L0pht Advisory: release of L0phtCrack for NT

From: owner-bugtraq () netspace org (Aleph One)
Date: Fri, 11 Apr 1997 18:07:56 -0400

```
                    L0pht Security Advisory
                  Advisory released April 11 1997

          Program: L0phtcrack - Windows NT password insecurities

                    Vulnerability Scope: Windows NT

         Severity: The L0pht is pleased to release L0phtcrack rev 1.
              This program recovers the LANMAN and/or NT Dialect
              MD4 plaintext password from output derived from the
              SAM registry.

                    Authors: mudge () l0pht com
                             weld () l0pht com

Intro:

  This tool, as with many others, can be used for breaking into systems
  in illegal fashions - THAT IS NOT WHAT IT IS INTENDED FOR! We had a
  working version done the same day that PWDump was released in order
  to audit some of our internal networks. However, as we started
  researching more into it we noticed many shortcomings in how MS
  security is handled and present some of these in our tool. We take
  no responsibility for misuse of this information. It is our belief
  that the only way to protect yourself is to fully understand your
  vulnerabilities. Unfortunately, for some of these problems we still
  don't see immediate solutions. Our particular solution has been to
  trust our users, and not let any of our NT machines talk to the internet
  (ie filtered very tightly at the perimiter). We are interested in
  other solutions.

Overview:

  Recently several NT password crackers have emerged. We offer this
  one with the belief that it offers some features and functionality
  that the current ones do not have.

  L0phtcrack will recover passwords from Windows NT registries in a
  variety of fashions.

  By feeding in the output from PWDump [by Jeremy Allison, jra () cygnus com]
  and a dictionary file, L0phtcrack rev 1 will attempt to retrieve:

    1) only the LANMAN plaintext password
```

```
      2) only the NT Dialect MD4 plaintext password [see reasoning below]
      3) Both the LANMAN and MD4 plaintext passwords (by deriving the
         MD4 password from the LANMAN output and running through up to
         2 to the Nth power permutations)

    Alternatively, L0phtcrack gives you the capability to _brute force_ the
    entire key space and recover ALL USER PASSWORDS up to 14 characters in
    length.

    By going through the entire keyspace available, this program
    WILL RETURN ALL OF THE PLAINTEXT PASSWORDS (both LANMAN and MD4) up to
    and including 14 characters in length (note that the User Login Dialog
    box on NT machines limits the amount of characters that can be typed
    to 14 for the MD4 dialect. Future releases of this software will enable
    brute forcing of up to 16 characters for MD4).

    L0phtcrack comes in three flavours:

      1) A nice Windows GUI interface so you can point and click.
      2) A CLI version for running in "DOS" windows.
      3) Source code that is generic enough to build on most Un*x's.

  Description:

    Here's how it works -

    For NT, LANMAN passwords are derived in the following fashion:

      . The user password is converted to UPPERCASE
      . If the user password is less than 14 bytes, the password is padded
        with NULL characters to 14 bytes.
      . If the user password is greater than 14 bytes, the password is
        truncated to 14 bytes.
      . The 14 byte string is split down the middle into two 7 byte strings.
      . One 8 byte odd parity des key is derived from each of the 7byte
        strings [note1].
      . The constant 'magic value' [note2] is then encrypted first
        with the first odd parity des key and then with the second. The results
        are concatenated. This is the LANMAN OWP [note3].

      [note1: There is a significant loss of bits in the str_to_key functions
       which derive the 8 byte odd parity DES keys from the 7 byte strings.
       This knocks down the possibly key space to attack DES substantially.
       Thanks to Hobbit () avian org for pointing this out to us]

      [note2: the constant 'magic value' is derived from the encryption
       of 0x4B47532140232425 with a key of all 1's ]

      [note3: quickly scanning the LANMAN OWP's it is easy to see who has
       passwords that are 7 characters or less. If the second half of the
       LANMAN OWP is 0xAAD3B435B51404EE the value for the last seven characters
       in the user password were all NULLs.]

    For NT, NT Dialect MD4 passwords are derived in the following fashion:

      . The users password is converted to Unicode [note4].
      . The unicode password is run through MD4 to return a 16 byte value.
        This is the MD4 OWP [note5] [note6].

      [note4: There is a large amount of confusion as to where Unicode stops.
       i.e. is "ABC", which is in actuallity 'A','B','C','\0', encoded
       as 'A' '\0' 'B' '\0' 'C' '\0' or 'A' '\0' 'B' '\0' 'C' '\0' '\0' '\0'.
       We find that in this situation the former is the case.

      [note5: You might say "why do you even bother having an option of doing
       _only md4_ when it is much quicker to derive it from the LANMAN
       password". To which we would reply "this gives us the ability to
       easilly roll in the ability to dictionary attack traffic that we
       see on the network. This will be particularly important if the
```

proposed changes to the CIFS spec go into place. See our S/Key cracker MONKEY for more of an idea on what's to come".]

[note6: For those who were building md4 crypt-n-compare engines from inside Microsoft's Visual C++ IDE. The VC++ does not by default define _MSDOS_, or 8086 which are necesarry to through the byte ordering into the correct mode in md4.c]

What we do in rev 1 -

In rev 1 of l0phtcrack the user must hand in a password file in the format of Jeremy Allison's PWDump output. From this the following actions can be taken.

LANMAN only -
A dictionary is fed in and each word is encrypted using the LANMAN one round DES format as described above. The list of users is checked against this encrypted OWP. Any that are found matching are flagged.

MD4 only -
A dictionary is fed in and each word is encrypted using md4. The list of users is checked against this encrypted OWP. Any that are found matching are flagged. See the description of rev 2 for why this option is important.

LANMAN and md4 -
A dictionary is fed in and each user is first checked against the LANMAN one round DES OWP. If a match is found, the word is run through 2 to the power of strlen(word) case permutations in md4 to return the case sensitive md4 value.

Brute force -
An input string containing the list of valid characters is run through sequentially in all possible combinations up to 7 characters in length. The first half and second half of the LANMAN password are compared against these, thus returning all passwords up to 14 characters in total length. Since the logon screen will not allow you to enter more than 14 characters ,even though the NT MD4 dialect will allow up to 128, this should return all users passwords. When a match is found the word is run through 2 to the power of strlen(word).

By changing the default string that is processed through you can drastically change the amount of time it takes to brute through the entire keyspace. Keep in mind that the following characters are not valid in passwords so they don't need to be included: '/', '\', '[', ']', ':', ';', '|,' ,'=', ',', '+', '*', '?', '<', '>' [according to the MS technet information]. For example: if you just want to check all combinations of letters all you have to run through is ABCDEFGHIJKLMNOPQRSTUVWXYZ.

rev 2 will have this optimized a bit more, in addition to allowing a remote querry to our tables of precomputed hashes, thus reducing the problem to that of a table lookup.

Why is it important to be able to attack md4 only? That is much slower!

The changes being made to the CIFS spec imply that in the future a server will be able to force a client to use the NT dialect and not negotiate down. Based upon how the "key exchange" is done this will be attackable via the hooks put in for md4 only much in a similar way that our program "MONKEY" will attack s/key sessions based upon promiscuously viewed network traffic.

errata in rev 1 -

Several of the routines need to be optimized a bit more but the

```
      tool is quite usable and quite fast as it is (100 users and an
      an 8 meg dictionary file took under 1 minute on a PPro 200
      with the GUI version, the CLI is by nature a bit faster  -
      the bruting with a string of
      "ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789-_" took a little over 3 days
      on a P133).

      There are hooks to preen through the user list and instantly kick
      out whether a user has a password of 7 characters or less, or
      if a users password is greater than 7 chars.

      If you specify md4 only it just does a straight dictionary
      crypt and compare, if you specify any other method that returns
      md4 values it runs through all case possibilities.

      The brute forcer is not implemented in the windows GUI version. Use
      the command line version for this functionality.

   What you can expect to see in rev 2 -

     . The functionality of PWDump will be included in the l0phtcrack
       program so you won't need to run seperate programs.

     . You should be able to pull down registries from remote / local
       machines WITHOUT BEING ADMINISTRATOR and WITHOUT NEEDING TO
       KNOW THE ADMINISTRATOR's PASSWORD [read this bullet item again!!!]
       - we believe we are very close to being able to do this now.

     . You will be able to brute force the NT Dialect password up to
       16 characters in length for those tricky network users that
       never log in via the console.

     . The windows GUI will be multi-threaded to take advantage of
       multiple processors for dramatically improved brute forcing.

     . We should have pre-computed tables of the entire key-space
       available so all that needs to be done is a remote table look
       up.

  L0phtcrack is freely available from the l0pht advisories page:
    http://www.l0pht.com/advisories.html
    screenshots should be available on the web page in the next couple
    of days.

    A mirror of the packages will be available at
    ftp://dot.ishiboo.com/users/tfish/l0phtcrack.tar.gz
    and
    ftp://dot.ishiboo.com/users/tfish/l0phtcrack.zip

  If anyone makes modifications / improvements please mail the diffs to
  mudge () l0pht com.

  We hope this tool is usefull,

  mudge () l0pht com , weld () l0pht com

--------------
For other advisories check out http://www.l0pht.com/advisories.html
--------------
```

By Date By Thread

Current thread:

qualcomm POP server *David Sacerdote (Apr 09)*