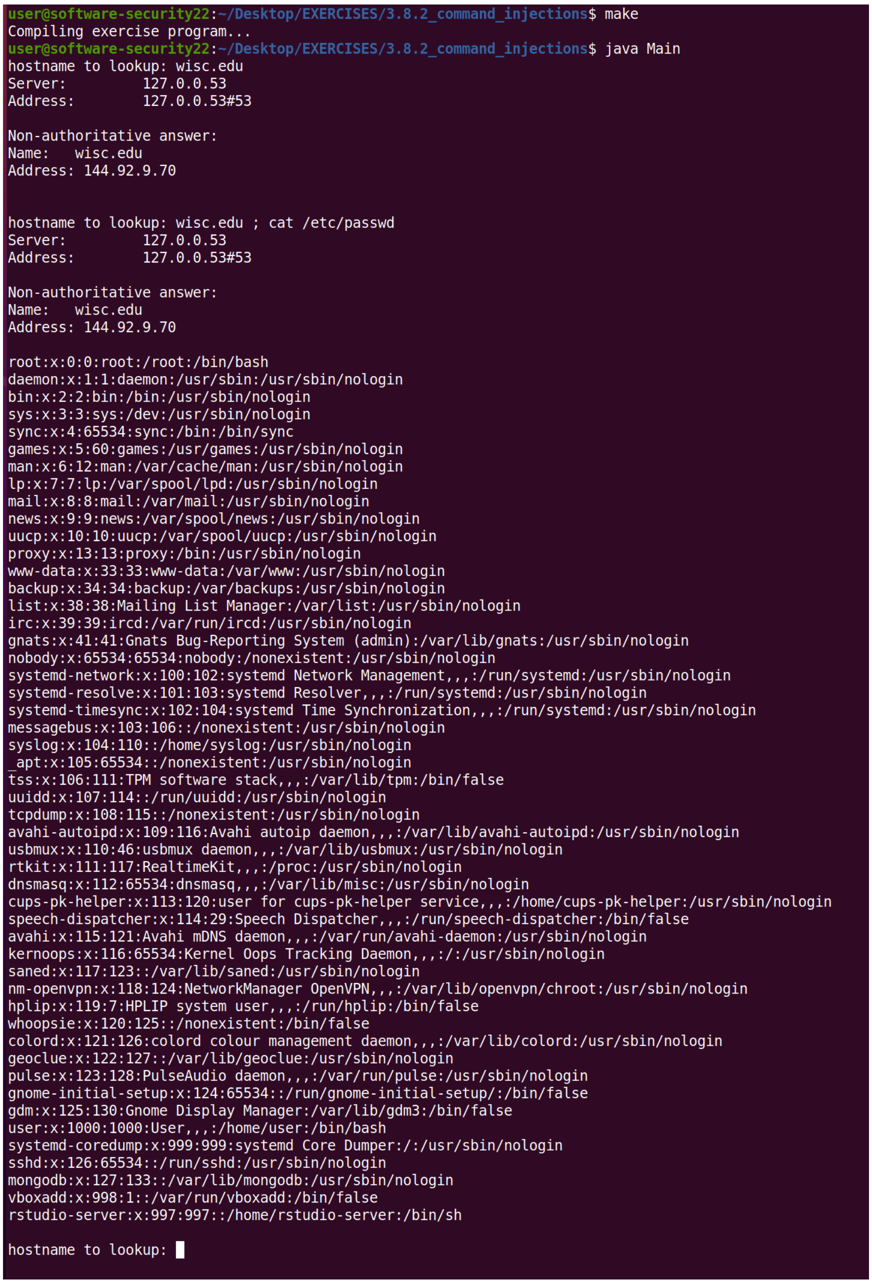
CS 542 – Introduction to Software Security
Exercise on Command Injection

Binhao Chen (bchen276@wisc.edu), Steven Yang (yang558@wisc.edu)

Due: October 18 at 2:30pm

1 Command Injection Vulnerability

1.1 Screenshots or printouts showing the inputs used for the attack, and the outputs you got from the system

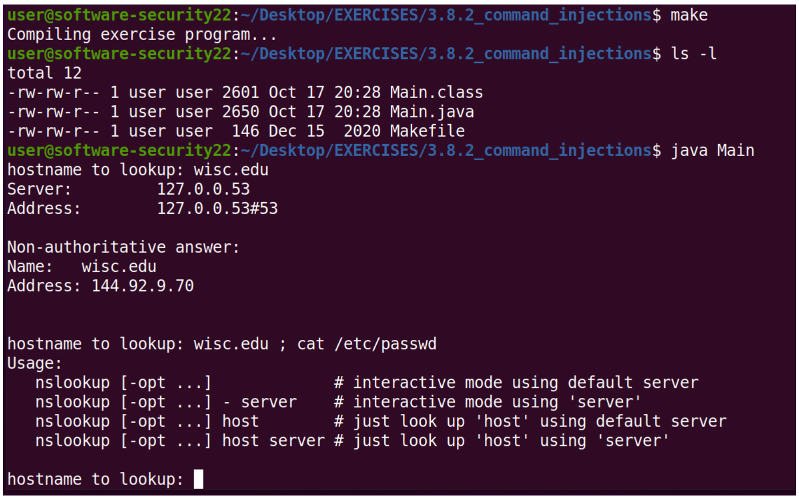


```
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ make
Compiling exercise program...
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ java Main
hostname to lookup: wisc.edu
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   wisc.edu
Address: 144.92.9.70


hostname to lookup: wisc.edu ; cat /etc/passwd
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   wisc.edu
Address: 144.92.9.70

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
user:x:1000:1000:User,,,:/home/user:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sshd:x:126:65534::/run/sshd:/usr/sbin/nologin
mongodb:x:127:133::/var/lib/mongodb:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
rstudio-server:x:997:997::/home/rstudio-server:/bin/sh

hostname to lookup: █
```

1.2 Your commented code for the 2 mitigation approaches

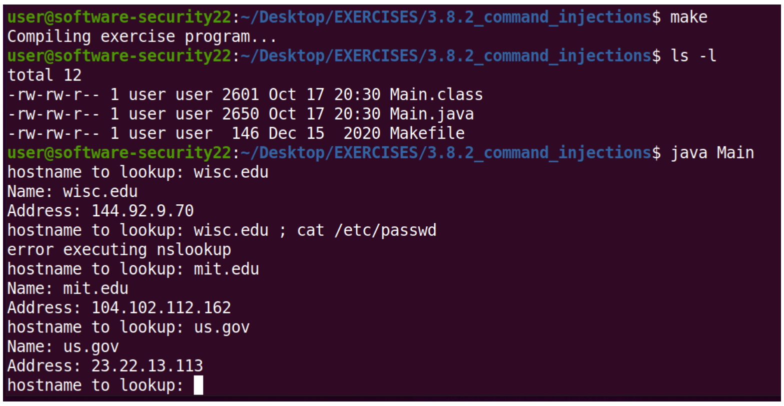
```java

import java.io.BufferedReader;
import java.io.Console;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.InetAddress;

/**
 * Main execution class for cmd_injection exercise. Prompts user for input to
 * the nslookup command and prints the output.
 *
 * @author Joseph Eichenhofer
 *
 */
public class Main {

    /**
     * Prompts user for hostname to lookup. Performs DNS resolution and prints
     * address/info for the given hostname.
     *
     * @param args
     *             n/a
     */
    public static void main(String[] args) {
        Console terminal = System.console();

        if (terminal == null) {
            System.out.println("Error fetching console. Are you running from an IDE?");
            System.exit(-1);
        }

        while (true) {
            String hostname = terminal.readLine("hostname to lookup: ");

            if (hostname.toLowerCase().equals("exit"))
                break;

            try {
                // System.out.println(rDomainName(hostname));

                // This is the second mitigation appraoch by calling
                // the new created method newDomainName;
                // the newDomainName method calls
                // getByName method in InetAddress class
                // to determines the IP address of a host,
                // given the host's name
                System.out.println(newDomainName(hostname));
            } catch (IOException e) {
                System.out.println("error executing nslookup");
            }
        }
    }

    /**
     * Lookup given hostname using getByName method in InetAddress class.
     * Return the output/error of the getByName method as string.
     *
     * @param hostname
     *                  hostname/domain to lookup
     * @return string output of nslookup command
     * @throws IOException
     *
     */
    private static String newDomainName(String hostname) throws IOException {
        // We first instantiate a InetAddress class called host,
        // By calling the getByName method, it returns
        // the IP address of a host, given the host's name
        InetAddress host = InetAddress.getByName(hostname);
        String temp = host.toString();

        // The IP address is split into two components, which are the
        // hostname, followed by its IP address;
        // The output is formatted as follows.
        String[] output = temp.split("/");
        return "Name: " + output[0] + "\nAddress: " + output[1];
    }

    /**
     * Lookup given hostname using nslookup command. Return the output/error of the
     * nslookup command as string.
     *
     * @param hostname
     *                 hostname/domain to lookup
     * @return string output of nslookup command
     * @throws IOException
     *                     if unable to execute the command or read its output
     */
    private static String rDomainName(String hostname) throws IOException {
        // execute the nslookup command
        // String[] cmd = { "/bin/sh", "-c", "nslookup " + hostname };

        // By constructing a new string with command ``nslookup''
        // and input ``hostname'' only,
        // this method will execute the intended program directly,
        // instead of executing a shell command (e.g., /bin/sh).
        // Therefore, we remove the shell interpreter's ability
        // to execute multiple programs
        // thereby mitigate the vulnerability.
        String cmd = "nslookup " + hostname;
        Process proc = Runtime.getRuntime().exec(cmd);

        // capture output from command
        BufferedReader stdOut = new BufferedReader(new InputStreamReader(proc.getInputStream()));
        BufferedReader stdErr = new BufferedReader(new InputStreamReader(proc.getErrorStream()));

        StringBuilder output = new StringBuilder();
        String currLine = null;
        while ((currLine = stdOut.readLine()) != null) {
            output.append(currLine + "\n");
        }
        while ((currLine = stdErr.readLine()) != null) {
            output.append(currLine + "\n");
        }

        // return the result
        return output.toString();
    }
```

1.3 Screenshots or printouts showing the inputs and outputs after fixing the vulnerability, for the 2 mitigation approaches.



```
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ make
Compiling exercise program...
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ ls -l
total 12
-rw-rw-r-- 1 user user 2601 Oct 17 20:28 Main.class
-rw-rw-r-- 1 user user 2650 Oct 17 20:28 Main.java
-rw-rw-r-- 1 user user  146 Dec 15  2020 Makefile
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ java Main
hostname to lookup: wisc.edu
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   wisc.edu
Address: 144.92.9.70


hostname to lookup: wisc.edu ; cat /etc/passwd
Usage:
   nslookup [-opt ...]             # interactive mode using default server
   nslookup [-opt ...] - server    # interactive mode using 'server'
   nslookup [-opt ...] host        # just look up 'host' using default server
   nslookup [-opt ...] host server # just look up 'host' using 'server'

hostname to lookup: 
```

Figure 1: Mitigate by executing the intended program directly



```
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ make
Compiling exercise program...
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ ls -l
total 12
-rw-rw-r-- 1 user user 2601 Oct 17 20:30 Main.class
-rw-rw-r-- 1 user user 2650 Oct 17 20:30 Main.java
-rw-rw-r-- 1 user user  146 Dec 15  2020 Makefile
user@software-security22:~/Desktop/EXERCISES/3.8.2_command_injections$ java Main
hostname to lookup: wisc.edu
Name: wisc.edu
Address: 144.92.9.70
hostname to lookup: wisc.edu ; cat /etc/passwd
error executing nslookup
hostname to lookup: mit.edu
Name: mit.edu
Address: 104.102.112.162
hostname to lookup: us.gov
Name: us.gov
Address: 23.22.13.113
hostname to lookup: 
```

Figure 2: Mitigate by creating a new method that replaces rDomainName() and generating the appropriate output using java.net.InetAddress

1.4 An explanation on your attack and your mitigations

Attack: We attack by passing in a host name followed by a semicolon, then enter the second command that we want to execute (here we use the innocuous“cat /etc/passwd”). The semicolon ends the nslookup command and allows the second command to be executed to print out sensitive information.

Mitigation 1: The first way to mitigate is to execute the intended program(nslookup) directly, instead of executing a shell command(e.g., /bin/sh). We directly pass in the string ”nslookup ” + hostname to execute the program. In this way, attacker cannot use the shell interpreter’s ability to execute multiple programs.

Mitigation 2: The second way to mitigate is to use an internal API. We create a new method that utilizes the java.net.InetAddress and the method getByName(hostname) to retrieve the IP address of the host. In this way, it will not allow multiple commands to be executed and the attacker’s input will become a strange string. The method will throw an IOException if the input string is not appropriate.