Secure scripting (SES)

**Module Description**: This module focuses on basic scripting using the Bourne-Again SHell (BASH). It focuses on the basic constructs and useful commands such as conditional, looping, and logical constructs as well as variables and user and system interfaces. Students create basic scripts and have a foundation upon which to learn more advanced bash scripting, or other scripting languages. One of the exercises is to implement a simple file system scanner, so the students will practice scripting security applications and administrative scripts.

**Prerequisite Knowledge:** Basic knowledge of programming (variables, loops, conditionals); knowledge of a Linux or other UNIX-like system including pipes and input and output redirection. In particular, students should be able to read a manual page with guidance. Instructors should conduct a pre-assessment to verify that students have the above background knowledge. If students do not know these concepts and have no experience with them, much of the module will be more difficult than intended.

**Length of Completion:** 8-9 contact hours.

**Level of Instruction:** Freshman level. This module is well-suited to inclusion in a CS1 class.

**Learning Setting:** This module is intended for in-class (face-to-face) instruction.

# learning outcomes

MODULE LEARNING OUTCOMES

• Understand how to approach a problem to solve it using scripting

• Understand bottom-up programming techniques suitable for use in scripting

• Demonstrate proficiency in scripting by using elements of the Bourne shell scripting language, and of scripting languages for various Linux tools

# Module Details

**Interconnection:** This module includes the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Lessons | Presentations | Lab Title and Number of Exercises | Advanced Challenges |
| The Basics | X | “Looking for Strings”  4 lab exercises | 2 |
| Advanced Control | X | “System Scanner”  5 exercises | 2 |
| Advanced Scripting | X | “Analyzing a Spreadsheet”  5 exercises | 2 |

**Instructional Files and Online Resources that are Needed:**

Lesson 1: The Basics

02. SeS\_Lesson1\_TheBasics\_Presentation.pptx

03. SeS\_Lesson1\_TheBasics\_Lab.docx

04. SeS\_Lesson1\_TheBasics\_DataFiles

05. SeS\_Lesson1\_TheBasics\_LabSolutions.docx

06. SeS\_Lesson1\_TheBasics\_LabSolutionFiles

or

07. SeS\_Lesson1\_TheBasics\_LabSoltuionPDFs

Lesson 2: Advanced Control

08. SeS\_Lesson2\_AdvancedControl\_Presentation.pptx

09. SeS\_Lesson2\_AdvancedControl\_Lab.docx

10. SeS\_Lesson2\_AdvancedControl\_DataFiles

11. SeS\_Lesson2\_AdvancedControl\_LabSolutions.docx

12. SeS\_Lesson2\_AdvancedControl\_LabSolutionsFiles

or

13. SeS\_Lesson2\_AdvancedControl\_LabSolutionsPDFs

Lesson 3: Advanced Scripting

14. SeS\_Lesson3\_AdvancedScripting\_Presentation.pptx

15. SeS\_Lesson3\_AdvancedScripting\_Lab.docx

16. SeS\_Lesson3\_AdvancedScripting\_DataFiles

17. SeS\_Lesson3\_AdvancedScripting\_LabSolutions.docx

18. Ses\_Lesson3\_AdvancedScripting\_LabSolutionFiles

or

19. SeS\_Lesson3\_AdvancedScripting\_LabSolutionsPDFs

**Assessment:** The assessment guide contains 33 multiple choice and fill in the blank test questions mapped to the lesson learning outcomes as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 1.1 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 |  | X | X | X |  |  |  | X |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.3 |  | X |  |  | X | X | X | X |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.4 |  |  |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.5 |  | X |  |  |  |  |  | X |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| 2.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |  |
| 2.6 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |
| 3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |  |  |
| 3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |  |  |
| 3.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |  |  |
| 3.4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |
| 3.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |

See the assessment guide files:

20.SeS\_AssessmentGuide. Files containing scripts that are the correct answers to assessment questions are provided in the directory 21.SeS\_AssessmentAnswerFiles. Each script is available in either plain text (“.sh”) format or PDF format, so instructors can use whichever file format they find easier to manage.

# lessons

**Overview of Lessons:**

Lesson 1: The Basics

Lesson 2: Advanced Control

Lesson 3: Advanced Scripting

### **Lesson 1. The Basics**

**Learning Outcomes:**

Upon completion of Lesson 1:

1.1 Students will be able to analyze a problem and develop a script to solve it.

1.2 Students will be able to create and execute a Bourne shell script.

1.3 Students will be able to use command-line arguments in that script.

1.4 Students will be able to use conditional (*if* … *elif* ... *fi*) statements to test for various conditions and act accordingly.

1.5 Students will be able to perform basic error checking in the script.

**Warm Up:**

**Lesson:** 02.SeS\_Lesson1\_TheBasics\_Presentation.pptx

To demonstrate the examples included in the presentation, you will need the following data files from the directory 04.SeS\_Lesson1\_TheBasics\_DataFiles:

* abcscript — Use for the example shown on Slides 6–7.
* dict.txt — This file contains a list of words, one per line. It’s first used starting on Slide 13.
* add — This script is introduced on Slide 25.
* mycat.sh — This script is discussed on Slides 28–31. It is provided so students can experiment with the different quotation marks.
* x, y, x y — These three files are used with Slides 28-31 to demonstrate the proper use of quotation marks.

If you want the students to try the exercises themselves, copy the files to the students’ systems or make available for them to download.

**Active Learning Activity/Lab:** 03.SeS\_Lesson1\_TheBasics\_Lab.docx

Students will need the following data files from the directory 04.SeS\_Lesson1\_TheBasics\_DataFiles. These files can be copied to the students’ systems or made available for them to download.

* dict.txt
* mycat.sh
* x
* y
* x y

Solutions to lab questions are located in 05.SeS\_Lesson1\_TheBasics\_LabSolutions.

### **Lesson 2. Advanced Control**

**Learning Outcomes**

Upon completion of Lesson 2:

2.1 Students will be able to merge two or more scripts into one that performs the same functions.

2.2 Students will be able to use *for* loops in the script.

2.3 Students will be able to use variables in the script.

2.4 Students will be able to use Boolean operators to test conditions in the script.

2.5 Students will be able to process options given to the script.

2.6 Students will be able to perform basic error checking in the script.

**Warm Up:**

**Lesson:** 08.SeS\_Lesson2\_AdvancedControl\_Presentation.pptx

Files in the directory 10.SeS\_Lesson2\_AdvancedControl\_DataFiles are used in the examples.

* for1.sh — This script is an example of a basic *for* loop (see Slide 15).
* myls.sh — This script lists each named file, giving attribute information as well as the filename(s). See Slide 10.
* Subdirectory “sample” — These files are used in the demonstration of *for* loops on Slides 16–18. See notes below under “Lab.”
* for2.sh — This script is an example of a *continue* statement (see Slide 20).
* var1.sh, var2.sh, var3.sh — These show some things about variables. They are referenced on Slides 23–25.
* bool1.sh, bool2.sh — These demonstrate various Boolean operators. See Slides 30–31.
* x1, x2 — These two files containing very similar content are used to demonstrate the output from *diff*. They are referenced on Slide 34.

**Active Learning Activity/Lab:** 09.SeS\_Lesson2\_AdvancedControl\_Lab.docx

Students will need a copy of the directory “sample”. It can be copied to the students’ systems or put out for them to download. Please do not change anything in that directory – one of the tests the students run depends on those files being unchanged.

* abc xyz — This file is empty. Note the blank space; that is part of the filename.
* abcde — This file is also empty.
* demofor.sh, demofor2.sh, demofor3.sh — These scripts demonstrate various things about *for* loops.

Solutions to lab questions are located in 11.SeS\_Lesson2\_AdvancedControl\_LabSolutions

**Suggestion**

If you want to split this Lesson into two parts, we recommend using Slides 1 through 21 or 22 as the first part; use Slide 3 then Slides 22 through 26 as the second part. There is a natural breaking point after Slide 21, because the class has finished the loops and done an exercise. Slide 22 is a high-level overview of the next few slides and might be useful as a “teaser”; if your students respond well to that sort of slide or discussion ender, go ahead and include Slide 22 at the end of the first part. You should begin the second part with Slide 22, though, to remind students of where the class left off. Also, if you split this Lesson, we strongly recommend beginning the second part with Slide 3 to remind students of the problem being solved.

### **Lesson 3. Advanced Scripting**

**Learning Outcomes**

Upon completion of Lesson 3:

3.1 Students will be able to analyze data in a file using a script.

3.2 Students will be able to use while loops in the script.

3.3 Students will be able to do simple arithmetic in the script.

3.4 Students will be able to edit values of variables and data.

3.5 Students will be able to perform pattern matching.

**Warm Up:**

**Lesson:** 14.Ses\_Lesson3\_AdvancedScripting\_Presentation.pptx

Files in the directory 16.SeS\_Lesson3\_AdvancedScripting\_DataFiles are used in examples during the presentation.

• connect.csv — This is a comma-separated value (CSV) version of the spreadsheet used in Module 8, Lesson 5.

• gleep — This is used to demonstrate reading fields from a file. It is referenced on Slides 9 and 10.

* fn1, fn2, round, midmin — These files are only used with the extra material for advanced students, on Slides 35, 36, 37, and 38.

**Active Learning Activity/Lab:** 15.SeS\_Lesson3\_AdvancedScripting\_Lab.docx

Students will need the following data file from the directory 16.SeS\_Lesson3\_AdvancedScripting\_DataFiles. This file can be copied to the students’ systems or made available for them to download. PowerPoint slides and data files are included for each Lesson. (Verdana, 12 pt)

* connect.csv — This is a comma-separated value (CSV) version of the spreadsheet used in Module 8, Lesson 5.

**Note:** The material in Lesson 3 includes the use of regular expressions for pattern-matching in programs such as *expr* (1) and *sed* (1). If students have never seen this, the instructor should be prepared to expand on what is in the slides. This may increase the amount of time needed to cover this Lesson.

**Suggestion**

If you want to split this module into two parts, we recommend using Slides 1 through 22 as the first part and Slides 23 through 46 as the second part. There is a natural breaking point after Slide 22, because the class has finished counting and the *while* loop and done two exercises on the subject. The second part begins with editing variable values, which is different enough that a break here makes sense.