Module 2 Cheat Sheet: Security Testing and Mitigation Strategies

Skills Network

<table>
<tr><th>Package/Method</th><th>Description</th><th>Code Example</th></tr>
<tr><td>bash</td><td>Bash, or the Bourne Again Shell command, is a command-line interpreter commonly used in Unix-based operating systems. It runs in a text window where the user can interpret commands to carry out various actions.</td><td>Example: This generates a list of numbers and prints them:<pre>#!/bin/bash
# Loop from 1 to 3 and print the numbers
for i in {1..3}; do
    echo "Number: $i"
done</pre>Copied!</td></tr>
<tr><td>alias</td><td>Lets you create a shortcut name for a command, file name, or any shell text. Using aliases saves a lot of time when performing frequent tasks.</td><td>Basic syntax of <code>alias</code> command<pre>alias [new-name[=command]</pre>Copied!
Example 1: Replaces command <code>cd C:\Users\Videos</code> with new alias <code>cdv</code>; so instead of typing <code>cd C:\Users\Videos</code>, one can type <code>cdv</code> to execute the same command<pre>alias cdv="cd C:\Users\Videos"
cdv</pre>Copied!
Example 2: Use <code>-p</code> option to view all your alias commands<pre>alias -p</pre>Copied!
Example 3: Use unalias with <code>-a</code> option to remove all your alias commands<pre>unalias -a</pre>Copied!
Example 4: Use unalias command with name of alias on to remove specific alias command<pre>unalias cdv</pre>Copied!</td></tr>
<tr><td>cd</td><td>Used to move efficiently from the existing working directory to different directories on your system.</td><td>Basic syntax of <code>cd</code> command<pre>cd [options] [directory]</pre>Copied!
Example 1: Change directory location to <code>folder1</code><pre>cd /usr/local/folder1</pre>Copied!
Example 2: Get back to previous working directory<pre>cd -</pre>Copied!
Example 3: Move up one level from present working directory tree<pre>cd ..</pre>Copied!</td></tr>
<tr><td>dependency-check</td><td>OWASP dependency-check is a software composition analysis utility that detects publicly disclosed vulnerabilities in application dependencies.</td><td>Basic syntax of <code>dependency-check</code> command<pre>dependency-check.bat [options] --scan[directory] --out [directory]

`Options
project: The name of your project as it should appear in the report
scan [directory]: The folder which contains the 3rd party dependency libraries
out [directory]: The folder where the vulnerability analysis reports should be exported to`</pre>Copied!
Example 1: In <code>Windows</code>, use command as given below<pre>dependency-check.bat --project "my_project" --scan "c:\java\application\lib"</pre>Copied!
Example 2: In <code>Linux</code>, use command as given below<pre>dependency-check.sh --project "my_project" --scan "/java/application/lib"</pre>Copied!</td></tr>
<tr><td>docker network</td><td>You can use this code to manage networks. The subcommands can be used to create, inspect, list, remove, prune, connect, and disconnect networks.</td><td>Create a <code>docker network</code><pre>docker network create my_network</pre>Copied!
Verify Network Connection<pre>docker network inspect my_network</pre>Copied!
List docker Network<pre>docker network ls</pre>Copied!
Remove docker network<pre>docker network rm NETWORK_NAME_OR_ID</pre>Copied!
Prune docker network<pre>docker network prune</pre>Copied!
Connect Docker Network<pre>docker network connect NETWORK_NAME CONTAINER_NAME_OR_ID</pre>Copied!
Disconnect Docker Network<pre>docker network disconnect NETWORK_NAME CONTAINER_NAME_OR_ID</pre>Copied!</td></tr>
<tr><td>docker ps</td><td>Lists the running containers by default. We can use different flags to get the list of other containers that are in stopped or exited status.</td><td><pre>docker ps [OPTIONS]</pre>Copied!
If you want to see all containers, including the stopped ones, you can use the <code>-a</code> or <code>-all</code><pre>docker ps -a</pre>Copied!</td></tr>
<tr><td>docker pull</td><td>You can download Docker images from the internet.</td><td><pre>docker pull [OPTIONS] IMAGE_NAME[:TAG]</pre>Copied!</td></tr>
<tr><td>docker run</td><td>It runs a command in a new container, getting the image and starting the container if needed.</td><td><pre>docker run [OPTIONS] IMAGE [COMMAND] [ARG...]</pre>Copied!</td></tr>
<tr><td>git clone</td><td>You can create a copy of a specific repository or branch within a repository.</td><td><pre>git clone REPOSITORY_URL [DESTINATION_DIRECTORY]</pre>Copied!</td></tr>
<tr><td>jake</td><td>Jake is a simple JavaScript build program with capabilities similar to the regular make or rake command. It has the following features:
Jakefiles are in standard JavaScript syntax
Tasks with prerequisites
Namespaces for tasks
Async task execution</td><td><pre>jake ddt</pre>Copied!</td></tr>
<tr><td>jq</td><td>Used to transform JSON data into a more readable format and print it to the standard output on Linux.</td><td>Basic command syntax<pre>jq [options] [filter] [file]</pre>Copied!
Let's consider a simple <code>example.json</code> file that describes an array as below -<pre>[
    {
        color: "red",
        value: "#f00"
    },
    {
        color: "green",
        value: "#0f0"
    },
    {
        color: "blue",
        value: "#00f"
    }
]
</pre>Copied!
Example 1: The identity filter <code>.</code> takes the input and produces prints all output unchanged<pre>jq '.' example.json</pre>Copied!
Example 2: Extract the name of each color from each object in the array<pre>jq '.[].color' example.json
jq 'map(has("color"))' example.json</pre>Copied!</td></tr>
<tr><td>ls</td><td>Basic Linux command used for listing information regarding files and directories within the file system.</td><td>Basic command syntax<pre>ls [options] [file/directory]</pre>Copied!
Example 1: Sorts the file names displayed in the order of last modification time. <code>r</code> is for displaying in reverse order<pre>ls -lt
ls -ltr</pre>Copied!
Example 2: Displays hidden files<pre>ls -a</pre>Copied!</td></tr>
<tr><td>pip</td><td>To ensure that requests will function, the pip program searches for the package in the Python Package Index (PyPI), resolves any dependencies and installs everything in your current Python environment.</td><td><pre>pip list</pre>Copied!</td></tr>
<tr><td>pip install</td><td>The pip install <package> command looks for the latest version of the package and installs it.</td><td><pre>pip install example_package</pre>Copied!</td></tr>
<tr><td>sonar-scanner</td><td>The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system.</td><td>Basic Syntax of the sonar-scanner command; commonly used options are:<pre>-D,--define <arg> Define property
-h,--help Display help information
-v,--version Display version information
-X,--debug Produce execution debug output</pre><pre>sonar-scanner [options]</pre>Copied!
Example: verify your installation by executing the command<pre>sonar-scanner -h</pre>Copied!</td></tr>
<tr><td>wget</td><td>Stands for web get. The `wget` is a free non-interactive file downloader command. Non-interactive means it can work in the background when the user is not logged in.</td><td>Basic Syntax of the wget command; commonly used options are <code>[-V], [-h], [-b], [-e], [-o], [-a], [-q]</code><pre>wget [options]</pre>Copied!
Example 1: Specifies to <code>download file.txt</code> over HTTP website url into the working directory.<pre>wget http://example.com/file.txt</pre>Copied!
Example 2: Specifies to download the <code>archive.zip</code> over HTTP website url in the background and returning you to the command prompt in the interim.<pre>wget -b http://www.example.org/files/archive.zip</pre>Copied!</td></tr>
<tr><td>which</td><td>Used to locate the executable file associated with the given command by searching it in the path environment variable</td><td>Basic syntax of <code>which</code> command<pre>which [option] [filename1] [filename2]</pre>Copied!
Example 1: To know where exactly the java program is located, execute the command as below<pre>which java</pre>Copied!
Example 2: To know exact location of multiple programs, execute the command as below<pre>which java python</pre>Copied!
Example 3: By default which command will display the path of the first occurrence, but if we want to display all the occurrences of the program, then we can use <code>-a</code> option.<pre>which -a python</pre>Copied!</td></tr>
</table>

Changelog

Date	Version	Changed by	Change Description
2023-08-16	0.1	Gagandeep	Initial version created
2023-08-17	0.2	Shilpa Giridhar	Updated Cheat Sheet