CS-410: Project proposal
Prof: Bart Massey
05/21/2023

Stash: an encrypted folder manager

Team members:

- Jacob Bentley: jpb7@pdx.edu
- Richard Duffy: rduffy@pdx.edu

GitLab repo URL:

https://gitlab.cecs.pdx.edu/cs410-rust/stash

Topic area:

Filesystems, encryption, security

Project vision:

Stash is a command-line tool that allows the user to create and manage encrypted folders on their local Linux filesystem. The idea is to add an extra layer of privacy and security for sensitive files or values such as API keys.

Basically, stash provides a set of terminal commands that allow the user to quickly encrypt a given file or set of files into an encrypted folder (called a stash), and also to decrypt a file or files from a stash into the current directory.

For encryption and decryption, stash will use a Rust implementation of the AES-GCM-SIV algorithm which can be found at:

https://crates.io/crates/aes-gcm-siv.

Stash would handle creation of a new stash with the command:

stash init <stash_name> <path/to/stash>

The basic syntax of the primary commands will be:

```
stash <cmd> <file> <stash_name>
```

So, to encrypt a copy of a given file and move it into the default stash, one could use:

```
stash cp <file>
```

One could also move that file into the default stash by using:

```
stash mv <file>
```

To restore an encrypted file from the default stash to the current directory, one can use:

```
stash grab <file>
```

The contents of a given stash will be viewable with:

```
stash ls <stash_name>
```

To list those contents without decrypting all files in the stash, each stash will maintain an encrypted text file that records all filenames in the stash. Upon receiving the above command, stash will decrypt that text file and print its contents to the terminal.

Flex goals:

Stash will start off using a single stash as a proof of concept or minimum viable product. Ideally, however, it will grow to include multiple stashes which can be tucked away in different project directories for easier access and greater segmentation.

Another stretch goal will be to include some initialization that creates a Linux user called `stash` which has its own password and set of permissions.

A final flex goal would be to incorporate some kind of tagging system for easier grabbing. This would consist of a hash map whose keys are tags and whose values are filepaths. The hash map would be serialized into JSON, encrypted, and stored in a stash; upon retrieval it would be decrypted, deserialized, and then accessed by means of whatever key was passed to the command.