

CORE-2007-1212: SILC pkcs_decode buffer overflow

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2008-03/msg00333.html>

- *From:* Core Security Technologies Advisories <advisories@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 25 Mar 2008 18:09:10 -0200
-

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Core Security Technologies – CoreLabs Advisory

<http://www.coresecurity.com/corelabs/>

SILC pkcs_decode buffer overflow

Advisory Information

Title: SILC pkcs_decode buffer overflow

Advisory ID: CORE-2007-1212

Advisory URL: <http://www.coresecurity.com/?action=item&id=2206>

Date published: 2008-03-25

Date of last update: 2008-03-25

Vendors contacted: SILC development team

Release mode: Coordinated release

Vulnerability Information

Class: Arbitrary memory corruption

Remotely Exploitable: Yes

Locally Exploitable: Yes

Bugtraq ID: 28373

CVE Name: N/A

Vulnerability Description

Secure Internet Life Conferencing (SILC) is open protocol aimed at providing encrypted and authenticated communications over an insecure medium such as the Internet. The SILC application of the same name implements the protocol as an open source project. SILC is generally used as a more secure replacement for Internet Relay Chat (IRC) networks and other open and publicly accessible as well as private instant messaging networks. A remote buffer overflow vulnerability found in a

CORE-2007-1212: SILC pkcs_decode buffer overflow

library used by both the SILC server and client to process packets containing cryptographic material may allow an un-authenticated client to execute arbitrary code on the server with the privileges of the user account running the server, or a malicious SILC server to compromise client systems and execute arbitrary code with the privileges of the user account running the SILC client program.

Vulnerable Packages

- . SILC server up to 1.1.1
- . SILC client up to 1.1.3

Non-vulnerable Packages

- . SILC server 1.1.2
- . SILC client 1.1.4

Vendor Information, Solutions and Workarounds

Fixed versions are available at <http://silcnet.org/software/download/>.

Credits

This vulnerability was discovered by Core Security Technologies team "Los Plomeros vs. Blue Demon" during Bugweek 2007: Ariel Weissbein, Pedro Varangot, Martin Mizrahi, Oren Isacson, Carlos Garcia and Ivan Arce.

Technical Description / Proof of Concept Code

Upon initial connection with a SILC server, mutual authentication between peers (client, routers and servers) is performed and a key negotiation protocol is executed to obtain a shared key that is subsequently used to encrypt communications. While a detailed analysis of key exchange protocol used by SILC may be appropriate for further study, what is relevant for the issue at hand is that cryptographic material is exchanged between peers with data packets encoded using the PKCS #1 1.5 standard [1].

SILC's PKCS1 encoding functionality is implemented in the 'silccrypt' library in file 'silcpkcs1.c'. The specific code to decode PKCS#1 packets is implemented in the 'silc_pkcs1_decode' function shown below:

/-----

```

SilcBool silc_pkcs1_decode(SilcPkcs1BlockType bt,
const unsigned char *data,
SilcUInt32 data_len,
unsigned char *dest_data,
SilcUInt32 dest_data_size,
SilcUInt32 *dest_len)
{
int i = 0;

SILC_LOG_DEBUG(("PKCS#1 decoding, bt %d", bt));

/* Sanity checks */
if (!data || !dest_data || dest_data_size < 3 ||
data[0] != 0x00 || data[1] != (unsigned char)bt) {
SILC_LOG_DEBUG(("Malformed block"));
return FALSE;
}

/* Decode according to block type */
switch (bt) {
case SILC_PKCS1_BT_PRIV0:
/* Do nothing */
break;

case SILC_PKCS1_BT_PRIV1:
/* Verification */
(1) for (i = 2; i < data_len; i++)
if (data[i] != 0xff)
break;
break;

case SILC_PKCS1_BT_PUB:
/* Decryption */
(2) for (i = 2; i < data_len; i++)
if (data[i] == 0x00)
break;
break;
}

/* Sanity checks */
(3) if (data[i++] != 0x00) {
SILC_LOG_DEBUG(("Malformed block"));
return FALSE;
}
if (i - 1 < SILC_PKCS1_MIN_PADDING) {
SILC_LOG_DEBUG(("Malformed block"));
return FALSE;
}
if (dest_data_size < data_len - i) {
SILC_LOG_DEBUG(("Destination buffer too small"));
return FALSE;
}

```

```

}

/* Copy the data */
(4) memcpy(dest_data, data + i, data_len - i);

/* Return data length */
if (dest_len)
*dest_len = data_len - i;

return TRUE;
}

```

-----/

In the above code, a maliciously forged data packet with valid PKCS#1 encoding of all bytes set to '0xff' or non-'0x00' when transmitting private ('BT_PRIV1') or public ('BT_PUB') key material (respectively) will make the execution flow to exit the loops at '(1)' and '(2)' with the value of the unsigned integer variable 'i' set to 'data_len'. Next, at '(3)' the same variable 'i' is incremented by one and thus set to 'data_len+1'. A carefully crafted packet that passes the sanity check in '(3)' will eventually cause memory corruption due to an integer overflow in the third argument passed in the 'memcpy()' function call at '(4)'. Since 'i' is set to 'data_len+1' the 'data_len - i' expression used to calculate the value of the third argument (amount of bytes to be copied from the source buffer) will evaluate to '-1' causing the process memory to be overwritten due to a Unsigned to Signed conversion error [2]. Exploitation of this vulnerability leads to a direct Denial of Service to the vulnerable program or to arbitrary code execution on the vulnerable system with the privileges of the SILC program (either the client or the server) being attacked.

Report Timeline

- . 2008-03-19: Initial notification sent to the SILC project.
- . 2008-03-20: SILC project team acknowledges the bug and patches their source tree [3].
- . 2008-03-20: Core confirms the SILC project the advisory will be published on March 25th.

References

- [1] PKCS #1: RSA Encryption Version 1.5
<http://www.ietf.org/rfc/rfc2313.txt>
- [2] Unsigned to Signed Conversion Error – Common Weakness Enumeration (CWE) Definition :
<http://cwe.mitre.org/data/definitions/196.html>
- [3]

CORE-2007-1212: SILC pkcs_decode buffer overflow

<http://git.silcnet.org/gitweb/?p=silc.git;a=commitdiff;h=b36495161037e52ad993202da5d3df1837235d24>

About CoreLabs

CoreLabs, the research center of Core Security Technologies, is charged with anticipating the future needs and requirements for information security technologies. We conduct our research in several important areas of computer security including system vulnerabilities, cyber attack planning and simulation, source code auditing, and cryptography. Our results include problem formalization, identification of vulnerabilities, novel solutions and prototypes for new technologies. CoreLabs regularly publishes security advisories, technical papers, project information and shared software tools for public use at: <http://www.coresecurity.com/corelabs/>.

About Core Security Technologies

Core Security Technologies develops strategic solutions that help security-conscious organizations worldwide develop and maintain a proactive process for securing their networks. The company's flagship product, CORE IMPACT, is the most comprehensive product for performing enterprise security assurance testing. CORE IMPACT evaluates network, endpoint and end-user vulnerabilities and identifies what resources are exposed. It enables organizations to determine if current security investments are detecting and preventing attacks. Core Security Technologies augments its leading technology solution with world-class security consulting services, including penetration testing and software security auditing. Based in Boston, MA and Buenos Aires, Argentina, Core Security Technologies can be reached at 617-399-6980 or on the Web at <http://www.coresecurity.com>.

Disclaimer

The contents of this advisory are copyright (c) 2008 Core Security Technologies and (c) 2008 CoreLabs, and may be distributed freely provided that no fee is charged for this distribution and proper credit is given.

GPG/PGP Keys

This advisory has been signed with the GPG key of Core Security Technologies advisories team, which is available for download at http://www.coresecurity.com/files/attachments/core_security_advisories.asc.

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.6 (MingW32)

CORE-2007-1212: SILC pkcs_decode buffer overflow

Comment: Using GnuPG with Mozilla – <http://enigmail.mozdev.org>

iD8DBQFH6VvlyNibggitWa0RAtlbAKCiPlkRO8bUdgGJjwjPOd59smSPVQCgkMDb
XgZGnl122WKkgCEMRyhBfcA=
=VSVf
-----END PGP SIGNATURE-----