jamf | NATION

User Conference

OCTOBER 24-26

2017

Josh Roskos

Professional Services Engineer

github.com/kc9wwh

Bryson Tyrrell

Systems Administrator

github.com/brysontryrrell

© JAMF Software, LLC

API Scripting Lab

Tips & Tricks

Loopity Loops & Arrays

Modifying Scope

Let’s Script Something

© JAMF Software, LLC

House Keeping Items…

Computers

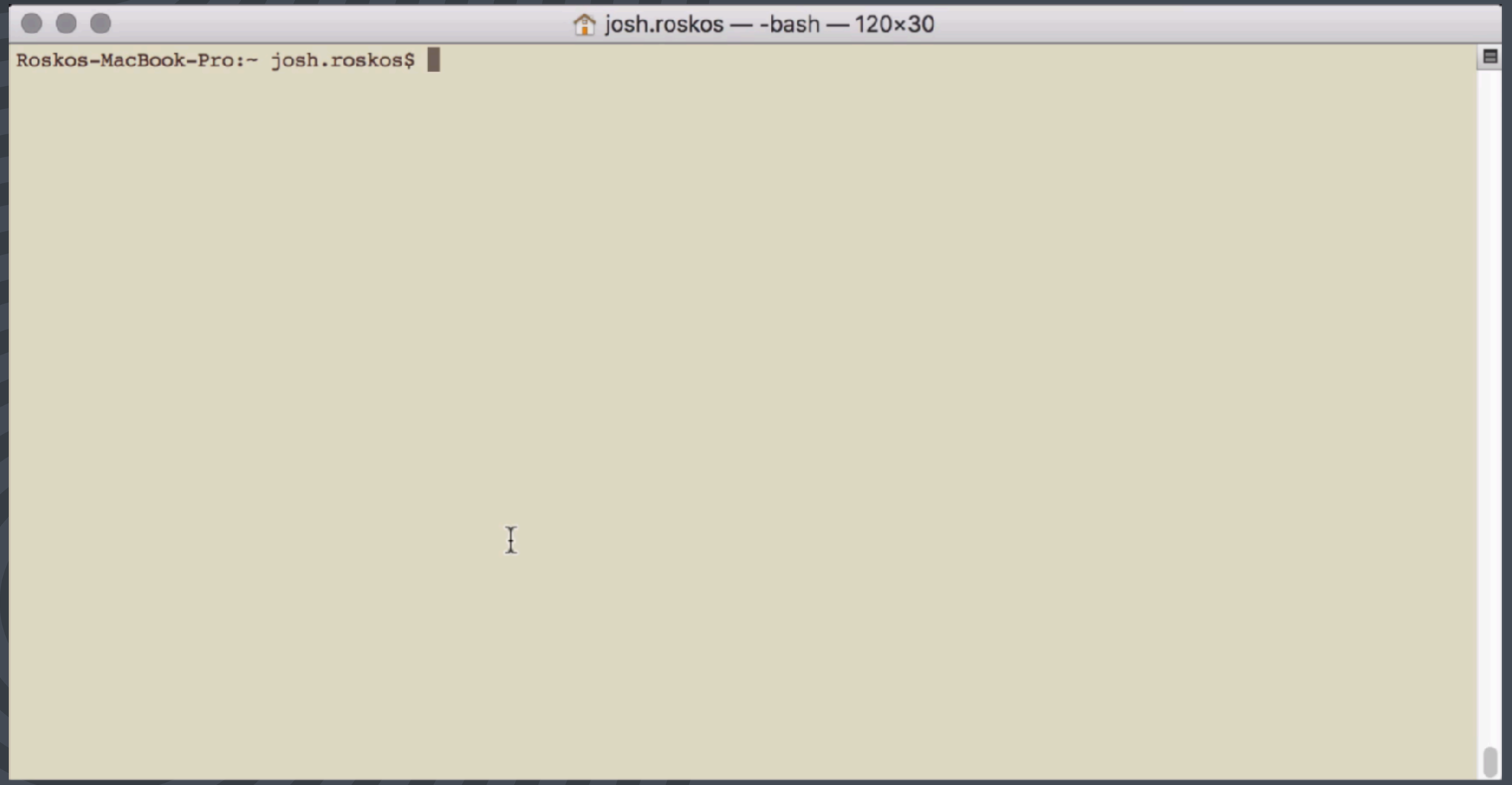
- Logon: localadmin / jamf1234
- /Users/Shared/Resources/…
  - Code Editors
  - Sample API Script

Jamf Pro server

- https://pro.jamf.training:8443/jnuc##
- Logon: jamfadmin / jamf1234

© JAMF Software, LLC

Tips & Tricks

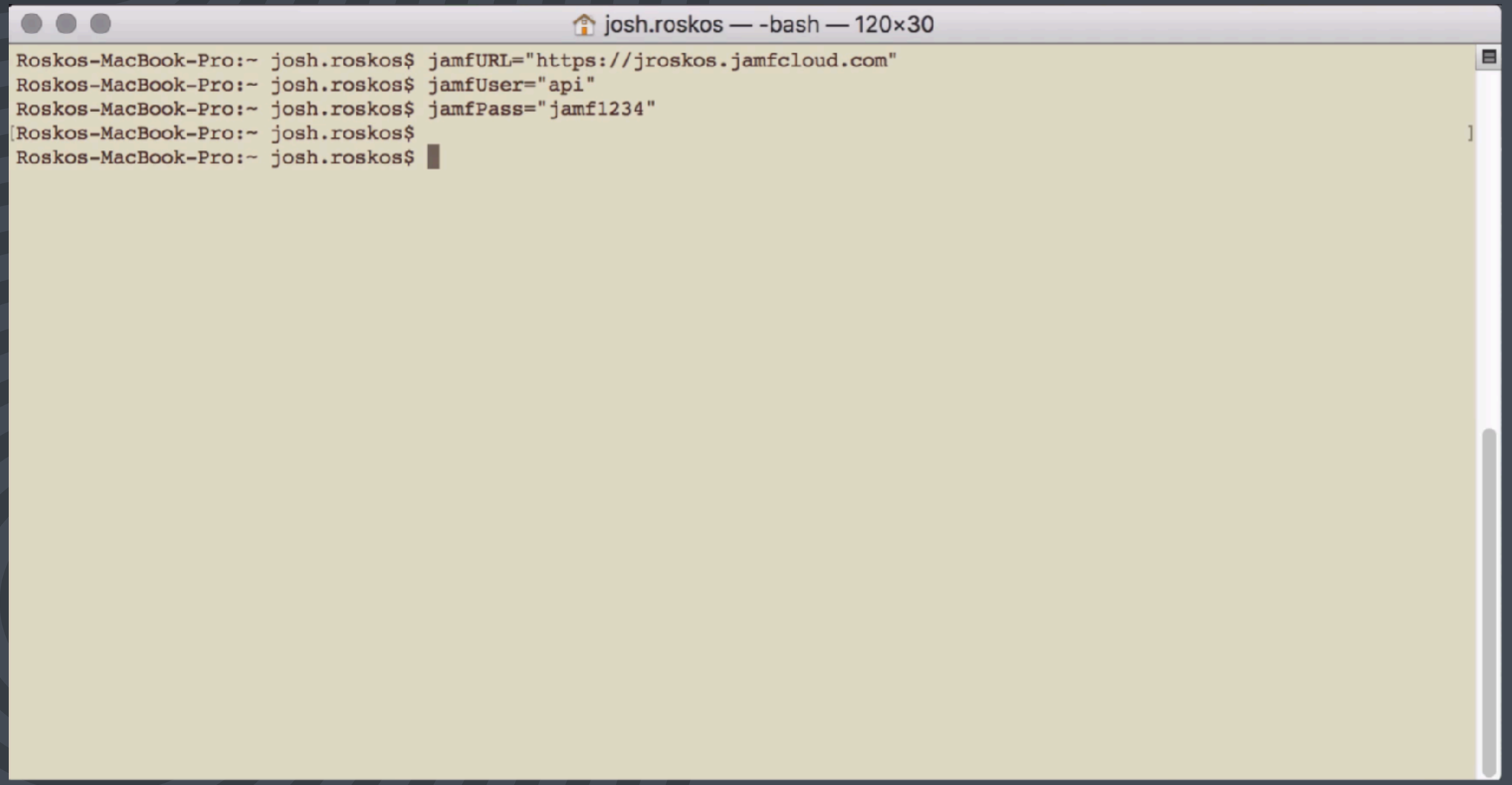


© JAMF Software, LLC

Tips & Tricks - xpath

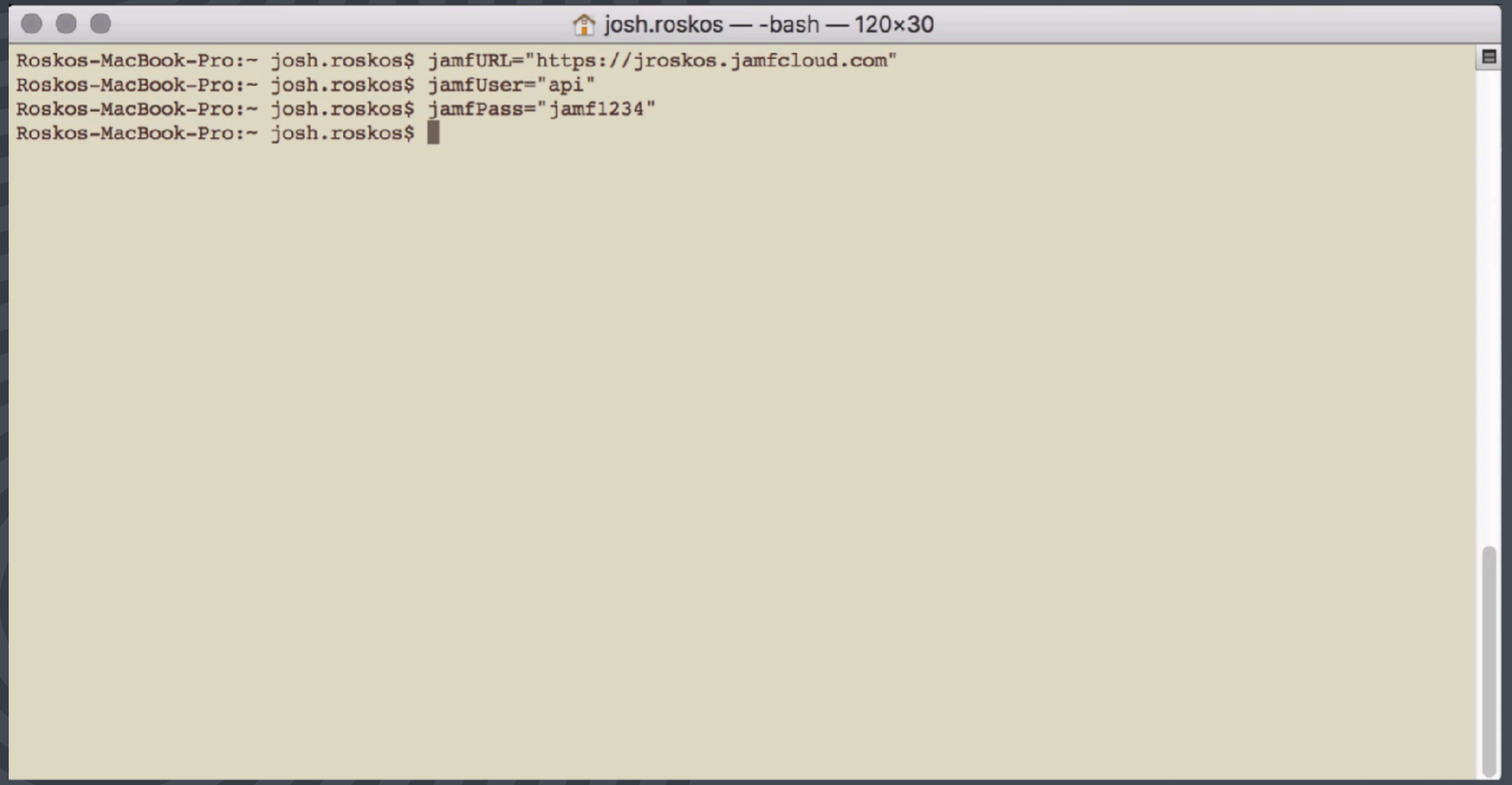
© JAMF Software, LLC

Tips & Tricks - xpath



© JAMF Software, LLC

Loopity Loops & Arrays



© JAMF Software, LLC

Modifying Scope

Ways to modify Scope:

- Directly add/remove devices
- Add/remove Smart/Static Groups
- Modify Static Groups
- Modify Extension Attributes for Smart Groups

© JAMF Software, LLC

Modifying Scope

Gotchas! Don’t do it this way:

PUT /JSSResource/policies/id/{x}

```
<policy>
    <scope>
        <computers>
            <computer>
                <id>1</id>
            </computer>
        </computers>
    </scope>
</policy>
```

© JAMF Software, LLC

Modifying Scope

Where supported, do this:

PUT /JSSResource/policies/id/{x}

```
<policy>
    <scope>
        <computer_additions>
            <computer>
                <id>1</id>
            </computer>
        </computer_additions>
    </scope>
</policy>
```

© JAMF Software, LLC

Modifying Scope

Where supported, do this:

PUT /JSSResource/policies/id/{x}

```
<policy>
    <scope>
        <computer_deletions>
            <computer>
                <id>1</id>
            </computer>
        </computer_deletions>
    </scope>
</policy>
```

© JAMF Software, LLC

Modifying Scope

Similar gotcha with Groups:

PUT /JSSResource/policies/id/{x}

```
<policy>
    <scope>
        <computer_groups>
            <computer_group>
                <id>1</id>
            </computer_group>
        </computer_groups>
    </scope>
</policy>
```

© JAMF Software, LLC

Modifying Scope

Modify the Group, not the object:

PUT /JSSResource/computergroups/id/{**x**}

PUT /JSSResource/mobiledevicegroups/id/{**x**}

```
<mobile_device_group>
    <mobile_device_additions>
        <mobile_device>
            <id>1</id>
        </mobile_device>
    </mobile_device_additions>
</mobile_device_group>
```

```
<computer_group>
    <computer_additions>
        <computer>
            <id>1</id>
        </computer>
    </computer_additions>
</computer_group>
```

© JAMF Software, LLC

Modifying Scope

You don’t modify Smart Groups:

PUT /JSSResource/computers/id/{**x**}

```
<computer>
  <extension_attributes>
    <extension_attribute>
      <id>1</id>
      <value>MyNewValue</value>
    </extension_attribute>
  </extension_attributes>
</computer>
```

© JAMF Software, LLC

Let’s Script Something Already!

How can I see what is scoped to my computer groups?

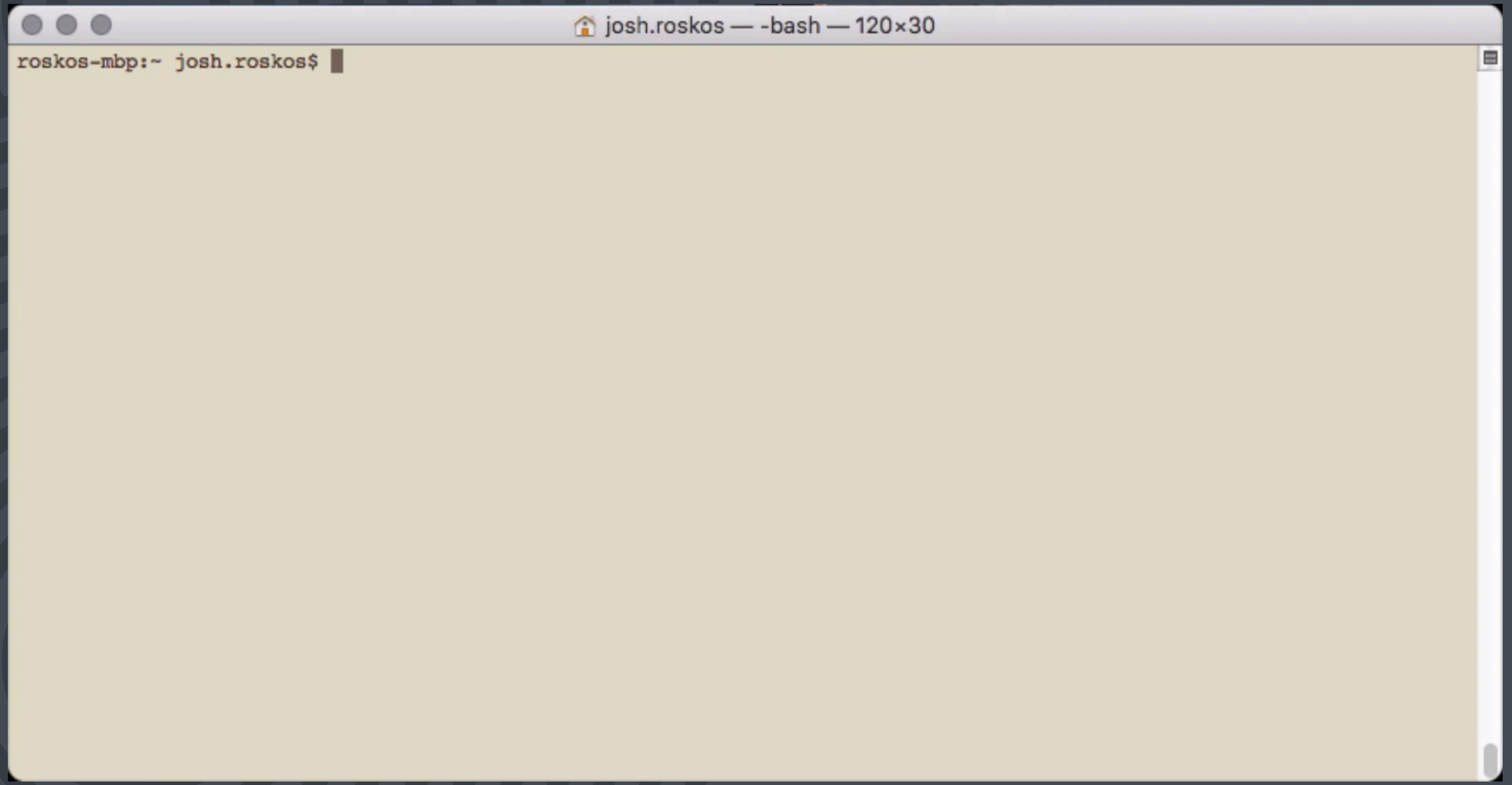
© JAMF Software, LLC

```
#!/bin/bash

#########################################################################
#
# What is scoped to my Computer Groups?
#
# In this script we will utilize the Jamf Pro API to determine what Policies are assigned
# to your Computer Groups.
#
#
# OBJECTIVES
#       - Create a list of all Smart Groups
#       - Provide a list of what is scoped to each Smart Group
#
#########################################################################

#########################################################################
# USER VARIABLES
```

© JAMF Software, LLC

```
################################################
# APPLICATION
################################################

## Retrieve & Extract Computer Group IDs/Names & Build Array
compGroupData=$( curl -k -s -u "$jamfUser":"$jamfPass" $jamfURL/JSSResource/
computergroups -H "Accept: application/xml" -X GET )
compGrpSize=$(echo $compGroupData | xpath "//computer_groups/size/text()")
index=0
declare -a compGrpNames
declare -a compGrpIDs
while [ $index -lt ${compGrpSize} ]; do
    element=$(($index+1))
    compGrpName=$( echo $compGroupData | xpath "//computer_groups/
computer_group[${element}]/name/text()" )
    compGrpID=$( echo $compGroupData | xpath "//computer_groups/computer_group[$
{element}]/id/text()" )
    echo "Computer Group ID: $compGrpID"
```

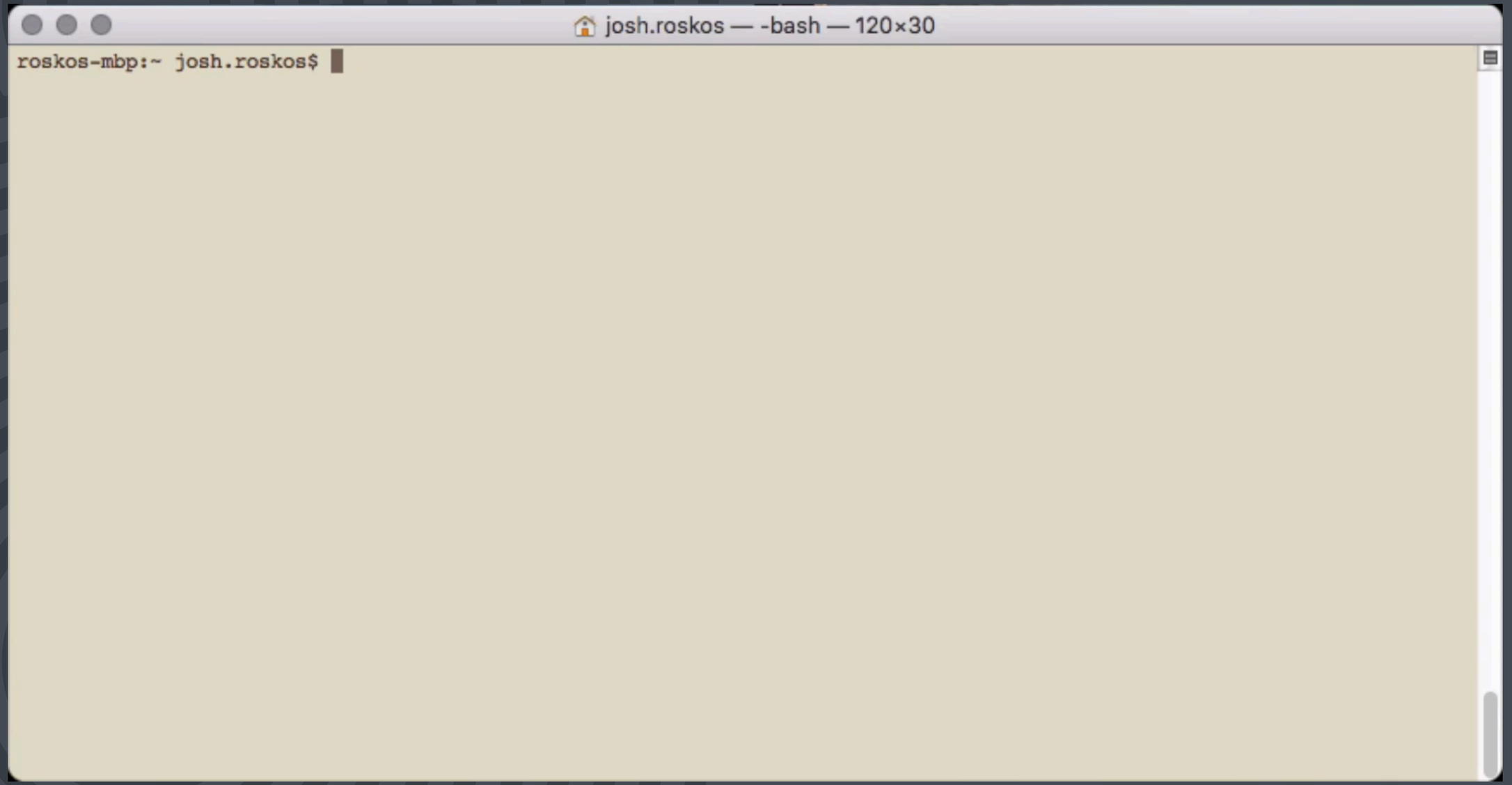
© JAMF Software, LLC



© JAMF Software, LLC

```
## Retrieve & Filter Policy Data
echo "Retrieving List of All Policy IDs..."
unset policyIDs
policyIDs=( $( curl -k -s -u "$jamfUser":"$jamfPass" $jamfURL/JSSResource/policies -H
"Accept: application/xml" -X GET | /usr/bin/perl -lne 'BEGIN{undef $/} while (/<id>(.*?)<V
id>/sg){print $1}' ) )
for i in "${policyIDs[@]}"; do
    echo "Retrieving Policy ID ${i}'s' Data..."
    policyData=$( curl -k -s -u "$jamfUser":"$jamfPass" $jamfURL/JSSResource/policies/id/$
{i} -H "Accept: application/xml" -X GET )
    policyName=$( echo $policyData | xpath "//policy/general/name/text()" )
    ## Check if is a Jamf Remote Policy
    echo "Checking if this is a Jamf Remote Policy..."
    if [[ $policyName == $( echo $policyName | egrep -B1 '[0-9]+-[0-9]{2}-[0-9]{2} at [0-9]
{1,2}:[0-9]{2,2} [AP]M \| .* \| .*' ) ]]; then
        ## This is a Jamf Remote Policy
        ## Setting policy name in array to "JamfRemotePolicy-Ignore"
        echo "    This is a Jamf Remote policy"
        continue
    else
```

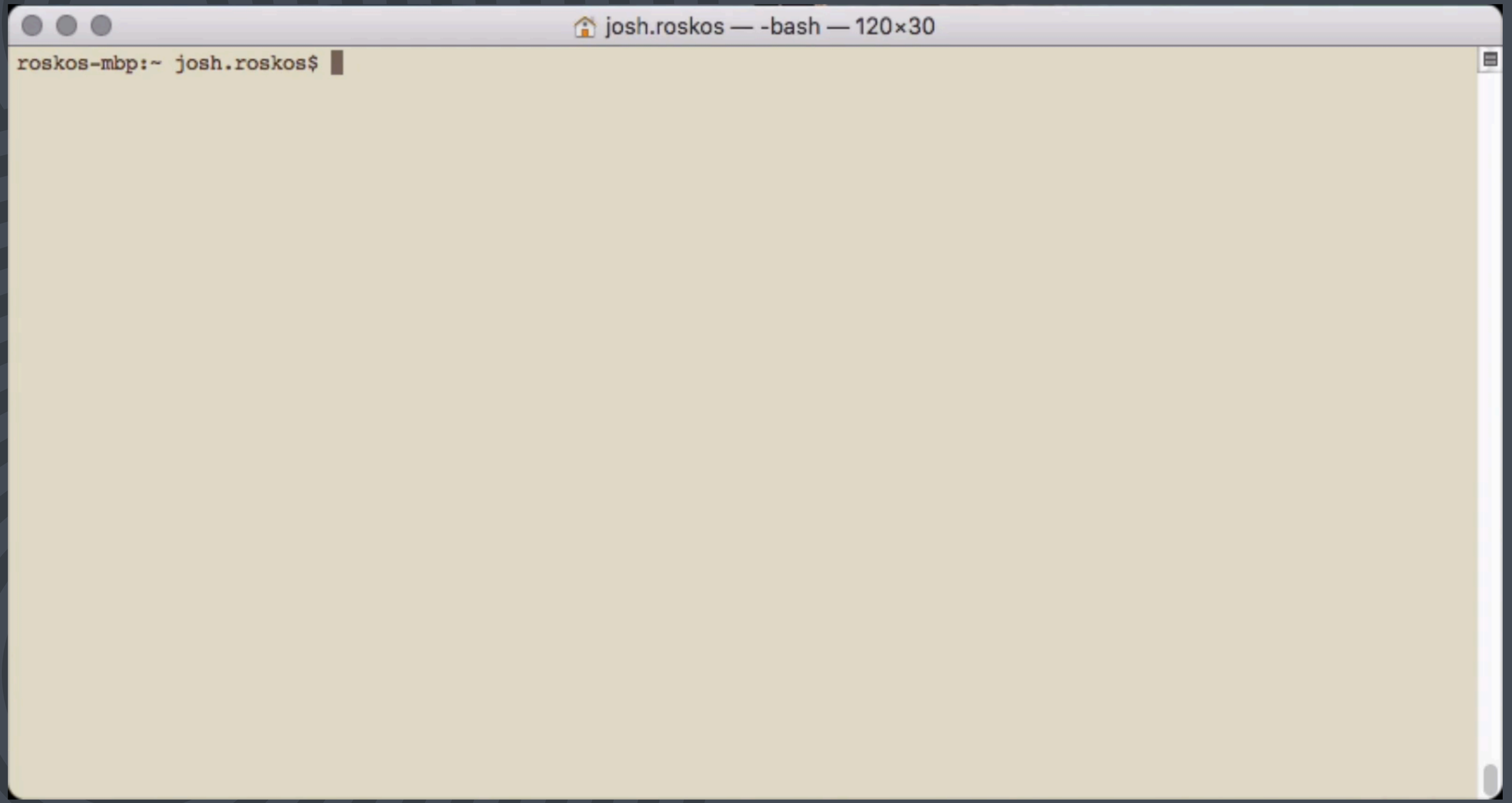
© JAMF Software, LLC



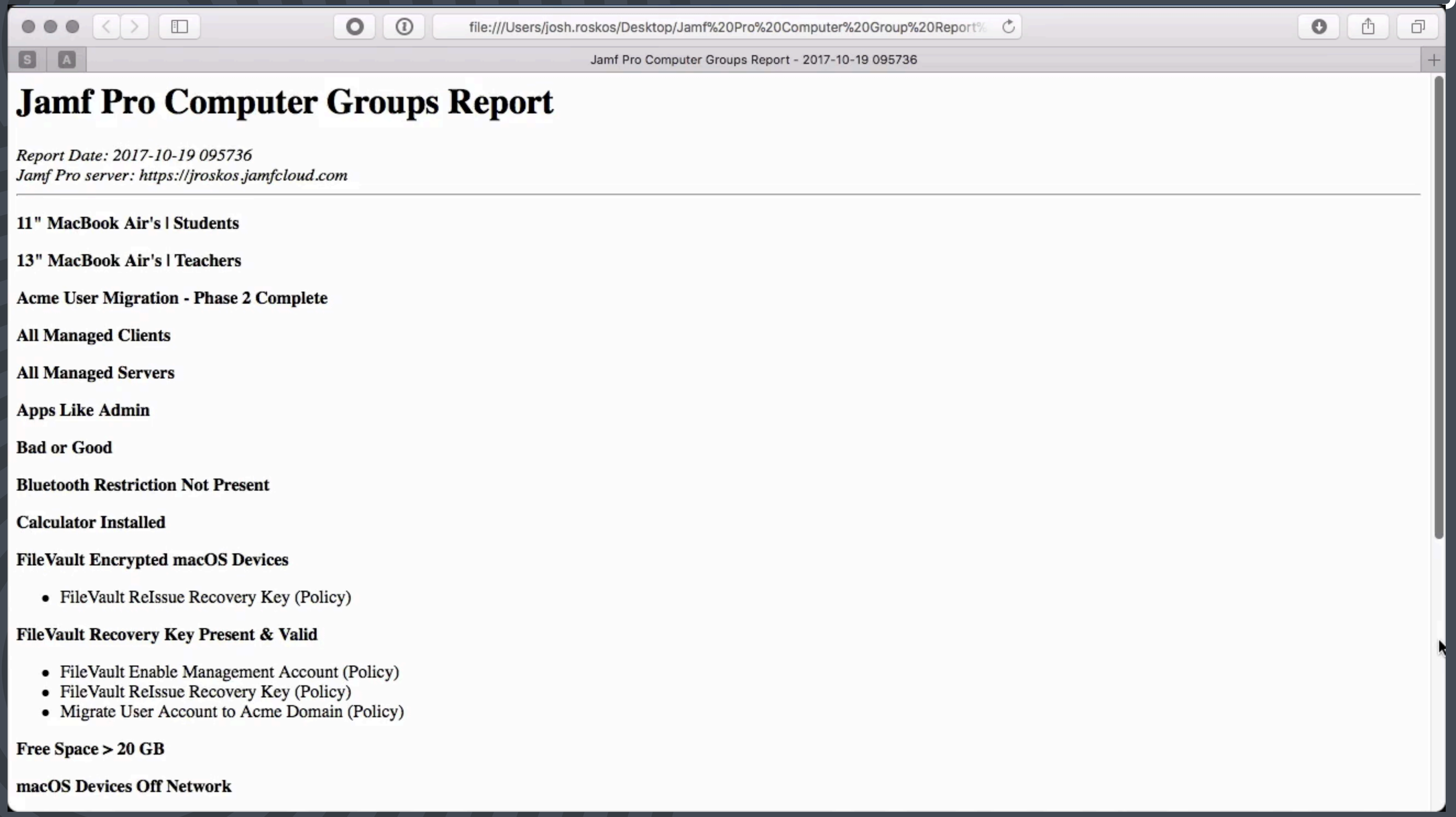
© JAMF Software, LLC

```
################################################################
# BUILD HTML REPORT
################################################################

reportDate=$(/bin/date "+%Y-%m-%d %H%M%S")
reportName="/Users/${currentUser}/Desktop/Jamf Pro Computer Group Report -
$reportDate.html"
echo "<html>
<head>
<title>Jamf Pro Computer Groups Report - $reportDate</title>
</head>
<body>
<h1>Jamf Pro Computer Groups Report</h1>
<i>Report Date: $reportDate<br/>Jamf Pro server: $jamfURL</i>
<hr/>
<p/>" > "$reportName"
for (( x = 0 ; x < ${#compGrpNames[@]} ; x++ )); do
    echo "<b>${compGrpNames[$x]}</b><br/><ul>" >> "$reportName"
    groupID="${compGrpIDs[$x]}"
```

© JAMF Software, LLC



© JAMF Software, LLC



Jamf Pro Computer Groups Report

Report Date: 2017-10-19 095736
Jamf Pro server: https://jroskos.jamfcloud.com

11" MacBook Air's | Students

13" MacBook Air's | Teachers

Acme User Migration - Phase 2 Complete

All Managed Clients

All Managed Servers

Apps Like Admin

Bad or Good

Bluetooth Restriction Not Present

Calculator Installed

FileVault Encrypted macOS Devices

- FileVault ReIssue Recovery Key (Policy)

FileVault Recovery Key Present & Valid

- FileVault Enable Management Account (Policy)
- FileVault ReIssue Recovery Key (Policy)
- Migrate User Account to Acme Domain (Policy)

Free Space > 20 GB

macOS Devices Off Network

© JAMF Software, LLC

Q&A

© JAMF Software, LLC

Thank you!

© JAMF Software, LLC