Fortinet Technologies (Canada) Inc.

Software Development QA Engineer

Instructions for the Off-site Written Interview

November 2015

Introduction

The successful candidates for the open position of Software Development QA Engineer will take on various tasks, including:

- manual system-level product testing

- setting up test environments

- reproducing customer-reported issues

- learning and understanding of various networking protocols

- developing new automated test systems

- maintaining existing automated test systems

These written interview questions focus on exploring the candidates knowledge of scripting languages used for some of our automated test systems.

This is an open-book test. A list of resources is provided towards the end of this document. You are encouraged to make good use of them. The on-site interview will ask questions that allow each candidate the opportunity to demonstrate their level of knowledge in the same area.

Structure of the Test Questions

This test places the candidate in a situation similar to what the team handles regularly. The NIST organisation provides a data feed of publicly announced software vulnerabilities on their website: https://nvd.nist.gov/download.cfm

The team is tasked with extracting this information on a regular basis.

An initial implementation is provided in Perl. You may review it and rewrite in any other scripting language available on Microsoft Windows or Linux, such as Python, Ruby, Tcl, etc.

Here is the directory structure used by the reference implementation:

nvd

|

--- download # files downloaded from the internet for the task

|

--- lib # Perl modules created for this task are saved here

|

--- log # log messages generated by scripts may be saved here

|

--- scripts # top level scripts for this task

A Perl module exists in the lib directory. It is used by the script in scripts directory. One of the XML files provided by the NIST feed is saved into the download directory.

Required Perl Modules

Perl is usually installed by default on regular Linux desktops. For Microsoft Windows, ActiveState Perl is recommended: http://www.activestate.com/activeperl

The following Perl modules are used:

- strict # standard Perl module. It is always installed.

- warning # same as strict, this is a standard Perl module

- Dumpvalue # Perl core module used for introspection

- FindBin # should also always be available if Perl is installed

- XML::LibXML # needs to be installed to parse the NIST XML files.

Perl Documentation

Here are some useful Perl documents:

- Perl programming landing page

https://www.perl.org/

- CPAN - Perl module pages

http://search.cpan.org

https://metacpan.org/

- XML::LibXML tutorial

http://search.cpan.org/~shlomif/XML-LibXML-2.0122/LibXML.pod

https://metacpan.org/pod/distribution/XML-LibXML/LibXML.pod

the Perl module is a wrapper around the C module:

http://www.xmlsoft.org/

- Dumpvalue manual page

http://perldoc.perl.org/Dumpvalue.html

- DBI

http://dbi.perl.org/

http://search.cpan.org/~timb/DBI-1.634/DBI.pm

https://metacpan.org/pod/DBI

Perl provides module documentation as Perldoc. If the module is installed, the documentation can be accessed using:

perldoc <module\_name>

perldoc FindBin

where <module\_name> is the name of the module. An example for FindBin is

provided.

The implementation provided in lib uses some not-so-common Perl data structures. The following Perl documentation may be useful:

- Perl references tutorials

http://perldoc.perl.org/perlreftut.html

http://perldoc.perl.org/perlref.html

- Perl Object-Oriented Programming

http://perldoc.perl.org/perlootut.html

This is not required for this task. But may be useful for

understanding how to use some of the external modules

(XML::LibXML, Dumpvalue, DBI)

- Perl debugging tutorial

http://perldoc.perl.org/perldebtut.html

Very helpful for debugging Perl scripts and modules. Otherwise, not

required for this task.

- Package

https://perldoc.perl.org/functions/package.html

Most of the common Perl modules can be installed on Linux using yum, dnf or apt-get. On Microsoft Windows, ppm installs most modules.

Relational Database Documentation

The team uses both MySQL and SQLite. Candidate may choose to use either. May also use any other relational database.

MySQL documentation home is at: http://dev.mysql.com/doc/. An installation of MySQL server will be useful for testing.

SQLite documentation is available at http://www.sqlite.org/. There is no need for a database server. This is one advantage of SQLite over other relational database systems.

The Perl DBI module is often used for database implementations from Perl scripts and modules. It presents a consistent interface to Perl applications, irrespective of the underlying relational database server. A tutorial is available at: http://www.mysqltutorial.org/perl-mysql/

Other than the difference in how the connect() method is called, most of the tutorial will apply to other database servers, such as SQLite or PostgreSQL. The module documentation listed earlier is very detailed and useful.

Turning in your work

Place all work, downloaded files and logs in the working directory provided. Retain the directory structure. Here is the suggested final directory layout:

<candidate\_name>

|

--- nvd

|

--- download

|

--- lib

|

--- log

|

--- scripts

where <candidate\_name> is the candidate's full name without spaces or special characters. If you need any new directories, create them under nvd, at the same level as lib and scripts.

It will be helpful to create Answers.txt under the nvd directory. You can provide a short summary of your response to each of the items in Questions.txt.

When all is done, compress the top level directory: <candidate\_name> with the appropriate extension: .zip or .gz. Submit the compressed file back to Fortinet. If you find that the compressed file is too large to send by e-mail, you may upload it to a shared drive like Dropbox or Google Drive, and send information on how to retrieve it. You may also remove the download and log directories from the directory structure before compressing it and sending by e-mail. These two are likely to be the largest directories by size.

Comments and Feedback

If you run into issues, create a plain text file in the top level directory of your submission, at the same level as nvd, to explain how you addressed them.

Good luck!