https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

第42回FreeBSDワークショップ

佐藤　広生 <hrs@FreeBSD.org>

東京工業大学/ FreeBSD Project

2018/6/29

開催背景

- **日本国内の＊BSDユーザ活動を活発化させましょう**
  - 月1回、東京近辺で定期的な会合を。
  - 講演を聞くだけでなく、話を持ち寄って双方向に議論しましょう
  - 困ったことや要望などはなるべく拾っていきます

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

オーガナイザの自己紹介

- 名前：佐藤 広生
- FreeBSD コアチームメンバ(2006-2020)、
  リリースエンジニア(2006-)
- FreeBSD Foundation 理事(2008-)
- その他の＊BSD／オープンソース関連の活動いろいろ
- 東京工業大学助教(2009-)

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

本ワークショップの進行

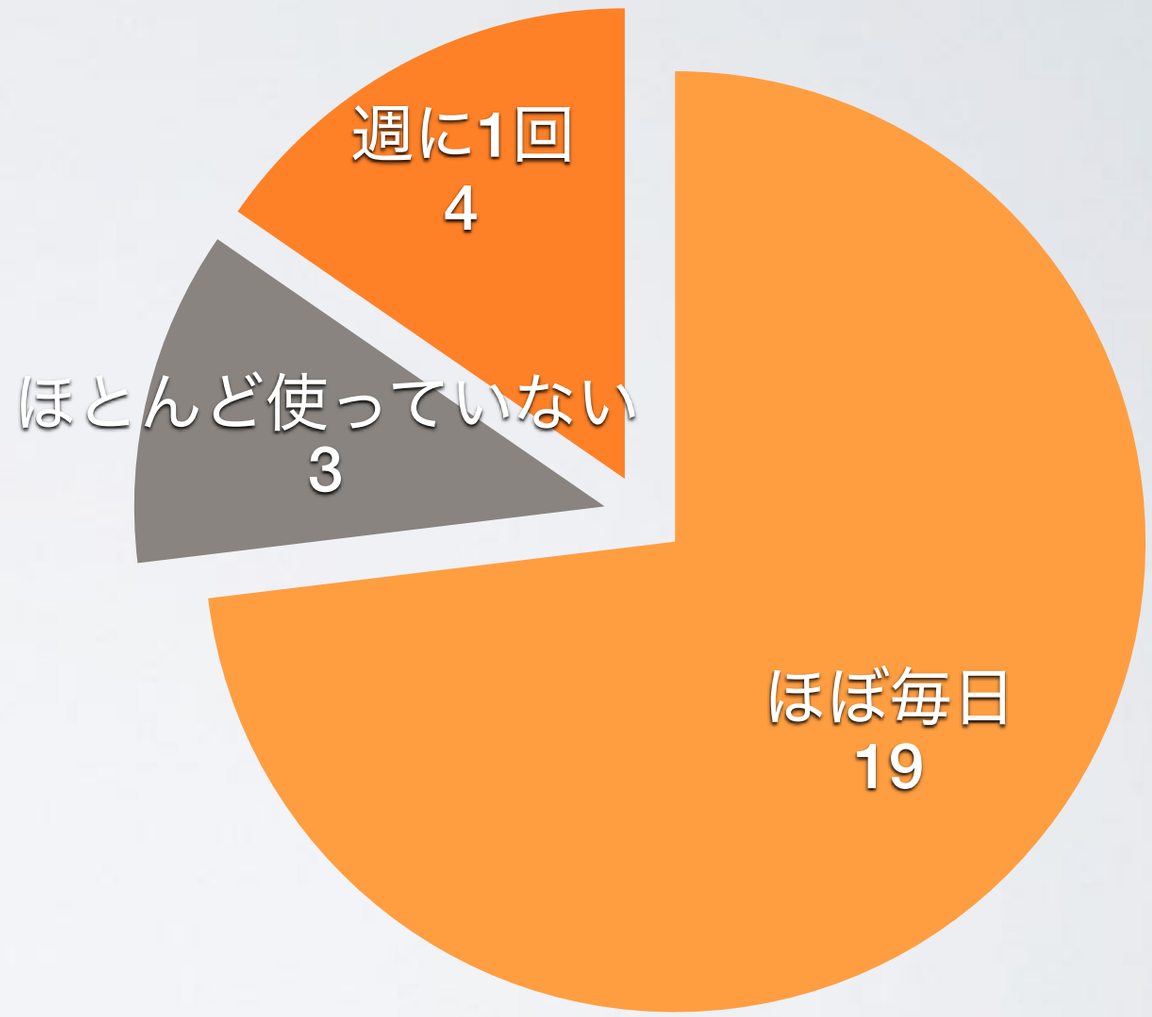
- 19:00〜19:20　自己紹介＋話題にしたいトピック
- 19:20〜21:00　ライトニングトーク

意見は自由に発言ください！

自己紹介タイム

- 名前（所属）
- 開発者 or 利用者
- 興味がある／話題に
  したい内容

をどうぞ



今回の出席者内訳：新規2名、再参加者24名

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

メモ

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

告知

- FreeBSDワークショップ（ほぼ月一回）
- 次回は7月26日
- 発表の提案、具体的な解説の要望は随時歓迎

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

FreeBSDワークショップ

30分で分かるKerberos

佐藤 広生 <hrs@allbsd.org>

東京工業大学 / FreeBSD Project

2018/6/29

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

Kerberos

- 名前は知ってるけど使ったことはない
- なんか面倒そう
- 管理しているユーザ数が少ないので不要と思う

Kerberos

何ができるのか

- /etc/{shadow,master.passwd}の
  パスワード部分の一元管理？

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

Kerberos

何ができるのか

- /etc/{shadow,master.passwd}の
  パスワード部分の一元管理？

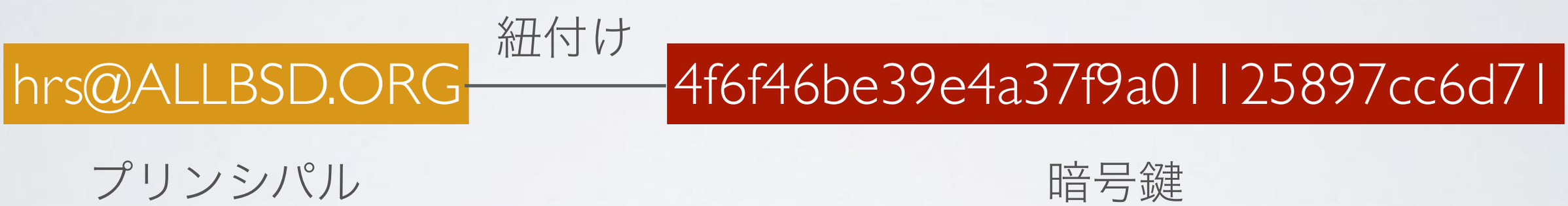
実現できるが、それだけが特長ではない
（というか、それが主目的だと思うと面倒）

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

Kerberos

何ができるのか

- サービスパスワード（暗号鍵）の管理



- ログインに使う、メールサーバのアクセスに使う、色々可能。
- 置き換えではなく、補完的にも使える

簡単セットアップ

- **構成要素**
  - kdc（鍵サーバ、必須）: 88/tcp
  - kadmind（リモート管理サーバ）: 749/tcp
  - kpasswdd（管理者以外が使えるパスワード変更サーバ）:464/tcp
- **設定ファイル・ディレクトリ**
  - `/var/heimdal/kadmind.acl`: **kadmind** のアクセス制御 (0640)
  - `/var/heimdal/m-key`: マスター暗号鍵(0600)
  - `/var/heimdal/heimdal.db`: 暗号鍵データベース(0600)

簡単セットアップ

- **設定前に決めなければならないもの**
  - **レルム：認証の名前空間**
    - 普通はドメインパートを大文字にしたものを使う
    - 例：ALLBSD.ORG
  - **管理者のユーザ名（プリンシパル）**
    - 自分の普段使っているユーザ名に /admin を付ける
    - ユーザ名の後ろに「@レルム」を付ける
    - 例：hrs/admin@ALLBSD.ORG
    - 書式は「プリンシパル名/インスタンス名@レルム」

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

簡単セットアップ

- ステップ
  - マスター暗号鍵を設定
    - 管理者がデータベースを操作できるように設定
    - 管理者を追加
    - デーモン起動
    - 一般ユーザを追加

```
# kstash
Master key:
Verifying - Master key:
kstash: writing key to `/var/heimdal/m-key'
#
```

簡単セットアップ

- ステップ
  - マスター暗号鍵を設定
  - 管理者がデータベースを操作できるように設定
  - 管理者を追加
  - デーモン起動
  - 一般ユーザを追加

```
# (echo "hrs/admin@ALLBSD.ORG all *";
   echo "hrs/admin@ALLBSD.ORG all */admin";
   echo "hrs/admin@ALLBSD.ORG all host/*@*";
) > /var/heimdal/kadmind.acl
#
```

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

簡単セットアップ

- ステップ
  - マスター暗号鍵を設定
  - 管理者がデータベースを操作できるように設定
  - 管理者を追加
  - デーモン起動

```
# kadmin -l add hrs/admin@ALLBSD.ORG
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
hrs/admin@ALLBSD.ORG's Password:
Verifying - hrs/admin@ALLBSD.ORG's Password:
#
```

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

簡単セットアップ

- ステップ
  - マスター暗号鍵を設定
  - 管理者がデータベースを操作できるように設定
  - 管理者を追加
  - デーモン起動

```
# (echo "kdc_enable=YES";
   echo "kadmind_enable=YES";
   echo "kpasswdd_enable=YES";
  ) >> /etc/rc.conf
# service kdc start
# service kadmind start
# service kpasswdd start
```

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

簡単セットアップ

- **ステップ**
  - マスター暗号鍵を設定
  - 管理者がデータベースを操作できるように設定
  - 管理者を追加
  - デーモン起動
  - 一般ユーザを追加

```
% kadmin -p hrs/admin@ALLBSD.ORG add hrs@ALLBSD.ORG
```

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

ここまでできたこと

- **プリンシパル（ユーザ）と暗号鍵のペアを記録するサーバの構築**
  - kadmin コマンドで追加・削除できる
  - 「ユーザアカウントそのものの追加」とは関係なし
  - 管理対象は「暗号鍵」

- **Kerberos が提供するもの**

  ＝ プリンシパルの所有者であることの証明

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

- **この暗号鍵を使ってSSHでアクセスできるようにしたい**

- **SSHのアクセスで保証されないといけないこと**

  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

- **この暗号鍵を使ってSSHでアクセスできるようにしたい**

- **SSHのアクセスで保証されないといけないこと**
  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？

- **hrs@ALLBSD.ORG の正当な所有者とは？**
  ＝パスワードを知っている人（鍵を持っている人）

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

- この暗号鍵を使ってSSHでアクセスできるようにしたい

```
% kinit hrs@ALLBSD.ORG
hrs@ALLBSD.ORG's Password:
% klist
Credentials cache: FILE:/tmp/krb5cc_20001
        Principal: hrs@ALLBSD.ORG

  Issued                Expires               Principal
Jun 29 12:59:07 2018  Jun 29 22:59:07 2018  krbtgt/ALLBSD.ORG@ALLBSD.ORG
```

- hrs@ALLBSD.ORG の正当な所有者とは？
  ＝パスワードを知っている人（鍵を持っている人）
- kinit コマンドで、正当な所有者であることを証明できる。
  証明するとチケットが発行される。ssh(1)はチケットを利用する

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

- **この暗号鍵を使ってSSHでアクセスできるようにしたい**

- **SSHのアクセスで保証されないといけないこと**

  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？

- **ではアクセス先マシンの正当な所有者とは？**

  → 対応するプリンシパルを作成する。パスワードを
  手入力できないので、暗号鍵をサーバにコピーしておく

使い方

- この暗号鍵を使ってSSHでアクセスできるようにしたい

```
% kadmin -p hrs/admin@ALLBSD.ORG add host/machine.allbsd.org@ALLBSD.ORG
hrs/admin@ALLBSD.ORG's Password:
...

# kadmin -p hrs/admin@ALLBSD.ORG ext_keytab \
    -k /etc/krb5.keytab host/machine.allbsd.org@ALLBSD.ORG
```

- kadmin add "host/FQDN": SSH等に使う「マシン」証明用の名前
- kadmin ext_keytab ... : 暗号鍵を取り出して /etc/krb5.keytab に追加

使い方

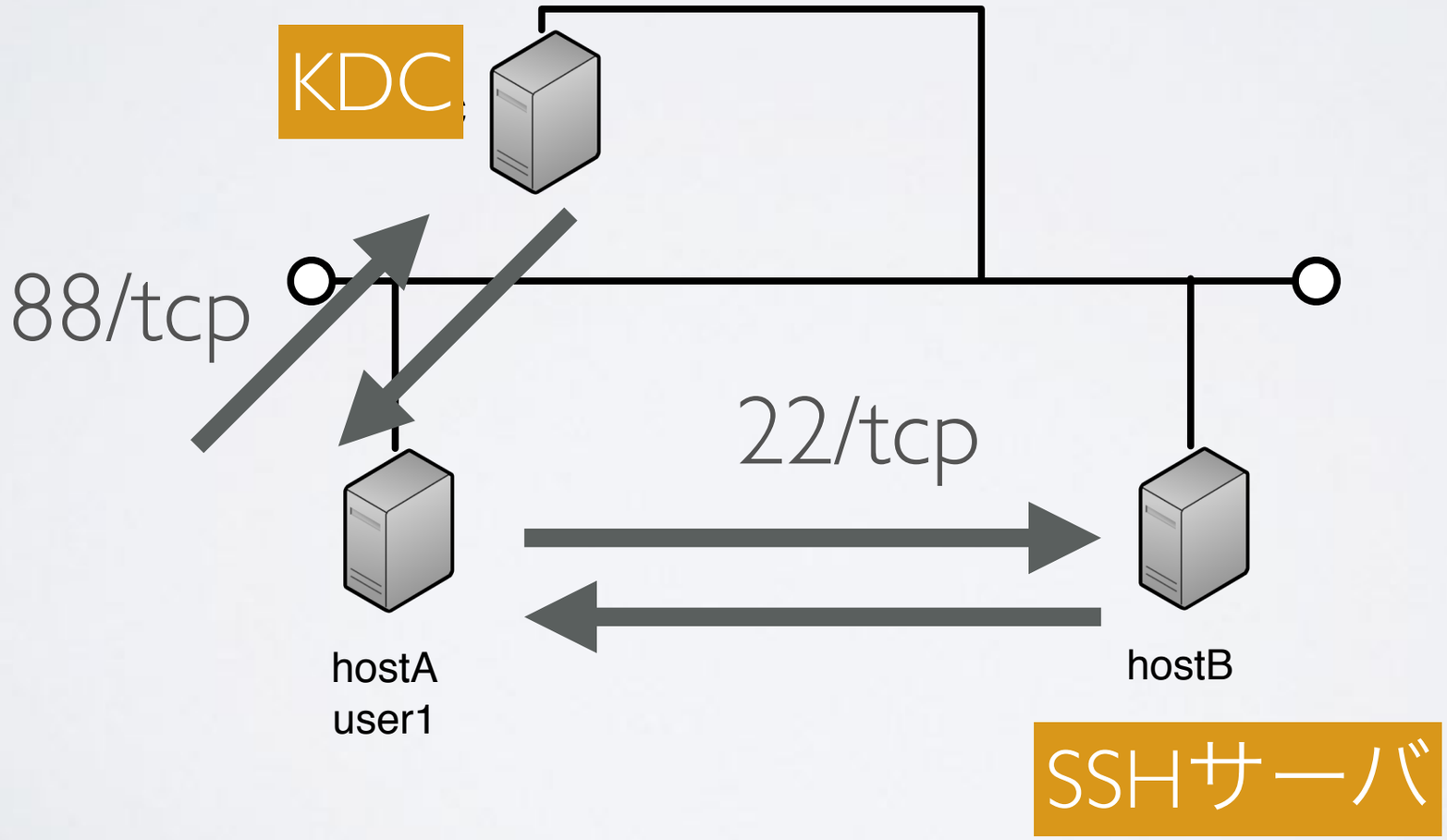
- **SSHのアクセスで保証されないといけないこと**
  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？

    - **ssh は、host/machine.allbsd.org@ALLBSD.ORG の鍵を KDC サーバに要求する (88/tcp) 。要求すると、「要求したユーザ名を、要求した鍵で暗号化したデータ」が得られる**
    - **sshd は、host/machine.allbsd.org@ALLBSD.ORG の鍵を /etc/krb5.keytab から読み込む。**
    - **sshd は暗号化したデータを受け取り、鍵で復号する**
    - 復号できれば、ユーザとマシンは両方とも真正である

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

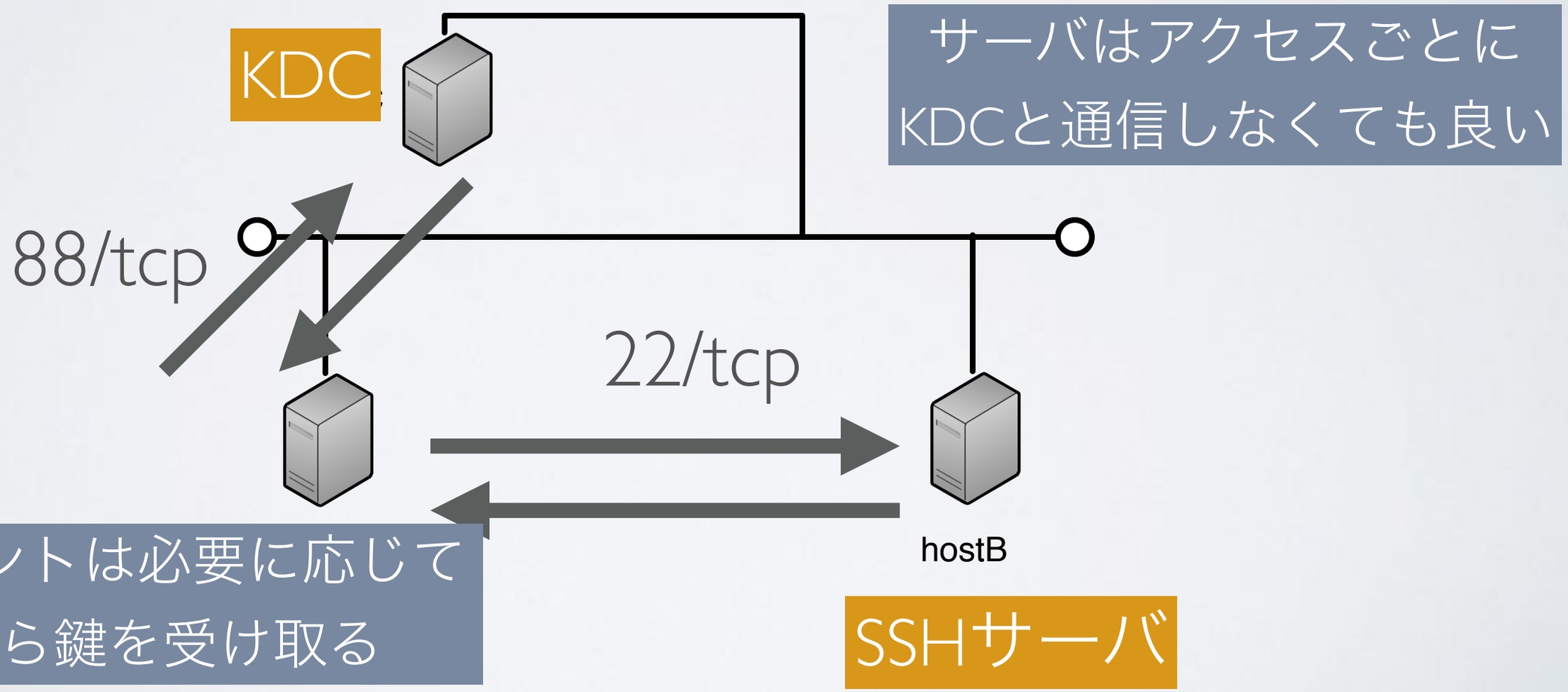
- **SSHのアクセスで保証されないといけないこと**
  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？



https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

使い方

- **SSHのアクセスで保証されないといけないこと**
  1. アクセス先（sshd 動いているマシン）は、本物か？
  2. アクセスしようとしているユーザは、本物か？



SSHの設定

- デフォルトでは Kerberos 認証は off になっている

```
GSSAPIAuthentication=yes（sshd_config と ssh_config 両方）
GSSAPIDelegateCredentials=yes（ssh_config：フォワードを許可）
```

- sshがKDCの場所を知っていないといけない

```
# /etc/krb5.conf
[libdefaults]
    default_realm = ALLBSD.ORG
[realms]
    ALLBSD.ORG ={
        kdc = kdc.allbsd.org
    }
```

- DNSのSRV RRとTXT RRで、レルムとKDCの設定ができる

SSHの設定

- **authorized_keys を使った公開鍵認証と比べて何が嬉しいの？**
  - 信用できなくなったユーザ・マシンからのアクセスを
    すぐに遮断できる
  - screen や tmux を使っていて ssh-agent と組み合わせて使うのが
    面倒だな、というケースでも kinit なら無問題
- ログイン時のパケット交換回数がpubkeyよりも少ないので
  ログインが速い。

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

一般的な設定方法

- **サービスに対応するプリンシパルを追加**
  - MTA (sendmail等）: smtp/mail.allbsd.org@ALLBSD.ORG
  - HTTPサーバ: HTTP/www.allbsd.org@ALLBSD.ORG
  - サービス・アプリケーションで決まっている
    - sshdがホスト鍵を作成するのと同じように、host/FQDNをよく使う
- **クライアントの設定**

  kinitでパスワードを入力してKDCからチケットを得るか、

  あらかじめkeytabファイルとして鍵を抽出しておく
- **サーバの設定**

  keytabファイルを抽出しておくことが多い

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

一般的な設定方法

- **FQDNが必要ということはmymachine.local とかやってるとだめ？**
  - /etc/hosts に書いてあれば良い
- **krb5.keytab とかに入っている鍵ってそもそも何なの？**
  - パスワードからKDF（鍵導出関数）で生成されたデータ
  - パスワードと1:1関係がある
  - 同じパスワードを使うと、同じ鍵になってしまう
  - host/FQDN とかは、kadmin add --random-key を指定して ランダムな鍵を自動設定することが多い

NFSの場合は？

- **NFSv4 は Kerberos が「使える」（必須ではない）**
- **サーバの設定**
  - ホスト名に対応するnfs/server.allbsd.org@ALLBSD.ORGを作成
  - 暗号鍵を/etc/krb5.keytabとして取り出しておく
  - nfsuserd(8) を起動 (-domain=ALLBSD.ORG を指定）
  - gssd(8) を起動（-h フラグを指定）
  - /etc/exports に -sec=krb5p:sys を追加
    （sysがあれば従来のアクセスも許可）

NFSの場合は？

- **NFSv4 は Kerberos が「使える」（必須ではない）**
- クライアントの設定
  - ホスト名に対応するnfs/client.allbsd.org@ALLBSD.ORGを作成
  - 暗号鍵を/etc/krb5.keytabとして取り出しておく
  - nfsuserd(8) を起動 (-domain=ALLBSD.ORG を指定）
  - mountオプションに -nfsv4,sec=krb5p,gssname=nfs を追加
    （gssname=nfs は、プリンシパル名に対応。hostでも可。）

NFSの場合は？

- **hrs@ALLBSD.ORG というユーザでのアクセスは許可されるのか？**
  - **mountはクライアントの鍵（nfs/client.allbsd.org）で行われる**
    → rootが/etc/krb5.keytabを使ってマウント
  - **アクセスはクライアントの鍵とユーザの鍵で行われる。**
    → kinitしないと、アクセスしても拒否される
- **nfsuserd(8)がプリンシパルをUID/GIDに変換して解釈する**
  → これが起動していないとアクセスしても拒否される
- **kinitして、nfsuserd(8)が上がっていれば、**
  プリンシパル名からレルムを削った名前をUID/GIDに変換

日々の管理

- **プリンシパルの追加・削除・変更**
  - KDCにアクセスできれば kadmin コマンドでどこからでも可能
- **パスワードだけ変更**
  - kpasswd コマンドで変更できる(kpasswdd が上がっている必要がある)
- **kinit するのが面倒**
  - /etc/pam.d/system にある pam_krb5 の行を有効にすると、ログイン時に自動でkinitするようになる
- **ログファイル**
  - kdc, kadmind, kpasswd は syslog に出せる。/etc/krb5.conf を設定

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

高度な話

- **KDCが落ちてたらログインできないぞ**
  - KDCは複数用意して冗長化・自動同期可能。ローカルに上げるのもあり。
- **クライアントに個々に設定を入れるのが面倒**
  - DNSサーバにTXTとSRVを入れましょう。
- **88/tcp なんて今時のネットワークじゃ通らない**
  - HTTP proxy を経由させることが可能。/etc/krb5.conf に設定

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

高度な話

- **KDCはbrute force attackできるので、パスワードがそもそも嫌。**
  - kinit は公開鍵認証でも可能（PK-INITと呼ばれる）
  - PK-INIT: クライアント証明書を配って、それを鍵として使う

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

まとめ

- **Kerberosは複雑なシステムではない**
  - 単に foo@REALM と鍵（パスワード）を格納したデータベース
  - ユーザ、マシン、サービス（＝プリンシパル）に鍵を割り当てる
  - 「本当に割り当てた対象本人なのか」を証明してくれる
  - 使用中の認証に追加する形で導入できる
- **Kerberosを使うサービスの設定は?**
  - 用意すべきプリンシパルと、その鍵をどこに置けば良いのかを
    マニュアルを読んで調べる
  - Active Directory等は、このあたりを全部
    意識せずに使えるように工夫してあるだけで、やってることは同じ

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

おしまい

- 質問はありますか？

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

FreeBSDワークショップ

90年代の設定ファイル管理

佐藤 広生 <hrs@allbsd.org>

東京工業大学 / FreeBSD Project

2018/6/29

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

設定ファイルの管理

- **何を使ってやる？**
  - テキストファイルを単にコピー＆ファイル名に日付
  - 昔懐かしRCS
  - CVS とか Subversion とか git とか。
- **VCSを使ったとして、設定ファイルのコピーとかは？**
  - 昔懐かし rdist
  - tar とか rsyncで強引に上書き
  - Puppet とか Chef とか Ansible とか？

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

設定ファイルの管理

- **何ができれば良いのか**
  - バージョン管理
  - 今の設定との差分をとる
  - 設定ファイルを更新した時の再起動等の定型処理

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

設定ファイルの管理

- **何ができれば良いのか**
  - バージョン管理
  - 今の設定との差分をとる
  - 設定ファイルを更新した時の再起動等の定型処理

Makefile でさくっと書く方法

http://people.allbsd.org/~hrs/FreeBSD/config.sample.20180629.tar.gz

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

設定ファイルの管理

- **bsd.config.mk**
  - bsd.prog.mk をベースにした wrapper
    - make install: インストール
    - make diff: 差分
    - make merge: マージ
  - 例：etc というディレクトリを作成して Makefile に次のように書く

```
FILESDIR= /etc
FILES= rc.conf

beforeinstall:
    install -o root -g wheel -d ${FILESDIR}

.include <bsd.prog.mk>
```

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

設定ファイルの管理

デモ

2018/6/29 (c) Hiroki Sato

https://people.allbsd.org/~hrs/FreeBSD/sato-FBSDW20180629.pdf

おしまい

- 質問はありますか？