Layering Boundary Protections: An Experiment in Information Assurance

Lee A. Benzinger
NAI Labs
Lee_Benzinger@nai.com

Dale M. Johnson
The MITRE Corporation
djohnson@mitre.org[1]

Abstract

The DARPA Information Assurance Program has the aim of developing and executing experiments that test specific hypotheses about defense in depth and dynamic defense capabilities. This paper describes the development and execution of an experiment in layering. The basic hypothesis was that layers of defense, when added in a careful and systematic way to a base system, lead to increased protection against attacks on the system. For the particular experiment, a mission and broad policy were defined and a base system was developed to support the mission and the policy. The boundary controller for the system was designed and developed as a series of layers; these elements became the main focus of experimentation on layering. The results tended to confirm the experimental hypothesis that layers have a cumulative effect on protection against outside attacks. However, there are often other opportunities for attackers to go around the layers or avoid them altogether. A broader methodological result was that the entire process of developing experiments needs to be carefully thought through. In addition, the experimental data resulting from this experiment provide only a limited corroboration for the given experimental hypothesis.

Introduction

This paper concerns the development and execution of an experiment dealing with layered defense. The experiment was motivated by the needs of the DARPA Information Assurance (IA) Program to investigate "grand hypotheses" about defense in depth and dynamic defense capabilities at the level of specific experiments both in thought (*Gedankenexperimente*) and in the laboratory. A prime grand hypothesis of the IA Program is that

> Layers of defense for a system compose to make the system more resistant to the attacks of an adversary.

This broad, admittedly vague hypothesis needs further articulation to make it testable. Defense layers can be put together in many ways in a system architecture and applied in various ways for protection.

One can think of layers of defense in the dimension of depth or of breadth. Yet, these dimensions are not necessarily independent. For the purposes of the experiment being discussed, layers of the architecture are being thought of in the dimension of depth. With this viewpoint successive layers in the architecture lead to a greater depth in the system defensive mechanisms. However, as the layers of architecture are traversed in depth by, say, a network packet, the protection mechanisms of the boundary controller check a variety of properties of the packet. This leads to greater breadth in the security attributes that are checked. So layers include breadth as well as depth.

Basic Plan of the Layering Experiment

A basic system was conceived as consisting of an image server with a set of images to be distributed to a set of clients. In other words, the mission was to distribute images unaltered from the server to the clients as requested by the clients. The image server was regarded as the central part of the inside of the

[1] Work was completed while the author was at NAI Labs.

system. The clients were taken as part of the outside of the system beyond the inside image server. The basic security aim was to distribute the images in a secure manner determined by a security policy.

The security policy for the system was the following:

Confidentiality: Only an authorized user can obtain an image.

Integrity: No one outside the system is authorized to change an image on the image server or an image being transmitted to an authorized client.

System Availability Protecting Against Denial-of-service: An authorized client receives the image it requests.

System Availability Protecting Against Denial-of-service: An authorized client has available to it an image request service.

This security policy was directly related to the set of flags for the red team. Each flag represented a failure of the system to enforce the security policy.

The focus of the experiment was a simple basic system with well understood security properties that could be incrementally changed. The system consisted of an inside, outside, and a boundary controller on the system boundary. The inside of the system contained an image server that was accessed from the outside by clients that were authorized to receive images that they requested.

Given the basic system, the objective was to develop the boundary controller as a sequence of layers. The layering for the experiment was achieved by incrementally changing the boundary controller by successively adding a series of layers of security mechanisms. The aim was to layer defenses as a part of the system policy enforcement on the boundary.

Initially some time was spent analyzing the experiment in the abstract. A methodology for a simple analysis of the system was followed. This analysis was based on the basic system given in Figure 1. The fundamental idea, a simple key to the experiment, was that as the layers at the boundary were increased, then the adversary would have to penetrate them to get to the image server. In this simple sense, the layers did compose or accumulate protections against the adversary's attacks.

Development of the Experiment

To make a real experiment it was necessary to proceed from the abstract architecture to a concrete design. See Figure 2. Given the existing laboratory resources as well as the analysis of the architecture, the following concrete design and implementation were developed.

- The image server was implemented as an IIS server running on a Windows NT 4.0 Server platform with authentication and authorization capabilities activated.
- The clients, two in number for the experiment, were Windows NT 4.0 machines. It was assumed that the clients were vulnerable to attack. The emphasis was not on protecting them or attacking them, since it was assumed that in an actual system such clients would have to be further strengthened. However, this was not the primary focus of the experiment.
- In addition, there was a Cisco (model 3640) router at the front of the inside domain, which was used initially only for routing and not for boundary protection.
- Layers for the boundary controller were successively constructed as follows:
  - For layer 0, the basic client-server system was taken as given.
  - For layer 1, the filtering capability on the Cisco router was activated and filtered by IP source and destination addresses, TCP ports, and protocol numbers.
  - For layer 2, an NAI Gauntlet Internet Firewall with proxies for HTTP was positioned and turned on.
  - For layer 3, the IPsec Encapsulated Security Payload (ESP) Protocol in tunnel mode [1] was activated between the Gauntlet Firewall and the clients. Terminating ESP at the Gauntlet qualified this protection as a boundary protection.
  - For layer 4, strong authentication with the Axent Defender was activated at the Gauntlet. In the end this layer was not tested in the laboratory.

Execution of the Experiment as a *Gedankenexperiment*

Before any execution in the laboratory, three major discussion sessions were held to analyze the experiment and consider in detail likely attacks on the system in its various layers. In effect, the experiment was designed and conceptually set up, analyzed, and then run on a white board. The first session was devoted to analyzing the concept of the experiment as proposed by the designers, the authors of this paper. The second session was concerned with shaking out the details of the implementation of the layers and developing the flags of the experiment. The third session was devoted to a detailed analysis

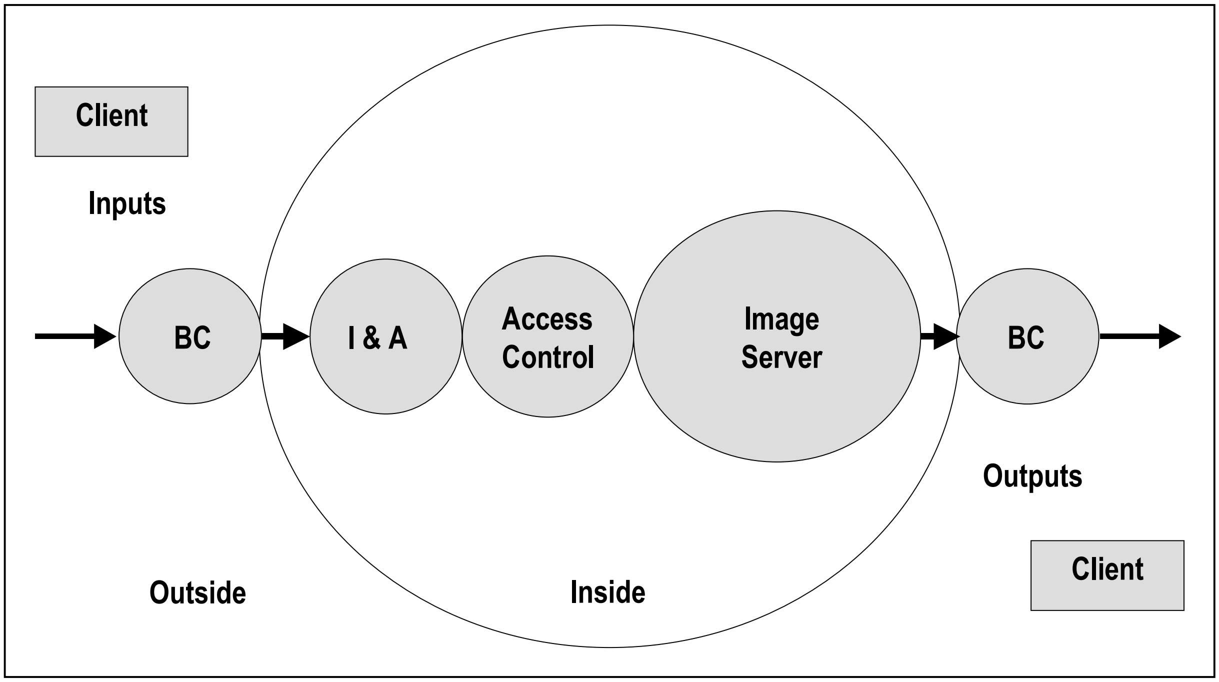


Figure 1: Basic System

of attack trees that the red team proposed as possible ways to attack and defeat the system.

The first session covering the basic conceptual design of the experiment was held in July 1999 with the experiment designers, prospective members of the red team, and members of the Experimentation Working Group supporting the DARPA program. The session lasted for a full day, the output of which was a briefing summarizing results. These were later discussed in a teleconference in August 1999.

The experiment designers (the authors of this paper) in proposing this experiment wanted to have a very clear idea of layering and to make the basic experimental system as simple as possible in order to expose the layers. The methodology and analysis as given above for this was presented at this first session. The key hypothesis that was settled on at this session for subsequent testing was as follows:

> **Assertion:** By composing successive security mechanisms forming independent layers, (independent decision mechanisms) the time to complete an attack shall be successively increased.

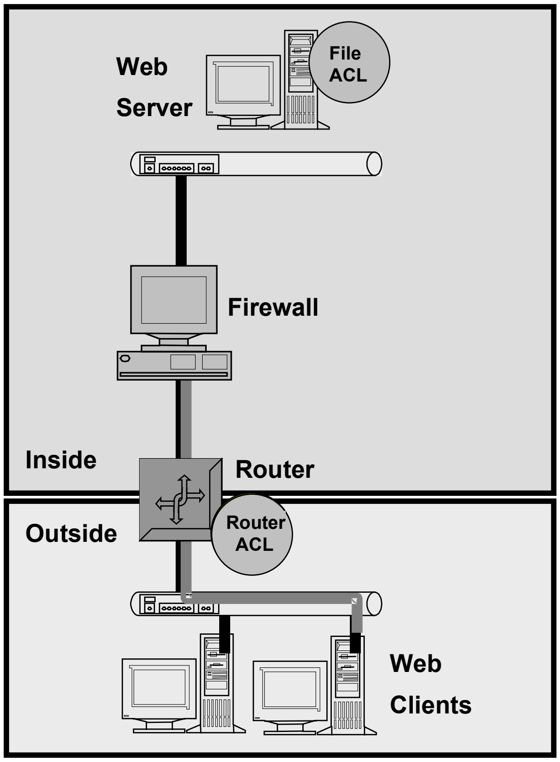
The measure associated with this assertion was as follows:

> **Primary measure:** Total time for the red team to complete an attack.
>
> **Secondary measure:** Total costs for the blue team to develop and implement the design.

After this session the experiment designers, when trying to construct a concrete experiment, realized how difficult it is to make the layers truly independent. Any system has some dependencies either through the related hardware platforms, the similar operating systems, the interoperable protocols, or other parts of the system. The teleconference brought to the fore the view that independence was relative. One could only strive for as much independence as seemed practical, recognizing that when analyzing the experiment, dependence factors might arise in an actual implementation.

During the time between the meetings for the first and second sessions the experiment was designed in much more detail. The types and specifics of the equipment were settled upon. In particular, the elements of the boundary controller—filtering router, proxying firewall, and IPsec ESP—were decided upon.

The next session dealing with the experiment in detail, including designers, red team members, and Experimentation Working Group members, took place in middle January 2000. The concrete design for the boundary controller elements was considered carefully. It was thought useful to add a fourth element, viz., strong authentication at the firewall, to complete the design. However, later it turned out that it was unnecessary to test this element. In all, the following elements were proposed: Cisco filtering router with specific filtering rules, Gauntlet Internet firewall with the HTTP proxy, IPsec ESP on the Gauntlet firewall and clients, and Axent Defender authentication on the Gauntlet firewall. The red team



Layers for Testing

- •Layer 0 (Server access controls and file ACLs)
- •Layer 1 (Router filtering
- •Layer 2 (Firewall proxying)
- •Layer 3 (IPsec)
- •Layer 4 (Strong authentication)

Figure 2: Implementation of Layers

concluded that the combination of filtering router and proxying firewall with IPsec ESP on different platforms was going to be very hard to defeat in order to get through to the image server.

During this second session it was decided that the initial position for the red team was physically and electronically on the network LAN on the outside with the clients. The flags for the red team were set as follows:

Confidentiality flag: Red team copies an image.

Confidentiality flag at the image server: Red team copies an image from the image server.

Integrity flag: Red team changes an image or images.

Integrity flag at the image server: Red team changes an image or images on the image server.

Denial-of-service flag: Red team denies an image to a client.

Denial-of-service flag at the server: Red team denies an image to a client from the image server.

Originally when planning for discussion sessions it was thought that after the second session testing could proceed in the laboratory. However, during the second session participants realized that a further discussion session was needed to go through the attack trees prepared by the red team for planning their attacks. Hence, a third session was held at the beginning of February 2000 to analyze in detail the attack trees that the red team would use for planning their attacks.

The attacks trees were detailed enough to indicate strategy and tactics for attacks. They encompassed attack sets for perpetrating the attacks.

Execution of the Experiment in the Laboratory

The red team came to the DARPA laboratory for a week in the latter half of February 2000 fully prepared to execute the experiment. They brought a toolkit of attack executables, attack scripts, and more general ideas for attacks. The experimental setup was all in place. Since there were no dynamic defenses, the various stages of layer deployment were treated statically. The blue team defenders did not have to react to red team attacks for this experiment.

Laboratory testing takes a significant amount of time to complete. Even with being given a week for laboratory testing the red team could not test all the layers. Hence, for experimental purposes testing was conducted on the following layers illustrated in Figure 2:

- Base system without boundary controller protections (layer 0)

- System with the filtering router (layer 1)
- System with the proxying firewall having IPsec ESP to the clients (layer 3)
- System with both filtering firewall and proxying firewall having IPsec ESP to the clients in place (layers 1+3)—to a limited extent.

To capture the flags the red team tried attacks on both the image server and the clients.

For their attacks the red team used the following typical tools [2]. They had experimented with them prior to arriving in the laboratory.

- IIS-RDS exploit: a special-purpose tool used against Microsoft IIS and its Remote Data Service (RDS) that exploits a buffer-overflow vulnerability within a default-installed CGI script; ultimately this exploit allows for remote execution of commands as a privileged user on a server
- Squid: a proxy caching server for Web clients that can be used for capturing and changing data or images between a server and client
- ICMP: a tool that provides a full range of Internet Control Message Protocol (ICMP) functionality and manipulation, including route redirects of IP packets
- HUNT: a tool designed for exploiting weaknesses in TCP/IP that allows for poisoning of a client's ARP cache
- IIShack: a special-purpose tool that exploits a buffer overflow vulnerability included with IIS and effectively shuts down the Web server process
- Sniffer: a protocol analysis tool that collects network packets on a network segment where it is installed
- L0phtCrack: a brute-force password-cracking tool that provides a dictionary of known password hashes for comparison with captured password hashes
- WinVNC (Windows Virtual Network Control): a tool that, when installed, provides for remote administration and misuse of a client machine.

Testing was performed systematically on the various layers of boundary controller defense. Attempts to capture the flags as listed above were made one by one. Timings of attacks were recorded in spreadsheets. In this case the timings were in minutes. This level of granularity was sufficient for this experiment. In reality the majority of time was spent in preparing for the attacks before even arriving in the laboratory.

It is useful to divide up the red team's time as:

1. Preparation time for understanding the system that is the object of the attacks, general layout, configurations, and so forth
2. Preliminary testing and setting up time for assessing the appropriate tools relative to the supposed system prior to coming to the laboratory and setting up in the laboratory
3. Execution time in the laboratory for accomplishing the attacks
4. Analysis time for understanding the nature and effects of the attacks on the system.

Execution time is actually the smallest part of the exercise. It was realized during the course of execution, perhaps not surprisingly, that if more preparation time had been used, some of the attacks would have gone smoother and quicker. As it was, the execution times were mostly very short.

Results

The following is a selection of results for the experiment. Everything in this form of testing depends ultimately on detailed understanding of the system under test and finding the right tools to capture the flags. Server attacks are considered first, then client attacks.

Server Attacks

The Microsoft Internet Information Server was configured with minimal protections at the base layer 0, i.e., with simple access and authorization controls. As with all software it has vulnerabilities, which can be exploited. In particular, its Remote Data Service was exploitable as configured. The red team was able to use the RDS attack tool to take advantage of a CGI script and thus make the IIS do its bidding. Hence, it was possible to go into the server and change locations of the images, so that the normal client saw different images than it should have seen. This was a straightforward integrity attack. Similarly, the red team was able to go into the server and easily capture images and attain a confidentiality flag.

At layer 1 with the Cisco filtering router adding protection, the red team had to spoof the client to go into the IIS to perform the same RDS attacks to attain the integrity and confidentiality flags. The spoofing added a small amount of time to the red team's efforts, so ostensibly the layered defense worked as planned. In this case the specific times for attack execution were not as significant as the extra work needed for the red team to prepare for the attack.

At layer 3 with IPsec ESP in place between the Gauntlet Internet proxying firewall and the client, the red team had rather more difficulties. Defeating IPsec ESP on the Gauntlet in the given execution time was not easy; indeed the red team did not accomplish this goal. The results suggested that the combination of the Cisco filtering router and IPsec ESP on the Gauntlet proxying firewall was very daunting for a red team or adversary.

For denial-of-service attacks on the IIS server at layer 0 the red team used three different attacks. Causing denials of service is generally an easier matter than other types of attacks. First, the RDS tool was used to cause a denial of images to the clients. Such an attack was similar to those for integrity and confidentiality. Second, the IIShack.exe attack tool was used to shut down the http process on the remote server. Third, the denial of service from the server was accomplished by running ICMP redirects against server. At layer 0 these were all easily accomplished.

At layer 1 for denial of service for the IIS, additional spoofing was needed to carry out the attacks. This extra step meant a small amount of extra effort for the red team.

For layer 3 and denial of service for the IIS the Gauntlet with IPsec ESP again proved an insurmountable hurdle in the time given to defeating it. Such a roadblock naturally led to considering attacks on the clients.

Client Attacks

Attacks against the clients were not the main interest of this layering experiment. In effect, such attacks for the most part avoid the protection layers. Nevertheless, an adversary could well exploit such attacks even if they need to be directed against one client at a time.

A simple integrity attack against a client is to bind the legitimate server's IP address to the red team's rogue server to replace the good server, then use ICMP redirects to have the client point to the rogue server, and finally send incorrect images to the client. This works easily for layers 0 and 1 and takes less than five minutes. In the case of layer 3, when IPsec ESP is enabled, there is extra work to disable IPsec ESP. The red team was able to do that by gaining root access to the client and then turning off IPsec ESP on the client.

Many other attacks are possible against a client. The red team succeeded with several. However, since such attacks were not part of the layering experiment as such, they will not be further summarized.

Conclusions

The following items capture the main conclusions about the experiment.

- The experimental tests tended to support the result that layers add to the work factor for the red team, when red team is forced through the layers.

  This result seems very natural when one considers the traversal of the architecture of layers of the boundary controller from the perspective of any user at the client or on the client LAN initiating network packets that must go from the outside through the boundary controller to the inside server. The red team in organizing an attack through the layers of architecture has to deal with each layer as it proceeds to the inner image server. Hence, at least some minimal effort is needed to pierce each layer. This result is supported by a direct analysis. There may, however, be cases when a layer can be passed through without much effort.

  This is not a strictly rigorous result from an empirical standpoint. The experiment execution did not yield a wide range of data. However, the measured data collected tended to corroborate it.
- When the client on the outside was attacked directly, minimizing the effect of or bypassing the layers of the boundary controller devices, then the layers either did not add or added only modestly to the work factor of the red team.

  In this case the protection mechanisms provided by the layered architecture for the boundary controller is bypassed, so one would not expect the layers to add to the red team work factor. This too appears to be a very natural result by analysis.
- Adding IPsec ESP, which includes strong encryption, proved to be the most significant hurdle in terms of red team work factor.

  The encryption itself is very hard to penetrate, in the sense of direct cryptanalysis of the ciphertext. Exploitation of weaknesses in the management of the encryption is likely to be an easier way to get through the protection provided by encryption. Using this approach the red team was able to break into the client and shut off the IPsec encryption.
- Since the natural inclination for an attacker is to attack the weakest point, as the boundary controller is strengthened, the client becomes the most attractive target for red team attack.

In this experiment there are both attacks directly against the client and attacks through the client's legitimate connection to the server.

The most likely strategy for any adversary trying to defeat a system is to attack the weakest points, assuming this form of attack is consistent with the adversary's goals. This may not be the case when the adversary is very risk adverse and an attack on a weak target is highly visible. In the system designed for this experiment, the client was not especially hardened, though it might have been. For the red team it was worth looking at the vulnerabilities of the client, though the results from exploiting these were not the main results. One could draw the simple conclusion that it is really necessary to look at the overall system security architecture to develop more uniform ways of protecting the entire system. For example, each client might have been put in a kind of enclave like the server with a boundary controller protecting it. A personal firewall could have been used to guard each client. The experiment designers and red team discussed these kinds of possibilities, and the red team conjectured that the attacks would have been much harder for them. The clients were "giveaways" in most respects. However, they were not part of the layers of the boundary controller.

- An indirect result of this experiment that is noteworthy is that the combination of the Cisco filtering router and IPsec ESP on the Gauntlet proxying firewall was very daunting for a red team or adversary. For future experimentation it would be interesting to gain more insight into the robustness in the face of attacks of this combination of two commercial products.

The experiment under discussion is quite simple. It was intentionally chosen to be a simple testing of layering, because experimentation in the field of information assurance is a very recent endeavor. Experience is only now being gained on how to conduct such experiments. In this experiment we were dealing with human/system interaction. As such, there is ongoing debate as to what to measure and how to make measurements, especially since such experiments are dependent upon human behavior. The dependency is all the greater when dealing with the behavior of a red team. The experience and expertise of various red teams or red team members will vary, so measured results will in part be measures of human reactions to given system events. In addition, for a given red team, experimental results for the same attack on the same system will change as the red team gains experience.

In this experiment, the measurements of red team attack times as the layers were pierced no doubt would vary over different red teams with different skills. However, the notion that each layer had to be overcome to get to a flag at the inside server suggested that the layers add in protection.

Currently there is no well-defined methodology for information assurance experimentation. It is likely that a methodology could take quite some time to develop. The field of computer security development has for the most part been a form of art with methods that tend to regularize the ways systems are conceived, designed, build, tested, penetrated, and maintained. Putting scientific methods of experimental testing into the mix is not necessarily easy. However, it is worth a try, especially if the potential is there for the results to lead to a better way of building secure systems.

References

[1] RFCs 2401, 2402, 2406, 2408, 2409, 2412, http://www.ieft.org/.

[2] http://www.bo2k.com/, http://www.cri.cz/kra/index.html, ftp://ftp.st.ryukoku.ac.jp/pub/security/tool/icmp/, http://www.datafellows.com/v-descs/iishack.htm, http://www.l0pht.com/advisories.html, http://www.l0pht.com/~weld/netcat/, http://www.securiteam.com/windowsntfocus/IIS_RDS_vulnerability.html, http://squid.nlanr.net/, http://www.uk.research.att.com/vnc/winvnc.html.

Acknowledgements

The authors wish gratefully to acknowledge the help in planning, critically assessing, and executing this experiment of Eli Dart (Sandia National Laboratories), Ted Farrell (BBN), Steve Hurd (Sandia National Laboratories), Dorene Kewley (BBN), John Lowry (BBN), Eric Monteith (NAI Labs), Dan Schnackenberg (Boeing), Gregg Schudel (BBN), and Ken Theriault (BBN).

This work was supported under the DARPA Information Assurance Program. The authors are thankful for the support under the Integration Contract for the IA Program.