Network Working Group
Request for Comments: 2289
Obsoletes: 1938
Category: Standards Track

N. Haller
Bellcore
C. Metz
Kaman Sciences Corporation
P. Nesser
Nesser & Nesser Consulting
M. Straw
Bellcore

February 1998

A One-Time Password System

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

1.0 ABSTRACT

This document describes a one-time password authentication system (OTP). The system provides authentication for system access (login) and other applications requiring authentication that is secure against passive attacks based on replaying captured reusable passwords. OTP evolved from the S/KEY (S/KEY is a trademark of Bellcore) One-Time Password System that was released by Bellcore and is described in references [3] and [5].

2.0 OVERVIEW

One form of attack on networked computing systems is eavesdropping on network connections to obtain authentication information such as the login IDs and passwords of legitimate users. Once this information is captured, it can be used at a later time to gain access to the system. One-time password systems are designed to counter this type of attack, called a "replay attack" [4].

The authentication system described in this document uses a secret pass-phrase to generate a sequence of one-time (single use) passwords. With this system, the user's secret pass-phrase never needs to cross the network at any time such as during authentication

or during pass-phrase changes. Thus, it is not vulnerable to replay attacks. Added security is provided by the property that no secret information need be stored on any system, including the server being protected.

The OTP system protects against external passive attacks against the authentication subsystem. It does not prevent a network eavesdropper from gaining access to private information and does not provide protection against either "social engineering" or active attacks [9].

3.0 INTRODUCTION

There are two entities in the operation of the OTP one-time password system. The generator must produce the appropriate one-time password from the user's secret pass-phrase and from information provided in the challenge from the server. The server must send a challenge that includes the appropriate generation parameters to the generator, must verify the one-time password received, must store the last valid one-time password it received, and must store the corresponding one-time password sequence number. The server must also facilitate the changing of the user's secret pass-phrase in a secure manner.

The OTP system generator passes the user's secret pass-phrase, along with a seed received from the server as part of the challenge, through multiple iterations of a secure hash function to produce a one-time password. After each successful authentication, the number of secure hash function iterations is reduced by one. Thus, a unique sequence of passwords is generated. The server verifies the one-time password received from the generator by computing the secure hash function once and comparing the result with the previously accepted one-time password. This technique was first suggested by Leslie Lamport [1].

4.0 REQUIREMENTS TERMINOLOGY

In this document, the words that are used to define the significance of each particular requirement are usually capitalized. These words are:

- MUST

  This word or the adjective "REQUIRED" means that the item is an absolute requirement of the specification.

- SHOULD

  This word or the adjective "RECOMMENDED" means that there might exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before taking a different course.

- MAY

  This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor might choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

5.0 SECURE HASH FUNCTION

The security of the OTP system is based on the non-invertability of a secure hash function. Such a function must be tractable to compute in the forward direction, but computationally infeasible to invert.

The interfaces are currently defined for three such hash algorithms, MD4 [2] and MD5 [6] by Ronald Rivest, and SHA [7] by NIST. All conforming implementations of both server and generators MUST support MD5. They SHOULD support SHA and MAY also support MD4. Clearly, the generator and server must use the same algorithm in order to interoperate. Other hash algorithms may be specified for use with this system by publishing the appropriate interfaces.

The secure hash algorithms listed above have the property that they accept an input that is arbitrarily long and produce a fixed size output. The OTP system folds this output to 64 bits using the algorithms in the Appendix A. 64 bits is also the length of the one-time passwords. This is believed to be long enough to be secure and short enough to be entered manually (see below, Form of Output) when necessary.

6.0 GENERATION OF ONE-TIME PASSWORDS

This section describes the generation of the one-time passwords. This process consists of an initial step in which all inputs are combined, a computation step where the secure hash function is applied a specified number of times, and an output function where the 64 bit one-time password is converted to a human readable form.

Appendix C contains examples of the outputs given a collection of inputs. It provides implementors with a means of verification the use of these algorithms.

Initial Step

In principle, the user's secret pass-phrase may be of any length. To reduce the risk from techniques such as exhaustive search or dictionary attacks, character string pass-phrases MUST contain at least 10 characters (see Form of Inputs below). All implementations MUST support a pass-phrases of at least 63 characters. The secret pass-phrase is frequently, but is not required to be, textual information provided by a user.

In this step, the pass phrase is concatenated with a seed that is transmitted from the server in clear text. This non-secret seed allows clients to use the same secret pass-phrase on multiple machines (using different seeds) and to safely recycle their secret pass-phrases by changing the seed.

The result of the concatenation is passed through the secure hash function and then is reduced to 64 bits using one of the function dependent algorithms shown in Appendix A.

Computation Step

A sequence of one-time passwords is produced by applying the secure hash function multiple times to the output of the initial step (called S). That is, the first one-time password to be used is produced by passing S through the secure hash function a number of times (N) specified by the user. The next one-time password to be used is generated by passing S though the secure hash function N-1 times. An eavesdropper who has monitored the transmission of a one-time password would not be able to generate the next required password because doing so would mean inverting the hash function.

Form of Inputs

The secret pass-phrase is seen only by the OTP generator. To allow interchangeability of generators, all generators MUST support a secret pass-phrase of 10 to 63 characters. Implementations MAY support a longer pass-phrase, but such implementations risk the loss of interchangeability with implementations supporting only the minimum.

The seed MUST consist of purely alphanumeric characters and MUST be of one to 16 characters in length. The seed is a string of characters that MUST not contain any blanks and SHOULD consist of strictly alphanumeric characters from the ISO-646 Invariant Code Set. The seed MUST be case insensitive and MUST be internally converted to lower case before it is processed.

The sequence number and seed together constitute a larger unit of data called the challenge. The challenge gives the generator the parameters it needs to calculate the correct one-time password from the secret pass-phrase. The challenge MUST be in a standard syntax so that automated generators can recognize the challenge in context and extract these parameters. The syntax of the challenge is:

otp-<algorithm identifier> <sequence integer> <seed>

The three tokens MUST be separated by a white space (defined as any number of spaces and/or tabs) and the entire challenge string MUST be terminated with either a space or a new line. The string "otp-" MUST be in lower case. The algorithm identifier is case sensitive (the existing identifiers are all lower case), and the seed is case insensitive and converted before use to lower case. If additional algorithms are defined, appropriate identifiers (short, but not limited to three or four characters) must be defined. The currently defined algorithm identifiers are:

md4	MD4 Message Digest
md5	MD5 Message Digest
sha1	NIST Secure Hash Algorithm Revision 1

An example of an OTP challenge is: otp-md5 487 dog2

Form of Output

The one-time password generated by the above procedure is 64 bits in length. Entering a 64 bit number is a difficult and error prone process. Some generators insert this password into the input stream and some others make it available for system "cut and paste." Still other arrangements require the one-time password to be entered manually. The OTP system is designed to facilitate this manual entry without impeding automatic methods. The one-time password therefore MAY be converted to, and all servers MUST be capable of accepting it as, a sequence of six short (1 to 4 letter) easily typed words that only use characters from ISO-646 IVCS. Each word is chosen from a dictionary of 2048 words; at 11 bits per word, all one-time passwords may be encoded.

The two extra bits in this encoding are used to store a checksum. The 64 bits of key are broken down into pairs of bits, then these pairs are summed together. The two least significant bits of this sum are encoded in the last two bits of the six word sequence with the least significant bit of the sum as the last bit encoded. All OTP generators MUST calculate this checksum and all OTP servers MUST verify this checksum explicitly as part of the operation of decoding this representation of the one-time password.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.