Keymaker Documentation

Release 0.0.1

Andrey Kislyuk

Jul 16, 2018

Contents

Keymaker is **the missing link between SSH and IAM accounts on Amazon AWS**. It's a stateless synchronization engine that securely manages the process of SSH public key sharing and verification, user and group synchronization, and home directory sharing (via optional EFS integration). You, the AWS account administrator, define or import user and group identities in IAM, and instances in your account dynamically retrieve and use those identities to authenticate your users. Keymaker is the modern, minimalistic alternative to **LDAP** or **Active Directory** authentication.

CHAPTER 1

Installation

Run `pip install keymaker`.

On instances that accept SSH logins:

- Run `keymaker install`.
- Ensure processes launched by sshd have the IAM permissions `iam:GetSSHPublicKey`, `iam:ListSSHPublicKeys`, `iam:GetUser`, `iam:ListGroups`, `iam:GetGroup`, `iam:ListGroupsForUser`, `iam:GetRole`, and `sts:GetCallerIdentity`. The easiest way to do this is by running `keymaker configure --instance-iam-role ROLE_NAME` as a privileged IAM user, which will create and attach a Keymaker IAM policy to the role `ROLE_NAME` (which you should then assign, via an IAM Instance Profile, to any instances you launch). You can also manually configure these permissions, or attach the IAMReadOnlyAccess managed policy.

Keymaker requires OpenSSH v6.2+, provided by Ubuntu 14.04+ and RHEL7+.

CHAPTER 2

Usage

Run `keymaker` with no arguments to get usage information. In client mode (running on a computer that you will connect from), you can run `keymaker <subcommand>`, where subcommand is:

```
upload_key          Upload public SSH key for a user. Run this command for each user␣
↪who will be accessing EC2 hosts.
list_keys           Get public SSH keys for a given or current IAM/SSH user.
disable_key         Disable a given public SSH key for a given or current IAM/SSH␣
↪user.
enable_key          Enable a given public SSH key for a given or current IAM/SSH user.
delete_key          Delete a given public SSH key for a given or current IAM/SSH user.
configure           Perform administrative configuration tasks on the current AWS␣
↪account.
```

Use `keymaker <subcommand> --help` to get a full description and list of options for each command.

CHAPTER 3

Principle of operation

Amazon Web Services IAM user accounts provide the ability to add SSH public keys to their metadata (up to 5 keys can be added; individual keys can be disabled). Keymaker uses this metadata to authenticate SSH logins. Keymaker provides an integrated way for a user to upload their public SSH key to their IAM account with `keymaker upload_key`.

Run `keymaker install` on instances that you want your users to connect to. This installs three components:

- An `AuthorizedKeysCommand` sshd configuration directive, which acts as a login event hook and dynamically retrieves public SSH keys from IAM for the user logging in, using the default boto3 credentials (which default to the instance's IAM role credentials).
- A `pam_exec` PAM configuration directive, which causes sshd to call `keymaker-create-account-for-iam-user` early in the login process. This script detects if a Linux user account does not exist for the authenticating principal but an authorized IAM account exists with the same name, and creates the account on demand.
- A cron job that runs on your instance once an hour and synchronizes IAM group membership information. Only IAM groups whose names start with a configurable prefix (by default, `keymaker_*`) are synchronized as Linux groups.

As a result, users who connect to your instances over SSH are given access based on information centralized in your AWS account. Users must have an active IAM account with active matching SSH public keys in order for authentication to succeed. Users' UIDs and group memberships are also synchronized across your instances, so any UID-based checks or group-based privileges remain current as well.

CHAPTER 4

Cross-account authentication

Some AWS security models put IAM users in one AWS account, and resources (EC2 instances, S3 buckets, etc.) in a family of other federated AWS accounts (for example, a dev account and a prod account). Users then assume roles in those federated accounts, subject to their permissions, with sts:AssumeRole. When users connect via SSH to instances running in federated accounts, Keymaker can be instructed to look up the user identity and SSH public key in the other AWS account (called the "ID resolver" account).

Keymaker expects to find this configuration information by introspecting the instance's own IAM role description. The description is expected to contain a list of space-separated config tokens, for example, `keymaker_id_resolver_account=123456789012 keymaker_id_resolver_iam_role=id_resolver`. For `sts:AssumeRole` to work, the role `id_resolver` in account 123456789012 is expected to have a trust policy allowing the instance's IAM role to perform sts:AssumeRole on `id_resolver`.

Run the following command in the ID resolver account (that contains the IAM users) to apply this configuration automatically: `keymaker configure --instance-iam-role arn:aws:iam::987654321098:role/ROLE_NAME --cross-account-profile AWS_CLI_PROFILE_NAME`. Here, 987654321098 is the account ID of the federated account where EC2 instances will run, and AWS_CLI_PROFILE_NAME is the name of the AWS CLI role profile that you have set up to access the federated account.

CHAPTER 5

Requiring IAM group membership

Group membership is asserted if the instance's IAM role description contains the config token `keymaker_require_iam_group=prod_ssh_users`. The user logging in is then required to be a member of the **prod_ssh_users** IAM group. Apply this configuration automatically by running `keymaker configure --require-iam-group IAM_GROUP_NAME`.

CHAPTER 6

Security considerations

Integrating IAM user identities with Unix user identities has implications for your security threat model. With Keymaker, a principal with the ability to set SSH public keys on an IAM user account can impersonate that user when logging in to an EC2 instance. As an example, this can expand the scope of a compromised AWS secret key. You can mitigate this threat with an IAM policy restricting access to the UploadSSHPublicKey method.

CHAPTER 7

EFS integration

Email kislyuk@gmail.com for details on the EFS integration.

CHAPTER 8

Authors

- Andrey Kislyuk

CHAPTER 9

Links

- Project home page (GitHub)
- Documentation (Read the Docs)
- Package distribution (PyPI)

9.1 Bugs

Please report bugs, issues, feature requests, etc. on GitHub.

CHAPTER 10

License

Licensed under the terms of the Apache License, Version 2.0.

CHAPTER 11

API documentation

CHAPTER 12

Release Notes

CHAPTER 13

Changes for v1.0.7 (2018-07-16)

- Avoid trimming username if the suffix length is zero (#47)

CHAPTER 14

Changes for v1.0.6 (2018-05-25)

- Fix logic error in keymaker sync_groups
- Allow username suffix to be set in keymaker configure

CHAPTER 15

Changes for v1.0.5 (2018-05-21)

Fixup for get_uid with user suffix

CHAPTER 16

Changes for v1.0.4 (2018-05-18)

- Allow configurable username suffix in keymaker role config
- Auto-configure assume role permissions in keymaker configure
- Add missing iam:GetGroup permission for keymaker sync_groups (#42)

CHAPTER 17

Changes for v1.0.3 (2018-04-13)

- Remove unused dependency

CHAPTER 18

Changes for v1.0.2 (2018-04-12)

- Produce more readable log line when no config is found in role description

CHAPTER 19

Changes for v1.0.1 (2018-04-12)

- Fix user autovivification

CHAPTER 20

Changes for v1.0.0 (2018-04-05)

- For the avoidance of doubt, this tool is stable.

CHAPTER 21

Changes for v0.5.3 (2018-04-05)

- Remove unnecessary PAM config. Fixes #23
- Fix group sync on default iam_linux_group_prefix. Fixes #40

CHAPTER 22

Changes for v0.5.2 (2018-03-26)

- Make get_user, get_group, sync_groups cross account aware (#38)
- Add keymaker –version

CHAPTER 23

Changes for v0.5.1 (2018-02-05)

- keymaker configure: account autoconfiguration support (#30)

CHAPTER 24

Changes for v0.5.0 (2017-12-21)

- Introduce cross-account auth capability and group membership requirement (#29)
- PAM JIT user vivifier: Change ‘requisite’ to ‘optional’ (#22)
- Changing PAM behaviour to stop ugly output on first connection (#19)
- Adding shell and create-home to useradd command (#18)

CHAPTER 25

Changes for v0.4.3 (2017-05-25)

- Make the SSH hook work with RHEL-based distributions (#16)
- Test and documentation improvements

CHAPTER 26

Changes for v0.3.3 (2016-09-25)

- Fix release script

CHAPTER 27

Changes for v0.3.0 (2016-09-25)

- Python 2.7 support

27.1 Version 0.2.1 (2016-03-09)

- Further cleanup and documentation improvements.

27.2 Version 0.2.0 (2016-03-09)

- Cleanup and documentation improvements.

27.3 Version 0.1.0 (2016-03-06)

- Complete work on first iteration of CodeDeploy-compatible SSH public key API.

27.4 Version 0.0.2 (2016-03-06)

- Begin work on CodeDeploy-compatible SSH public key API.

27.5 Version 0.0.1 (2015-04-11)

- Initial release.

CHAPTER 28

Table of Contents