dnsmasq SOS Fund Audit Fix Log

Identified Vulnerabilities

DM-01-001 Uninitialized buffer leads to memory leakage (Medium)

Fix:
http://thekelleys.org.uk/gitweb/?p=dnsmasq.git;a=commitdiff;h=294d36df4749e01199ab220d44c170e7db2b0c05

VERIFIED

Mike Wege (Cure53): *Because of potential further information leaks, we still recommend a more general approach of dealing with the problem by clearing the shared buffers on reuse. While there is certainly an argument against such an approach because of performance reasons, we believe even the lowest performance platforms today are fast enough to allow for such measures.*

Simon Kelley (dnsmasq): Buffer clearing implemented:
http://thekelleys.org.uk/gitweb/?p=dnsmasq.git;a=commitdiff;h=fa78573778cb23337f67f5d0c9de723169919047

EXTENDED FIX VERIFIED

DM-01-003 Makefile lacks security parameters for gcc (Low)

Simon Kelley (dnsmasq): I reject this. The makefile is carefully constructed to work standalone on a lowest-common-denominator system: It doesn't assume any particular tools or compilers or libraries other than a completely vanilla Unix system. Specifying complex gcc-specific flags like PIE would break this.

The Makefile doesn't preclude setting C compiler flags, and the Debian packaging, which I maintain and is held in the same git repo, does set the CFLAGS to implement PIE. I would expect that installations in other distributions would do the same. Output from Debian package build:

```
gcc -g -O2 -fPIE -fstack-protector --param=ssp-buffer-size=4 -Wformat
-Werror=format-security -D_FORTIFY_SOURCE=2 -Wall -W -DHAVE_DBUS
-DHAVE_CONNTRACK -DHAVE_DNSSEC -DLOCALEDIR='"/usr/share/locale"'
-DVERSION='"2.76-5-g04cb536"' -I/usr/include/dbus-1.0
-I/usr/lib/x86_64-linux-gnu/dbus-1.0/include -c cache.c
```

REJECTION ACCEPTED

Mike Wege (Cure53): *The argument that the Makefile is supposed to work on low-end as well as on higher-performance systems makes sense. It would be nice to make a note about*

the PIE-flags and it's security-relevance at an appropriate place in the documentation, though.

DM-01-006 Allocated memory is not cleared (Low)

Fix:
http://thekelleys.org.uk/gitweb/?p=dnsmasq.git;a=commitdiff;h=d55f81f5fd53b1dfc2c4b3249b542f2d9679e236

VERIFIED

Miscellaneous Issues

DM-01-002 Unchecked return value can lead to NULL pointer dereference (Low)

Fix:
http://thekelleys.org.uk/gitweb/?p=dnsmasq.git;a=commitdiff;h=ce7845bf5429bd2962c9b2e7d75e2659f3b5c1a8

VERIFIED

DM-01-004 Wrong assumption about return value of snprintf()/vsnprintf() (Low)

Simon Kelley (dnsmasq): On careful inspection, these three uses of snprintf() are correct.

In log.c the length of data used has the correct ceiling applied here:

```
entry->length = len > MAX_MESSAGE ? MAX_MESSAGE : len;
```

and in the DHCP logging code, the line is terminated on the condition

```
(q - daemon->namebuff) > 40)
```

which is always true when the buffer is too small for the output of snprintf().

REJECTION ACCEPTED

Mike Wege (Cure53): *Further inspection of the respective code revealed that the described case is impossible to exploit. There are a few more occurences of snprintf()/vsnprintf() usage in the source code, which when put under closer scrutiny are also safe.*

DM-01-005 Hardcoded values in fscanf() format strings with aliased buffers (Low)

Simon Kelley (dnsmasq): I confess to not really understanding the suggestion here. Note that the string-size limits in the scanf format strings are determined by the specifications of the file format being parsed. It's necessary to ensure that the re-purposed buffers are always larger than the string-length lints, but is most cases they are already much larger, and

cannot be made smaller without breaking their primary use. The single case where the buffer is close to the limit has an explanatory comment.

Mike Wege (Cure53): *The problem we see here is the fact that buffers are reused in several locations and the size constants are hard-coded upon definition of the respective buffers. Later in the code different hard-coded values are used to limit access to the buffers. We do acknowledge that there are no apparent problems with this right now, but want to point out that a sleight of hand or careless change can easily lead to major problems and therefore recommend using constants defined in a single combined location and furthermore modulo-limiting it to the absolute size of the buffer.*

Simon Kelley (dnsmasq): Fix committed:

http://thekelleys.org.uk/gitweb/?p=dnsmasq.git;a=commitdiff;h=bf4e62c19e619f7edf8d03d58d33a5752f190bfd

This is fairly readable, and should make it rather difficult to reduce any of those buffers below the assumed sizes accidentally.

VERIFIED

Mike Wege (Cure53): *I personally would have defined additional constants for the values 64 and 764 at a location right next to where you define DHCP_BUFF_SZ and moved them into the format strings with the stringify (#) operator, but using the preprocessor to check for overly large values in relative vicinity to the actual usage also solves the potential problem.*