least two hours. The adversary's success in reaching specific end-states was mixed. In four cases, the adversary achieved the goal either directly or through concession by the defender. In two cases, the adversary gave up, with general agreement that additional exploit development was necessary. In two other runs, the adversary was unable to reach a desired end-state.

This kind of success-or-failure scoring is not a valid evaluation of the hypotheses. If scoring were important, the experiment would have been constructed differently. In fact, the experiment was designed so that the adversary had a high likelihood of traversing major portions of the attack tree while encountering enough difficulty to force real-time adaptation or development of alternative courses of action. In this sense, the experiment construction worked as expected and gave us insight into the effectiveness of this kind of defense layering in changing adversary behavior.

4.1. Layering Results

The NAI Gauntlets provided enclave protection in the form of an enclave-to-enclave IPsec VPN with stateful inspection. The VPN was present for all four phases of the experiment. The system architecture did not require authentication from client hosts; consequently, any traffic that came from what appeared to be a valid user on a valid host was allowed to pass into the enclave.

Further, each enclave was assumed to have a mix of users and missions, some requiring VPN protection and others that could not use the VPN. Such a “mixed-mode” environment certainly is not ideal for security but reflects many real-world situations.

Internal to External Proxy	
*idip ²	56508
*cvs	2401
http	80
pop3	110
ssh	22
https	443
smtp	25
icmp ²	-
IPSec ^{1,2}	Snicker:500
ftp	20/21
RealAudio	7070
External to Internal Proxy	
*idip ²	56508
Pop3	110
ssh	22
smtp	25
icmp ²	-
IPSec ^{1,2}	York:500
IPSec Enclave to IPSec Enclave	
Everything	Everything
¹ IP Proto 50/51 for IPSec	
² filter rule	
*red text - off limits to adversary	

Figure 7. External firewall configuration rules.

Figure 7 depicts the external or enclave firewall configuration for the DLA LAN. Piercing this enclave boundary was a problem for the adversary. Analysis indicated that attack paths that involved the communications of the “Consultant” would be most fruitful. Consequently, many of the exploits were aimed at SSH or at gaining access to the “Consultant” remote host (York). Success at penetrating the enclave boundary was dependent solely on boundary configuration.

Personal firewalls were used for two of the four phases. These turned out to be much more effective than expected.

Traffic Type	Host: Direction: Host: Protocol
Consultant	York:* ? Cheeto: SSH

	Cheeto:* ? Mound: Telnet Cheeto:* ? Snicker: Telnet
Order Traffic	Twix :* ? Snicker: HTTP Kitkat:* ? Snicker:HTTP
DNS	Snicker:* ? Tsetse-internal: DNS Mound:* ? Tsetse-internal: DNS Cheeto:* ? Tsetse-internal: DNS
FTP	Snicker:* ? Mound: FTP
EDI	Mound:* ? Almondjoy: IIO/CORBA

Figure 8. DLA LAN personal firewall configuration, phase 1.

Figure 8 shows a representative configuration for personal firewalls during phase 1 for the DLA LAN. This configuration is much more restrictive than the enclave boundary configuration and specifies not only protocols, but also direction of traffic origination (where applicable) and specific hosts to communicate with. The additional restrictions are in keeping with the experiment’s assumption that tighter policies will be desired within an enclave for selected systems.

The adversary found that personal firewalls made the host look invisible to scans. This led them to try some “noisier” mapping techniques that they normally would have avoided. Further, each violation of personal firewall policy (e.g., unauthorized host) was logged and reported. This had the effect of turning each host into a kind of intrusion detection sensor capable of reporting to a central analysis and reporting function.

The adversary defeated the personal firewall by turning it off with a malicious email attachment. In this case, the firewall exited silently. The adversary was then able to access the host using a number of well-known host exploits.

Windows 2000 IPsec VPN was used for host-to-host connections within the community of interest (COI) for phases three and four of the experiment. The community of interest consisted of the hosts supporting the logistics mission and the consultant. Every host in the COI used a separate and distinct IPsec policy specifying permissible pairwise communication partners. The “Consultant” permitted additional protocols outside of IPsec, including SMTP, POP3, HTTP (80 and 8080), HTTPS, and FTP. These protocols were covered by the personal firewall policy that defined direction and communicating partners. In the phases without the personal firewall, or when the personal firewall was disabled, those protocols were exposed to attack.

This configuration and the network architecture allowed the adversary to run well-known exploits on the outside “Consultant” host (York) and use that access to gain direct access to the host-to-host VPN. This gave access to the DLA LAN and the order processing system. Further, this approach simplified the adversary’s problem by making it unnecessary to perform further scanning or

sniffing inside the DLA LAN, because all the information necessary to identify the target of interest was present on the inside “Consultant” machine (Cheeto).

In this experiment, the host-to-host VPN did not make much difference by itself, because the “Consultant” provided a virtual back door into the enclave. The VPN was much more effective when coupled with a network-level access controller like the personal firewall. **Adversary Planning Results**

The model adversary used in this experiment starts with an initial plan (attack tree) and tries to stick with it. The plan requires substantial effort in preparation and practicing. Therefore, in general, it takes a significant obstacle to force this adversary to modify the plan.

However, in this experiment, each attack tree had multiple attack paths, and often there was little apparent difference between the most preferred attack path and the next most preferred attack path. The adversary was much more willing to try already developed alternative paths than to develop new ones in real time.

During the planning phase, we identified a new variable that appears to help determine adversary choice of exploit and attack path. The variable related to the degree of direct control over the exploit. In this experiment, the control desired was over the timing of the exploit execution. One attack path led through a node (exploit) that would disable the personal firewall through delivery and execution of an email attachment. There were alternatives with identical values for *risk*, *cost*, and *likelihood of failure*, and the adversary preferred these, e.g., sniffing for a password. It was only after these alternatives proved unfruitful during attack execution that the path with the email exploit was (successfully) followed.

4.3. Initial Analysis of Adversary Attack Path Selection

Our ability to analyze attack path selection is still at a preliminary stage, and only suggestive observations can be made. Still, it appears that fairly simple decision analysis tools and rules can account for gross characterizations in attack path selection.

Two tools are being used for analysis. A commercial product called Decision Pro has been helpful but contains a wealth of capabilities that haven’t been used. Most of the exploration of the data has been done using a spreadsheet and simple JAVA tools.

The phase 1 attack tree has over 100 nodes and yields about 2900 possible attack paths. A fairly simple model is able to exclude the majority of paths and leave a small number among which are the top four preferred paths chosen by the adversary. This is significant, because the

model adversary indicated that paths the red team identified beyond the top three were generated only to meet experimental requirements and would not be developed under other circumstances.

In analyzing the attack tree, cost, risk, and likelihood of failure are treated as orthogonal axes of a Cartesian space. The default measurement for a node in the attack tree is the square of the distance from the origin to that node. The measurement for a path is the sum of the measurements for the nodes on the path. Current analysis results are, at best, approximate, because we do not have a way to normalize paths that have different numbers of nodes or where the nodes differ in complexity.

A function $m(c,r,f)$ defines the metric coefficients for cost, risk, and likelihood of failure. The variables v_c , v_r , and v_f , represent the values assigned by the adversary at each node. The measure for a node is defined as $measure(node) = c*v_c^2 + r*v_r^2 + f*v_f^2$. The measure for the path is defined as $measure(path) = Sum(measure(node))$ for all nodes on a path.

Adjusting the coefficients has little effect in identifying the adversary’s preferred paths with the exception of *likelihood of failure*. Adjusting the weighting for *likelihood of failure* filters the vast majority of paths leaving a small number of attack paths including the top four preferred paths identified by the adversary. This suggests that the variable *likelihood of failure* is a significant factor in determining attack path selection

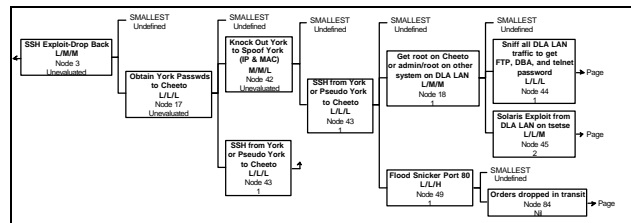


Figure 9. Decision tree analysis, branch view.

A significant amount of analysis and questions remain. Among the remaining questions are:

- ?? Is there an impact in path selection due to the total number of nodes?
- ?? What is the value of the total number of entry or exit points on a node? Does it influence path selection?
- ?? Are there particular intermediate nodes (intermediate states) of such significant value to the adversary that previous nodes with greater risk, cost or likelihood of failure are still worth attempting?

- ?? Are there additional variables that need to be identified? I.e., utility factors, time, risk of detection.
- ?? What accounts for the inclusion of non-selected paths with selected paths?
- ?? How do we normalize the weightings across the various nodes?

4.4. Observations on Adversary Course of Action Selection

The model adversary developed several alternative attack nodes and branches when exploits failed to work or when the adversary made assumptions about system behavior or configuration. These alternatives were always a modification to the attack path already followed. We had expected to see the adversary first try alternate paths or to jump from one path to another when two paths intersected at a node.

Two simple algorithms emerged while observing the adversary achieve a node or develop alternate nodes and branches.

- ?? Upon reaching a node,
- If the risk of detection is high, the node is easy to regain, or the end-state is within reach, then continue to the next node;
 - Else, modify the system to regain access easily, e.g., install a backdoor.

Figure 10. Adversary behavior at a node.

Figure 10 captures a behavior that was not anticipated from generation of the attack trees, i.e., during attack planning. The behavior is to consolidate the effort expended and risk endured in achieving the node. The caveats are listed next and have no quantifiable values. These are fundamentally ‘expert judgments’ made by the adversary in real-time.

- ?? If the node is unachievable,
- Use other obvious or easy exploits.
 - Or, withdraw and prepare further.
 - Or, use alternative paths.

Figure 11. Adversary top-level CoA process.

Figure 11 captures the high-level adversary behavior and requires some additional explanation. As noted above, this adversary would like to persist on the preferred attack path. The preferred path is perceived to represent a combination of the best chance to achieve the end-state

and, as more of the path is traversed, an investment in cost and risk. When the exploits to achieve the next node fail, the adversary is inclined to explore options. The definition of what is an obvious or easy exploit is again a qualitative judgment based on the expertise of the adversary. Note that the adversary stated he would prefer to withdraw rather than attempt alternative paths.

Understanding the adversary’s choice of options and exploits is impossible to quantify with the data from this experiment. The adversary knows from experience that attacks will not go perfectly and so is prepared with three additional high-level approaches. First is to try exploits that are believed should not work, because sometimes they do. For example, in this experiment, the adversary successfully exploited a misconfigured SSH application that was not expected to be vulnerable. Second is to look for “gifts”. Frequently, a system will be misconfigured, lack security patches, or have tools like compilers or script engines. These “gifts” often can be used by the adversary and may impact the selection of the next node on a path. Finally, the adversary will sometimes consider branching from a failed node to a node that has low cost and low risk, but higher probability of failure. This option is only considered when the additional ‘leverage’ provided by attaining the node is high and gaining the preferred node has failed.

The adversary also considers the issues of timeliness and stealthiness. Timeliness relates to considerations of overall time to complete an attack and total time spent at the previous node. Overall time to complete an attack seems to have been more important in this experiment since each attack run was planned to be limited to 2 hours. This time limitation may have analogues in the real world, but that should not be assumed. Time spent at the previous node relates to the adversary’s determination of overall risk. Generally, this adversary would like to spend as little time as possible actively working in the system because certain classes of node and exploit are estimated to carry greater inherent risk of discovery and failure.

Finally, when confronted with an unexpectedly difficult or unachievable node, the adversary would usually prefer to return to the lab and prepare further. Additional preparation was not possible for this experiment, and such may be the case for a number of real-world attack scenarios with this class of adversary. However, it is assumed that a different class of adversary, e.g., a foreign intelligence agency, would rank stealth and long-term exploitation as very desirable and would be much more likely to quietly withdraw to prepare further.

4.5. Lessons Learned

This model adversary follows a defined process, and one interpretation of the planning phase is that the adversary executes the plan the red team developed.

Future experiments should use different adversaries that have different planning processes. A comparison could be done of the two planning processes to see whether similar vulnerable nodes and attack paths are identified.

The attack trees developed as part of this model adversary's process is dependent on the skills of the adversary team. During the experiment hot wash (i.e., immediate after-action review), it was noted that a different skill mix on the team would probably have resulted in additional nodes being identified and alternative branches or paths being selected for attack. A future experiment that only built attack trees, but uses multiple teams with different skills, might reveal whether the overall differences are significant. This team identified low-level socket and stack programming as a skill that would have been helpful.

This experiment showed that mixed-mode IPsec—that is, the use in a system of both IPsec communications and unprotected communications—is very complex to configure and protect. The adversary will use well-known exploits of unprotected communications to gain access to the host. The presence of configuration data that identifies the members of an IPsec community on an exploited host and the ability of the adversary to hide in the IPsec channel certainly represents a significant vulnerability in that it helps to hide the adversary as well as identify important targets.

Personal firewalls made a significant contribution when tightly configured and monitored. They helped to protect the hosts from scanning and well-known exploits to the extent that the adversary made the personal firewall a significant initial target. Unfortunately, these firewalls are vulnerable to being turned off or having their configurations changed and the user receives no alerts. Instrumenting these personal firewalls to report to a central monitoring system would have tipped-off the defender that the system was being probed and attacked.

The goals for this experiment were probably too large and broad. This made control and measurement difficult and made the final analysis more difficult than anticipated. It would have made sense to limit the experiment to a single phase with a single focus on either layering or adversary planning and CoA.

The experiment scenario was fairly rich, but the architecture for “consultant” support on the DLA LAN provided a backdoor that caused the adversary to focus on particular branches of the tree. This narrowed the area of the tree for analysis and probably made it difficult to distinguish the significance of the variables and whether other unidentified variables were contributing. A future experiment designed to differentiate between paths should carefully control for “easy” branches.

5. Acknowledgements

The author gratefully acknowledges the work of the entire RT0001 team. Many people contributed to this experiment behind the scenes that aren't named below. People listed below made significant contributions in planning and execution:

?? *Red Team:* Julie Bouchard, Ray Parks, Dave Duggan, Brad Wood.

?? *Blue Team:* Dorene Kewley, Sara Kaufman, Tom Hash, Dale Johnson, David Levin, Gregg Schudel.

?? *White Team:* Dave Smith, Don Faatz, Ken Theriault.

This work is supported by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-98-C-0012.