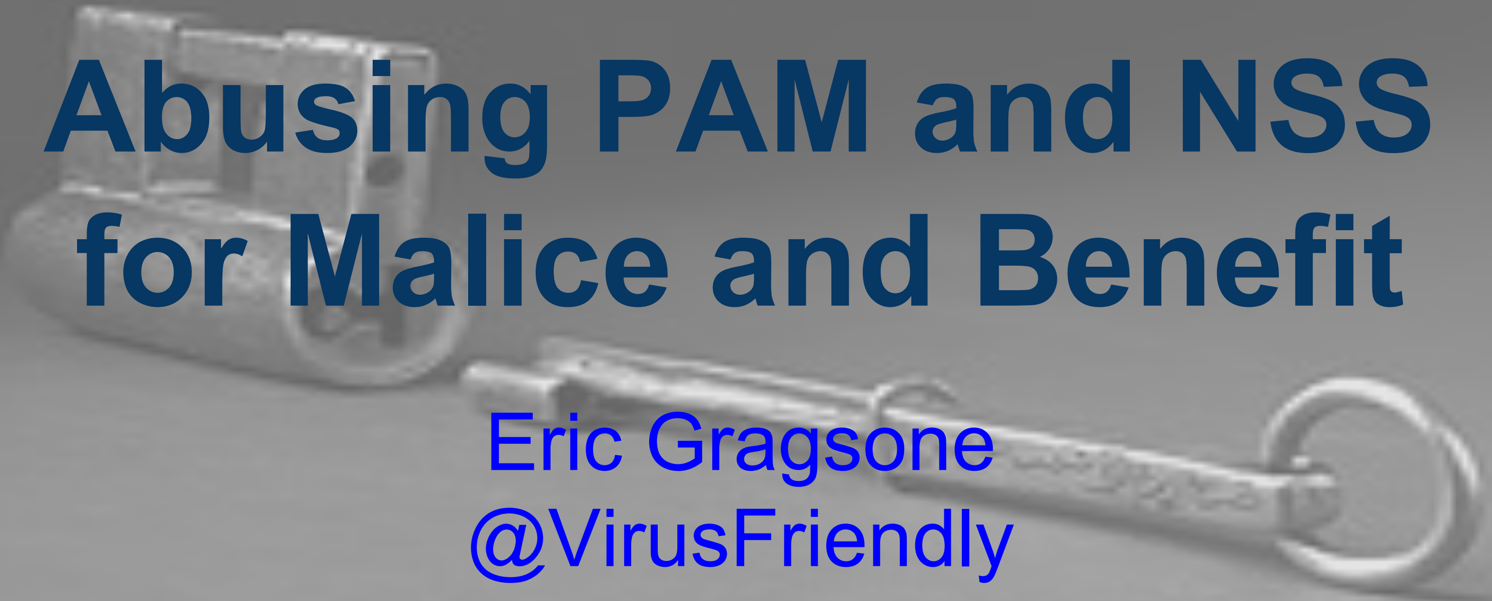
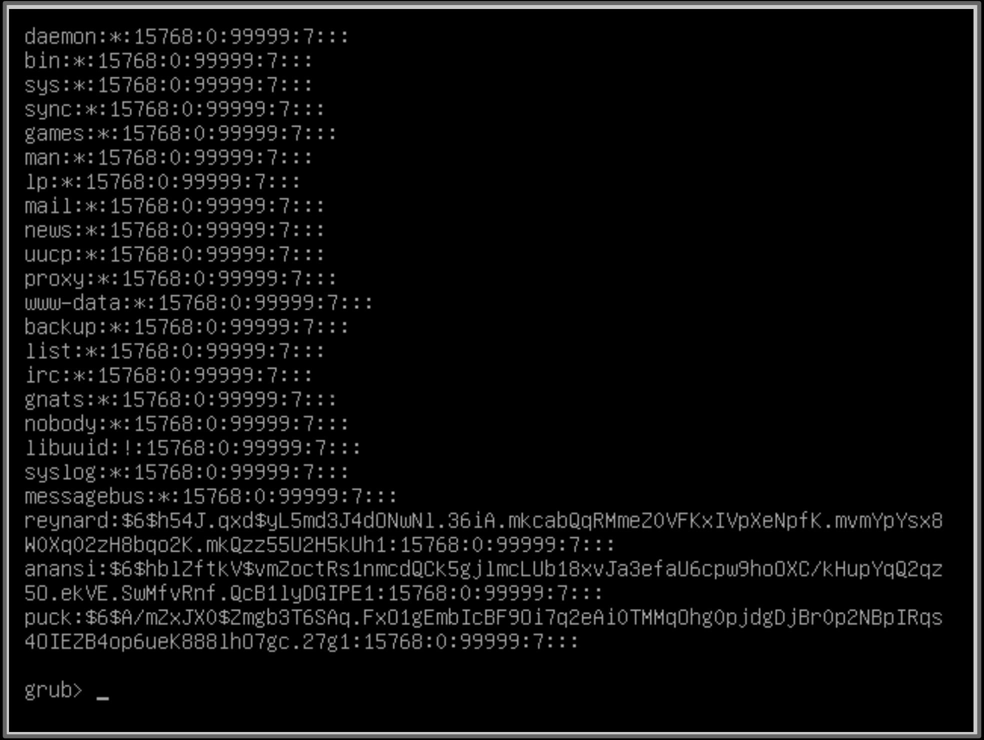


Abusing PAM and NSS for Malice and Benefit

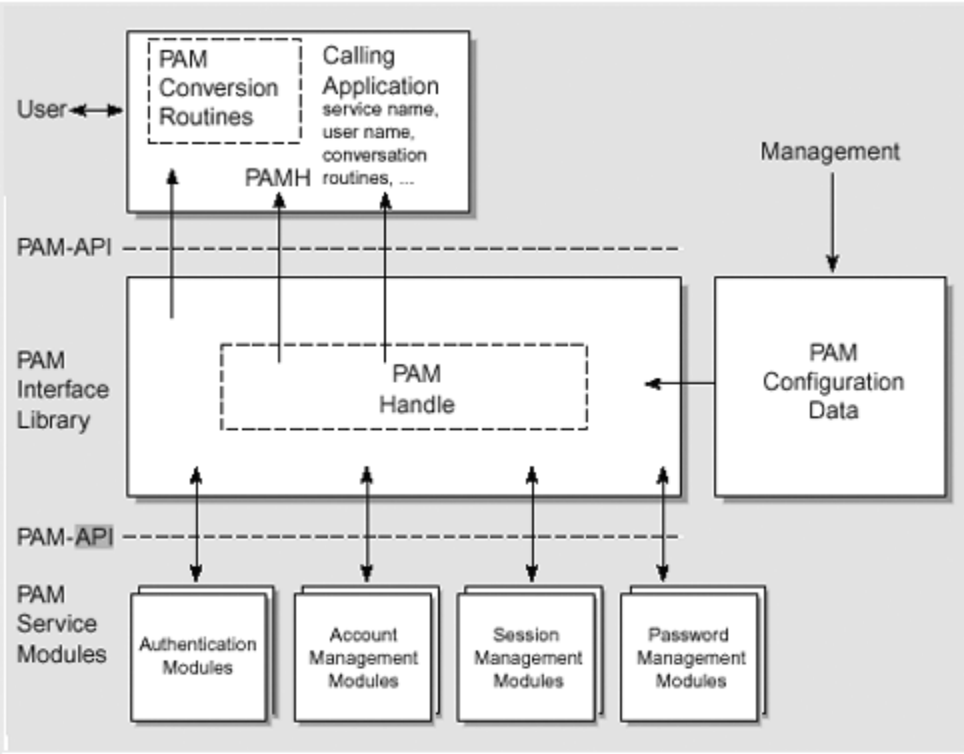
Eric Gragsone
@VirusFriendly

passwd, shadow, group



```
daemon:*:15768:0:99999:7:::
bin:*:15768:0:99999:7:::
sys:*:15768:0:99999:7:::
sync:*:15768:0:99999:7:::
games:*:15768:0:99999:7:::
man:*:15768:0:99999:7:::
lp:*:15768:0:99999:7:::
mail:*:15768:0:99999:7:::
news:*:15768:0:99999:7:::
uucp:*:15768:0:99999:7:::
proxy:*:15768:0:99999:7:::
www-data:*:15768:0:99999:7:::
backup:*:15768:0:99999:7:::
list:*:15768:0:99999:7:::
irc:*:15768:0:99999:7:::
gnats:*:15768:0:99999:7:::
nobody:*:15768:0:99999:7:::
libuuid:!:15768:0:99999:7:::
syslog:*:15768:0:99999:7:::
messagebus:*:15768:0:99999:7:::
reynard:$6$h54J.qxd$yL5md3J4dONwNl.36iA.mkcabQqRMmeZOVFKxIVpXeNpfK.mvmYpYsx8
WOXqO2zH8bqo2K.mkQzz55U2H5kUh1:15768:0:99999:7:::
anansi:$6$hblZftkV$vmZoctRs1nmcdQCk5gjlmcLUb18xvJa3efaU6cpw9hoOXC/kHupYqQ2qz
5O.ekVE.SwMfvRnf.QcB1lyDGIPE1:15768:0:99999:7:::
puck:$6$A/mZxJXO$Zmgb3T6SAq.FxO1gEmbIcBF9Oi7q2eAiOTMMqOhgOpjdgDjBrOp2NBpIRqs
4OIEZB4op6ueK888lhO7gc.27g1:15768:0:99999:7:::

grub> _
```

PAM Architecture



PAM - Pluggable Authentication Modules

Basic config line:

[action] [priority] [module]

http://www.linux-pam.org/

PAM - Pluggable Authentication Modules

Actions

- account - access limitations
- auth - checks authentication
- password - updates authentication
- session - post login (logs, etc)

PAM - Pluggable Authentication Modules

Priority

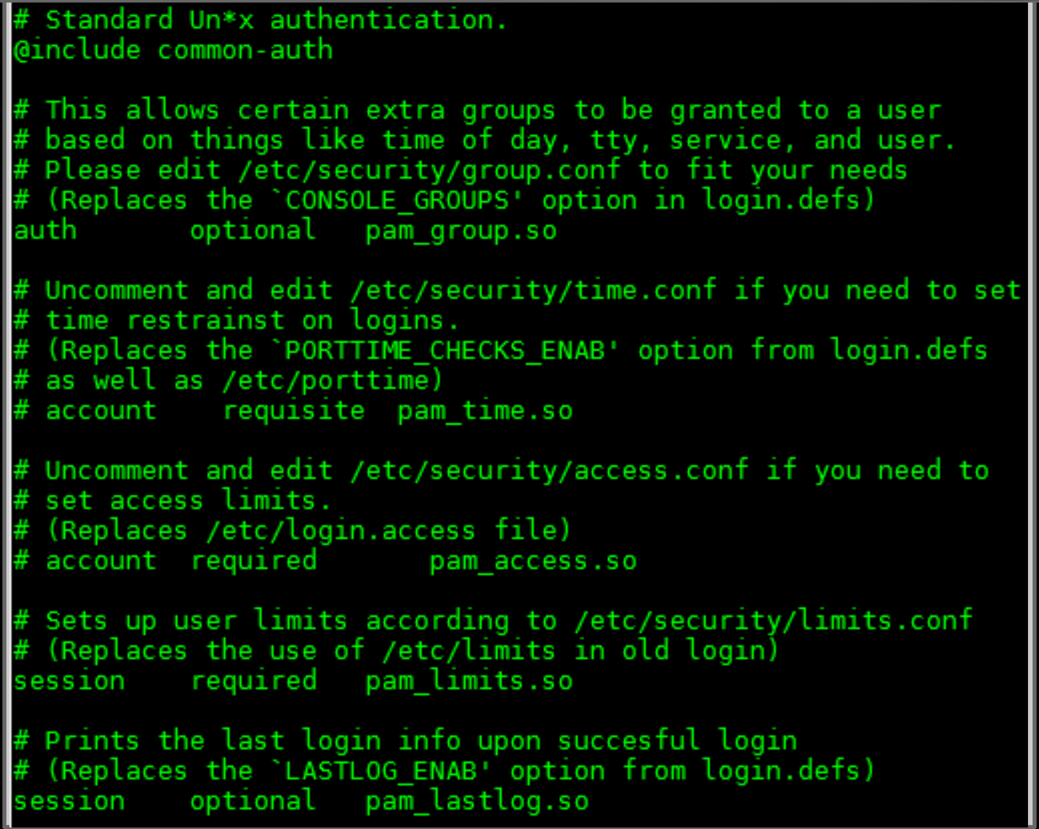
- requisite - If fail, end stack. Stack fails.
- required - If fail, continue. Stack fails.
- sufficient - End stack if no required fails
- optional - ignored if one non-optional runs

PAM - Pluggable Authentication Modules

Basic PAM Modules

- pam_deny
- pam_exec
- pam_nologin
- pam_permit

PAM - Pluggable Authentication Modules



```
# Standard Un*x authentication.
@include common-auth

# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
auth       optional   pam_group.so

# Uncomment and edit /etc/security/time.conf if you need to set
# time restraint on logins.
# (Replaces the `PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)
# account    requisite  pam_time.so

# Uncomment and edit /etc/security/access.conf if you need to
# set access limits.
# (Replaces /etc/login.access file)
# account  required       pam_access.so

# Sets up user limits according to /etc/security/limits.conf
# (Replaces the use of /etc/limits in old login)
session    required   pam_limits.so

# Prints the last login info upon succesful login
# (Replaces the `LASTLOG_ENAB' option from login.defs)
session    optional   pam_lastlog.so
```

PAM - Pluggable Authentication Modules

Custom PAM Module ideas

- Backdoor without modify daemon
- Fix authentication to a specific passwords
- Get a copy of passwords when changed
- Log passwords in general

CatchAll - Logging Nonexistent Accounts

Pronto’s SSH-Rankings

https://sshrank.in/g/lists/24hr

Twitter @sshbrute

Logs user names attempted by attackers

Organized by IP

CatchAll - Logging Nonexistent Accounts

CatchAll inspired by SSH-Rankings

https://github.com/maetrics/CatchAll

- Attribution based on passwords
- Enhancing password dictionaries

CatchAll - Logging Nonexistent Accounts

Password grabbing is done by pam_catchall.c

pam_sm_authenticate()

API counterpart of pam_authenticate()

sm - service module

CatchAll - Logging Nonexistent Accounts

CatchAll avoid legitimate accounts to prevent leaking real data.

Only affects accounts with no passwords or CatchAll in the GECOS field.

Performs fake crypt() call to prevent timing attacks

CatchAll - Logging Nonexistent Accounts

Kept receiving #010#012#015#177INCORRECT for all passwords.

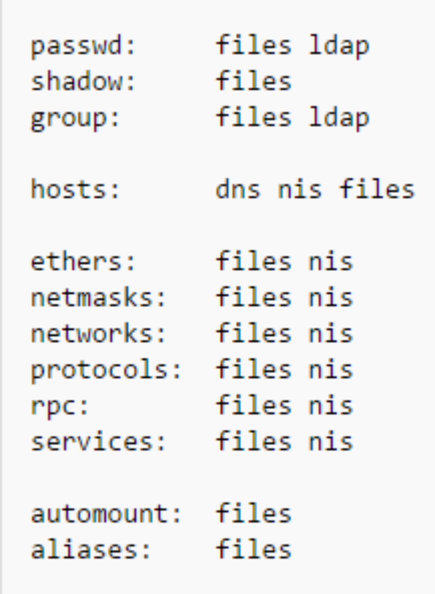
Research discovered this is how OpenSSH makes calls to prevent timing attacks.

NSS - Name Service Switch

Needed to trick OpenSSH into thinking user exists.

Enter NSS. Responsible for Name and common DB resolution.

NSS - Name Service Switch



```
passwd:     files ldap
shadow:     files
group:      files ldap

hosts:      dns nis files

ethers:     files nis
netmasks:   files nis
networks:   files nis
protocols:  files nis
rpc:        files nis
services:   files nis

automount:  files
aliases:    files
```

NSS - Name Service Switch

NSS is used for the following calls

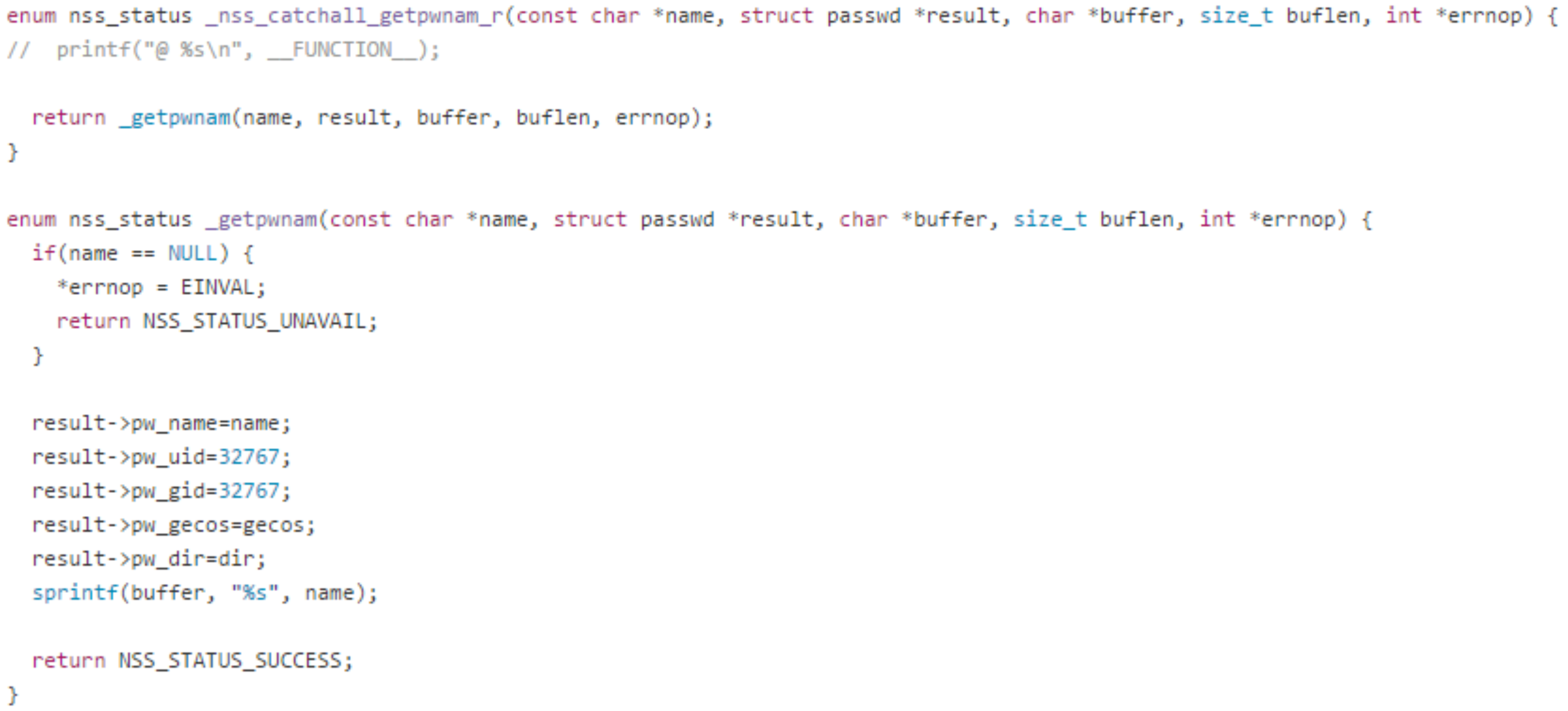
- setpwent()
- endpwent()
- getpwent()
- getpwnam()
- and more!

NSS - Name Service Switch

NSS is used for the following calls

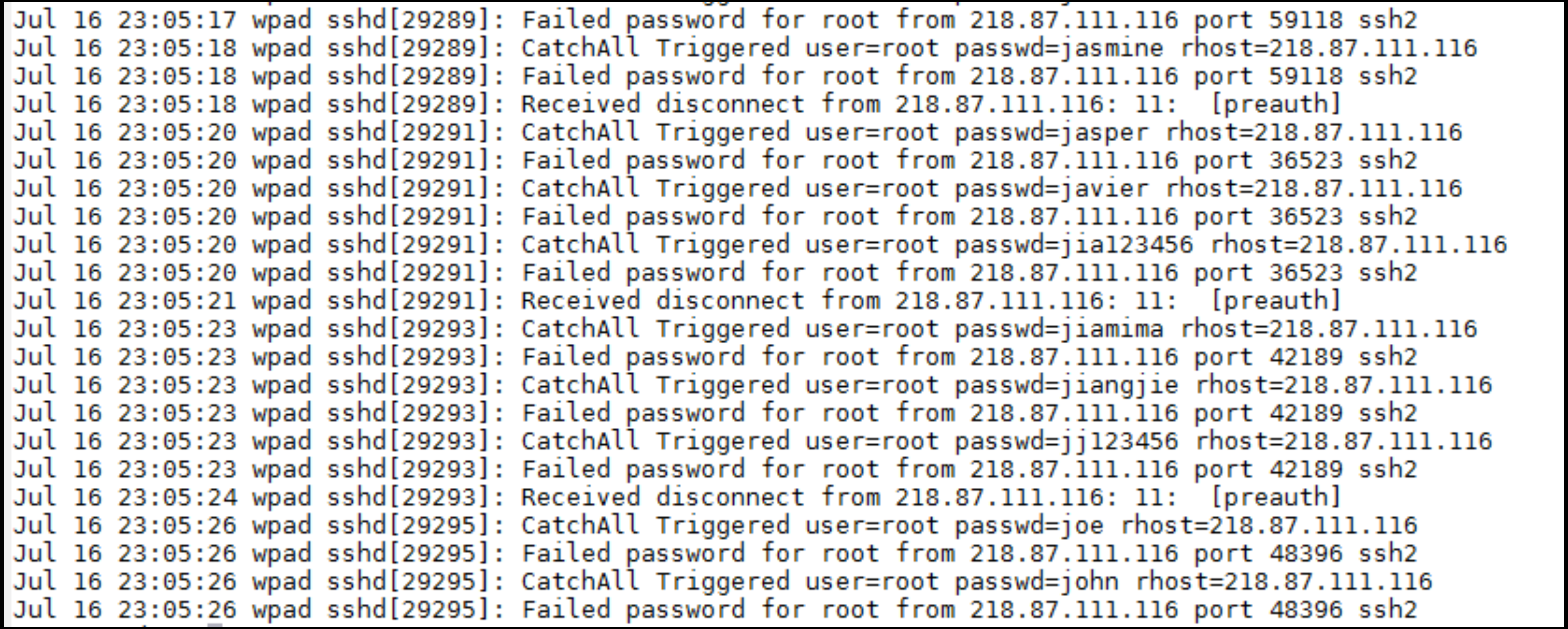
- _nss_catchall_setpwent()
- _nss_catchall_endpwent()
- _nss_catchall_getpwent()
- _nss_catchall_getpwnam()
- and more!

NSS - Name Service Switch



```c
enum nss_status _nss_catchall_getpwnam_r(const char *name, struct passwd *result, char *buffer, size_t buflen, int *errnop) {
//  printf("@ %s\n", __FUNCTION__);

  return _getpwnam(name, result, buffer, buflen, errnop);
}

enum nss_status _getpwnam(const char *name, struct passwd *result, char *buffer, size_t buflen, int *errnop) {
  if(name == NULL) {
    *errnop = EINVAL;
    return NSS_STATUS_UNAVAIL;
  }

  result->pw_name=name;
  result->pw_uid=32767;
  result->pw_gid=32767;
  result->pw_gecos=gecos;
  result->pw_dir=dir;
  sprintf(buffer, "%s", name);

  return NSS_STATUS_SUCCESS;
}
```

CatchAll - Logging Nonexistent Accounts



```
Jul 16 23:05:17 wpad sshd[29289]: Failed password for root from 218.87.111.116 port 59118 ssh2
Jul 16 23:05:18 wpad sshd[29289]: CatchAll Triggered user=root passwd=jasmine rhost=218.87.111.116
Jul 16 23:05:18 wpad sshd[29289]: Failed password for root from 218.87.111.116 port 59118 ssh2
Jul 16 23:05:18 wpad sshd[29289]: Received disconnect from 218.87.111.116: 11:  [preauth]
Jul 16 23:05:20 wpad sshd[29291]: CatchAll Triggered user=root passwd=jasper rhost=218.87.111.116
Jul 16 23:05:20 wpad sshd[29291]: Failed password for root from 218.87.111.116 port 36523 ssh2
Jul 16 23:05:20 wpad sshd[29291]: CatchAll Triggered user=root passwd=javier rhost=218.87.111.116
Jul 16 23:05:20 wpad sshd[29291]: Failed password for root from 218.87.111.116 port 36523 ssh2
Jul 16 23:05:20 wpad sshd[29291]: CatchAll Triggered user=root passwd=jia123456 rhost=218.87.111.116
Jul 16 23:05:20 wpad sshd[29291]: Failed password for root from 218.87.111.116 port 36523 ssh2
Jul 16 23:05:21 wpad sshd[29291]: Received disconnect from 218.87.111.116: 11:  [preauth]
Jul 16 23:05:23 wpad sshd[29293]: CatchAll Triggered user=root passwd=jiamima rhost=218.87.111.116
Jul 16 23:05:23 wpad sshd[29293]: Failed password for root from 218.87.111.116 port 42189 ssh2
Jul 16 23:05:23 wpad sshd[29293]: CatchAll Triggered user=root passwd=jiangjie rhost=218.87.111.116
Jul 16 23:05:23 wpad sshd[29293]: Failed password for root from 218.87.111.116 port 42189 ssh2
Jul 16 23:05:23 wpad sshd[29293]: CatchAll Triggered user=root passwd=jj123456 rhost=218.87.111.116
Jul 16 23:05:23 wpad sshd[29293]: Failed password for root from 218.87.111.116 port 42189 ssh2
Jul 16 23:05:24 wpad sshd[29293]: Received disconnect from 218.87.111.116: 11:  [preauth]
Jul 16 23:05:26 wpad sshd[29295]: CatchAll Triggered user=root passwd=joe rhost=218.87.111.116
Jul 16 23:05:26 wpad sshd[29295]: Failed password for root from 218.87.111.116 port 48396 ssh2
Jul 16 23:05:26 wpad sshd[29295]: CatchAll Triggered user=root passwd=john rhost=218.87.111.116
Jul 16 23:05:26 wpad sshd[29295]: Failed password for root from 218.87.111.116 port 48396 ssh2
```

CatchAll - Logging Nonexistent Accounts

The future

- Integration into SSH-Rankings
- Splunk Module
- ???

CatchAll - Logging Nonexistent Accounts

Questions?