Global Information Assurance Certification Paper

GCIH ADVANCED INCIDENT HANDLING AND HACKER EXPLOITS CURRICULUM PRACTICAL ASSIGNMENT
DNS Query Storm Exploit
Version 2.0

By

Tim O'Neil
April 16, 2002

Preparation

Part 1

Background

It was during the on-site interview for the position of Information Security Manager that my soon-to-be- boss mentioned that they were encountering a series of Denial of Service Attacks. It seemed that the company was undergoing an attack on their DNS infrastructure that was causing them to have to reboot their nameserver, located on the firewall load balancer as Recursive DNS queries where flooding the machine.

The load balancer was responsible for ensuring that the twin firewalls did an equal share of work. It also served as a nameserver DNS Server, used to resolve outside DNS request delegated to it from the main DNS server located on the firewall.

I thought little more of the incident until I was hired on in the position. I arrived a short week later to find that the company was in the midst of another attack, having again suffered a DoS incident against the same device.

"The event that shut us down today is recorded as a "**DNS non-Internet lookup**" according to the Network Intrusion Detection Device" explained one of the Level III Network Engineers, a specialist with close to 20 years experience in IT. The IDS was a Black Ice Network Intrusion Detection System.

He went on to say that our IDS vendor described the incident as a DNS attack, but he and others in the office weren't so sure. "Perhaps it is a misconfigured DNS Server as the Internet is known to be full of them," he ventured. This was definitely a question that we would be pondering over the coming weeks. For the immediate future however, it mattered little, as the results were the same, customers couldn't connect to our Internet or Extranet sites.

Were the incidences real DoS attacks? Yes, they were causing our network to experience a denial of service. Whether the incidents were caused by a seasoned hacker, a pimply faced sixteen-year-old script kiddy or several misconfigured DNS server was an issue that remained to be answered. But that wasn't as important as resolving the situation by configuring the DNS to defend itself.

Throughout this paper I have attempted to enumerate the characteristics of this incident that lends credibility to my belief that we experienced not just a DoS, but a Distributed Denial of Service Attack. Additionally, the possibility exists that we were not the

intended final victim in the attack, but possibly an intermediate victim used to launch an attack against the true intended target.

Protocol Description: DNS

DNS, the Domain Name Service, translates from domain names used by people to the corresponding IP addresses required by all network software. Data is stored in a distributed database where each nameserver is responsible (authoritatively) for its own piece of the naming tree. Delegation of authority occurs via NS (nameserver) records that must be consistent between parent nodes and children in the naming tree.

The DNS mechanism is made up of 2 essential agents:

The resolver that is the agent responsible for asking a DNS question. For example, a machine that needs to know the IP address of www.company.com uses its' resolver to ask the question "what is the IP address of www.company.com".

(1) The name server- the agent that is responsible for answering a DNS question. This is the agent present in DNS servers. When asked a question like "what is the IP address www.company.com", the name server answers to the best of its ability.

All basic Internet hosts and TCP/IP stacks contain the resolver. DNS Servers on the other hand, contain both a resolver and a name server. The resolver is necessary in a DNS server in case it is asked a question it cannot answer. The resolver gives the DNS server a way to find answers to questions it doesn't know the answer to. However, the DNS server resolver is a bit more intelligent than a normal host resolver is. Therefore, there are two types of resolvers: client resolvers and server resolvers. Still the basic purpose of a resolver is to ask a question.

There are also two kinds of DNS questions or queries that can be asked within the DNS mechanism. Remember that an example of a question is "what is the IP Address of www.company.com"

1. Iterative: An iterative query can be answered with an absolute answer or referral. This is like posing the following question: " I need to know the IP address of www.company.com. If you know the answer great! If not, can you refer me to someone who knows better than you."
2. Recursive: A recursive query must be answered with an absolute answer. This is like posing the following question: I need to know the IP address of www.company.com. If you know the answer, great! If not, please find the answer for me and let me know. As I stated above, the flag for whether this query is a Recursive Query or not is set to 1 for Yes. Therefore, the query is recursive and therefore, implicitly more load intensive for a recipient DNS Server than an iterative query as the latter may only figuratively shrug its shoulders if it doesn't know the answer to a query. The

Recursive query recipient must do everything in its power to find the answer and this includes querying other DNS servers[i].

Preparation

The company had an ad hoc incident response team in place. The Level III Network Engineers had decades of experience in the field and they knew the network. They both had contributed to its construction and knew the individual configurations of all its components. They were both technically proficient with multiple routing technologies and firewall design. Not to mention that they both had security certifications; one had a CISSP, the other a GIAC Security Essentials Certification.

One of the engineers had supervised the installation of the BlackICE Intrusion and detection system. The other had designed and installed the redundant firewall configuration, including the fail-over loadbalancers that hosted the victim DNS servers.

There was a Security Policy in rough draft form, but it was as yet unpublished. The following procedural components were in place:

- Warning Banners advised all users of the following:
    - Access to the systems was limited to company authorized employees
    - Access was limited to authorized activities
    - The information and resources are the property of the company
    - The system may be monitored
    - Misuse or unauthorized access may lead to civil or criminal prosecution

Additionally, there was an Electronic Communication policy in place that each employee was required to sign. Since we were sure that the attack came from the outside, there was little applicability for that policy, but it is important to note that it was in place, since according to current estimates, approximately 50% of computer related incidence are the fault of the "trusted insider."

Unfortunately, there was no standard operating procedure for incident handling within the organization. Once the incident had been detected, one of the Network Engineers contacted SANS and obtained an Incident Handling Guide. The team used the guide as their bible to facilitate a favorable resolution to the incident.

The Director of IT Infrastructure was a hands-on manager with more than a cursory understanding of the networking technologies in his department. He was detail oriented, a cautious thinker, and had considerable technical depth. He assumed the mantle of leadership without hesitation. I joined the team late on March 24th, but was able to get in on the Eradication, Recovery and Lesson Learned Phases of the Process.

The wider team consisted of the Vice President of IT, and representatives from both Corporate Communications and the Legal Department. Corporate Communications assisted in the preparation of a statement that was to be given out to our strategic and other business partners in the event that they were unable to connect to our site.

A conscious decision was made to notify our partners and B to B customers that the company was experiencing intermittent Internet outages and was working to resolve the problem. This addressed the issue at hand; if the situation got more serious the specifics of the notification could be revised.

The legal department advised upper management on the legal implications of notifying the authorities of the incident, allowing them to make the best business decision in the matter. The company served a critical business function, but a conscious decision was made not to notify law enforcement, at least until the incident had been resolved to a point where the attackers could no longer shut down access to our Internet based resources.

A notification to the media and general public was also considered, but also rejected. A myriad of reasons factored into that decision, but the short answer is that such a notification would have benefited no one. The safety of the public was not a concern and our business could be hurt by such a disclosure.

Upper Management was kept constantly informed of new incidences and the on going effort to resolve the problem. Management was kept informed, but trusted in the ability of the task organized Incident Handling Team to favorable resolve the problem without their active involvement.

Most of our business partners that used our site that were effected by our outage were small businesses with no incident response capabilities. We deemed that it be important that they be notified of the outage and be told nothing more for the time being. Should the situation increase in severity, a decision would then be made to give them further information.

Identification

According to DNS Measurement at a Root Server by Nevil Brownlee:

> **The root of the DNS distributed database is managed by 13 root nameservers. We passively measure the performance of one of them: f.root-servers.net**
>
> **These measurements show an astounding number of bogus queries: from 60-85% of the observed queries were repeated from the same host within the measurement interval. Over 14% of a root server's query load is due to**

> **queries that violate the DNS specification. Denial of Service attacks using root servers are common and occurred throughout our measurement period (7-14 Jan 2001). Though not targeted at the root servers, DoS attacks often use root servers as reflectors towards a victim network.[ii]**

So we have some research material that indicates DNS are often victims of DoS attacks.

The exploits described during Incident Handling concerning DNS centered around Cache Poisoning of DNS Servers, but precious little about a DoS directed against a network's external DNS servers was presented (unless I missed that instruction).

I knew from my interaction with others in the field that such attacks were common, because the entire Internet depended on DNS for name resolution and therefore DNS servers had to be located in a very accessible and vulnerable location.

Were there other DNS Exploits and were they available? Our friends at packetstorm had a few. Specifically, snoof.tar.gz which the site describes as a " A DNS spoofer based on ADM's "ADMsnOOfID" code. This, according to the site "has been almost completely recoded, for better performance, BIND 8 compatibility, and user defined TTL. By Doc_Chaos [RoC]." I found the user defined TTL to be very interesting, yet these exploits did not exactly describe our situation.

http://www.cotse.com/name.htm also listed several DNS Mass Query Exploits ready for download including dnsscan and BIND-496.c. There was also the program "jizz" which colorful name aside, is a very effective DNS Cache poisoning.

None of the exploits seemed to exactly fit the characteristics of the incident we were experiencing. There was also dnsloop.tar.gz, which fits some of the characteristics of our DOS situation, although we were not using TCPdump.

Finally, after much searching, I was rewarded with two documents detailing a very similar situation to the one we had been experiencing. One of the documents was found on CERT.ORG and the other was on the AUSCERT site. What follows is a description of how the exploit might have been intended to work:

Denial of Service Attacks using Nameservers

Overview

Intruders are using nameservers to execute packet-flooding denial of service attacks.

Description

We are receiving an increasing number of reports of intruders using nameservers to execute packet-flooding denial of service attacks. The most common method we have seen involves an intruder sending a large number of UDP-based DNS requests to a nameserver using a spoofed source IP address. Any nameserver response is sent back to the spoofed IP address as the destination. In this scenario, the spoofed IP address represents the victim of the denial of service attack. The nameserver is an intermediate party in the attack. The true source of the attack is difficult for an intermediate or a victim site to determine due to the use of spoofed source addresses.
Because nameserver responses can be significantly larger than DNS requests, there is potential for bandwidth amplification. In other words, the responses may consume more bandwidth than the requests. We have seen intruders utilize multiple nameservers on diverse networks in this type of an attack to achieve a distributed denial of service attack against victim sites.
In incidents we have seen as of the date of publication, the queries are usually crafted to request the same valid DNS resource record from multiple nameservers. The result is many nameservers receiving queries for resources records in zones for which the nameserver is not authoritative. The response of the nameserver depends on its configuration.

- **If the target nameserver allows the query and is configured to be recursive or to provide referrals, the nameserver's response could contain significantly more data than the original DNS request, resulting in a higher degree of bandwidth amplification.**
- **A target nameserver configured without restrictions on DNS query sources may not log malicious queries at all.**
- **If the target nameserver is configured to restrict DNS queries by source, and the source IP address is not allowed to make queries, the nameserver's response will be a reject message with little to no bandwidth amplification. Also, the nameserver can log the malicious queries. An example syslog entry looks like this:**

Apr 27 14:26:12 intermediary.example.com named[pid]: unapproved recursive query from [10.1.2.3].udp-port for resource.example.net

In this example, the IP address "10.1.2.3" represents the victim of the denial of service attack. The name "intermediary.example.com" represents an

> **intermediary nameserver used in the attack. The name "resource.example.net" represents the DNS resource record being queried in the DNS request. Some reports we have received indicate logging malicious DNS queries at a rate as high as 5 per second during an attack.**
>
> **The intermediary nameserver may receive packets back from the victim host. In particular, ICMP port unreachable packets may be returned from the victim to the intermediary in response to an unexpected UDP packet sent from the intermediary nameserver to the victim host.**
>
> **Impact**
>
> **Sites with nameservers used as intermediaries may experience performance degradation and a denial of DNS service as a result of an increase in DNS query traffic. It is also possible to experience higher bandwidth consumption and a bandwidth denial of service attack on the intermediary nameserver's network. Victim sites may experience a bandwidth denial of service attack due to a high volume of DNS response packets being forwarded by one or more intermediary nameservers.**[iii]

The preceding article stated that the syslog servers had recorded rates of up to 5 malicious queries a second, whereas we were receiving them at a rate of over 110 a second—definitely indicative of a multi-server attack. Additionally, the article mentioned authoritative queries that should not be received by a low-level root server—this was definitely similar to our situation.

That report had a link for a similar report for an AUSCERT advisory that got more in-depth as to possible defenses to the action and provides special instructions for ISPs to filter out this attacked based on installing access rules on their routers which would half filter; that would drop any pack from an IP address not within the range of addresses assigned to that subnet:

> **There are three parties: the target, the traffic multiplying DNS servers (amplifiers), and the attacker. Any platform connected to the Internet may be the target of the denial of service. Service is denied by occupying all link bandwidth with responses to bogus DNS queries and potential ICMP port unreachable responses to these bogus responses. Any DNS server may be used to multiply the denial of service attack. Usually several DNS servers on networks with good bandwidth to the target are required to effectively attack the target, however the same effects can be achieved by using a larger number of amplifiers with smaller bandwidth capabilities. The attack is launched from a remote location with moderate bandwidth to the amplifiers.**

IMPACT: Small DNS queries are sent from the attacker to each of the DNS servers. These queries contain the spoofed IP address of the target. The DNS servers respond to the small query with a large response. These responses are routed to the target, causing link congestion and possible denial of Internet connectivity. If the number of DNS queries from the attacker is large, then the traffic may congest the DNS server's Internet link or degrade the DNS server's response time. Although no different in principle from ICMP ECHO ("ping") flooding, DNS query traffic cannot be traffic-shaped by network routers without greatly inconveniencing legitimate users. Information regarding this vulnerability has been made publicly available. A tool to exploit this vulnerability was posted to the BUGTRAQ mailing list on 30 July 1999 by smaster@sail.it in message <199907310000.AA154206596@sail.it>. AusCERT members have observed this tool in use against hosts on their networks.

SOLUTION: Since this attack relies upon spoofed source IP addresses, source address checking by ISPs originating traffic is the only means to entirely defeat this form of denial of service attack. Appendix A of CERT advisory CA96.21 "TCP SYN Flooding and IP Spoofing Attacks" gives more information on configuring networks to defeat IP address spoofing. That advisory is available from: http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html Additional information may also be found in RFC2267 "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing". That RFC is available from your local RFC repository including: ftp://ftp.auscert.org.au/pub/mirrors/ftp.isi.edu/in-notes/rfc2267.txt

WORKAROUND: The current tools and attacks are very straightforward and administrators can prevent their DNS servers from being used as amplifiers by configuring their servers to answer queries from unexpected sources with a small REFUSED response rather than a much larger name resolution response. [In the discussion below, "trusted" sources are defined as hosts for which the DNS server provides recursive DNS name resolution. These hosts would usually lie within an ISP's or enterprise's network. These hosts usually have the DNS server listed in a configuration file such as /etc/resolv.conf or supplied to it in a PPP, DHCP or BOOTP response.] For the purposes of refusing queries from unexpected sources, DNS queries directed to a particular name server can be categorized into the following types. For each type of query a typical access control configuration is given. In addition, we suggest controls for zone transfers. While restricting access to zone transfers is not directly related to the denial of service attack described in this alert, it may provide additional security for some sites. (1) QUERIES FOR ANY NAME. Allow queries from "trusted" sources only. Allow no zone transfers. (2) QUERIES FOR NAMES IN PRIMARY ZONES. A "primary

zone" is a zone for which a server has the DNS master file described in RFC1035 and the server is one of the name servers that has been delegated the domain. Allow queries from all sources. Allow zone transfers from official and stealth secondaries. (3) QUERIES FOR NAMES IN OFFICIAL SECONDARY ZONES. An "official secondary zone" is a zone for which the server has a zone transfer of the DNS master file and is one of the name-servers that has been delegated the domain. Official secondary zones exist to add robustness to the Domain Name System. Allow queries from all sources. Possibly allow zone transfers from official and stealth secondaries. (4) QUERIES FOR NAMES IN STEALTH SECONDARY ZONES. A "stealth secondary zone" is a zone for which the server has a zone transfer of the DNS master file but is not one of the name servers that has been delegated the domain. Stealth secondary zones are often used to add performance to DNS resolution, especially at sites reachable only across slow wide-area network links or on machines containing DNS-intensive applications such as e-mail. Allow queries from "trusted" sources only. Allow no zone transfers. It may be administratively convenient to allow queries from all sources, as this minimizes the risk of outages if official secondary zones and stealth secondary zones are confused or if all the users of the stealth secondary zone are not known. If queries are limited to "trusted" sources only, then a careful eye should be kept on the DNS server log. An exception to the guidelines in (2) to (4) above is that within the configuration of each DNS server a sub-domain cannot service a smaller range of query sources than its parent domain. If a DNS server allows queries from any source for the domain "example.com", then it must also allow queries from any source for the delegated sub-domain "instructive.example.com". It is administratively useful to allow DNS zone transfers between all primary and secondary DNS servers. This eases the debugging of zone transfer faults. Similarly, allowing DNS zone transfers to a limited number of hosts used by network administrators may also be convenient. Allowing zone transfers to all "trusted" users may be too trusting in environments such as Internet service providers or universities. "Stealth primary zones" also exist. As these are mainly used inside firewalled environments, it is not useful to describe their configuration in this document.

LIMITATIONS: There are obvious limitations to the workarounds presented in this alert. As stated previously, the only solution to this problem currently is that all sites implement source address checking to prevent packets with spoofed IP addresses leaving their networks. Nonetheless, these suggestions will assist in mitigating current forms of the attack and may provide some additional security. [iv]

I came to the conclusion that yes there were readily available DNS exploits in the form of mass query programs that at least on the surface, fit the description of the incidences we were experiencing. That would due for now. The team would proceed as if we had identified the incident as a DoS.

Part 2 - The Attack – The Redundant Firewall Configuration

Our network had a redundant firewall array (See Figure 2) configured so that two Raptor Proxy Firewalls worked in a loadbalanced array to manage connections to the Internet and Extranet sites. Two Linkproof loadbalancers handled the loadbalancing of these firewall connections. The four port loadbalancers that were arrayed redundantly, with one unit in a hot standby mode. The units maintained a heartbeat connection via a network connection. In theory, if one loadbalancer shut down, the other would step in and take the place of the failing box.

A series of webservers that frontended for a large data center database repository was located in our DMZ. The site was protected through an SSL encrypted log-in and session, but service denial to the site was a major concern. Monitoring the interior and exterior networks were two Network Intrusion Detection Devices with six sensors arrayed at critical points throughout the network.

Apparently, we had allowed our DNS server to answer recursive requests! Meaning that the server was forced to seek an answer to each and every query it received. A fairly labor intensive endeavor. Conclusion: We were indeed the victims of a DNS based DDoS Attack. But was that the entire story?

How the Exploit Worked:

It seems that our DNS server had received thousands of recursive Start of Authority (SOA) queries within the period of less than six minutes. Apparently, this deluge of queries, (over 39,000 during one incident) had caused the client table to fill up and the box had subsequently stopped responding to new queries. The huge number of queries was stored in the loadbalancer's client table, which is stored in volatile memory. The client table and hence the memory would fill up and the box would freeze. Any new connection attempts would then fail.

The firewall load balancers were arrayed redundantly in hot standby, as opposed to the redundant firewalls, which did an equal share of the filtering work, one load balancer stood on in hot standby, ready to receive a come on line in the event of a failure of the primary unit.

The stand-by firewall load balancer was connected via a network connection heartbeat to the primary box. Should the primary fail the heartbeat signal would go silent and the standby load balancer would take over for the failing box.

During all of the recent DoS attacks against the Load balancer however, the first box would subsequently fail, but the second box would apparently fail to take over. The reason for this failure remained unclear until we realized the second box probably failed for the same reason as the first: It was configured to do a hot cutover, without a loss of connectivity to the clients. This meant that the client table is mirrored on the redundant box. Since the amount of memory is the same on both boxes, we surmised that a similar failure was taking place during the cutover attempt. Over the course of the event we experienced the same incident on five different occasions, as follows:

- Events occurred on Feb 21, Feb 27, March 8th & 25th (twice) of 2002
- 196 minutes of total downtime
- Thousands of DNS Recursive Queries Packets over the course of minutes, usually not more that 5-6 minutes until the DNS server stopped responding.
- DNS in the load balancer stopped responding
- Reboot of the load balancer corrected the problem

The downtime was caused by the fact that it took us time to discover the DoS, and time to respond, but the remedial action was very quick to execute.

Below is a copy of the IDS records from our probe relating to the most current incident. Notice the amount of packets that entered the DNS server over a relatively short period of time. It appears that we were the victims of a DDoS Attack. As the total amount of queries received equaled 39,555 from six different IP Addresses over the course of approximately 5:09 minutes. (The IP Address field has been deleted in an effort to protect the innocent and the allegedly guilty parties).

Signature of the Attack

The outages occurred on the following dates and times:

Date Time
Feb 27: 23:48 - Feb 28 00:06
Mar 12: 10:25 - 10:45
Mar 25: 7:06 - 8:42
Mar 25: 9:22 - 10:24.

Below is a graph detailing the amount of packets received during one of the March 25 incidents:

Date	Time	No. Packets
2002-03-25	08:06:01	1,360
2002-03-25	08:05:56	4,838
2002-03-25	08:05:56	5,664
2002-03-25	08:05:56	2,919
2002-03-25	08:05:56	4,436
2002-03-25	08:01:48	2,945
2002-03-25	08:01:45	5,621
2002-03-25	08:01:45	4,891
2002-03-25	08:01:43	4,367
2002-03-25	08:01:29	595
2002-03-25	08:00:55	598
2002-03-25	08:00:52	1,321

The "DNS non-Internet lookup" events which caused outages

Source IP: XX8.142.192.66
Destination: XXX-redirect-2.mycompany.com (XXX.XXX.XXX.X)

Delta Time DNS Query
+66.078126 DNS Standard query SOA net
+66.171876 DNS Standard query SOA edu
+66.234376 DNS Standard query SOA edu
+66.250000 DNS Standard query SOA org
+66.265626 DNS Standard query SOA edu
+66.296876 DNS Standard query SOA edu
+66.312500 DNS Standard query[malformed packet]
+66.343750 DNS Standard query SOA usc.edu
+66.406250 DNS Standard query SOA net
+66.312500 DNS Standard query[malformed packet]

It seemed to us that we were being inundated with requests for information as if we were the authoritative server for the zone and that our nameserver was simply choking on the requests. We checked the source of the attack through a "Whois" search on the ARIN "Whois" site and the range of addresses differed during each separate attack.

Each attack followed the same basic formula: A bunch of separate computers, some identifiable as DNS servers would inundate our DNS server with thousands of queries in the space of a few minutes causing the DNS Server to become unresponsive and the hot standby loadbalancer to fail as well. New connections to the Internet would be lost.

This appeared to be a Distributed Denial of Service Attack. The IP addresses differed on each attack, but the geographic point of origin of the attacks appeared to be Australia and Asia Pacific. Of course this meant little as no self-respecting hacker used his own computer to execute an attack. He simply used poorly configured servers regardless of the location. These intermediate victims would be compromised and turned into zombies so they could be used as platforms to launch an attack.

The case is still being investigated and we are considering using the SANS Report Back Program to notify the ISP of the problem, but no final decisions have been made as of yet.

The Packet

- The flag was set for a Recursive query. That meant that if the DNS server was set to accept recursive requests we had a problem.

- The malformed nature of the packet is still a mystery. Initial speculation led us to believe that this packet was some type of exploit disguised as a DNS query designed to exploit a buffer overflow, but that is merely speculation.

- In this case the port that the request or query originated from is a seemingly random high port. According to applicable RFC's, the source port is usually a high numbered random port, so this is not a suspect characteristic.

- The header is 20 bits in length while a standard header is 16 bits in length.

- The Time to Live for the packet, may or may not be suspect. Standard TTL for a TCP/IP packet is 255. Each router hop decrements the TTL by one. This is in effect a hop count. Several DNS exploits allow the hacker to set the TTL of the packet to any number in order to better mask the exploit's point of origin.

- SOA (Start of Authority) requests: I also see no reason for external entities to be making SOA requests from us. This should be blocked as well. Most of the offending packets are SOA request for top level domains.

(destination port = 53) and (offset x36 = 03)

Below is a copy of the actual packet:

```
Frame 25304 (62 on wire, 62 captured)
   Arrival Time: Mar 25, 2002 07:20:45.296874000
   Time delta from previous packet: 0.015624000 seconds
   Time relative to first packet: 783.656250000 seconds
   Frame Number: 25304
   Packet Length: 62 bytes
   Capture Length: 62 bytes
Ethernet II
   Destination: 00:00:b0:80:66:c1 (ns2.mycompany.com)
   Source: 00:10:7b:67:2c:01 (Cisco_67:2c:01)
   Type: IP (0x0800)
Internet Protocol, Src Addr: XXX.142.1.242 (198.142.1.242), Dst Addr:
ns2..mycompany.com (XX.170.144.2)
   Version: 4
   Header length: 20 bytes
   Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
   Total Length: 48
   Identification: 0x8000
   Flags: 0x00
      .0.. = Don't fragment: Not set
      ..0. = More fragments: Not set
   Fragment offset: 0
   Time to live: 203
   Protocol: UDP (0x11)
   Header checksum: 0xd58f (correct)
   Source: XXX.142.1.242 (198.142.1.242)
   Destination: ns2.mycompany.com (xx.xxx.144.2)
User Datagram Protocol, Src Port: 20226 (20226), Dst Port: domain (53)
   Source port: 20226 (20226)
   Destination port: domain (53)
   Length: 28
   Checksum: 0x0000 (none)
Domain Name System (query)
   Transaction ID: 0xea22
   Flags: 0x0100 (Standard query)
      0... .... .... .... = Query
```

.000 0... .... = Standard query
.... ..0. .... = Message is not truncated
....1 .... = Do query recursively
....0 .... = Non-authenticated data is unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
[Malformed Packet: DNS]

[Hex Deleted]

The use of recursive mode is limited to cases where both the client and the name server agree to its use. The agreement is negotiated through the use of two bits in query and response messages: The recursion available, or RA bit, is set or cleared by a name server in all responses. The bit is true if the nameserver is willing to provide recursive service for the client, regardless of whether the client requested recursive service. That is, RA signals availability rather than use. Queries contain a bit called recursion desired or RD. This bit specifies whether the requester wants recursive service for this query. Clients may request recursive service from any name server, though they should depend upon receiving it only from servers which have previously sent an RA, or servers which have agreed to provide service through private agreement or some other means outside of the DNS protocol.[v]

Apparently, we had allowed our DNS server to answer recursive requests! Meaning that the server was forced to seek an answer to each & every query it received. A fairly labor intensive endeavor. Conclusion: We were indeed the victim of a DNS based DDoS Attack, but was that the entire story?

Part 3 – The Incident Handling Process

That was my first day on the job. Ironically, I had just graduated from the SANS Incident Handling Course, yet all they had taught me had failed to coalesce into my gray matter. Lesson number 1 should have been an experienced Incident Handler always travels with his reference materials. All the SANS material detailing DNS and other exploits, was safely but uselessly secured in my household goods slowly making their way east to my new home.

Data from several sources was collected and copied to CDROM. These included Network Intrusion Detection Sensors and the two redundant Firewalls, as well as plugging in a console into the Internet connection so we could see the packets as they arrived into the dirty side of the network. We also had a Syslog server, but that was not yet in production.

From these sources we were able to gather copies of the packets, source IP Addresses, but not much else. The IP Addresses came from several different IP Addresses, so it appears we were victims of a Distributed Denial of Service Attack. It would have been nice to recreate the attack in a test environment, but that would take time that we didn't have. The source IP Addresses always changed after each attack, as we were adding the source addresses to that ACLs in the gateway routers after each attack. The hackers were simply moving or spoofing accordingly.

Containment / Eradication - Immediate Actions

Initially, the offending subnets were directed to the unused public address on each of the Internet routers, which basically directed them to a black hole. Static routes were used, as they don't require the router to examine each packet as an ACL would. The down side to using a static route is that packets are still let in, they just can not return. The attacker(s) would still be able to flood our DNS servers if they were persistent. I would imagine that they would notice that they don't get any form of unreachable message back and realize that they are blocked one-way.

On the positive side, they should realize we are taking measures against them and stop. On the other hand once they realize it is a one-way block, they could still try the same attack.

An ACL is much more effective, although that would drive our CPU utilization up. Since current CPU usage during peak hours was at a modest 20% a decision was made to add the offending IP addresses to ACLs on our Internet Gateway routers, specifically denying the address ranges. Of course, subsequent attacks came from different IP Addresses, which were in turn added to the ACL's.

Additionally, we added ACLS that would block any DNS 53/udp requests that had a source port lower than 1024 that is not 53. According to a security consultant we had consulted, DNS queries normally originated from a high random port, and that a DNS query that can from a port below 1024 that was not 53 were invalid.

At the same time, we blocked all SNMP traffic for other reasons independent of the current situation, but in response to the vulnerability inherent in having routers pass that traffic.

Tolerances/ Default Settings Lowered

- Connection tolerances lowered:
- Client table aging time-out lowered by 50% to 30 seconds
- Lowered connection time-out for DNS
- Syn time lowered blocked offending IP Addresses from connecting

- Minimum TTL - The minimum time-to-live value applies to all resource records in the zone file. This value is supplied in query responses to inform other servers how long they should keep the data in cache. The default value is 3,600 or one hour. This was lowered from the default to a TTL of 30 minutes.
- Syn time-out was lowered from 60 to 5 seconds.
- The client table timeout for DNS to 10 seconds on the DNS/firewall loadbalancers
- Recursive query refused flag set on all DNS servers

One of the things we believed was that this was a DDoS that simply overloaded the DNS Server with more request than it had the capacity to address and it simply overloaded the buffer. Additionally, since the requests were recursive, the DNS had to service the request.

Recursive queries: There is no valid reason for external entities to make recursive queries to our DNS servers. All of the attacking packets have the recursive bit set on. A decision was made to block all packets meeting the following conditions:

(destination port = 53) and (offset x2C = 01)

SOA (Start of Authority) requests: I also see no reason for external entities to be making SOA requests from us. This should be blocked as well. Most of the offending packets are SOA request for top level domains.

(destination port = 53) and (offset x36 = 03)

Evidence Collection

We believed the best course of action was to insure we had good packet traces of any future events. This, we felt would allow us to see exactly what is occurring. Unfortunately, initial attempts at proper collection proved to be elusive—the laptop we setup to capture console data had "write error" and did not record the log data we need.

The network intrusion detection nodes; BlackICE Sentries, were configured to do packet captures on a rotating file basis. Both sensors were configured to do 4MB captures over a rotation of 500 files (2GB of data per node). Both NIDS had about 20GB of space available for logging, although ideally, we should have been making pains to save all of the data rather than allowing it to be overwritten.

A Digitech network protocol analyzer was set up on a port mirrored to the first loadbalancer with a rotating capture buffer. If an event where to occur, this trace would have to be stopped quickly as the unit doesn't have much hard drive space to allocate to this.

Additionally, a laptop was set up to record via a telnet session with the loadbalancer in order to have the ability to record the events as they occurred to determine what was happening to the box. The results could be copied to a floppy disk from the laptop.

With the MAC address and the time from the unit, we can verify which unit was active through syslog records. This and the logfile may provide some valuable clues.

Many enterprising Incident Handlers use traceroute to trace the packet back to its point of origin. Some of the IP Addresses we collected from the Firewall log indicated the several of the suspect DNS queries originated from as far away as Australia. A nice graphic program called Neotrace from McAfee allowed us to trace back the source IPs, through numerous router hops. We managed to trace them back to their ISP, but it covered most of Asia, as well as Australia. So, although we did whittle the source down a bit, we were still along way from catching the hackers red-handed. ARIN "Whois" also provided information about the IP Address range. This might have also led us back to a host of zombies.

Lessons Learned

As of the date of the submission of this article, we have experience no further DDoS attacks. I believe that the immediate adjustments to the routers and DNS Servers solved our problem; at least for the immediate future. After the incident had been resolved to a point where we were no longer experiencing outages, I prepared and presented a briefing to our executive leadership that addresses the specific incidents and how we proposed to resolve the situation in the event of future attacks:

- Organization and Training of a Computer Incident Response Team and procedures
- Approval of capital expenditures
- Final tool selection and implementation
- Recurring vulnerability test of the network
- Continue development of defense in depth strategy throughout the network
- Proactive monitoring & report back on suspect activity

The second through fourth bullet points concern the deployment of specific DDoS defensive tools and firewall feature set IOS to address provide specific protection at the Internet Gateway and the DMZ. Additionally, we proposed a continuing vulnerability analysis to annually check for vulnerabilities.

- **Layered Defenses**

Defenses should be layered. Don't depend on a Firewall to be the sole response to an attack. Defenses should start at the farthest reaches of your network, which to me means at your ISP. Normally, this means practicing Due Diligence in ensuring they are doing everything possible to mitigate the threat. Then look at your Internet Router. Is it hardened? Consider the installation of a Firewall Feature Set. This can stop many attacks

at the Internet Gateway Routers. Cisco IOS Firewall Feature Set for the Internet router made good cost effective sense. Although it only has a limited number of attack signatures in its database, a properly configured Context Based Access Control (CBAC) could mitigate a host of DoS attacks at the farthest reaches of our network by simply dropping the packets. DoS / DDoS Mitigation Products:

Many company now make DoS Mitigation tools specifically designed to stop an attack in progress by recognizing the attack signature, either through a signature database or dynamically, through a learning ability. It will move to reset the connection, insuring it the suspect connection and the victim's machine are issued a reset. We quickly began to investigate these solutions. We were remiss in not insuring that we had an item in place to do this, but we won't make the same mistake again.

Also, as far as having a layered defense we thought that it made a great deal of sense to stop the attacks, or whatever they are as far outside the perimeter of the network as possible. Cisco IOS Firewall Feature set for the Internet router made good cost effective sense. Although it only has a limited number of attack signatures in its database, the Context Based Access Control CBAC feature made properly installed could mitigate a host of DoS attacks at the farthest reaches of our network by simply dropping the packets.

Default Setting

Default settings on any device whether exposed to the Internet or not are ripe playgrounds for Hackers. Any time a device is added to your network, a hardened configuration should be researched and implemented. There are off the shelf hardening for many operating systems. Others have to be painstakingly modified to ensure that there are no openings for hackers to exploit. Look specifically at services running, connection time outs and open ports.

As an Information or Network Security Manager, part of your job should be to research these issues and bring them up any time there is a new addition to the network infrastructure. Research the recommended settings versus the default setting for network routers, firewall etc. Review the rule set of the firewall and stay current on threats and what patches can be used to mitigate the threat.

- Memory & other resources: Remember that DoS attacks use resources, including memory and bandwidth. Insure that your configuration is robust enough to handle a sudden surge in the use of system resources, whatever they may be.

- Time-outs: Time-outs for connection Syncs or connection times should be analyzed, to determine what is the best (lowest) Time-out for the particular service or connection.

Over the long term, we came to the conclusion that DoS and DDoS attacks were going to increase in number and sophistication and to defeat the attacks we had to take a longer-term approach. New application type defense tools can be used as stand-alone units or in conjunction with products that sit on the "dirty side of the network, such as Internet routers, load balancers, Intrusion Detection Sensors, and firewalls. New tools can recognize attacks and dynamically update ACLs, or issue resets to suspect connections.

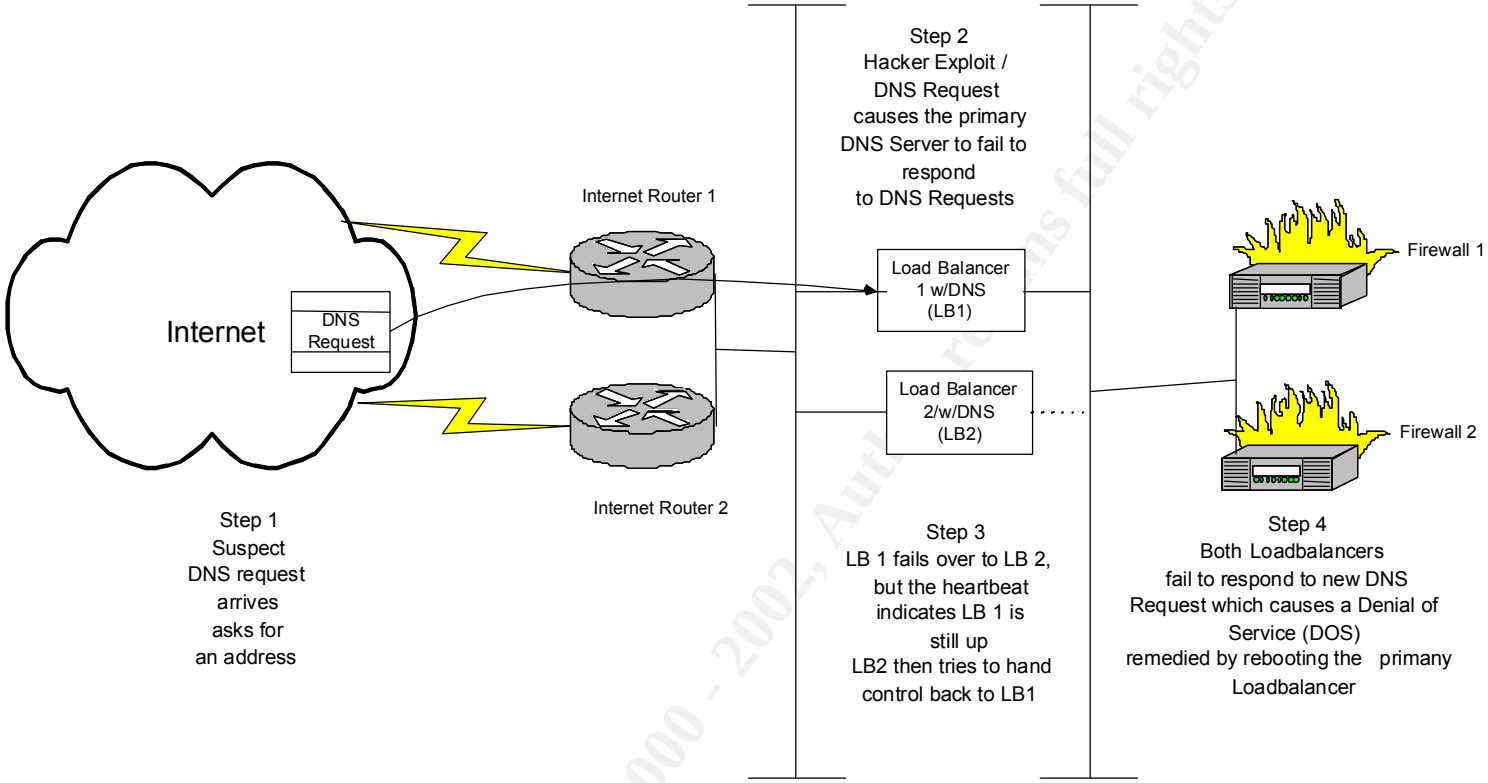
We selected a product that would allow expansion of our Business to Business Extranet site and also identify and reset suspect signatures that it could identify from an internal database that housed a library of known signatures, but that also allowed us to program out own.

Additionally, we organized a Computer Incident Response Team from within the network group that would have the responsibility for the implementation of a refined Incident Response Plan, so that the next time this happens (And we felt there would be a next time) we could better respond to the challenge.

Probe network perimeters for security holes at least annually. Had we probed for exploitable holes in our perimeter, the DNS configuration on the loadbalancer might have been discovered earlier.

The fact of the matter is that knowing the cause is secondary to resolving the problem to a point where we are safe from that particular incident. That is all we can hope for, because there is always another exploit on the pike, and there is no one who can claim to have eradicated all risks to their network. We learned some valuable lessons from the experience that allowed us to refine the way we do our job. Under the leadership of the IT Director, the two experienced Network Engineers resolved the incident in a timely and cost effective manner. I offered some input on a layered defense and documented what they did to resolve the issue.

Diagram of the Attack (Appendix 1)



References

[i] Request for Comments: 1034, Network Working Group's detailing Domain Name Service; http://asg.web.cmu.edu/rfc/rfc1034.html

[ii] DNS Measurement at a Root Server by Nevil Brownlee, The University of Auckland and CAIDA, SDSC,UC San Diego.

[iii] Denial of Service Attacks using Nameservers Updated: Monday, January 15, 2001 (changed RFC 2267 to RFC 2827/BCP 38) Date: Friday, April 28, 2000 CERT http://www.cert.org/incident_notes/IN-2000-04.html

iv A U S C E R T A L E R T AL-1999.004 -- AUSCERT ALERT Denial of Service (DoS) attacks using the Domain Name System (DNS) 13 August 1999 Last revised: 13 August 1999 ftp://ftp.auscert.org.au/pub/auscert/advisory/AL-1999.004.dns_dos

[v] Ibid

Last Updated: May 28th, 2017

Upcoming Training

SANS Madrid 2017	Madrid, Spain	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GA	May 30, 2017 - Jun 04, 2017	Live Event
Community SANS Virginia Beach SEC504*	Virginia Beach, VA	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Mentor Session - SEC504	Reston, VA	Jun 13, 2017 - Aug 01, 2017	Mentor
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Seattle SEC504	Seattle, WA	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS ICS & Energy-Houston 2017	Houston, TX	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Sacramento SEC504	Sacramento, CA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Ottawa SEC504	Ottawa, ON	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Des Moines SEC504	Des Moines, IA	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Annapolis SEC504	Annapolis, MD	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Phoenix SEC504	Phoenix, AZ	Jul 24, 2017 - Jul 29, 2017	Community SANS
Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Aug 07, 2017 - Aug 12, 2017	Community SANS
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event