Privileged Account Management

Martin Cannard, Security Solutions Architect
Customer Use Cases - Introduction

- A US-based Natural Gas and Electric company serving multiple states

- Project Requirements
  - Only grant access to shared administrative accounts with pre-approval based on established policy
  - Need to provide ‘firecall’ functionality
  - Needed to delegate administrative access for Separation of Duty (SoD)
  - Required logging of Windows administrator activity
  - Needed to consolidate Unix identities into Active Directory to streamline provisioning, password management and privilege account management
Customer Use Cases - Introduction

• A global leader in payment processing

• Project Requirements
  – Needed to centralize accounts and get control over passwords and user lifecycles
  – Needed to replace NIS and provide centralized authentication
  – Needed to restrict and audit what users could do but at the same time provide for users to carry on with their day-to-day jobs
  – Needed to provide controls around shared administrative passwords
  – Needed to rotate administrative account passwords regularly
  – Needed to correlate and audit administrative activity with the actual end user
Allow the user to use their AD credentials to access Unix/Linux hosts

Shared Passwords

Automate the management of functional account passwords

Delegation

Once the user is logged on, manage what they can do

Managed list of resources the user is authorized to access. Audit of all session activity

Session Management
Allow the user to use their AD credentials to access Unix/Linux hosts.

Once the user is logged on, manage what they can do.

Managed list of resources the user is authorized to access. Audit of all session activity.

Automate the management of functional account passwords.
Use Case – Utility Company

- Needed to consolidate Unix identities into Active Directory to streamline provisioning, password management and privilege account management

Use Case - Payment Processing

- Needed to centralize accounts and get control over passwords and user lifecycles
- Needed to replace NIS and provide centralized authentication
Allow the user to use their AD credentials to access Unix/Linux hosts

Automate the management of functional account passwords

Once the user is logged on, manage what they can do

Managed list of resources the user is authorized to access. Audit of all session activity
Use Case – Utility Company

- Only grant access to shared administrative accounts with pre-approval based on established policy
- Need to provide ‘firecall’ functionality

Use Case - Payment Processing

- Needed to provide controls around shared administrative accounts
- Needed to rotate administrative account passwords regularly
Shared Password Management
Shared Password Management

- Password Request
- Password Release
- Password Reset
- Root
PAM Sub-Categories

PRIVILEGES

- AD Bridge
- Shared Passwords
- Privilege Sessions
- Delegation
Allow the user to use their AD credentials to access Unix/Linux hosts. Once the user is logged on, manage what they can do. Managed list of resources the user is authorized to access. Audit of all session activity.
PAM Sub-Categories

Use Case – Utility Company

• Required logging of Windows administrator activity
Privileged Session Management - Windows

Request session

Retrieve password

RDP
Privileged Session Management - Unix

Request Unix session

Retrieve password

shell
Privileged Session Management – Network Enclaves

Request RDP session to “X1”

Retrieve password

Session goes straight back to the user

PCI Network

SSH Enclave

Privileged Session Manager

Session details passed to Proxy

RDP

X1

UNIX

UNIX

SSH
Allow the user to use their AD credentials to access Unix/Linux hosts

Once the user is logged on, manage what they can do

Managed list of resources the user is authorized to access. Audit of all session activity

Automate the management of functional account passwords
PAM Sub-Categories

Use Case – Utility Company

- Needed to provide fine-grained delegation of administrative (root) access for Separation of Duty (Sod)

Use Case - Payment Processing

- Needed to restrict and audit what users could but at the same time provide for users to carry on with their day-to-day jobs
- Needed to correlate and audit administrative activity with the actual end-user
How do I allow users to perform elevated tasks on Unix without losing control of the root password?

- Pair a password vault with a delegation solution

**Common delegation solutions**

- Native OS solutions (RBAC implementations)
- The open source Sudo project
- The Commercial Unix Security space
What did we discover?

Privilege

AD Bridge

Shared Passwords

Delegation

Native OS options

Commercial 3rd party solutions

~3,000 customers

Result? Companies would:
- Purchase a PAM sol’n only for their highest risk machines
- Hate having to re-train admins & help desk staff on a new syntax
- “Bend” sudo in crazy ways

Native OS options

**sudo**

- Linux: 7.5M servers
- Unix: 2.8M servers
- Mac: 2.0M servers

No focus on this segment!
Sudo v1.7 and earlier

$I need to run a command as the 'oracle' user$

```
$ sudo -u oracle ./backup /var/oracle
```

Execute the command

Sudoers

Privileges
- AD Bridge
- Shared Passwords
- Privilege Sessions
- Delegation
Field Feedback: Common Pain and Trends

- How do I easily provide access control reports?
- How do I deal with sudoers?
  - How to manage it, distribute it, etc
- How do I enable central keystroke logging?
- How do I know what is going on across lots of systems?
- How do I provide more fine-grain control in the policy?
Sudo v1.8 and the new plug-in API
Example architecture using plug-in API

PRIVILEGES

- AD Bridge
- Shared Passwords
- Privilege Sessions
- Delegation
Example pain points that the plug-in API can assist with

- Sudo Reporting
  - Access Control Report
  - Event Activity
  - Commands run
  - Policy changes
- Deployment
  - Preflight and sudo plug-in installation
- Policy Management
  - Editor, Versioning, Rollback
- Keystroke Logging
  - Search, Playback
- Separation of Duty
SUDO v2.0: Design Phase

- New security policy format
  - Designed for the needs of the enterprise
- Include an API to support analysis and reporting tools
- Support grouping of commands and options in logical units
- Facility management of sudoers by multiple stake-holders
- Time based policy rules
- Data source plug-ins
SUDO v2.0: Design Phase

 PRIVILEGES

- AD Bridge
- Shared Passwords
- Privilege Sessions
- Delegation
Complete Identity & Access Management

Manage Access to Business Critical Information

Simplify Account Management

Access Governance

Privileged Account Management

Quest® One
Identity Solutions

Identity Administration

User Activity Monitoring

Understand & Control Administrator Activity

Audit User Activity
Thank You
Centrally view sudo event activity

<table>
<thead>
<tr>
<th>User</th>
<th>Host</th>
<th>Command</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>lihwa</td>
<td>neo.example.com</td>
<td>reboot</td>
<td>7/23/11 3:08 PM</td>
</tr>
<tr>
<td>ssset</td>
<td>neo.example.com</td>
<td>reboot</td>
<td>7/23/11 3:09 PM</td>
</tr>
<tr>
<td>ssset</td>
<td>neo.example.com</td>
<td>/usr/local/init</td>
<td>7/23/11 3:09 PM</td>
</tr>
<tr>
<td>doallimore</td>
<td>neo.example.com</td>
<td>/usr/local/init</td>
<td>7/23/11 3:10 PM</td>
</tr>
<tr>
<td>doallimore</td>
<td>neo.example.com</td>
<td>/usr/local/init</td>
<td>7/23/11 3:10 PM</td>
</tr>
<tr>
<td>doallimore</td>
<td>neo.example.com</td>
<td>/usr/local/init</td>
<td>7/23/11 3:10 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>bash</td>
<td>7/23/11 9:10 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>vi/vi/cd</td>
<td>7/23/11 9:12 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>cd/home/edmzpar</td>
<td>7/23/11 9:13 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>show -R edmzpar edmzpar</td>
<td>7/23/11 9:13 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>ls -lJ</td>
<td>7/23/11 9:14 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>show -R jhbash jhbash</td>
<td>7/23/11 9:14 PM</td>
</tr>
<tr>
<td>root</td>
<td>neo.example.com</td>
<td>passwd root</td>
<td>7/24/11 12:39 AM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>cd/home/shadow</td>
<td>7/26/11 11:20 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>/opt/qube/bin/astool -u host/search (objectid=amgroup)</td>
<td>7/26/11 1:00 PM</td>
</tr>
<tr>
<td>jhbash</td>
<td>neo.example.com</td>
<td>/opt/qube/bin/astool -u host/search (objectid=amgroup)</td>
<td>7/26/11 1:00 PM</td>
</tr>
</tbody>
</table>
Search and filter sudo event logs
Manage local accounts

### Quest® One Management Console for Unix

**Getting Started** | **Hosts** | **Sudo** | **All Local Users** | **Active Directory** | **Reporting** | **QAS Agent Status**
---|---|---|---|---|---|---

<table>
<thead>
<tr>
<th>Name</th>
<th>UID</th>
<th>Comment (GECOS)</th>
<th>Host</th>
<th>Joined Domain</th>
<th>AD User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brittan</td>
<td>1039</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>abri</td>
<td>485</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>scorbett</td>
<td>1013</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>adm</td>
<td>3</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>alexi</td>
<td>1025</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>apache</td>
<td>48</td>
<td>Apache</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>avahi</td>
<td>70</td>
<td>Avahi mDNS/DNS-S...</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>avahi-autopd</td>
<td>170</td>
<td>Avahi IPv4LL Slack</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>ben</td>
<td>1024</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>bin</td>
<td>1</td>
<td>bin</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>daemon</td>
<td>2</td>
<td>daemon</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>dbus</td>
<td>81</td>
<td>System message bus</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>dcmimcore</td>
<td>1015</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>dcmzar</td>
<td>0</td>
<td></td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
<tr>
<td>ftp</td>
<td>14</td>
<td>FTP User</td>
<td>rhe6.example.com</td>
<td>EXAMPLE.COM</td>
<td></td>
</tr>
</tbody>
</table>

**Local User Statistics**

- Users requiring AD logon: 0
- Users not requiring AD logon: 29
- System Users: 32
- Total Local Users: 61

**Require AD Logon**

To require a local user to logon to a host that is joined to an AD domain using their AD password, select a local user on the left and the AD account to use for access below, then click 'Require AD Logon to Host'.

Search by name:

Search in: Entire directory

<table>
<thead>
<tr>
<th>Account</th>
<th>Name</th>
</tr>
</thead>
</table>
Replay sudo sessions

ch/root@rhe16 /home
[root@rhe16 home]# cd questusr
[root@rhe16 questusr]# ls -lart
total 68
-rw-r--r-- 1 questusr questgrp 124 Jun 22 2010 .bashrc
-rw-r--r-- 1 questusr questgrp 176 Jun 22 2010 .bash_profile
-rw-r--r-- 1 questusr questgrp 16 Jun 22 2010 .bash_logout
drwxr-xr-x 2 questusr questgrp 4056 Jul 14 2010 .gnome2
drwxr-xr-x 4 questusr questgrp 4056 Jul 15 06:10 .mozilla
drwxr-xr-x 2 questusr questgrp 4056 Jul 16 17:55 .ssh
drwxr-xr-x 3 questusr questgrp 1001 4056 Jul 16 18:45 .subversion
drwxr-xr-x 6 root questgrp 4056 Jul 19 10:32 ...
-rw-r--r-- 1 questusr questgrp 736 Jul 19 13:10 .viminfo
drwxr-xr-x 7 questusr questgrp 4056 Jul 10 13:44 ...
drwxr-xr-x 2 questusr questgrp 4056 Jul 21 15:13 .scratch
-rw-r--r-- 1 questusr questgrp 24331 Jul 21 22:45 .bash history
[root@rhe16 questusr]# cd .ssh
[root@rhe16 .ssh]# ls -lart
total 24
-rw-r--r-- 1 questusr questgrp 406 Jul 16 17:52 id_rsa.pub
-rw-r--r-- 1 questusr questgrp 1671 Jul 16 17:52 id_rsa
-rw-r--r-- 1 questusr questgrp 414 Jul 16 17:53 known_hosts
drwxr-xr-x 2 questusr questgrp 4096 Jul 16 17:53 ...
-rw-r--r-- 1 questusr questgrp 605 Jul 19 07:50 authorized_keys
drwxr-xr-x 7 questusr questgrp 4096 Jul 19 13:44 ...
[root@rhe16 .ssh]# touch /home/questusr/.ssh/authorized_keys
Detailed Sudo access control reporting

### Access and Privileges by Host

<table>
<thead>
<tr>
<th>Host</th>
<th>User ID</th>
<th>Run User</th>
<th>Run Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>jones2-ubuntu</td>
<td>Hijones2</td>
<td>root</td>
<td>PROCESSES</td>
</tr>
<tr>
<td>jones2-ubuntu</td>
<td>Hijones2</td>
<td>root</td>
<td>/usr/bin/zypper</td>
</tr>
<tr>
<td>jones2-ubuntu</td>
<td>Hijones2</td>
<td>root</td>
<td>SUGI_COMMANDS</td>
</tr>
<tr>
<td>jones2-ubuntu</td>
<td>Hijones2</td>
<td>secure</td>
<td>/usr/bin/zypper</td>
</tr>
<tr>
<td>jones2-ubuntu</td>
<td>Hijones2</td>
<td>trusted</td>
<td>/usr/bin/zypper</td>
</tr>
</tbody>
</table>

### Privileges from User Alias: AD_USERS

### Privileges from User Alias: SECURE_USERS

### Privileges from User Alias: SUGI_USERS

### Privileges from User Alias: WEB_USERS
## Console Access and Permissions

This report lists the users who have access to the console based on membership in a role and the permissions assigned to the role.

### Audit Sudo Policy

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console User</td>
<td>This Permission allows console login access</td>
</tr>
<tr>
<td>View Hosts</td>
<td>This Permission allows viewing of hosts</td>
</tr>
<tr>
<td>View IO Logs</td>
<td>This Permission allows viewing of sudo keystroke logs</td>
</tr>
<tr>
<td>View Sudo Change Reports</td>
<td>This Permission allows viewing of sudoers change reports</td>
</tr>
<tr>
<td>View Sudo Privileges Reports</td>
<td>This Permission allows viewing of sudo access reports</td>
</tr>
</tbody>
</table>

### Members

- HAdministrator

### Console Administration

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console User</td>
<td>This Permission allows console login access</td>
</tr>
<tr>
<td>Manage Active Directory</td>
<td>This Permission allows management of Active Directory</td>
</tr>
<tr>
<td>Manage Console Settings</td>
<td>This Permission allows management of console settings</td>
</tr>
<tr>
<td>Manage Supervisor Password</td>
<td>This Permission allows resetting of the supervisor password</td>
</tr>
<tr>
<td>View Active Directory</td>
<td>This Permission allows viewing of Active Directory</td>
</tr>
<tr>
<td>View Console Settings</td>
<td>This Permission allows viewing of console settings</td>
</tr>
<tr>
<td>View Hosts</td>
<td>This Permission allows viewing of hosts</td>
</tr>
</tbody>
</table>