Upgrading the reverse-shell (and understanding it)

**Step 1. What kind of python do you have?**



just checks the type of python the system is running.

**Step 2. Create your improved shell.**



First Part: This is the command that “spawns” a more interactive shell from your basic reverse shell, the breakdown is:

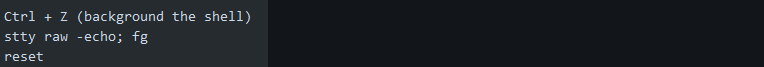
* python -c: This invokes the Python interpreter with the -c option, which allows you to specify a Python command as a string.
* 'import pty; pty.spawn("/bin/bash")': This is the Python code executed using the -c option. It imports the pty module and uses the pty.spawn function to spawn a new interactive shell ("/bin/bash" in this case). -PTY stands for pseudo-terminal

The command essentially tricks the system into thinking that you are interacting with a terminal, providing you with a more feature-rich shell experience.

Second Part: “SHELL=/bin/bash script -q /dev/null”, is an alternative for systems that don't have Python installed. It also aims to upgrade a basic shell to a more interactive one. Here's a breakdown:

* SHELL=/bin/bash: This sets the SHELL environment variable to "/bin/bash," indicating that the shell to be used is Bash.
* script -q /dev/null: The script command is used to make a typescript of a terminal session. In this case, it's used to start a new shell session with input and output logged to /dev/null (a special file that discards all data written to it).

**Step 3. Configure your upgraded shell.**



a common technique used to background a shell that is running in the foreground and then bring it back to the foreground with improved interactivity. Let's break down each part:

* **Ctrl + Z** (background the shell):
  + Pressing Ctrl + Z in a terminal sends a SIGTSTP signal to the foreground process, effectively stopping it and putting it in the background.
* **stty raw -echo; fg:**
  + stty raw -echo: This changes the terminal settings. stty is a command-line utility to change and print terminal line settings. The options raw and -echo are used to set the terminal to raw mode (passing characters through without processing) and turn off echoing (not displaying entered characters), respectively.
  + fg: This command brings the most recent background job to the foreground.

These commands are used to make the shell more interactive after it has been backgrounded. The stty commands are used to modify terminal settings to allow proper interaction, and fg brings the backgrounded process back to the foreground.

1. **reset:**
   * The reset command is often used to initialize the terminal to its default state. This can be helpful if the terminal appears to be in a strange state after backgrounding and bringing the shell back to the foreground.

In summary, the entire sequence is a set of commands to background a shell, modify terminal settings for improved interactivity, bring the shell back to the foreground, and reset the terminal if needed.

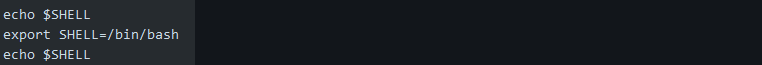
**Step 4. What kind of Terminal do you want?**

Sometimes the computer asks you what type of TERM you want to use. When the terminal prompts for the "TERM" to use, it asks you to specify the type of terminal emulation you want to use. The "TERM" environment variable defines the type of terminal that your shell session should emulate. There are many types, but we’ll use xterm:



The echo $TERM command is used to display the current value of the TERM environment variable in your terminal. The export TERM=xterm-256color command sets the TERM variable to "xterm-256color." Finally, echo $TERM is used again to confirm the updated value. After this step you should be able to clear your terminal.

**Step 5. Validate your new shell:**



The echo $SHELL command is used to display the current shell that is running in your terminal. The export SHELL=/bin/bash command sets the SHELL environment variable to "/bin/bash." Finally, echo $SHELL is used again to confirm the updated value.

* **echo $SHELL:**
  + Displays the current value of the SHELL environment variable, which represents the path to the user's preferred shell.
* **export SHELL=/bin/bash:**
  + Sets the SHELL environment variable to "/bin/bash." This change doesn't actually switch your shell; it just updates the environment variable to indicate that your preferred shell is Bash.
* **echo $SHELL:**
  + Displays the updated value of the SHELL environment variable, which should now be "/bin/bash."

**Step 6. Gather the info about your machine (attacker) and your target’s machine (victim):**

A close up of a screen

Description automatically generated

These commands are used to gather information about the terminal environment on both the attacker (the machine where you're running these commands) and the victim (the machine you're attacking). Let's break down each part:

**Attacker's Machine:**

* **echo $TERM:**
  + Displays the current value of the TERM environment variable, which represents the type of terminal emulation in use.
* **tput lines:**
  + Outputs the number of lines (rows) in the terminal.
* **tput cols:**
  + Outputs the number of columns in the terminal.

**Victim's Machine:**

* **stty rows 38 columns 116:**
  + Uses the stty command to set the number of rows to 38 and columns to 116 on the victim's machine. This is an attempt to configure the terminal size on the victim's side.