The AFP Project:
Secure Cloud Authentication for Machines and Humans

Valentin Haenel @esc____

2016-03-19 @ Chemnitzer Linux Tage

Version: v2.0 https://github.com/immobilienscout24/afp-talk

Outline

About Company

- Immobilienscout24
- @IS24_Developer

- 2000 (virtual-)machines in data-center
- 18 years in IT

About Me

IMMOBILIEN SCOUT 24

- Senior System Developer
- Doing Python and Cloud
- `@esc___`

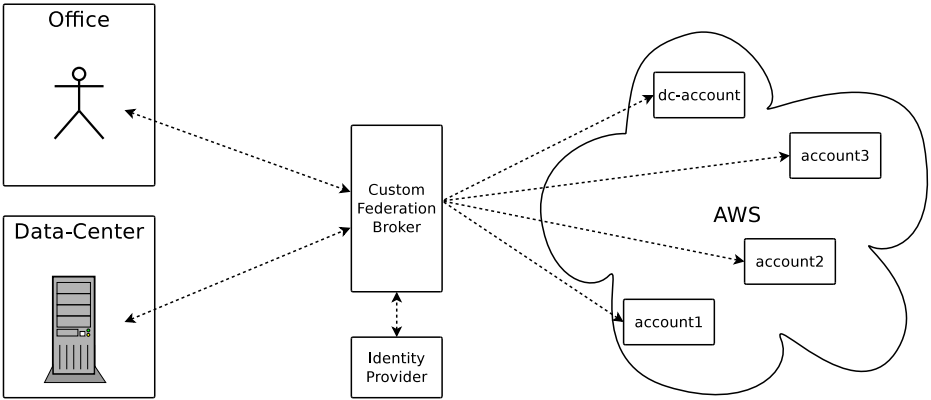
Outline

Our Problem

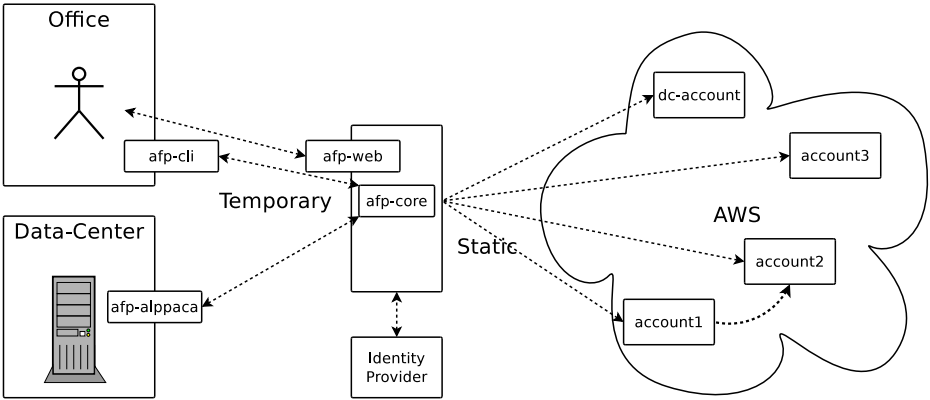
IMMOBILIEN SCOUT 24

- Connect the office to the cloud (no VPN)
- Making the 2000 machines in the data-center *cloud-ready*
- Easy and secure AWS service usage
- Temporary credentials for machines and humans

Our Solution



Enter: The AFP Project



IAM Users and Static Credentials

IMMOBILIEN SCOUT 24

- *Identity and Access Management* (IAM) users
- Static credentials
  - `AWS_ACCESS_KEY_ID` etc...
  - ssh-key
  - Basically anything w/o an expiry date
- How everyone starts with AWS
- This is Bad™

- → Mine Bitcoins if lost
- → Doesn't scale to larger organizations

What are Temporaray Credentials?

IMMOBILIEN SCOUT 24

- AWS feature: *Secure Token Service* (STS)
- Configure roles in you account that can be assumed
- A call to `STS:AssumeRole` yields temporary credentials
- Still needs a single technical IAM user
- However: this can be masqueraded (no shared password)

Multi-Account Paradigm

IMMOBILIEN SCOUT 24

- We use multiple AWS accounts
- (45 as of 2016-03-19)
- Goals:
  - Reduce blast radius in case of compromise
  - Modularise our data-center(s)
  - Feasible and incremental exit-strategy
- Rules:
  - Use well-defined interfaces
  - No VPN connections between accounts
  - Use secure puplic APIs

Outline

- *AWS Federation Proxy* (AFP)
- *Custom Federation Broker*
- Different providers
  - Linux group provider (humans)
  - IP provider (machines)

- A (backend-)webapplication
- One of the few places left that *does* have static credentials
- Uses *Secure Token Service* (STS) and *AssumeRole*

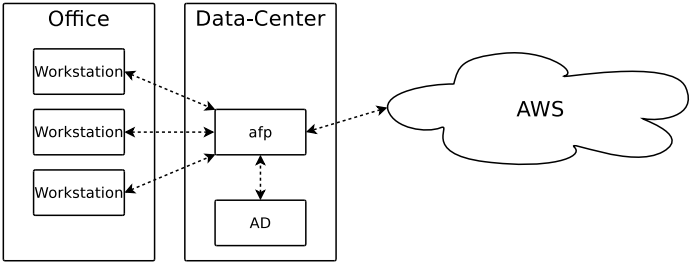
Connect *Active Directory* (AD) → AWS (Humans)

IMMOBILIEN SCOUT 24

- Special groups in AD: `aws-<ACCOUNT>-<ROLE>`
- AD users are members of such groups
- E.g. `aws_account1_read-only`
- → role: `read_only` in accout `account1`
- (Yes, the regex is configurable)

- Note: in our case it is AD, but it could be anything
- LDAP, Linux Users and Groups etc...

Schematic

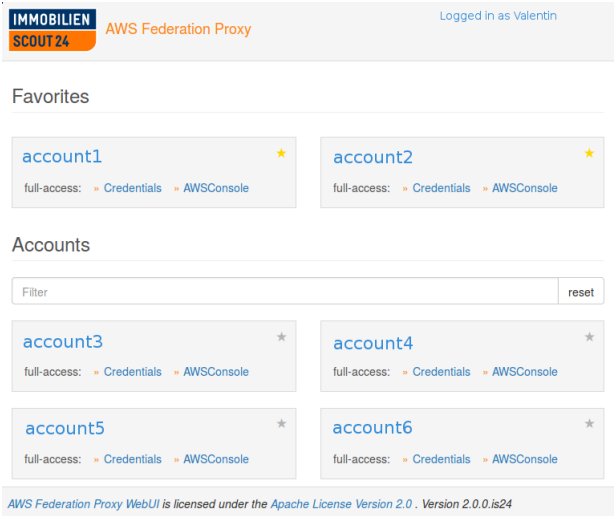

Office
Workstation
Workstation
Workstation
Data-Center
afp
AD
AWS

Outline

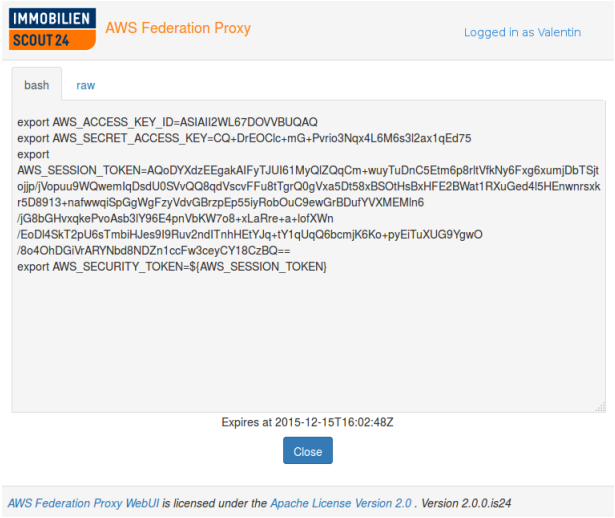
- Web-frontend for afp-core
- Give an overview of all available accounts and their usable roles for the logged-in user
- Get and show temporary credentials for the selected account and role
- Automatic login to the AWS management console for the desired account

- See also: Creating a URL that Enables Federated Users to Access the AWS Management Console (Custom Federation Broker)

Screenshot


IMMOBILIEN SCOUT 24
AWS Federation Proxy
Logged in as Valentin
Favorites
account1
full-access: » Credentials » AWSConsole
account2
full-access: » Credentials » AWSConsole
Accounts
Filter
reset
account3
full-access: » Credentials » AWSConsole
account4
full-access: » Credentials » AWSConsole
account5
full-access: » Credentials » AWSConsole
account6
full-access: » Credentials » AWSConsole
AWS Federation Proxy WebUI is licensed under the Apache License Version 2.0 . Version 2.0.0.is24

Screenshot


IMMOBILIEN SCOUT 24
AWS Federation Proxy
Logged in as Valentin
bash
raw
export AWS_ACCESS_KEY_ID=ASIAII2WL67DOVVBUQAQ
export AWS_SECRET_ACCESS_KEY=CQ+DrEOClc+mG+Pvrio3Nqx4L6M6s3l2ax1qEd75
export
AWS_SESSION_TOKEN=AQoDYXdzEEgakAIFyTJUI61MyQIZQqCm+wuyTuDnC5Etm6p8rltVfkNy6Fxg6xumjDbTSjt
ojjp/jVopuu9WQwemIqDsdU0SVvQQ8qdVscvFFu8tTgrQ0gVxa5Dt58xBSOtHsBxHFE2BWat1RXuGed4I5HEnwnrsxk
r5D8913+nafwwqiSpGgWgFzyVdvGBrzpEp55iyRobOuC9ewGrBDufYVXMEMln6
/jG8bGHvxqkePvoAsb3lY96E4pnVbKW7o8+xLaRre+a+lofXWn
/EoDl4SkT2pU6sTmbiHJes9I9Ruv2ndITnhHEtYJq+tY1qUqQ6bcmjK6Ko+pyEiTuXUG9YgwO
/8o4OhDGiVrARYNbd8NDZn1ccFw3ceyCY18CzBQ==
export AWS_SECURITY_TOKEN=${AWS_SESSION_TOKEN}
Expires at 2015-12-15T16:02:48Z
Close
AWS Federation Proxy WebUI is licensed under the Apache License Version 2.0 . Version 2.0.0.is24

Outline

afp-cli

IMMOBILIEN SCOUT 24

- Console/shell frontend for afp-core
- List available accounts and roles
- Open subshell with exported AWS creds
- Alternatively: display or write

Install

```
$ pip install afp-cli
```

- At our company: this just works!
- (Mainly because `https://afp` can be resolved)
  - Otherwise use:
  - → `api-url`(config)
  - → `--api-url`(command-line)

Use

```
$ afp
account1      read-only
account2      access-all-areas, read-only
$ afp account2
...
(AWS: account2/access-all-areas 59 min) $ aws s3 ls
...
```

Alternatives → see `README`

Outline

Instance Metadata Service (IMS)

IMMOBILIEN SCOUT 24

- Available on `169.254.169.254:80` on every EC2 instance
- Provides information about the machines
- Also provides temporary security credentials

```
$ curl 169.254.169.254/latest/meta-data/iam/security-credentials/
test_role
$ curl 169.254.169.254/latest/meta-data/iam/security-credentials/test_role
'{"Code": "Success",
  "AccessKeyId": "ASIAI",
  "SecretAccessKey": "oieDhF",
  "Token": "6jmePdXNehjPVt7CZ1WMkKrqB6zDc34d2vpLej",
  "Expiration": "2015-04-17T13:40:18Z",
  "Type": "AWS-HMAC"}'
```

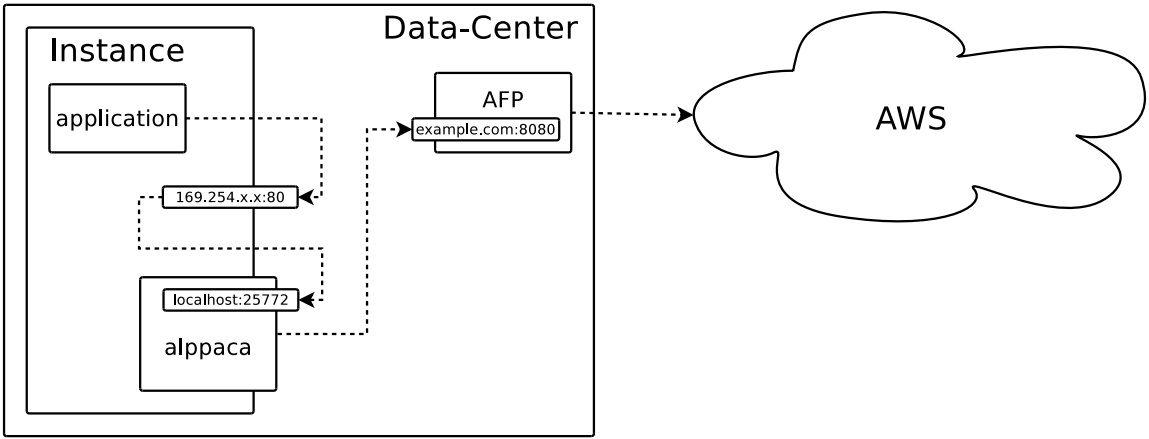
afp-alppaca

IMMOBILIEN SCOUT 24

- *A Local Prefetch Proxy for Amazon CredentiAls* (alppaca)
- Enables temporary credentials for machines
- Provides an *Instance Metadata Service* (IMS) compatible interface

- → Machines in data-center are *cloud-ready*
- (Yes, all 2000 of them can do this today!)

Schematic


Data-Center
Instance
application
169.254.x.x:80
localhost:25772
alppaca
AFP
example.com:8080
AWS

Details

IMMOBILIEN SCOUT 24

- Listens on `169.254.169.254:80`
  - Uses Linux firewall magic (`iptables`)
- Prefetches temporary credentials and stores them in memory
- Always has a valid set of credentials
  - Always able to deliver them in sub 1 second
  - No changes to Amazon SDKs needed
  - Attempt to refresh *before* the really expire
- Roles are in the special `dc-account`

Special Feature: Under the Hood `AssumeRole`

- afp-alppaca can be configured to perform an `AssumeRole` under the hood
- The data-center machine can appear to be in any one of our accounts
- Requires a role for that machine in that account and a *trusted principle* to enable the `AssumeRole` call

Further Uses: on EC2 in the Cloud

IMMOBILIEN SCOUT 24

- Run afp-alppaca on an EC2 instance?
- Yes, to allow cross-account access

- More Linux magic needed...

Outline

Alternatives?

IMMOBILIEN SCOUT 24

- hashicorp/vault
- AdRoll/hologram

- You tell me?

Kudos

Boris Petersen, Sebastian Spoerer, Michael Kuehne, Thomas Lehmann, Marco Hoyer, Oliver Schoenherr, Konrad Hosemann, Schlomo Schapiro, Robert J. Will, Sebastian Stiehl, Sebatian Herold, Enrico Heine, Ravi Yadav, László Károlyi, Sergej Domme, Stefan Neben, Tobias Vollmer, Tobias Höynck, Stefan Nordhausen, Valentin Haenel

As per `git-log` on 19.03.2016

Links

IMMOBILIEN SCOUT 24

- All of it is open-source!
- https://immobilienscout24.github.io/afp/

- Questions?