

www.securecomputing.com

Final Report: Targeted Security Assessment of the Sidewinder G2 Security Appliance

Secure Computing® has been solving the most difficult network and application security challenges for over 20 years. We help our customers create trusted environments both inside and outside their organizations.

This report details a targeted assessment from Black Hat Consulting of the Sidewinder G2 Security Appliance and how it handles a comprehensive set of fundamental real-world attack methodologies, ranging from layer 2 to layer 7 attack methods as referenced against the OSI Model.

For Secure Computing Corporation
By Black Hat Consulting

Table of contents:

Key findings and introduction..... 2

BHC methodology 2

Testing environment..... 3

Sidewinder G2 Security Appliance successes 3

 Denial-of-service..... 4

 Rule bypass 4

 Information acquisition 4

 Exploits 4

 Spoofing 4

 Information leakage and acquisition 4

 Testing anomalies..... 4

 Protocol failures..... 4

Conclusions 5

Appendix one: Line item results per the coverage 5

Appendix two: Report data files 7

Appendix three: Network layout..... 7

Appendix four: Interpreted anomalies..... 8

Appendix five: Incomplete test issues..... 8

Secure Computing Corporation

Corporate Headquarters

4810 Harwood Road
San Jose, CA 95124 USA
Tel +1.800.379.4944
Tel +1.408.979.6100
Fax +1.408.979.6501

European Headquarters

East Wing, Piper House
Hatch Lane
Windsor SL4 3QP UK
Tel +44.1753.410900
Fax +44.1753.410901

Asia/Pac Headquarters

1604-5 MLC Tower
248 Queen's East Road
Wan Chai Hong Kong
Tel +852.2520.2422
Fax +852.2587.1333

Japan Headquarters

Level 15 JT Bldg.
2-2-1 Toranoman Minato-Ku
Tokyo 105-0001 Japan
Tel +81.3.5114.8224
Fax +81.3.5114.8226

Key findings:

- The testing of the Secure Computing® Sidewinder G2® Security Appliance was a complete success. We note a few minor procedural exceptions at the end of this report, which cannot be attributed to Sidewinder G2, but rather, to the testing process itself.
- The system successfully defended against all of these attacks methodologies across OSI layers 2 through 7.
- We were very impressed with the overall functionality, control, and reliability of the Sidewinder G2 Security Appliance and its ability to thwart all types of attacks.

Introduction

In this day of sophisticated attacks and increasing reliance upon the Internet for day-to-day commerce, one must pay exceptionally keen attention to the network, and the security environment in place that is meant to protect it. Merely adding any commodity firewall and anti-virus software is inadequate; good security calls for best-of-breed tools, constant vigilance, and a bit of paranoia. The firewall must be able to stop both known and unknown, attacks, and it must be capable of application-layer filtering. It is with this in mind that we tested Secure Computing's Sidewinder G2 Security Appliance.

Secure Computing engaged Black Hat Consulting (BHC) to test the Sidewinder G2 Security Appliance against the most common type of "hacking" attacks. The goal of the testing was to test each of the seven layers of the Open Systems Interconnect (OSI) network model. Attacks at all layers of the OSI model were tested independently and then together, as each incrementally higher network protocol relies on the stability of the previous protocol. This approach to testing has become essential in this day and age, and thoroughly examines Sidewinder G2's ability to filter out single layer attacks as well as sophisticated blended attacks that attempt to exploit more than one layer in a single attack up to and including the application layer.

In recognition of the fact that attacks can come from anywhere, and often do originate from within the network itself, all tests were performed with a basic dual network configuration with internal and external zones. The appliance acted as the defensive point to stop attacks from the hacker directed to the server, and attacks from a hacker inside the internal network. Due to the nature of most protocol-based attacks, we felt it was an added bonus to leave the protocol

attacks running overnight and over weekends. We feel this added brute-force-attack resiliency as an element of the study, thus delivering more comprehensive results for those given tests.

Based on the standard Ethernet frame size, it is not feasible to generate all possible packets onto a network. We have attempted to address this problem with our protocol attacks by producing a massive amount of packets and randomly seeding the fields that are being tested. The volume of traffic generated is large, making it impossible to record it all; however, we include all the generative data with our deliverable so that results can be duplicated as necessary.

BHC methodology

The majority of network-based attacks are based on malformed packets, and various exploits of the rules that protocols use to function properly. When asserting the functionality and security of a system such as the Sidewinder G2 Security Appliance, we test based on the assumptions and rules defined for network connectivity and usage. The layers of the OSI model are tested independently and then together, as each higher network protocol relies on the stability of the previous one. A system that can defend against one layer may fail defending against another. It is for this reason that simple, packet-filtering firewalls are no longer adequate to address the threats that exist today. If the rules to determine the validity of one layer can be bypassed, then it may significantly affect the entire system's stability. This is why our test suite focuses on layers, instead of merely listing specific exploits. The majority of exploits in existence today use the protocols tested below.

The ability to correctly parse and limit malformed packets from being handled and routed is of significant importance. When we approach the application layer, the routing system may no longer specifically and categorically stop an exploit that is using well-formed packets. Instances of those attacks include the worms and other exploits we use at the conclusion of the testing phase. Given the nature of many of the exploits and worms that exist today, systems such as the Sidewinder G2 Security Appliance, which are built from the ground up to stop attacks at every layer, deliver the enhanced protection that is necessary for the enterprise. This sort of functionality, and the ability to stop these dangerous, application-based attacks, becomes quite valuable in maintaining a high level of security and addressing every possible type of attack or exploit. Our testing will assert and verify that network protocol-based exploits will be stopped and that malformed packets will not be routed or accepted as valid.

Testing environment

For the majority of the tests our rule base was configured with almost everything open on the appliance. This allowed us to ensure that the invalid packet structures, which took up a majority of our test coverage, were not going to be routed or accepted. Specific application tests and rule set bypass tests exercised the more strict versions of the appliance's rule capabilities. The malformed protocol packets were the easiest to generate and allowed us to get a baseline feel for how the appliance was going to react to more rigorous tests. Specifics of the rules for each test are outlined within the detailed output report along with capture files and packet structures.

We executed many tests, both from inside and outside the appliance, in order to get the most coverage from possible internal and external attacks. We sent gigabytes of data to and through the appliance as a result of bombarding it with high volumes of denial of service and malformed packets over an extended period of time. We bombarded the appliance with a wide variety of attacks that an enterprise is most likely to face.

Our test application builds custom packets based on a serialized packet format file. Since network packets have fields that are often less than 8 bits, our program interprets a single byte for input and automatically places those bits in the correct position within the packet upon generation. For example, given an IP packet, the first two fields are 4 bits each. Within the packet format file, you will see the IP Version Field defined as 0x04. The 4 bits for the version are automatically shifted to their significant location and then XOR is used to set the bits for that field in the outgoing packet data structure. Therefore, the length of the field describes the number of bits that are valid in the source byte or bytes. The bit length of the IP Version Field is 4 bits, so only the least significant 4 bits of the IP Version Field byte are used.

Since we generate each packet before it is sent, our randomizer can act upon the bytes for the current value of a field regardless of the actual number of valid bits, since all fields will have at least 1 byte as the current value, even if the packet assembler only uses specific bits.

Many of the attack tests we ran required custom packet creation methods. These baseline packet outlines could now be used in future tests to replay the same generative attacks with nothing more than some minor configuration changes for network addressing. And some of the attack tests we ran required manual step-by-step processes to definitively

ascertain whether the appliance successfully defended against certain given attacks. These manually intensive tests were generally based around attempting to hijack sessions over TCP and HTTP, including race-condition attacks where we as the attackers attempted to preemptively respond to a global request before the valid system answered the request.

We feel overall that the protocol-level tests have shown that the appliance discards invalid and malformed packets with ease and efficiency. Any valid packets (i.e., not malformed by our packet assembler) that were blocked by the security appliance fell under the defined rules of the appliance. In other words, Sidewinder G2 only stopped valid packets that the rules were defined to stop; and it did so in every instance, on every layer, and without blocking any legitimate traffic.

Although the following comment was not in the scope of this testing, we found the administrator user interface to the appliance to be an extremely useful and efficient way to view the real-time auditing information that helped us assert many successful tests.

Sidewinder G2 Security Appliance successes

The testing of the Sidewinder G2 Security Appliance was a complete success. The appliance blocked, filtered or ignored the attacker's packets that were transmitted to both internal and external interfaces. Intelligent application-level proxies and secured servers protect DNS, SMTP mail, FTP, HTTP and other high-use Internet services in ways that no "stateful-inspection-only" firewall could deliver.

We mark the Sidewinder G2 Security Appliance as the most stable and reliable firewall we have tested. During our testing, our local router was not able to handle the packet load and malformed packets that were being transmitted. The router continually crashed while testing packets that would be used during testing of the appliance and we had to remove it from the network. But the appliance stood up to very rigorous testing, and after completion of the tests, we were still able to connect from inside and outside the appliance to the services that we had open on the network.

In terms of interface and administration, Sidewinder G2 is highly efficient. A firewall with a difficult interface that is hard to administer often results in a situation where exploits occur simply because the device itself has been misconfigured, or paths inadvertently left open. **A misconfigured firewall**

is worse than no firewall at all in many respects, since it leaves the administrator with a false sense of security. This is certainly not the case with the Sidewinder G2 Security Appliance. Sidewinder G2 provides solid administration abilities and delivers excellent reporting facilities as well, with clear and precise reports that are easy to understand.

The system performs well in all respects, preventing attacks and exploits, and in initiating subsequent action whenever an attack is attempted.

Denial-of-Service

The appliance performs exceptionally well, exceeding our expectations based on the amount of traffic sent, both well and malformed. The detection mechanisms were useful and detected our flooding attempts, especially over valid channels through the appliance.

Rule bypass

The appliance rules as they were implemented were flawless.

Information acquisition

An attacker will often first attempt to gather information about the network and the firewall environment in order to better understand and exploit it. Sidewinder G2 protects this information well. No information that was deemed a security risk or information that could be used beyond the understanding of the functional operation of the appliance could be found.

Exploits

The Sidewinder G2 Security Appliance performed well under our rigorous testing environment. There are some competing appliance products that, unlike Sidewinder G2, have allowed the most basic exploits through, even years after their introduction to the public. Those products don't even detect, let alone filter, invalid packets. As more exploit definitions are added to the appliance it will become even more secure at protecting systems not yet patched or unaware.

Spoofing

When we send well-formed packets that are within the constraints of the rule sets, it is not possible to determine if the packet is spoofed or real. The appliance did well to raise alerts when flooding was done, and even when spoofing with random source addresses, the appliance did a good job signaling flood and probe attacks. When such an attack is

detected and Sidewinder G2 generates an alarm, action can then be taken by the administrator; and automatic reactions can also be programmed. The appliance performs admirably, while still adhering to the functional requirement to allow good traffic and yet detect bad traffic from multiple sources over time.

Information leakage and acquisition

The information leakage tests were deemed successful. It is obvious when connecting to the appliance that the given source machine violates some manner of the rule set defined on the appliance because you are immediately disconnected. We do not feel that any specific information leakage from the appliance exists that would lead to the determination of exactly what the rule sets are, without performing a large denial-of-service attack, spoofing many and all fields of standard protocol fields. And as mentioned earlier, **denial-of-service attacks and spoofs are handled exceptionally well by the Sidewinder G2 Security Appliance. An attack of this size would be detected by the appliance immediately, which would then escalate the notification to people who can act based on the details of the event and prevent the attack from succeeding.**

All floods, port scans and even random source IP, source MAC, and destination port scans were detected by the appliance. Based on the configuration of the response mechanisms inside the appliance, any network can act as needed, based on these attacks and queries. The appliance does a great job at anomaly detection for incoming scans.

Testing anomalies

In a few circumstances, minor issues arose, which must be mentioned. These issues do not reflect any flaws whatsoever in the overall security of the Sidewinder G2 Security Appliance; rather, these were the result of procedural anomalies in the testing environment, or situations that the Sidewinder G2 Secure Appliance can easily be configured to address.

Protocol failures

1. Hijacking session cookies (7-HTTP-HIJACK-1)
2. Session ID to IP address mapping (7-HTTP-HIJACK-2)
3. Hijacking UID (7-HTTP-HIJACK-2)

This set of hijacking tests isn't exactly fair to the appliance. Let us be clear why. Given the widespread use of proxies that aggregate traffic services at ISP sites, that exist all over corporate networks, are included in numerous different kinds of security devices, and given the widespread use of terminal

services, no security appliance could stop the hijacking of a stateless session between a hacker and user who are not present on the same network as the appliance. Generally, the appliance will always see the same IP address as the source proxy for multiple sessions and that proxy's own NAT will be required to perform a check. However, even that proxy has to submit to the fact that the source IP addresses it is receiving could be from another proxy also performing NAT. Hence, it is not feasible for any firewall security appliance to make this check. As well, a hacker could just as easily spoof an IP address, and given that an HTTP session has no protocol-level state (assuming correct HTTP implementation and each request has its own TCP session) if a hacker did gain access to the session ID, it would be impossible for the security appliance to detect the difference.

Hijacking sessions outside of SSL cannot be mitigated 100%, and the majority of applications don't even try. Even with the best implementation of Layer 7 HTTP URL hashing or some other form of Token transfer with each request, it is still possible to hijack if the client can be poisoned or denied service to the network when the hacker makes the hijack attempt. In fact, if an appliance enforced any Session ID to IP address convention, it would in fact deny service and cause false alarms given the configuration and use of HTTP over the Internet.

In the future, it would be useful to perform an SSL keying when the appliance is the SSL endpoint to a Session ID and IP address mapping test. Though this does not mitigate hijacking completely, it would be important to show that a hacker cannot hijack a Session ID and then create a new SSL connection through the appliance to the destination server.

Hijacking sessions is something that can generally be mitigated (by a decent percentage) by an application that maintains good state and uses encrypted sessions. Though hijacking cannot be completely mitigated, there is good documentation available within the development world on methods to detect and deter hijacking attempts.

This discussion, although interesting, can be considered a non-issue as it relates to our testing of the Sidewinder G2 appliance.

Conclusions

Having successfully penetrated and exploited many systems deemed and marketed as "secure" over the last several years, some in development, and many already live and publicly accessible, the Sidewinder G2 Security Appliance is the by far the sturdiest system we've audited.

Having stood up to very rigorous testing, we are confident about the stability of the appliance. Given that we could take down all the other systems on our network that we used during test—but not the appliance—we were very impressed with the overall functionality, stability, control, and reliability of the Sidewinder G2 Security Appliance. We mark the Sidewinder G2 Security Appliance as the most stable and reliable firewall we have tested.

Appendix one: Line item results per the coverage

If you view the Coverage in order in the Agreement you will be able to correlate the test names to the test acronyms below. They are listed in the same order as in the Agreement.

DOS	Denial of Service Attacks
SPF	Spoofing Attacks
RSB	Firewall Rule Set Bypass Attacks
EXP	Exploit
BF	Brute Force
HIJACK	Hijacking Sessions
DT	Directory Transversal

Table 1

Tests	Description	Status
2-ETH-DOS-1	Invalid packet lengths	Success
2-ETH-DOS-2	Malformed packet headers	Success
2-ETH-DOS-3	Incorrect packet frame sizes	Success
2-ETH-DOS-4	Invalid packet padding	Success
2-ETH-DOS-5	Invalid packet FCS	Success
2-ETH-DOS-6	Invalid packet addresses	Success
2-ETH-DOS-7	Various malformed packets	Success
2-ETH-DOS-8	Flooding w/ random MAC addresses	Success
2-ETH-SPF-1	Spoofing destination MAC	Success
2-ETH-SPF-2	Spoofing source MAC	Success
2-ETH-SPF-3	Spoofing MAC in and out of sessions	Success
3-IP-DOS-1	Invalid IP versions	Success
3-IP-DOS-2	Invalid TTL	Success
3-IP-DOS-3	Invalid source addresses	Success
3-IP-DOS-4	Invalid CRC	Success

Table 1 continued, page 6

Table 1 continued from page 5

Table 1

Tests	Description	Status
3-IP-DOS-5	Mixed options	Success
3-IP-DOS-6	Random padding values/lengths	Success
3-IP-DOS-7	Invalid/random flags	Success
3-IP-DOS-8	Fragmentation of packets	Success
3-IP-SPF-1	Spoofing into created sessions	Success
3-IP-SPF-2	Loop back source address spoofing	Success
3-IP-RSB-1	Complete rule set bypass attack	Success
3-IP-RSB-2	Random data rule set bypass attack	Success
3-ICMP-DOS-1	Invalid type values	Success
3-ICMP-DOS-2	Invalid code values	Success
3-ICMP-DOS-3	Invalid CRC values	Success
3-ICMP-DOS-4	Random data in unused/main fields	Success
3-ICMP-DOS-5	Responding to pings not sent	Success
3-ICMP-DOS-6	Ping of death invalid field value attack	Success
3-ICMP-SPF-1	Spoofing internet headers	Success
3-ICMP-SPF-2	Spoofing w/ original datagram data	Success
3-ICMP-SPF-3	Race condition responses	Success
4-TCP-DOS-1	Invalid port numbers	Success
4-TCP-DOS-2	Invalid sequence numbers	Success
4-TCP-DOS-3	Sequence rollover	Success
4-TCP-DOS-4	Out of state responses flags	Success
4-TCP-DOS-5	SYN floods	Success
4-TCP-DOS-6	Invalid data offsets	Success
4-TCP-DOS-7	Invalid window values	Success
4-TCP-DOS-8	Invalid CRC values	Success
4-TCP-DOS-9	Random padding	Success
4-TCP-DOS-10	Random data as TCP header	Success
4-TCP-DOS-11	ACK storms	Success
4-TCP-DOS-12	Fragmentation	Success
4-TCP-DOS-13	Partial ACK	Success
4-TCP-DOS-14	No flags	Success
4-TCP-DOS-15	Zero Length	Success
4-TCP-DOS-16	Out of window packets	Success
4-TCP-SPF-1	Source and destination ports	Success
4-TCP-SPF-2	Src/Dst ports after session started	Success
4-TCP-SPF-3	Source IP with wrong ports	Success
4-TCP-SPF-4	Invalid Source IP with correct ports	Success
4-TCP-SPF-5	Sequence replay	Success
4-TCP-SPF-6	Shotgun detection of hijack attempts	Success
4-TCP-RSB-1	Rule set bypass attacks against rules	Success
4-TCP-RSB-2	Random TCP data to bypass rules	Success
4-TCP-RSB-3	Information acquisition	Success
4-TCP-RSB-4	Port scanning detection	Success
4-TCP-RSB-5	Anomaly detection	Success
4-TCP-RSB-6	Connection prediction	Success

Tests	Description	Status
4-UDP-DOS-1	Invalid source addresses and ports	Success
4-UDP-DOS-2	Invalid destination addresses and ports	Success
4-UDP-DOS-3	Sending random data	Success
4-UDP-DOS-4	Blocking IP/MAC addresses	Success
4-UDP-DOS-5	Invalid Lengths	Success
4-UDP-DOS-6	Invalid CRC values	Success
4-UDP-SPF-1	Source addresses and ports	Success
4-UDP-SPF-2	Destination addresses and ports	Success
4-UDP-SPF-3	New/Open state values w/ all ports	Success
4-UDP-RSB-1	Rule set bypass attacks against rules	Success
4-UDP-RSB-2	Random UDP data to bypass rules	Success
4-UDP-RSB-3	Port scanning detection	Success
4-UDP-RSB-4	Anomaly detection	Success
4-UDP-RSB-5	Connection prediction	Success
7-DNS-DOS-1	Incorrect DNS packets	Success
7-DNS-DOS-2	Invalid field contents	Success
7-DNS-DOS-3	False DNS records	Success
7-DNS-DOS-4	Shotgun DNS packets	Success
7-HTTP-DOS-1	Multiple GET requests	Success
7-HTTP-DOS-2	Multiple GET requests w/ large buffers	Success
7-HTTP-DOS-3	GET to grab false items	Success
7-HTTP-DOS-4	POST w/ large buffers	Success
7-HTTP-DOS-5	POST w/ multiple responses	Success
7-HTTP-DOS-6	PUT w/ large buffers	Success
7-HTTP-DOS-7	PUT w/ file	Success
7-HTTP-DOS-8	PUT w/ ASCII file	Success
7-HTTP-DOS-9	CONNECT random ports/attempts	Success
7-HTTP-DOS-10	HEADER info w/ large buffer	Success
7-HTTP-DOS-11	HTTP keywords and large buffers	Success
7-HTTP-DOS-12	BINARY data in HTTP packets	Success
7-HTTP-DOS-13	(-binary) data in headers	Success
7-HTTP-HIJACK-4	XSS redirection (UNICODE)	Success
7-HTTP-BF-1	Massive GET/POST requests	Success
7-HTTP-BF-2	Anomaly detection	Success
7-HTTP-DT-1	Try “...” and other traversal methods	Success
7-HTTP-DT-2	UNICODE traversal methods	Success
7-HTTP-DT-3	Mixed strings	Success
7-SMTP-EXP-1	Send mail Header Overflow	Success
7-SMTP-EXP-2	Microsoft Exchange Heap Overflow	Success
7-SMTP-EXP-3	IA Web Mail 2.x Remote Exploit	Success
7-FTP-EXP-1	WFTPD Overflow	Success
7-FTP-EXP-3	ProFTPD 1.2.7 Remote root	Success
7-FTP-EXP-4	Wu-FTPD Format String	Success

Table 1 continued, page 7

Table 1 continued from page 6

Table 1

Tests	Description	Status
7-FTP-EXP-5	Wu-IMAPD Overflow	Success
7-FTP-EXP-6	ProFTPD ASCII File Overflow	Success
7-FTP-BOUNCE-1	FTP Bounce Attack (Dos)	Success
7-TELNETD-EXP-1	x86/BSD TelnetD Remote Root	Success
7-ORACLE-EXP-1	Man-In-The-Middle Oracle Attack	Success
7-WORM-1	MyDoom	Success
7-WORM-2	Apache Worm (1.3.2.x)	Success
7-WORM-3	Email Worm (worm.cpp)	Success
7-WORM-4	CodeRed	Success
7-WORM-5	SQL Slammer	Success
7-SNMP-EXP-1	MS Windows 2K SNMP Printer DoS	Success
7-SNMP-EXP-2	Oracle DBSNMP Overflow	Success
7-SNMP-EXP-3	Cisco Malformed SNMP DoS	Success
7-SNMP-DOS-1	Overlong OID	Success
7-SNMP-DOS-2	Incorrect fields	Success

Appendix two: Report data files

This report contains four directories.

\Server attack captures

This directory contains real and example test data to demonstrate the type of packets that were sent during the course of our testing of the Sidewinder G2 Security Appliance.

The packet captures were performed with Ethereal™. This packet analyzer is freely available which allows easy sharing of packet captures without requiring the purchase of new software. Some packet captures are rather large and require a machine with at least 512mb of RAM to load some of them into memory.

This protocol analyzer can be downloaded from <http://www.ethereal.com>.

Many of the packet captures within this directory may contain little to no data. All attacks that were performed through the appliance attempting to route or bypass rules defined within the appliance were stopped and therefore some packet captures contain only internal network traffic while the packet capture

was occurring. In addition, as a result of the sheer volume of traffic sent to the external and internal networks it was not possible to save the majority of the traffic history. For that reason, we have included example packet structures that were used in many of the tests.

\Spoofed example formats

The packet formats are literal dumps of base packet formats that are used by our packet spoofing and generation program. The literal value of a field changes and can be defined or randomized within our application. Though the values may change, the structure itself is what is important. Please note that the values reported are default values and do not reflect the values sent at runtime. These are structured patterns to build packets with.

For the first few denial-of-service tests we did not use correct packet checksums, either within IP, UDP or TCP. It was apparent that it was not possible to get malformed packets, specifically checksum-based malformed packets, through the appliance. After making changes and almost always calculating correct checksums we began getting packets where we wanted them to go as long as they didn't violate the appliance rules.

The Sidewinder G2 Security Appliance does a perfect job blocking checksum based malformed packets from routing based on our tests.

\Attacker exploit captures

This directory contains captures of exploits as they were transmitted from the hacker's system. They are named according to the exploit name defined in the Coverage of the Agreement.

\Reports

This directory contains this document and the executive summary report.

Appendix three: Network layout

Security Appliance

Appliance Domain: fw.bhtesting.com

Internal Nurb:

172.31.33.1
255.255.255.0

External Nurb:

10.10.0.1
255.255.255.0

Attackers Machine (we also used the server as attacker in some cases)

Windows 2000 or Red Hat Linux based upon the requirements to run tools.

10.10.0.2
255.255.255.0

10.10.0.3
255.255.255.0

Server Machine

172.31.33.2
255.255.255.0

Windows Advanced Server SP4

DNS

FTP

HTTP

SNMP

SMTP

SQL 2000

Appendix four: Interpreted anomalies

There were several anomalies that occurred and need to be mentioned when reviewing the packet captures; it is understood that these issues are not failures or problems with the appliance.

1. We had an issue with our DNS server being off when in a transparent configuration. Some captures contain DNS requests and ICMP failures in communication between 172.31.33.1 and our server at 172.31.33.2 that occurred in some packet captures. This issue was resolved and is included for completeness.

2. Often we had a window open to the Admin Console from the server viewing real time audit data. Sometimes we left this open and sometimes we did not. Several times when we thought all windows were closed, the process remained and continued sending data. This counts for much of the traffic to port 9003 from the server @ 172.31.33.2 to the appliance @ 172.31.33.1. The reason this was an issue was the audit process on the appliance was sending ACLDENY audit information to the server while we were performing a denial-of-service attack with DENY ALL rules set on the appliance. We saw traffic in the internal network when the test was running and we thought there was incorrectly routed traffic. However, this was due to the audit information being sent from the appliance to the audit process that did not close and was not incorrectly routed traffic.

Appendix five:

These planned exploits were not completed due to source code availability and schedule.

Email Worm (worm.cpp) corrupted

MS Windows 2K SNMP Printer DoS unrecoverable compiler issues

The *Man-In-The-Middle Oracle Attack* exploit, if possible to be provided by Secure Computing Corporation, was not received and therefore we have no results for that test.