



Traditional Authentication Practices Considered Dangerous

WolfeReiter Releases Open-Source, Cryptographically-secure Authentication Component

August 9, 2005

Long-term storage of logon credentials in networked business systems is a serious point of vulnerability. Most systems store credentials in an unencrypted or weakly encrypted form that system administrators or database power-users can view or edit. Historically this was fine as long as a set of conditions were true:

- The administrators of the system were known to be trustworthy and no untrusted users have the ability to view, alter or copy the password file or table.
- The normal execution flow of the system could not be used to reveal plaintext passwords.
- The online and backup repositories of authentication data were secure.

As individual companies grow these conditions begin to break down. They also more generally break down with the passage of time. Employee turnover, the increasing complexity of internet-connected systems, and the spread of knowledge and tools for executing simple security exploits all work together to change the rules of the game. Disgruntled employees are now widely acknowledged to be the single largest vulnerability for most companies, and are the most common attackers of computer systems. Defects in production software, runtimes and platforms can also disclose sensitive information to attackers. Recent headlines are indicative of the fact that systems that store "secret" logon information cannot be assumed to be safe:

- The Mozilla Foundation recently announced the personal information and passwords stored in their www.spreadfirefox.com advocacy site had been stolen by remote attackers.
- Several large companies including Citibank, Equifax and Time Warner have publicly disclosed losing sensitive backup tapes.

An important step toward hardening the security of business systems is to alter the authentication system so that the authentication file or table does not contain any secret information that must be protected in order

WolfeReiter

thoughtful computing

WolfeReiter provides comprehensive software services to business, nonprofit, and government clients.

- Software Design and Development
- Systems Integration
- Consultative ROI Analysis & IT Investment Planning
- Project Planning and Management Based on the Industry-Standard Software Development Life Cycle Approach
- Application Security Design and Implementation

Contact us today for a free consultation or visit www.thoughtfulcomputing.com to learn more.

941 O Street NW
Washington, DC 20001
info@thoughtfulcomputing.com

202.468.2752



to remain secure. This sounds like an oxymoron, but it is cryptographically feasible. Contemporary development environments include libraries such as Cryptographic Extensions for Java and System.Security.Cryptography for .NET and Mono that enable the use of strong cryptographic techniques to eliminate this type of vulnerability.

The key to hardening authentication systems is to use a technique known as cryptographic hashing. Hashing is a one-way process by which sensitive user input (such as a password) is converted to a number in such a way that it is extremely unlikely that any two inputs would generate the same number. Regardless of the size of the input, the hashed number is always the same size.

Hashes provide a way to authenticate an input without storing a copy of the input itself. This means a system can store cryptographic hashes of passwords instead of the passwords themselves.

When a user responds to the authentication challenge, the password she provides will be run through the cryptographic hash and compared to the one stored in the system for her account. If they match she is authenticated.

While this simple implementation of a hashed password system is much better than the ubiquitous plaintext version, it is still far too weak. Because the hash for any particular password is always the same, it is possible to pre-compute the hashes for a large number of possible passwords. Retrieving passwords from this dictionary of hashes is known as a *dictionary attack*.

Free software exists to perform this sort of dictionary attack without having to understand anything about how it works. To defend against the dictionary attack the hashing system must use the techniques of *salting* and *stretching* to make it so computationally intensive to calculate the hash dictionary that it becomes an infeasible attack for the foreseeable future.

Salting is the technique of combining the password to be hashed with a unique and cryptographically random number or *salt* so that a dictionary must be built to attack each individual password in the authentication system. The salt is not considered to be a secret and is stored along with the user name and password hash. Combining a password with salt removes the economy of scale from the dictionary attack.

WolfeReiter

thoughtful computing

The WolfeReiter BEDROCK System contains a strong authentication component that uses several cryptographic techniques to remove common vulnerabilities described in this article.

More information on BEDROCK is available at www.thoughtfulcomputing.com.

In order to promote more generally secure systems, WolfeReiter is making the HashManager class from BEDROCK that implements this functionality available for download from www.thoughtfulcomputing.com under the zlib/libpng open source license.

WolfeReiter provides comprehensive software services to business, nonprofit, and government clients.

- Software Design and Development
- Systems Integration
- Consultative ROI Analysis & IT Investment Planning
- Project Planning and Management Based on the Industry-Standard Software Development Life Cycle Approach
- Application Security Design and Implementation





Stretching is the technique of adding randomness or *entropy* to a cryptographic product by performing a series of cryptographic operations on the plaintext input. Stretching the hash makes it computationally expensive to calculate a single hash.

A strong hash algorithm combined with salting and stretching generates hashes that are so deeply encrypted that they cannot feasibly be attacked. The secrecy of the authentication system is no longer critical to the security of the system.

The WolfeReiter BEDROCK System contains a strong authentication system that uses these cryptographic salting and stretching techniques to process passwords with any hashing algorithm. Due to the phenomenal number-crunching power of contemporary personal computers and published weaknesses in various algorithms, cryptography researchers now consider the widely-used MD5, SHA, and SHA-1 hashing algorithms to be too weak for use in authentication systems. By default, BEDROCK uses the SHA-256 hash algorithm developed by the NSA as a part of the SHA-2 specification.

In order to promote more generally secure systems, WolfeReiter is making the HashManager class from BEDROCK that implements this functionality available for download from www.thoughtfulcomputing.com under the zlib/libpng open source license. WolfeReiter also provides security audits, consulting and developer mentoring services.

WolfeReiter

thoughtful computing

WolfeReiter provides comprehensive software services to business, nonprofit, and government clients.

- Software Design and Development
- Systems Integration
- Consultative ROI Analysis & IT Investment Planning
- Project Planning and Management Based on the Industry-Standard Software Development Life Cycle Approach
- Application Security Design and Implementation

Contact us today for a free consultation or visit www.thoughtfulcomputing.com to learn more.

941 O Street NW
Washington, DC 20001
info@thoughtfulcomputing.com

202.468.2752

