ClamAV Library RemØte Heap Overflows
Security Advisory

Date
July 24, 2005

Vulnerability
ClamAV is the most widely used GPL antivirus library today. It provides file format support for virus analysis. During analysis ClamAV Antivirus Library is vulnerable to buffer overflows allowing attackers complete control of the system. These vulnerabilities can be exploited remotely without user interaction or authentication through common protocols such as SMTP, SMB, HTTP, FTP, etc.

Specifically, ClamAV is responsible for parsing multiple file formats. At least 4 of its file format processors contain remote security bugs. Specifically, during the processing of TNEF, CHM, & FSG formats an attacker is able to trigger several integer overflows that allow attackers to overwrite heap data to obtain complete control of the system. These vulnerabilities can be reached by default and triggered without user interaction by sending an e-mail containing crafted data.

Impact
Successful exploitation of ClamAV protected systems allows attackers unauthorized control of data and related privileges. It also provides leverage for further network compromise. ClamAV implementations are likely vulnerable in their default configuration.

Affected Products
ClamAV – 0.86.1 (current) and prior

There are numerous implementations of ClamAV listed on their site which are likely vulnerable. One party of note is Apple. Apple includes ClamAV by default in Mac OS X Server. In addition, ClamAV has been ported to windows and a variety of other platforms by third parties who's implementations are also likely vulnerable. Refer to vendor for specifics.

Credit
These vulnerabilities were discovered and researched by Neel Mehta & Alex Wheeler.

Contact
security@remØte.com

ClamAV Library RemØte Heap Overflows
Security Advisory

Description
TNEF processing contains at least two integer overflows that result in a heap overflows. The following is vulnerable code from tnef_attachment() and tnef_message() in tnef.c. The length field is an arbitrary 32-bit integer. If length is -1, it will wrap and malloc() will return a small heap buffer which is overflowed on the following fread().

```
string = cli_malloc(length + 1);
if(fread(string, 1, length, fp) != length) {
        free(string);
        return -1;
}
```

CHM processing contains an integer overflow that results in heap corruption. The following is vulnerable code from read_chunk_entries() in chmunpack.c. If length is -1, it will wrap and malloc() will return small heap buffer which is overflowed on the following strncpy().

```
name_len = read_enc_int(&current, end);
file_e->name = (unsigned char *) cli_malloc(name_len+1);
if (!file_e->name) {
        free(file_e);
        return FALSE;
}
strncpy(file_e->name, current, name_len);
```

FSG processing contains a faulty boundary check that results in a buffer overflow. The following is vulnerable code from unfsg() in fsg.c. Specifically, backbytes and backsize are essentially encoded arbitrary 32-bit unsigned integers; and, if both are slightly negative values an attacker can trigger a heap overflow because of the integer wraps in the boundary check.

```
if (cdst-backbytes < dest || cdst+backsize >= dest+dsize)
        return -1;
while(backsize--) {
        *cdst=*(cdst-backbytes);
        cdst++;
}
```