

[UNIX] libkadm5srv Heap Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-12/0064.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 12/21/04

To: list@securiteam.com

Date: 21 Dec 2004 12:51:43 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

libkadm5srv Heap Buffer Overflow

SUMMARY

The MIT Kerberos 5 administration library (libkadm5srv) contains a heap buffer overflow in password history handling code that could be exploited to execute arbitrary code on a Key Distribution Center (KDC) host. The overflow occurs during a password change of a principal with a certain password history state. An administrator must have performed a certain password policy change in order to create the vulnerable state. (See MITIGATING FACTORS below.)

No exploits are known to exist at this time, though a public discussion of the bug took place during the first weeks of December 2004, containing sufficient detail that someone could infer how to perform an attack. Exploitation of this vulnerability is believed to be difficult, due to the limited extent of the overflow.

DETAILS

Impact:

An authenticated user, not necessarily one with administrative privileges, could execute arbitrary code on the KDC host, compromising an entire Kerberos realm.

Securiteam: [UNIX] libkadm5srv Heap Buffer Overflow

Mitigating Factors:

- * Typically, only a principal satisfying the following conditions can trigger the buffer overflow upon password change:
 - + Have changed its password fewer times than the history count in its password policy
 - + Had its password policy's history count subsequently reduced to equal the number of times it has changed its password

- * There are other means of producing the vulnerable state, though they are significantly more complex and much less likely. All of these other methods involve a reduction of the password history count in a password policy.

- * A workaround exists (see FIXES).

Affected Software:

- * KDC software on all releases of MIT krb5, up to and including krb5-1.3.5. The vulnerable library is libkadm5srv. Programs that use the vulnerable functionality of the library include:
 - + kadmind (administration daemon)
 - + kadmin.local (KDC-local administration client)
 - + kadmind4 (krb4 compatibility administration daemon)

Fixes:

- * **WORKAROUND:** Until your KDC programs and libraries have been patched, do not decrease the password history count on any policy in your Kerberos realm. Also, if you have already decreased the password history count on a policy at some point in the past, you should raise it to the maximum value that it has had in the past

- * The upcoming krb5-1.4 release (currently in beta test) will contain fixes for this problem. The krb5-1.4-beta3 release contains fixes for this problem

- * The upcoming krb5-1.3.6 patch release contains fixes for this problem

- * Apply the following patch to src/lib/kadm5/srv/svr_principal.c, and recompile the affected libraries and binaries. This patch was generated against krb5-1.3.5, and may apply, with some offset, to earlier releases.

This patch may also be found at:

<http://web.mit.edu/kerberos/advisories/2004-004-patch_1.3.5.txt>
http://web.mit.edu/kerberos/advisories/2004-004-patch_1.3.5.txt

The associated detached PGP signature is at:

<http://web.mit.edu/kerberos/advisories/2004-004-patch_1.3.5.txt.asc>
http://web.mit.edu/kerberos/advisories/2004-004-patch_1.3.5.txt.asc

Patch:

Index: svr_principal.c

Securiteam: [UNIX] libkadm5srv Heap Buffer Overflow

```

RCS file: /cvs/krbdev/krb5/src/lib/kadm5/srv/svr_principal.c,v
retrieving revision 1.26.2.1
diff -c -r1.26.2.1 svr_principal.c
*** svr_principal.c 2 Sep 2003 18:58:56 -0000 1.26.2.1
--- svr_principal.c 20 Dec 2004 19:47:29 -0000
*****
*** 1017,1022 ****
--- 1017,1025 -----

memset(&adb->old_keys[adb->old_key_len],0,sizeof(osa_pw_hist_ent));
    adb->old_key_len++;
+ for (i = adb->old_key_len - 1; i > adb->old_key_next; i--)
+ adb->old_keys[i] = adb->old_keys[i - 1];
+
memset(&adb->old_keys[adb->old_key_next],0,sizeof(osa_pw_hist_ent));
    } else if (adb->old_key_len > pol->pw_history_num-1) {
        /*
        * The policy must have changed! Shrink the array.
        *****
        *** 1039,1048 ****
            histp[i] = adb->old_keys[j];
        }
        /* Now free the ones we don't keep (the oldest ones) */
! for (i = 0; i < adb->old_key_len - (pol->pw_history_num - 1);
i++)
    for (j = 0; j < adb->old_keys[KADM_MOD(i)].n_key_data;
j++)
        krb5_free_key_data_contents(context,
            &adb->old_keys[KADM_MOD(i)].key_data[j]);
        free((void *)adb->old_keys);
        adb->old_keys = histp;
        adb->old_key_len = pol->pw_history_num - 1;
---- 1042,1053 -----
            histp[i] = adb->old_keys[j];
        }
        /* Now free the ones we don't keep (the oldest ones) */
! for (i = 0; i < adb->old_key_len - (pol->pw_history_num-1);
i++) {
    for (j = 0; j < adb->old_keys[KADM_MOD(i)].n_key_data;
j++)
        krb5_free_key_data_contents(context,
            &adb->old_keys[KADM_MOD(i)].key_data[j]);
+ free(adb->old_keys[KADM_MOD(i)].key_data);
+ }
        free((void *)adb->old_keys);
        adb->old_keys = histp;
        adb->old_key_len = pol->pw_history_num - 1;
        *****
        *** 1052,1061 ****
--- 1057,1070 -----

```

Securiteam: [UNIX] libkadm5srv Heap Buffer Overflow

```
    }  
  }  
  
+ if (adb->old_key_next + 1 > adb->old_key_len)  
+ adb->old_key_next = 0;  
+  
+ /* free the old pw history entry if it contains data */  
+ histp = &adb->old_keys[adb->old_key_next];  
+ for (i = 0; i < histp->n_key_data; i++)  
+   krb5_free_key_data_contents(context, &histp->key_data[i]);  
+ free(histp->key_data);  
  
+ /* store the new entry */  
+ adb->old_keys[adb->old_key_next] = *pw;
```

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-1189>>
CAN-2004-1189

Technical Details:

The vulnerable function is `add_to_history()` in `src/lib/kadm5/srv/svr_principal.c`. The password history is stored as a ring buffer, represented as an array of `osa_pw_ent_rec`, which is `adb->old_keys`. The "next" pointer is an index into the array, `adb->old_key_next`, and the length of the array is stored in `adb->old_key_len`. The array is dynamically resized as needed, and there is no separate head pointer.

The policy's history count is stored in `pol->pw_hist_num`, but the actual maximum number of keys stored in `adb->old_keys` is `pol->pw_hist_num-1`, since the "current" key data are also used for history comparisons when a password change occurs.

The index value `adb->old_key_next` is permitted to index to a position one past the end of the array `adb->old_keys` if `adb->old_key_next` is less than `pol->pw_hist_num-1`. This out-of-bounds indexing is usually fixed up when `add_to_history()` enlarges the array on a subsequent call.

If `pol->pw_hist_num` is reduced to `adb->old_key_next` after a password change that causes `adb->old_key_next` to index out of bounds, a subsequent password change will not run the resizing code, and `add_to_history()` will write a password history entry past the end of the array `adb->old_keys`.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:tlyu@mit.edu>> Tom Yu.

The original article can be found at:

<<http://web.mit.edu/kerberos/advisories/index.html>>

<http://web.mit.edu/kerberos/advisories/index.html>

Securiteam: [UNIX] libkadm5srv Heap Buffer Overflow

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.