

# Case Study: The Intrusion Tolerant JBI

Michael Atighetchi, Paul Rubel, Partha Pal, Jennifer Chong, Lyle Sudin  
*BBN Technologies*  
{matighet, prubel, ppal, jchong, lsudin}@bbn.com

## Abstract

*The same network infrastructure, that is essential for the operation of today's high valued distributed information systems, can also be misused by malicious attackers. Experience shows that implementing absolute security or completely preventing cyber attacks is infeasible when systems must be highly interconnected and are made of COTS components with unknown security characteristics. Therefore, focus is shifting towards making high value distributed systems survivable, so that they can continue to operate through attacks. This paper describes our experiences in making a DoD application survivable using the DPASA<sup>1</sup> survivability architecture focusing on the network aspects. In particular, we show how a survivable system can be built using sound design principals and a combination of COTS and research grade technologies.*

## 1. Introduction

The DPASA survivability architecture was recently tested in the context of a Joint Battlespace Infosphere (JBI) [1] system under sustained attacks from a sophisticated adversary (class A red team). The JBI is a distributed command and control information system (developed by the US Air Force) consisting of clients that communicate with each other over a network using the publish-subscribe paradigm. Network communication is not only fundamental for the functional aspect (i.e., the client's ability to publish and subscribe); it is also the primary facilitator of attacks mounted by intruders. In order to defend against this threat, the design of the survivability architecture introduces multiple mechanisms (representing both COTS and research grade technologies) to protect the confidentiality, integrity, and availability of network

communication. In addition, the design makes extensive use of early detection and reporting of network incidents, and supports recovery and graceful degradation to mitigate compromises in the network. Because of the distributed and networked nature of the system, management of these newly introduced "survivability" mechanisms themselves depends on network communication, necessitating self-protection.

The defense-enabled JBI aims to establish a new high watermark in survivable system design and implementation. Although this paper focuses on the JBI application, any distributed client-server application can be made survivable via the DPASA architecture with little or no customization. The architecture is designed not only to place a very high barrier to unauthorized entry from the outside (intrusion), but also to place similar resistance against an attacker who is attempting to expand his initial privilege or presence in the network. The architecture uses the principles of defense-in-depth and least privilege as much as is practical. Redundancy and modularization are used to facilitate tolerance and containment of attack effects.

Apart from the significant amount of internal evaluation of the design and implementation, the most externally visible evaluation of defense-enabled JBI was a weeklong red team exercise performed in March 2005. While the detailed results of the exercise are being tabulated as of this writing, it is clear that the survivability architecture raised the survivability bar significantly: the defended system ran successfully for over 12 hours and completed its mission despite sustained attacks from a sophisticated adversary. In addition, the adversary was forced to engage the outermost layers of defense, and to operate without much visibility into the system. While the exercise observed a pervasive disruption of the defended system's WAN communication when the adversary exploited a zero day attack on a COTS product from a privileged access point, let us also note that the same attack was unsuccessful when mounted from a less privileged access point.

---

<sup>1</sup> DPASA stands for Designing Protection and Adaptation into a Survivability Architecture

In this paper we present the network aspect of the DPASA survivability architecture, our experiences during the red team exercise, and lessons learned.

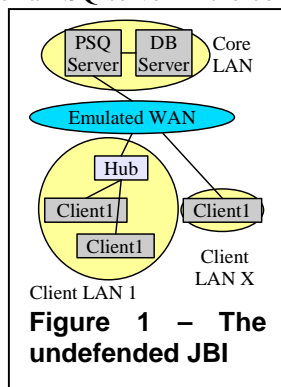
## 2. DPASA Overview

Many of today's security mechanisms such as firewalls or intrusion detection systems offer point solutions. In order to build a system that can withstand a wide range of threats, integration of multiple mechanisms is needed. The capabilities of existing solutions fall far short of what we sought to achieve in DPASA: while a COTS system can be taken down by an adversary in minutes, the defense-enabled JBI needs to survive sustained attacks from a sophisticated adversary over multiple hours. Consequently, an approach to combine the three major aspects of defense, namely protection, detection, and reaction, is needed. The DPASA survivability solution takes the form of a survivability architecture, which can be defined as the well-defined organization, placement, and interaction of a diverse set of defense mechanisms amongst the components of the undefended system. This allows for a careful organization of multi-dimensional layers of defense with each barrier backing up or managing the gaps of another.

The DPASA survivability architecture rests upon a foundation of a robust network infrastructure. This foundation consists of networking elements that support redundancy in the architecture and provide security services such as packet filtering, source authentication, link-level encryption, and network anomaly sensors. Upon detecting violations, middleware-based components within the architecture are used to support defensive responses that change the configuration and usage of the networking fabric.

A full description of the DPASA architecture is beyond the scope of this paper. We will introduce the relevant parts of the architecture by contrasting the defense-enabled JBI with the undefended version.

The *undefended JBI* (displayed in Figure 1) consists of a PSQ server in the core serving publish, subscribe, and query requests from clients combined with a database server. Air Force clients publish data into the JBI server and receive data via subscriptions and queries. The core and clients are organized into dedicated LANs



connected by a WAN emulating the SIPRNet<sup>2</sup>.

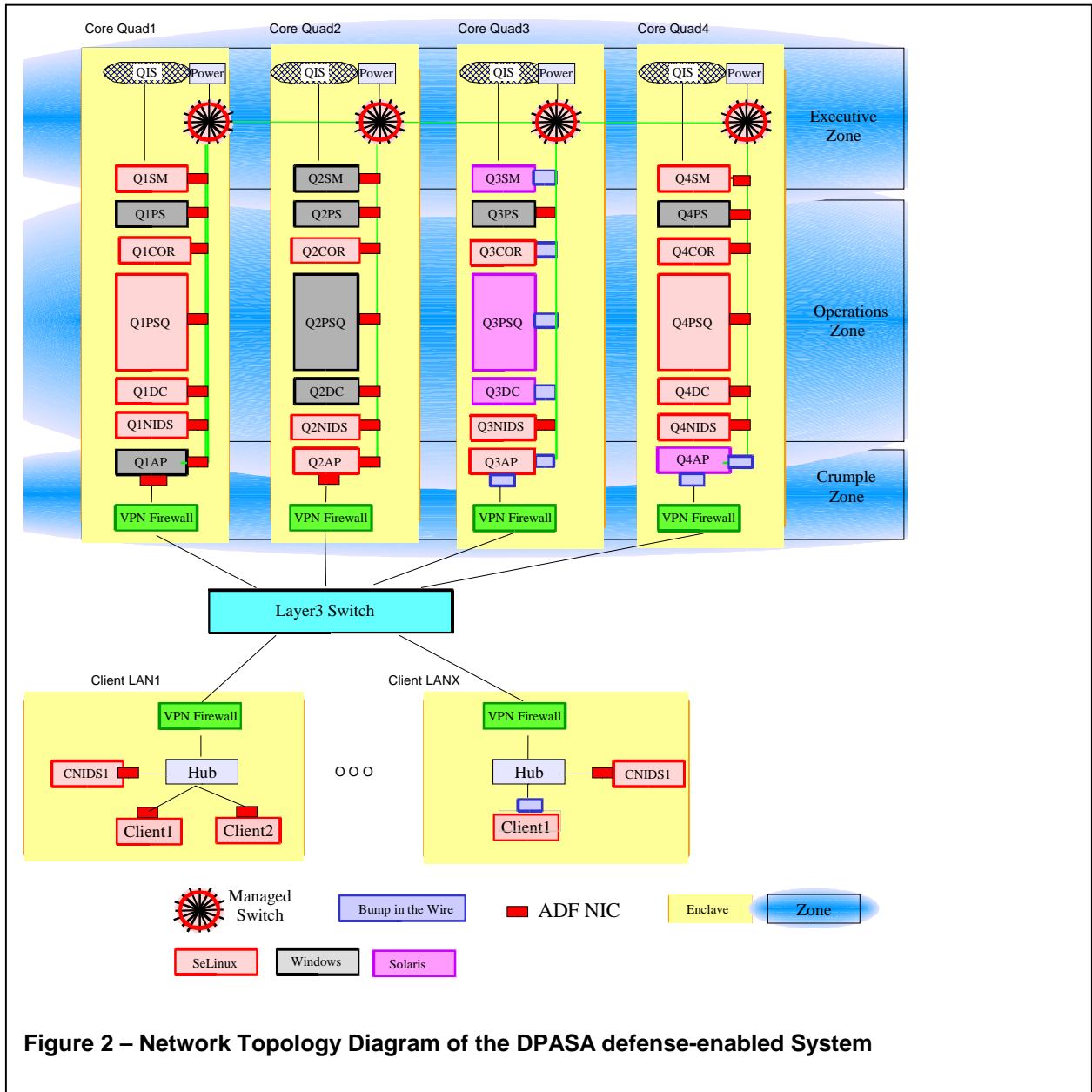
Figure 2 shows the *defended JBI* instantiated within the DPASA survivability architecture. We will use this defense-enabled JBI throughout the paper to introduce relevant architectural elements and to highlight network-level features of the architecture. The DPASA architecture introduces redundancy in the JBI core in the form of four core quadrants (quads) where each quad runs on a dedicated network LAN implemented as a VLAN. A layer 3 switch emulates the public IP networking infrastructure connecting the LANs<sup>3</sup>. Each LAN has a VPN firewall in front of it. Hosts in the four quadrants run three different operating systems, i.e. SELinux, Windows, and Solaris. All client hosts run SELinux except for legacy clients, which run on Solaris hosts. Each SELinux and Windows host is equipped with an Autonomic Distributed Firewall Network Interface Card (ADF) (see Section 3.2 for more details) that performs packet filtering and enforces encryption policies. The same functions are performed by an ADF equipped SELinux host configured as a bump-in-the-wire for each Solaris host<sup>4</sup>. As explained later in Section 3, some hosts are equipped with more than one ADF NIC. The core quads are organized into three zones. The executive (innermost) zone contains the overall management and control functions of the system. The operations (middle) zone contains hosts that are responsible for the main functional operations of the system, including the publish-subscribe-query service and supporting repositories (PSQ henceforth). The crumple (outer) zone acts as the region of first impact and proxies the operations zone functions for the clients. Zones physically impact network wiring (for the access proxies) and communication within a zone and across zones is strictly controlled via ADF policies and managed switches to limit attack propagation. The managed switches are powered via a custom device called the Quadrant Isolation Switch, which is explained further in Section 3.5.

The hosts in the crumple zone are called Access Proxies or APs (denoted as qXAP in Figure 2), and their role is discussed in Section 3.3. The hosts in the executive zone are called System Managers or SMs (denoted as qXSM in Figure 2). SMs are central components that gather system information and exert

<sup>2</sup> The SIPRNet is the DoD's classified version of the civilian Internet.

<sup>3</sup>This allows for deployment and red team testing of the system in a laboratory environment

<sup>4</sup> The driver for the NIC card was not available for the Solaris platform—one of the drawbacks of using research grade technologies



**Figure 2 – Network Topology Diagram of the DPASA defense-enabled System**

control on other components via adaptive algorithms for suggesting appropriate defensive actions to a human operator. More details on SMs, especially in terms of adaptive responses involving the networking elements, can be found in Section 3.8. As shown in Figure 2, the operations zone in each quad  $x$  consists of a Network Intrusion Detection System (denoted as  $qxNIDS^5$ ) and a Correlator (denoted as  $qxCOR$ ) both of which are discussed in Section 3.7; a Downstream Controller or a DC (denoted as  $qxDC$ ) that acts as an intermediary between the Access Proxies and the System Managers; a PSQ server hosting the PSQ function and associated repositories (denoted as  $qxPSQ$ ); and an ADF Policy

Server or a PS (denoted as  $qxPS$ ), which is a 3rd party control server for the ADF NICs.

With this brief introduction to the essential elements of the DPASA architecture, let us now explain how the architecture achieves the desired level of protection, detection, and adaptation at the network level by synergistically combining elements of COTS and research grade network technologies.

### 3. Network Design and Implementation

One of the objectives of the DPASA survivability architecture is to force the attacker to cross multiple defense barriers in order to be able to cause significant

<sup>5</sup> Client LANs also have a NIDS as shown in Figure 1

damage. This means that multiple barriers must be placed in the system and the system configuration must be hardened to prevent easy subversion or bypass of these barriers. Another advantage of having these multiple barriers is that it forces the attacker to spend more time, which in turn increases his chance of getting detected by one of the sensors embedded in these barriers. This section explains in detail the network-centric mechanisms in terms of their design, implementation, and how they engage or thwart attacks.

### 3.1 VPN Firewalls

VPN firewalls form the first line of defense against attackers coming from the untrusted public IP network (WAN) into LANs of the defended system. All communication between client and the four core LANs is intercepted by COTS a VPN firewall, which encrypts the traffic and forwards it over a crypto tunnel to the corresponding destination firewall. These VPNs effectively hide internal network addresses and payload content from packet sniffers on the WAN and deny opportunities for fine-grained traffic analysis. Any invalid or replay VPN traffic is dropped by the firewall before it reaches the inside LAN networks.

### 3.2 Autonomic Distributed Firewall Cards

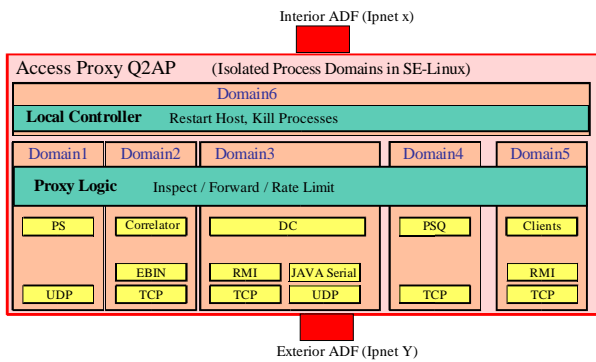
In a typical secure network environment, the distinction made between outsiders, the DMZ, and internal network is enforced at the DMZ by a firewall. This means that once an attacker defeats the firewall, he has relatively unrestricted access to the inside assets. By forcing the attacker to face the firewall hurdle as often as possible, the system's resistance can be improved. Recent advances in distributed firewall research enable finer grained firewalling via a custom firewall in each host. In the defense-enabled system, we use an Autonomic Distributed Firewall [2] (ADF) NIC on each host in the system (which are much more efficient and secure than software based personal firewall products). These firewalls are built into the network interface firmware, are separate from the operating system, and are controlled via encrypted messages by a Policy Server (see Section 2 and Figure 2). This separation means that even if an attacker compromises a host, the attacker cannot loosen the firewall policy on the compromised host. These firewalls enforce the separation required between the zones and also provide protection from unauthorized inter-zone communication. They only allow the network paths and protocols required by the system. Whereas a traditional firewall at a DMZ boundary needs to pass any traffic that any host behind it may

possibly need, DPASA enforces a least-privilege policy for network traffic on a per-host basis.

In addition to filtering out unwanted traffic both incoming and outgoing, ADFs also provide confidentiality and authentication using Virtual Private Groups (VPGs)[3]. VPGs encrypt messages sent between ADF NICs in the same group with a shared key. Non-members cannot send messages to the group without the key, and to prevent a group member from masquerading as another member, ADFs prohibit a host from sending spoofed packets. VPGs offer improvements over a traditional VPN in terms of multiple endpoints and finer-grained control. However, traditional VPN firewalls also have their place in the architecture as explained in Section 3.1. The VPN firewalls and the ADF NICs are an example of two layers of defense backing each other up. The coarse grained, but higher capacity (than the ADF NICs) VPN firewalls stop unauthorized traffic before it gets a chance to potentially overwhelm the ADF cards. This allows the ADF cards to deal solely with intra-VPG traffic.

### 3.3 Access Proxies

The JBI core is critical for the system's operation and therefore, it is a natural target for an attacker who has gotten past the outer VPN layers (for example by taking over a host on a client network which is legitimately able to talk within a VPN as well as certain VPGs). Attempts to compromise the JBI core from within the VPN layers are confronted by AP hosts. Figure 3 shows the overall process and network layout of a SELinux AP. Each AP is equipped with 2 ADF cards to protect its two network interfaces. The exterior ADF is configured to only allow VPG traffic from legitimate client ADFs in the same group, and further restricts access to the UDP and TCP ports on which the proxy processes are listening on. Each server in the operations zone is fronted by an access proxy application which processes incoming traffic from the exterior ADF, inspects the content, enforces rate-limiting if necessary, and then forwards the traffic to the corresponding server via the interior ADF. Since source IP addresses are guaranteed to be authentic via the ADF anti-spoofing feature, the rate limiting code maintains dedicated queues per source IP address for fair-queuing, making it hard for an attacker to cause legitimate traffic to be dropped in favor of attack packets. Furthermore, additional address checks are performed on alert payloads to crosscheck the source IP from the TCP packet with the source IP address from the alert payload.



**Figure 3 – Network Protection through Process Domains and ADFs on an Access Proxy**

Proxy processes, like most other DPASA processes, are started and executed in their own process protection domain, which limits their privilege and isolates them from each other via technologies such as SELinux, Cisco Security Agent (CSA), and Java Security Policies. Most of the protocols between clients and the core utilize all four APs in parallel and perform end-to-end encryption of their data. This means that the encrypted data flows through the access proxy without being decrypted, limiting the checks that can be performed to size and frequency expectation checks.

### 3.4 Core Switch Layout

Following the principle of least privilege the DPASA design calls for tight control of physical connectivity among the LANs and zones, especially in the context of the core quads. In addition to restrictions imposed by the VPGs, which are limited due to space limitations on the card, the managed switches can be configured to block any connection not permitted by the architecture at the link level. However, managed switches that support the source-port filtering needed to enforce such restrictions are relatively high end, and as a result this functionality was not implemented in the evaluated defense-enabled JBI.

### 3.5 Quadrant Isolation Switch

Each quad LAN is implemented on its own switch that provides the backbone for all communication within the LAN. A Quadrant Isolation Switch (QIS) allows human operators to isolate an entire quad if necessary. Usually, such a drastic measure is taken when other adaptive attempts to recover the quad have failed and graceful degradation, i.e. running without the quad, is deemed appropriate. The QIS, a custom piece of hardware, controls the power to each quad's switch and is cross-connected to the other 3 QISs and its local SM via serial ports. Each SM can vote whether or not to isolate any of the four quads. If three SMs vote to isolate a quad, power to that quad's switch is turned off by the QIS, effectively removing it from the network and denying those hosts to an attacker. The DPASA protocols are designed for such a situation and are capable of working in a degraded mode when a quad is

lost. Each QIS also has an override switch to restore connectivity in case manual intervention is required.

### 3.6 Network Hardening

In order to more readily detect suspicious traffic and remove potential attack points, any unnecessary network service is shut down or if possible uninstalled from the host, switch, or other network enabled device. This ensures that any alarms or traffic generated by the removed services are immediately noted as a possible attack. In previous red team exercises [3,4,5], low-level network attacks using ARP were used to great effect. Network traffic was redirected through attack machines and denied to its target through ARP manipulation. While ADFs prevent ARP spoofing, the architecture contains many non-ADF equipped components such as switches and VPN firewalls, which are vulnerable to this attack. In order to deny this attack-point to adversaries, static ARP tables are used wherever possible.

TCP/IP Stack hardening is utilized on all three operating systems making hosts more resistant to various attacks. Particularly, the TCP Stack is hardened to make hosts more resistant to SYN floods and other denial of service attacks. Various other commonly accepted network-hardening steps were employed, i.e. increasing TCP queue length and circuit establishment timers.

### 3.7 Network Detection

In order to detect malicious network activity NIDS appliances are deployed in each client LAN segment and in the crumple zone of each core quadrant. These systems analyze network traffic and report anything suspicious to the Correlator. Each client NIDS is fitted with three NICs: a non-ADF NIC used for sniffing, an ADF NIC used for sniffing, and an ADF NIC used to send alerts. The Core NIDS have four sniffing interfaces, one pair for the LAN and one pair for the external subnet. If the NIDS did not contain a non-ADF sniffing NIC, unencrypted traffic would be silently dropped at the NIC and never reach the NIDS for analysis. Since all hosts on the network have an ADF NIC, non-VPG traffic is a good indication of a possible attack.

In addition to standard signature-based network intrusion detection techniques, we use violations from various policy enforcement mechanisms for intrusion detection. At the application level, illegitimate network actions (such as opening a rogue server port) are detected and reported by application level policies. At the network level, each time an ADF equipped host sends an outgoing packet that violates the host's ADF

policy, the NIC drops the packet and generates an audit alert to the Policy Server. Auditing of incoming packets was disabled to deny an unprivileged attacker the ability to flood the system with audit traffic by simply sending a bad packet. Analogous mechanisms exist for handling violations of non-network policies, but these are outside of the scope of the present paper.

To deal with components abusing allowed communication paths, the DPASA components and protocols are designed to be robust against malformed data and report possibly malicious actions using custom alerts. These alerts include flooding, replay, and traditional access control violation reports.

In order to provide visibility into the network and to indicate component availability, most DPASA components are sending heartbeat messages in regular intervals. When a heartbeat fails to reach a quad for more than a few seconds, the SM alerts its operator. If heartbeat messages are failing to reach the core on account of network problems, there is a good chance that scenario traffic is similarly failing. If only one quad is reporting missing heartbeats, it is a further indication that the network to that quad is affected. Learning about a failure quickly allows operators to take action nearly as soon as problems occur and spend as much time diagnosing and fixing them as possible. During the red team exercise, missing heartbeats often indicated the first sign of an attack.

### 3.8 Automatic Adaptive Network Defenses

Auto-adaptive responses enable the DPASA system to continue to function even for cases in which the attacker has managed to break through protection layers and started to affect availability or integrity of the system. The overall goal is to engage the attacker through dynamic defenses and thus slow down attack propagation and keep the system operational, albeit at a reduced level (i.e., graceful degradation). Two kinds of adaptation are supported in the defense-enabled JBI: 1) SM-initiated “auto actions” which are executed upon observing correlated alerts about suspicious hosts and 2) local adaptations used to compensate for locally observed attacks which are implemented throughout the DPASA code base.

The main network-related auto action instructs the ADF Policy Server to isolate a host by putting all of the hosts ADFs into “block all” mode, which causes the NICs to drop all incoming and outgoing traffic. After inspecting and restoring the suspicious host, the core operator can bring back the NICs by putting them back into operational mode (i.e., recovery, as opposed to degradation). When a quad is severely compromised,

the operator can use the quad’s QIS to disconnect the entire quad from the network.

DPASA contains a variety of adaptive distributed algorithms that change behavior based on changes in the environment. Application-level queues with threshold schemes are used for sending out alerts so that alerts can be throttled in response to attacks that cause too many alerts. Another good example is access proxy selection and graceful degradation of the PSQ protocol. When a publish request is created by a client, the protocol determines which AP to send the request to according to past response times and trust, causing APs that have been fast in the past to be preferred over slow ones (note that the slowness could have been contributed by network delay). The client waits for positive acknowledgment or eventually times out and retries. This timeout is implemented using a bounded exponential back-off scheme to adapt to current system conditions.

## 4. Evaluation

Validation activities took place throughout the development of the DPASA survivability architecture. We identified ADF NICs as the main target for network attacks through internal white boarding sessions and stochastic model-based simulation [7]. Further empirical studies showed that ADF NICs are susceptible to denial-of-service attacks under certain circumstances [8], which generated various mitigation strategies, including the COTS VPN layer. A detailed discussion is beyond the scope of the paper; instead this section focuses on experiences and results from the most visible external evaluation of this project, the red team exercise that took place in March 2005.

### 4.1 Red Teaming of the Actual System

The main objective of the March 2005 red team exercise was to determine whether the defense-enabled JBI could survive 12 hours of sustained attacks and complete its mission. Two different red teams participated on separate days; each had complete knowledge of the system, and was fairly unrestricted in terms of rules of engagement. In addition, both red teams had access to a Switched Port Analyzer (SPAN) port, a VLAN port, and a trunk port on the layer 3 switch. The SPAN port enabled the red teams to see and capture all traffic through the switch (i.e., emulated SIPRNet). The trunk port, we later found out, gave the red team a trusted control channel into the emulated SIPRNet switching infrastructure. The first red team began their attacks from a VLAN on the layer 3 switch, giving them the same physical network access as any of

the VPN firewalls. Attacks were mainly targeted at the VPN firewalls in order to decrease system availability. Initially, the VPN firewalls were configured to drop any non-valid VPN traffic. However, this did not allow the red team any visibility into what happened to their attack packets. To continue testing with the first red team, a change was made in the VPN firewalls to allow all unauthorized traffic to be routed back to the emulated SIPRNet.

With this change, the red team was successful in flooding the system with massive amounts of seemingly valid replay traffic<sup>5</sup>. The flood condition caused lost heartbeat messages from clients and delay in PSQ requests. Multiple errors were also observed from the VPN firewalls under attack. However, the red team had no insight into the effectiveness of this attack, and moved on to different unsuccessful attacks. The system recovered and the mission was completed.

The second red team exploited the trunk port, which had Dynamic Trunking Protocol (DTP) enabled by default. This implied that the trunk port was highly trusted. The red team devised an attack to set up counterfeit endpoints for all DPASA VPN firewalls using their own hosts, together with an additional layer 3 switch connected to the trunk port. This attack was highly effective and all traffic sent from any DPASA LANs destined for the SIPRNet did not receive any replies, effectively disconnecting the LANs from each other and stopping the mission. Using the same attack over a port without DTP however, did not have any impact on the defended system.

In order to test defense in depth, the red team was next allowed to start the attack from an ADF enabled client host behind a VPN firewall. The red team started flooding through authorized and unauthorized paths, with no impact on the system. Next, the red team starting point was moved back to the emulated SIPRNet but with VPN firewalls disabled. Using corrupted replay packets, they were able to affect availability of one of the VPGs. The NIC on a Windows XP system was impacted by the attack and the host had to be rebooted to recover. The same attack had no impact on other Operating Systems (W2K, Solaris, SELinux).

## 4.2 Lessons Learned

Both red teams shared their frustration of "flying blind", as they were not able to gauge the impact of their attacks, which forced them to move on to other attacks. Also, since the system was hardened with only the

essential system utilities remaining, the red teams had difficulties installing and running their attack software and tools.

Although the analysis of the data collected from the exercise is still continuing, some lessons learned can be noted. We assumed that monitoring inside the two layers of VPNs was enough; consequently our NIDSes were not looking at what was being blocked or deflected by the VPN routers. On hindsight, it would have been beneficial to place a NIDS outside the VPN firewalls. In addition, since the VPN firewalls were the first target for attacks, fail over of VPN routers could have been useful for survival. This would have allowed legitimated WAN traffic to continue through an alternate router when the primary router was attacked. It should also be noted that the exercise was conducted using an emulated SIPRNet with no redundancy in the emulated part. The main goal was to defend protected LANs attached to the SIPRNet and not the emulated WAN network itself. Attacks that are solely targeted towards denying network service on the SIPRNet through low-level link flooding were deemed less interesting and therefore avoided in favor of other resource consumption attacks, including attacks on the TCP stack of the end systems.

## 5. Conclusions

The red team evaluation of the defense-enabled JBI demonstrated that it is possible to develop a well-configured adaptive system that can defend critical system functionality against sustained attacks from a sophisticated adversary over a reasonably long time. While some attacks caused disruptions during the mission, they required high-privilege local access on the emulated network, and potentially a flaw in the COTS VPN routers (based on preliminary red team reports). This level of access (equivalent to having SPAN and Trunk port access to a large number of routers in a public IP network simultaneously) would be difficult to obtain in a fielded system, but not impossible. It should be noted that the attacks did not impact integrity and confidentiality, and further did not compromise or damage any of the host operating systems.

Even though we raised the survivability bar, this achievement should be considered in the context of the big-picture view of cyber-security. The inherent asymmetry in the fight against attackers still remains. While attackers need to find only one flaw, defenders need to ensure that most (if not all) of them are addressed. The exercise highlighted the obvious fact that there will be flaws in the system implementation or

---

<sup>5</sup> mostly Internet Security Association and Key Management (ISAKMP) traffic

configuration, and a determined adversary will find and exploit that flaw.

We therefore view the successes of the DPASA effort as a continuing, but determined step forward in our fight against the threat of cyber insecurity. This experience shows that, in the context of defending the network, careful combination of appropriate technologies can make it very hard for attackers to find a successful attack. This is especially true when dealing with attacks aimed to breach integrity or confidentiality.

Examples where we expect to find the next generation of survivability tools and technologies include exploration of 1) regenerative ideas to replace lost capabilities with newer and improved ones, perhaps with different security policies and configuration settings, 2) how to combine adaptive security with protection measures that are grounded in hardware or sound cryptographic techniques, and 3) how to use policy violations and IDS alerts to compute trustworthiness of system components.

## 6. References

- [1] AFRL JBI homepage - <http://www.infospherics.org>
- [2] Payne, C., and Markham, T. "Architecture and applications for a distributed embedded Firewall". In 17th Annual Computer Security Applications Conference (December 2001).
- [3] Tom Markham, Lynn Meredith, and Charlie Payne. "Distributed embedded firewalls with virtual private groups". In DARPA Information Survivability Conference and Exposition -Volume II, Washington, D.C., April 2003. DARPA, IEEE.
- [4] Michael Atighetchi, Partha Pal, Franklin Webber, Richard Schantz, Christopher Jones, and Joseph Loyall. "Adaptive Cyberdefense for Survival and Intrusion Tolerance". IEEE Internet Computing, Vol. 8, No. 6, November/December 2004, pp. 25-33.
- [5] W. Nelson, W. Farrell, M. Atighetchi, S. Kaufman, L. Sudin, M. Shepard, and K. Theriault. "APOD Experiment 1: Final Report", BBN Technologies LLC, Technical Memorandum 1311, May, 2002.
- [6] W. Nelson, W. Farrell, M. Atighetchi, J. Clem, L. Sudin, M. Shepard, and K. Theriault, "APOD Experiment 2: Final Report" BBN Technologies LLC, Technical Memorandum 1326, Sep, 2002.
- [7] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J. F. Meyer, W. H. Sanders, P. Pal, "Model-Based Validation of an Intrusion-Tolerant Information System", from the 23rd Symposium on Reliable Distributed Systems (SRDS 2004), posted July 16, 2004.
- [8] M. Ihde and W. H. Sanders, "Barbarians in the Gate: Packet Flooding NIC-based Distributed Firewalls", submitted for publication