BIND 8 Buffer Overflow in TSIG

On January 29, 2001, CERT released their advisory CA-2001-02. This advisory contained a listing and description of 4 new vulnerabilities in ISC's (Internet Software Consortium) BIND (Berkley Internet Name Domain) Server. One of these vulnerabilities, that of a buffer overflow in the transaction signature (TSIG), posed the greatest concern, as it allowed for a root exploit on all affected servers. The purpose of this paper is to describe how the exploit works; how to determine if your server is vulnerable; why it is so critical to deal with the exploit; and how one can combat the exploit.

Introduction to DNS

Before beginning, the reader should have at least a rudimentary working knowledge of DNS and how important it is to the Internet and networks in general. DNS (Domain Name System) allows for mapping of a name to an IP. DNS is a hierarchical, distributed database, which allows for local control over segments of the database. This allows for a more efficient means of administering the different domains. At the top of the DNS hierarchy is the root level, or ".". Next, are the Top Level Domains (TLD's). These are the com, edu, gov, org, net, and mil domains. Below the TLD's begin the delegated domains. For example, microsoft.com, hp.com, uga.edu, etc. Ownership of these domains is delegated to companies or organizations. These companies and organizations are then responsible for managing the hosts and sub-domains below their domain.

DNS is also based on the client-server architecture. The name server constitutes the server portion of the relationship while your workstation's resolver constitutes the client portion. Thus, when you point your browser to a particular website, an exchange occurs between your resolver and the name server(s) you have listed in your resolver. Your resolver will request information about a specific domain name and the name server will provide the IP address for that domain name.

Why is DNS so important? The underlying need for DNS is that humans are much better at remembering names than numbers…especially numbers like IP addresses . When DNS is compromised, whole sites can be affected and corporations can see their finances and reputations suffer, as Microsoft did the week of January 22, 2001. Also, because a majority of DNS servers run ISC's BIND, which is well-known software, it is important to secure these servers to prevent DOS attacks and possible root exploits like the one described below. The bottom line is that without DNS, navigating the Internet or even your company's LAN would become very difficult if not impossible.

For more information on DNS and previous exploits, please refer to the SANS (System Administration, Security, Networking) papers submitted by Ken

Athanasiou (DNS Remote Root Exploit) and Sinéad Hanley (DNS Overview with a Discussion of DNS Spoofing). URLs are available below.

The TSIG Exploit

The particular vulnerability that this paper describes is a result of the resource record TSIG, introduced as a part of DNS Security Extensions (DNSSEC) in BIND 8. It is described, in detail, in RFC 2845. TSIG, or transaction signature, allows for transaction level authentication using shared secrets and one way hashing. It can be used to authenticate dynamic updates as coming from an approved client, or to authenticate responses as coming from an approved recursive name server.

As stated above, CERT released an advisory on January 29, 2001 that dealt with 4 new vulnerabilities in various versions of BIND. Here is the excerpt dealing with the TSIG buffer overflow:

> CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND
>
> Original release date: January 29, 2001
> Last revised: February 02, 2001
> Source: CERT/CC
>
> **VU#196945 - ISC BIND 8 contains buffer overflow in transaction signature (TSIG) handling code**
>
> During the processing of a transaction signature (TSIG), BIND 8 checks for the presence of TSIG's that fail to include a valid key. If such a TSIG is found, BIND skips normal processing of the request and jumps directly to code designed to send an error response. Because the error-handling code initializes variables differently than in normal processing, it invalidates the assumptions that later function calls make about the size of the request buffer.
>
> Once these assumptions are invalidated, the code that adds a new (valid) signature to the responses may overflow the request buffer and overwrite adjacent memory on the stack or the heap. When combined with other buffer overflow exploitation techniques, an attacker can gain unauthorized privileged access to the system, allowing the execution of arbitrary code.

This particular exploit affects all versions of BIND 8 prior to 8.2.3, including the 8.2.3TXB versions of BIND.

A buffer overflow vulnerability is formed when programs accept more data from an outside source (in this case, from a TSIG request), than they can store in the requested memory buffer. The extra data will overflow into the portion of memory where commands are executed and will then be executed as a part of the program. In BIND's case, this is very dangerous because BIND is generally run as the user root. Thus, if an overflow happens, then the extra instructions will be executed as the user root.

Further, in the report put out by Covert Labs on January 29, 2001 in which they describe the vulnerability, it is noted that the overflow occurs in the initial processing of a DNS request. This means that the attacker does not need to control an authoritative DNS server. Also, the attack is not dependent upon configuration options which means you cannot change your BIND configuration to block the attack.

It has also been determined that to use the exploit, the attacker must adhere to these two qualifications: the number of bytes past the end of the buffer is limited in number; and, the values of those bytes are mostly fixed.

Normally, when a server receives a DNS message, BIND will process it as either a response or request. If the DNS message is a request, the BIND server will determine if the message is a query, iquery, update, or notification. If a request is sent to the BIND server running 8.2, BIND will examine the message before processing the request to see if there is a TSIG resource record. If such a record is found, but no corresponding security key is available on the DNS server, then an error is signaled, and BIND bypasses the normal request processing. At this point, BIND makes an incorrect assumption about the size of the buffer. BIND makes this assumption based on a normal request, but this assumption is invalid due to the error returned and the fact that the request was never processed. As a result, BIND is now vulnerable to a buffer overflow. For more detailed information on how the attacks work, please refer to the Covert Labs report, [COVERT-2001-01] Multiple Vulnerabilities in BIND (see URL below).

Determining a System's Vulnerability

In order to attack a system, the server must be running BIND 8.2.X prior to 8.2.3 REL. There are several ways to find out the version of BIND you are running. As a system administrator, you can query your server by running the following command from your DNS server as root:

```
/usr/sbin/ndc status
```

This will return something like this:

```
# /usr/sbin/ndc status
named 8.2.3-REL [Linux 2.2.18 i686] Sat Jan 27 05:56:43 UTC 2001
compiled by Hostmaster <hostmaster@site.com>
config (/etc/named.conf) last loaded at age: Fri Dec 22 17:00:28 2000
number of zones allocated: 512
```

```
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is ON
server is up and running
```

What you are looking for is the statement: named 8.2.3-REL. If it refers to a version prior to 8.2.3-REL or 9.1.0, then you should upgrade immediately.

You can also query the server remotely to see what version of BIND is running. This can be done from a linux/unix box with the BIND utility dig. The command run would be:

```
dig @<server> chaos txt version.bind
```

This will return something like this:

```
# dig @<server> chaos txt version.bind

; <<>> DiG 8.3 <<>> @<server> chaos txt version.bind
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:
0
;; QUERY SECTION:
;;      version.bind, type = TXT, class = CHAOS

;; ANSWER SECTION:
VERSION.BIND.          0S CHAOS TXT     "8.2.3-REL"

;; Total query time: 67 msec
;; FROM: <server> to SERVER: <server> 10.10.10.10
;; WHEN: Tue Feb  6 11:01:43 2001
;; MSG SIZE  sent: 30  rcvd: 64
```

If you are looking for DNS servers that you can test for this vulnerability, you can run a scanner, like nmap, and look specifically for servers listening on port 53. From there, you can run dig against the servers to see if they are affected. The command would be:

```
nmap –v –PB –I –p53 –oN [PWD]/logfile 128.210.0-255.0-255
```

This would scan the entire class B network of 128.210.0.0 for hosts with port 53 open and dump the results to a text file called logfile.

If you are a network administrator, I would highly recommend that you scan your network (obtain permission first!!!) to see what name servers are operating, and then test for the version of BIND to see what servers need to be upgraded. For more information about nmap and the implications of running it on your network, you can refer to the SANS paper written by Brent Deterding (URL below).

Importance of the Exploit

Why is this exploit so important to network and system administrators? Well, within a week of the release of the CERT and Covert Labs advisories, at least two known scripts existed in the wild to take advantage of the exploit. Should an exploit be successfully carried out, an intruder will compromise the DNS server with root access. This, in turn, could lay your whole network open to attack and compromise. Both your system and network are vulnerable to confidentiality, integrity, and availability attacks. The exploit will allow for the integrity of your DNS server to be compromised. From there, confidential files, like the /etc/passwd or /home directories will be open since the attacker will have root access. Also, trusted systems can be made known and open to attack. Finally, availability of your network and systems can be compromised if the attacker chooses to shut your DNS server down.

System administrators should be running a file system integrity tool, like *Tripwire* and should be keeping a close eye on both successful and unsuccessful logins to their server. *Tripwire* will help the system administrator determine if any important binaries or configuration files have been altered, or if an attacker has installed a trojan or rootkit. Also, the system administrators should be exporting copies of the system log files on a regular basis. These log files will help you in determining if an exploit has occurred.

As stated in the ISC Advisory, there are active exploits of BIND 8.2.X available. One script, bind-tsig.c, was released on Bugtraq and claimed to be a script to test for the exploit. However, it was quickly determined that the script was a trojan that actually attacks dns1.nai.com. Also, another script, bugtraq.c was released. This script actually does perform the exploit in a limited form and shows to have been tested against Slackware 7.

How to Combat the Exploit

BIND administrators should immediately upgrade their 8.2.X servers to BIND 8.2.3-REL or 9.1.0. The current versions of BIND are available from:

http://www.isc.org/products/BIND/

Conclusions

DNS is considered to be an integral part of the network and Internet infrastructure. As such, it is very important as a DNS administrator to be aware of possible vulnerabilities to BIND as they are discovered. This can be accomplished by subscribing to CERT and Bugtraq for updates as well as the ISC's BIND mail lists.

CERT:	http://www.cert.org/contact_cert/certmaillist.html
Bugtraq:	http://www.securityfocus.com/frames/?content=/about/feedback/subscribe.html
ISC BIND:	http://www.isc.org/services/public/lists/bind-lists.html

BIND administrators should also keep their DNS servers up to date with the current versions of BIND and make sure that they have implemented some level of server security and/or IDS for protection against attacks.

Sources

Albitz, Paul and Cricket Liu. "Backgound." DNS and BIND. 1998.

Athanasiou, Ken. "DNS Remote Root Exploit." 17 April 2000. URL: http://www.sans.org/infosecFAQ/malicious/DNS_exploit.htm (5 February 2001).

"Bind.tsig.c." 1 February 2001. URL: http://packetstorm.securify.com/0101-exploits/bind-tsig.c (5 February 2001).

"BIND Vulnerabilities." 29 January 2001. URL: http://www.isc.org/products/BIND/bind-security.html (5 February 2001).

"Bugtraq.c." 5 February 2001. URL: http://packetstorm.securify.com/0102-exploits/bugtraq.c (5 February 2001).

"CERT Advisory CA-2001-02 Multiple Vulnerabilities in BIND." 2 February 2001. URL: http://www.cert.org/advisories/CA-2001-02.html (5 February, 2001).

"[COVERT-2001-01] Vulnerabilities in BIND 4 and 8." 29 January 2001. URL: http://www.pgp.com/research/covert/advisories/047.asp (5 February, 2001).

Deterding, Brent. "Nmap – The Tool, It's Author and It's Implications." 13 July 2001. URL: http://www.sans.org/infosecFAQ/audit/nmap.htm (5 February 2001).

Hanley, Sinead. "DNS Overview with a Discussion of DNS Spoofing." 6 November 2000. URL: http://www.sans.org/infosecFAQ/DNS/DNS.htm (5 February 2001).

Poulsen, Kevin. "BIND Holes mean Big Trouble." Security Focus. 29 January 2001. URL: http://www.securityfocus.com/frames/?content=/templates/article.html%3Fid%3D144 (5 February, 2001).

Vixie, Paul. "RFC 2845 Secret Key Transaction Authentication for DNS (TSIG)." May 2000. URL: ftp://ftp.isi.edu/in-notes/rfc2845.txt (5 February, 2001).