Global Information Assurance Certification Paper

Submitted By : Chen Leng, NG

DNS Buffer Overflow Exploit of Transaction Signatures

Background

BIND (Berkeley Internet Name Domain) is the open source version of the DNS protocol suite. DNS (Domain Name System) is an Internet Service that translates domain names into IP addresses, using a distributed database. DNS is an integral part that constitutes to the proper functioning of the whole internet, and at this time, BIND is almost exclusively used as the platform for DNS.

Scope

This paper aims to discuss about one of the most recent found vulnerability in BIND 8, a common version of BIND being run on the internet at this time, possible consequences of any exploit and also recommended fixes. The exploit is published in CERT advisory (http://www.cert.org/advisories/CA-2001-02.html), originating from Covert Labs (http://www.pgp.com/research/covert/advisories/047.asp)

CERT® Advisory CA-2001-02 Multiple Vulnerabilities in BIND (original release date 29 Jan 2001)

BIND 8 contains a buffer overflow that allows a remote attacker to execute arbitrary code. The overflow is in the initial processing of a DNS request and therefore does not require an attacker to control an authoritative DNS server. In addition, the vulnerability is not dependent upon configuration options and affects both recursive and non-recursive servers.

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. Overwriting this memory can allow an intruder (in conjunction with other buffer overflow exploit techniques) to gain unauthorized access to the vulnerable system.

The flawed program logic is distributed over several function calls within the BIND software. When the attacker sends a UDP request, the packet will be loaded into a buffer on the stack (u.buf) by the function datagram_read(). On the other hand, TCP requests are loaded into a buffer (sp->s_buf) on the heap by the function stream_getmsg(). Regardless of the protocol, each of these functions call dispatch_message(), which in turn calls ns_req().

The ns_req() function handles the request. A call to ns_find_tsig() determines if a transaction signature exists in the request, and find_key() is called thereafter to determine if a valid key has been included. In the case where a transaction signature is found but the key is NULL, msglen is computed to include only the portion of the request before the signature. This is where the problem occurs, because the variables buflen and msglen are assumed through most of the code to add up to the total size of the buffer allocated for holding the request.

BIND uses the same buffer for storing the request and generating the response. Specifically, the response is composed by appending an error code and a transaction signature to the existing request. Since the new transaction signature is supposed to overwrite the signature of the request, msglen was modified to reflect the request length minus the signature length. However, buflen was not modified to reflect the new value of msglen, causing subsequent function calls (specifically ns_sign) to cause BIND to overwrite memory adjacent to the packet buffer.

These overwrites may allow an intruder to create conditions required for the execution of arbitrary code. Because the overflows occur on the stack for UDP requests and on the heap for TCP requests, the specific details of the exploit begin to differ at this point. Both scenarios result in the same impact -- the attacker can execute arbitrary code on the vulnerable system.

***Covert Labs also identifies how it is possible for an attack on such systems*:**

One can perform a stack based buffer overflow, with two important qualifications: first, that the number of bytes past the end of the buffer that the attacker can overwrite is limited in length, and second, that the values of those bytes are mostly fixed. On the x86 architecture, the attacker can manipulate a sufficient number of bytes such that they can modify the saved frame pointer. Overwriting the least significant byte of the saved frame pointer can result in the execution of arbitrary code in certain predictable installations of the name server.

An attacker can also perform a heap overflow, overwriting malloc's internal variables. This method is very effective, though it requires that an operating system's

implementation of malloc stores internal data structures in the allocated memory. For this attack to be successful, TCP port 53 must be accessible.

Some sample programs was posted in securityfocus.com(BugTraq) (http://www.securityfocus.com) mailing lists that is supposed to exploit the said vulnerability.

Solution :

This Vulnerability affects systems prior to BIND 8.2.3. BIND 8 is still commonly used on the internet, version 8.2.3 of BIND resolve the vulnerability described in this advisory and BIND 9, a totally new version released by ISC(Internet Software Consortium) is not affected by the BUG.

So users still using older versions of BIND prior to 8.2.3 should either upgrade to version 8.2.3 or BIND 9.1.

To download the 8.2.3 version of BIND, the source can be downloaded from ISC ftp site

ftp://ftp.isc.org/isc/bind/src/

and the BIND 9 distribution can be downloaded from :

ftp://ftp.isc.org/isc/bind9/

Sources:

Paul Albitz & Cricket Liu "DNS and BIND" O'Reilly. Third Edition

CERT advisory (http://www.cert.org/advisories/CA-2001-02.html) released 29th January 2001

Internet Software Consortium, (http://www.nominum.com/news/press-releases/pr-bind9-upgrade.html)

COVERT Labs, (http://www.pgp.com/research/covert/advisories/047.asp)

Security Focus, (http://www.securityfocus.com/frames/?content=/templates/article.html?id=144)

Information of Transaction signatures:

RFC 2535: Domain Name System Security Extensions

http://www.ietf.org/rfc/rfc2535.txt

RFC 2845: Secret Key Transaction Authentication for DNS (TSIG)
http://www.ietf.org/rfc/rfc2845.txt