UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Informatics Institute of Technology

in collaboration with

the University of Westminster, UK

BEng (Hons) in Software Engineering

6COSCO008C Security and Forensics

Coursework Report

by

Dinuka Raneesh Anton Gomez

IIT ID: 2016087 | **UoW ID:** 16266986

Module Leader

Mr. Saman Hettiarachchi

Table of Contents

A Information Gathering 1

Question A - 1 1

Question A - 2 2

Question A - 3 2

Question A - 4 3

B Finding and Exploiting Vulnerabilities 5

Question B - 1 5

Question B - 2 7

Question B - 3 9

Question B - 4 11

C Man-in-the-Middle Attacks and Social Engineering 15

Question C - 1 15

Question C - 2 17

Question C - 3 20

D Protecting your Server 23

Question D - 1 23

Question D - 2 23

Question D - 3 24

Question D - 4 25

Question D - 5 26

References 27

List of Figures

Figure 1: Open ports on server....1
Figure 2: Services running on open ports on server....2
Figure 3: Verified admin trying to login....5
Figure 4: Viewing credentials after intercepting request....6
Figure 5: Tampering with credentials....6
Figure 6: Successful data tampering and denial of service....7
Figure 7: Verifying if the application is vulnerable to SQL injection....8
Figure 8: Testing unsanitized input....8
Figure 9: Successful SQL Injection attack....9
Figure 10: Testing for XSS vulnerability....10
Figure 11: Unencoded input in page source....10
Figure 12: Result of a sample XSS attack....11
Figure 13: Regular password change screen....12
Figure 14: Attacker gets password change markdown from page source....12
Figure 15: Attacker alters password change form with hidden fields....13
Figure 16: User is led to malicious page....13
Figure 17: Password successfully changed by attacker....14
Figure 18: Getting client credentials by ARP poisoning....15
Figure 19: Client submitting data to the form field....16
Figure 20: Monitoring form input from client to server using Wireshark....16
Figure 21: Client entering invalid username....17
Figure 22: Using Ettercap regex filter to modify username with correct credential....17
Figure 23: Edited markdown on login page....18
Figure 24: Attacker's custom PHP script to get and store the user's credentials....18
Figure 25: User creating a new account on the cloned site....19
Figure 26: User logging into the cloned site....19
Figure 27: Harvested credentials on the attacker's machine....20
Figure 28: Creating a reverse meterpreter shell....21
Figure 29: Creating a new listener with metasploit....21
Figure 30: Convincing the user to download the shell file....21

Figure 31: User running the reverse shell on the client 22
Figure 32: Attacker gaining access to the client's shell 22
Figure 33: Attacker accessing classified research directories.....22
Figure 34: Port knocking visualization (Arora, 2013) 23
Figure 35: IDS vs IPS visualization (IPWITHEASE, 2017) 24

Scenario

You are hired as a penetration tester for a private research and development company working on government projects. Their collaboration tool allows them to share confidential data and information with the relevant government organization. The collaboration tool does not hold personal information or financial information for anyone. Users credentials are stored on the database. The data stored are very confidential. Not all users have the same privilege.

Word Count

2487 words[1]

Note:

Given value is the length of the main body of content excluding the cover page, pages **i** to **iii**, titles, image captions and references.

[1] Results from https://wordcounter.net/

A Information Gathering

Question A - 1

The discovered open ports on the collaboration tool's server can be seen in Figure 1. Out of them, a few popular ports and their threats are described as follows.

- 22 – Attackers have the ability to remotely access the server, bypass firewalls and steal/corrupt classified research source files, undetected, by exploiting SSH port forwarding if any user has enabled local port forwarding on a client (Turner, 2019).
- 80 – Since HTTP is unencrypted, attackers could utilize cross-site scripting to change the layout of the website and lead users to interact with false components (Geer, 2017). This can lead to attackers stealing sensitive government data such as credentials during login process.
- 8080 – Usually, servers attached to port 8080 are legacy systems that haven't been updated in a long time (Geer, 2017). Remote attackers can perform Cross-Site Request Forgery (CSRF) attacks and access administrative controls during a session and even manipulate research contracts belonging to government organizations.

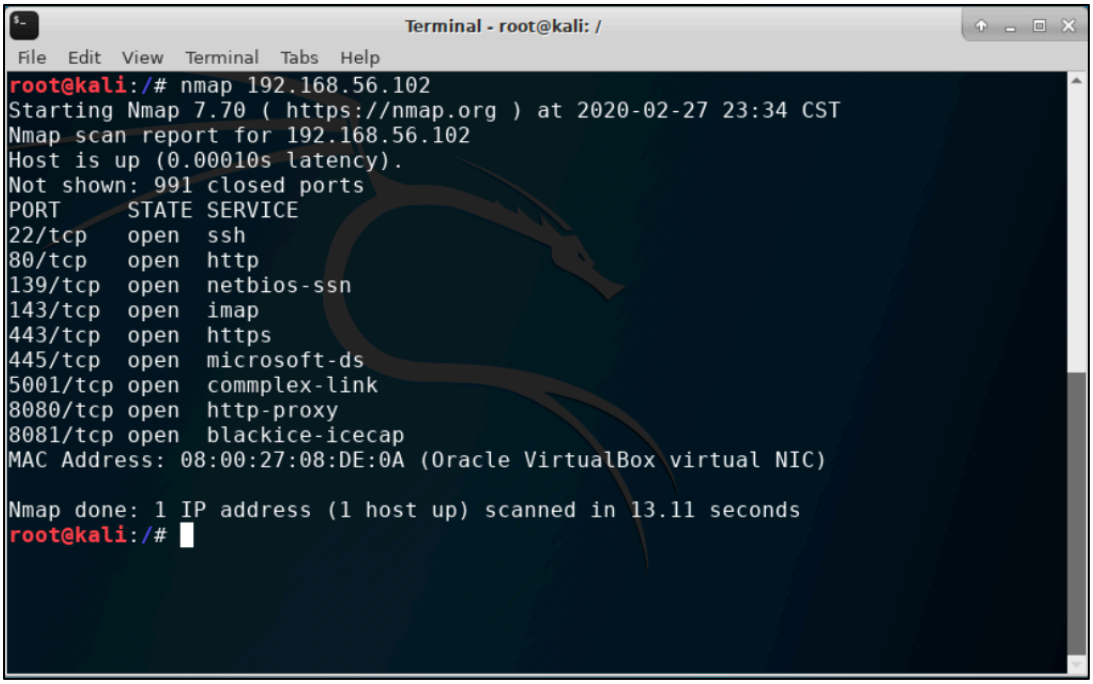

Figure 1: Open ports on server

Question A - 2

- netbios-ssn is an extremely important service to be protected because it is the session service for the NetBios protocol (Chandel, 2017) which allows devices on a Local Area Network to communicate with each other. Attackers can run NBSTAT over TCP/IP on port 139 (Geer, 2017) and gain access to device names, IP addresses, applications running on each system and even user IDs. This way, attackers have access to plenty of sensitive information to launch a coordinated attack and if null sessions are allowed they can even connect to the collaboration tool and access shared documents and research prototypes.
- ssh (port 22) is another important service to be protected because it allows direct access to the server remotely. There are multiple concerns that has to be looked at such as rogue host keys that haven't been authorized, terminated employees credentials that have not been invalidated and deprecated configurations which make it possible for attackers to access and corrupt running prototypes or pivot directly into government systems because of persistently trusted keys stored in the service (Turner, 2019).

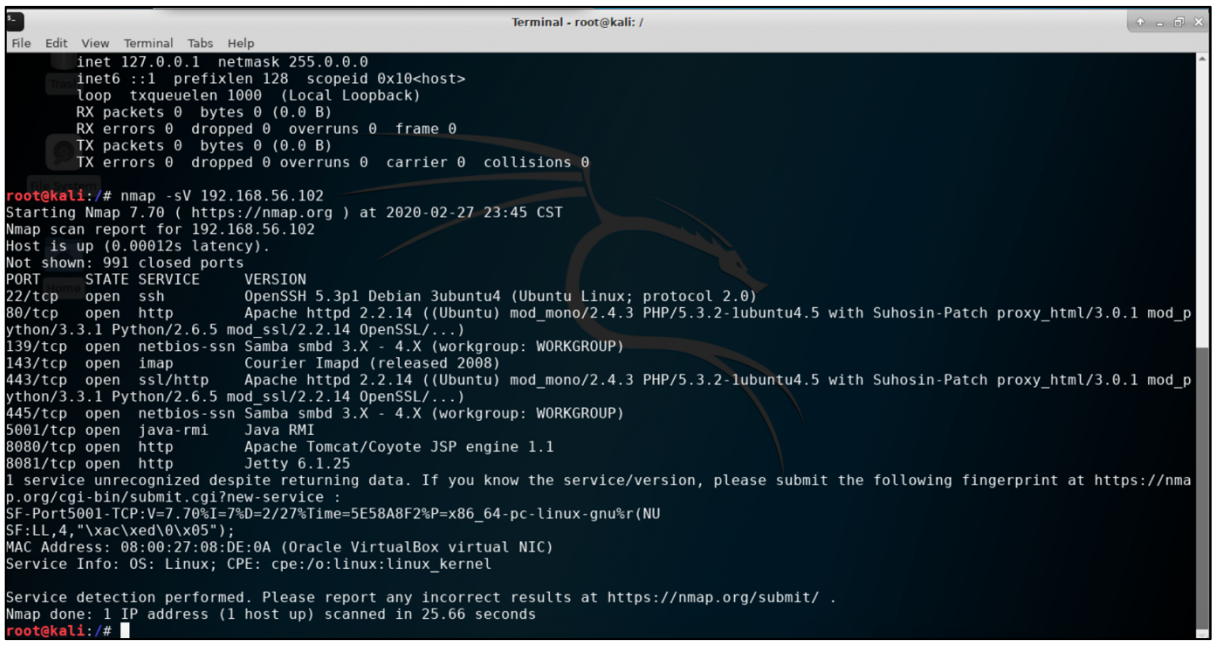

Figure 2: Services running on open ports on server

Question A - 3

1. netbios-ssn (Samba smbd 3.X to 4.X)

- In Samba versions 3.0.20 to 3.0.25rc3 there exists a command execution vulnerability which allows attackers to execute commands by giving a username containing shell meta-characters, when using the *username map script* configuration *(Samba 'username map script' Command Execution*, 2018).
- The bundled LDAP client library in Samba 3.X and 4.X did not recognize the *client ldap sasl wrapping* which allowed man in the middle attackers to perform LDAP protocol downgrade attacks (*Vulnerability Details : CVE-2016-2112*, 2016).
- Between versions 3.5 and 4.6.4, Samba is vulnerable to remote code execution, which allows a malicious client to upload a shared library to the server and cause the server to load and execute it (*Vulnerability Details : CVE-2017-7494*, 2017).

2. ssh (OpenSSH 5.3p1)
   - In OpenSSH versions before 7.6 the process_open function doesn't prevent write operations in read-only mode. This allows attackers to create files of zero-length (*Vulnerability Details : CVE-2017-15906*, 2017).
   - In OpenSSH versions 5.X to before7.1p2, the resend_bytes function allows remote servers to get sensitive information from memory by requesting for transmission of a full buffer (*Vulnerability Details : CVE-2016-0777*, 2016).
   - In OpenSSH versions before 7.4, sshd allows attackers to perform DOS attacks (null pointer dereferencing and daemon crashes) via an *out-of-sequence NEWKEYS message* (*Vulnerability Details : CVE-2016-10708*, 2018).

Question A - 4

According to (SecurityTrails Team, 2019), the four least secure ports out of the open ports on the server machine are the ssh (22), http (80), netbios-ssn (139) and imap (143) ports.

- OpenSSH (port 22)
  - Government organizations and research firms can easily generate millions of SSH keys for various tasks. It is very difficult to keep track of them in the midst of server upgrades, employee terminations etc. If attackers get hold of even one private key, they can gain a long-term network entry point and pose as a valid user and corrupt/steal time-sensitive and classified shared research data.
- NetBios (port 139)

  - Attackers can retrieve information about the OS, services and other applications running on the server through running an NBSTAT attack. The attacker will also receive user IDs (Olzak, 2007), and these can be used in SQL injection attacks to retrieve credentials of research employees and other sensitive user data.

- Courier Imapd (port 143)
  - If an anti-spam program is not implemented, the server can be subject to phishing scams through emails. Further, most IMAP servers have flexible buffer overflows during logins that can be exploited by attackers to cause Denial of Service attacks (*Port 143 Details*, 2009), and halt productivity of the research employees.
- Apache httpd (port 80)
  - Since http is an unsecure protocol, attackers can utilize XSS attacks to run background scripts to modify shared research documents as well as source files for prototypes through remote access. XSS attacks can be also executed to harvest user credentials and steal cookies during verified sessions as well (Cotten, 2016).

B Finding and Exploiting Vulnerabilities

Question B - 1

It was discovered that the application is vulnerable to data tampering. The pen-tester was able to intercept requests made to the server at a login instance and view/tamper with the credentials. This is a rather risky vulnerability as attackers can prevent verified users from accessing the application (DoS) and also gain access to classified government data and cutting-edge research that is shared on the application by pretending to be a verified user session.

In Figure 3, a verified admin user is seen trying to login to the collaboration tool. The attacker intercepts this request (see Figure 4) and steals the credentials while also tampering and altering the admin's credentials (see Figure 5). This causes a Denial of Service attack as the admin is unable to login to the application (see Figure 6).

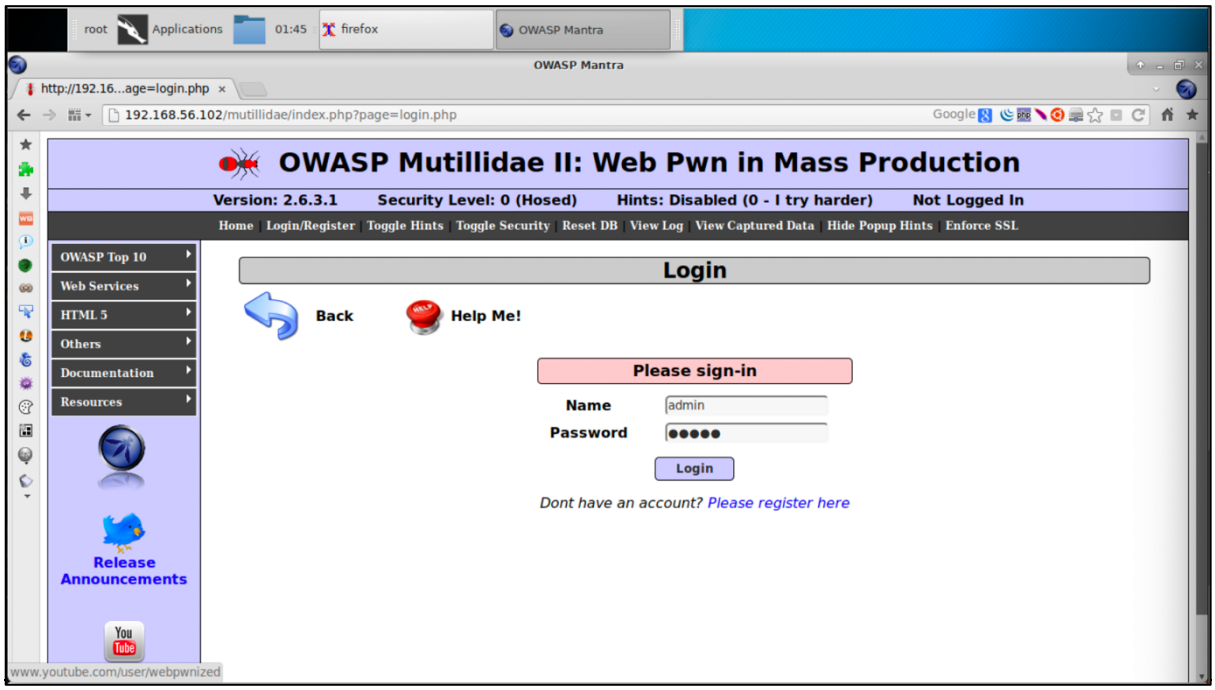

Figure 3: Verified admin trying to login

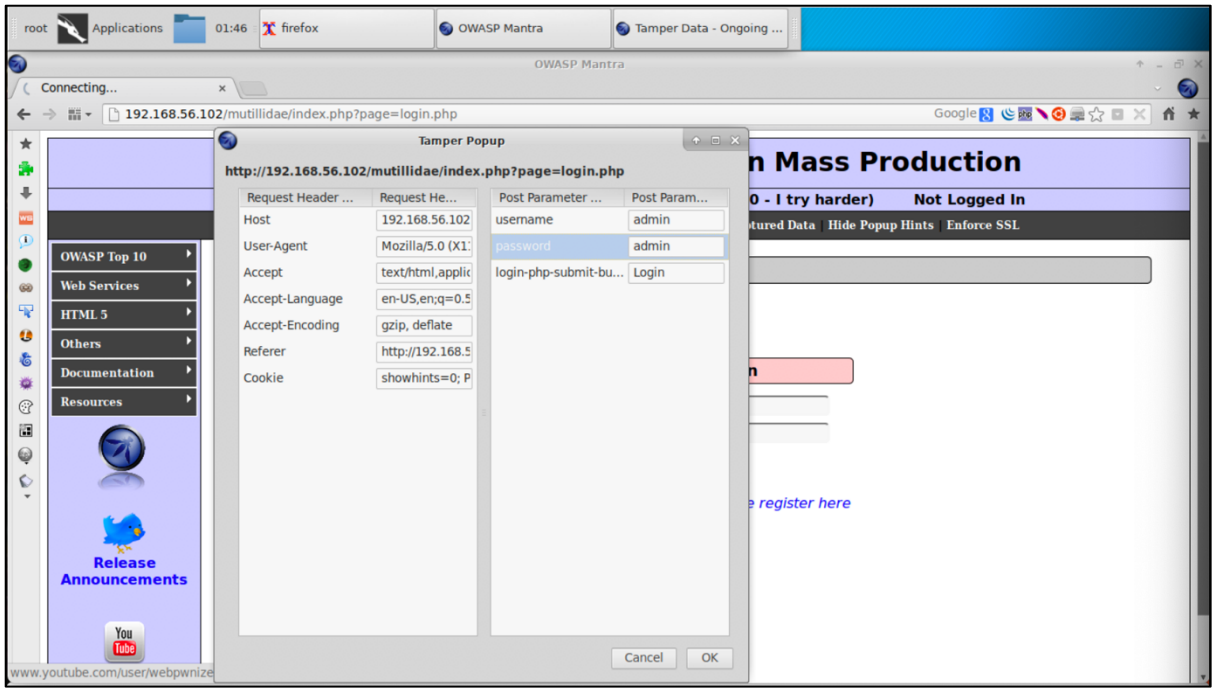

Figure 4: Viewing credentials after intercepting request

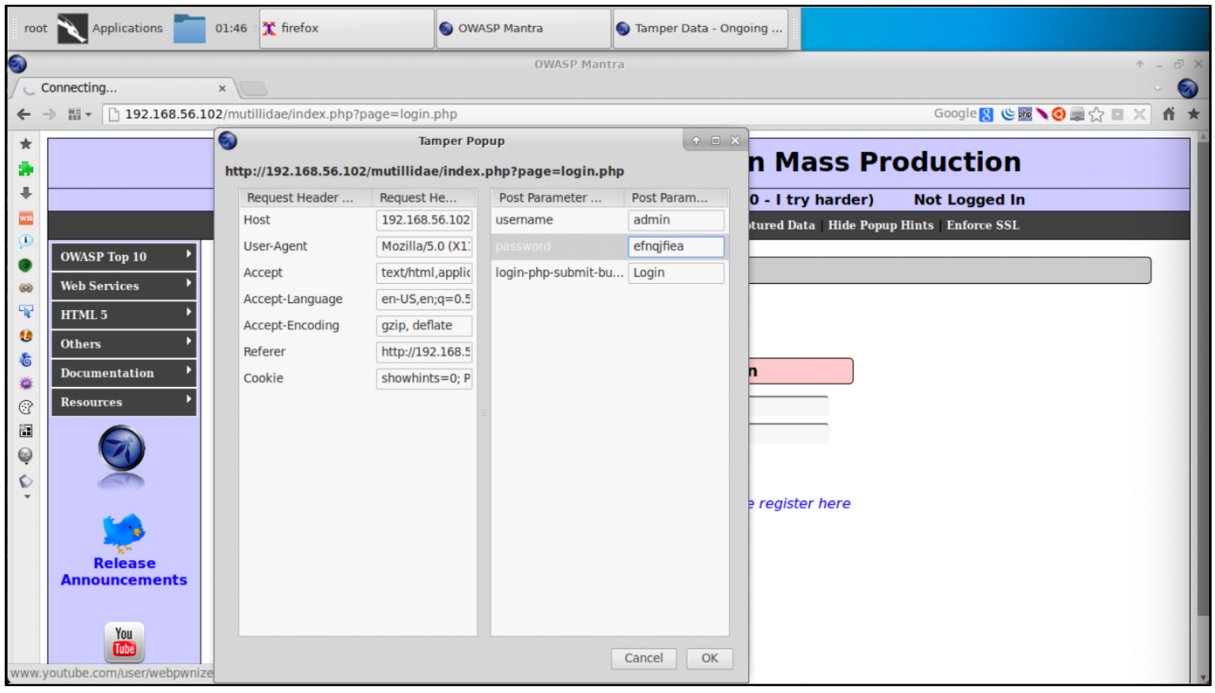

Figure 5: Tampering with credentials

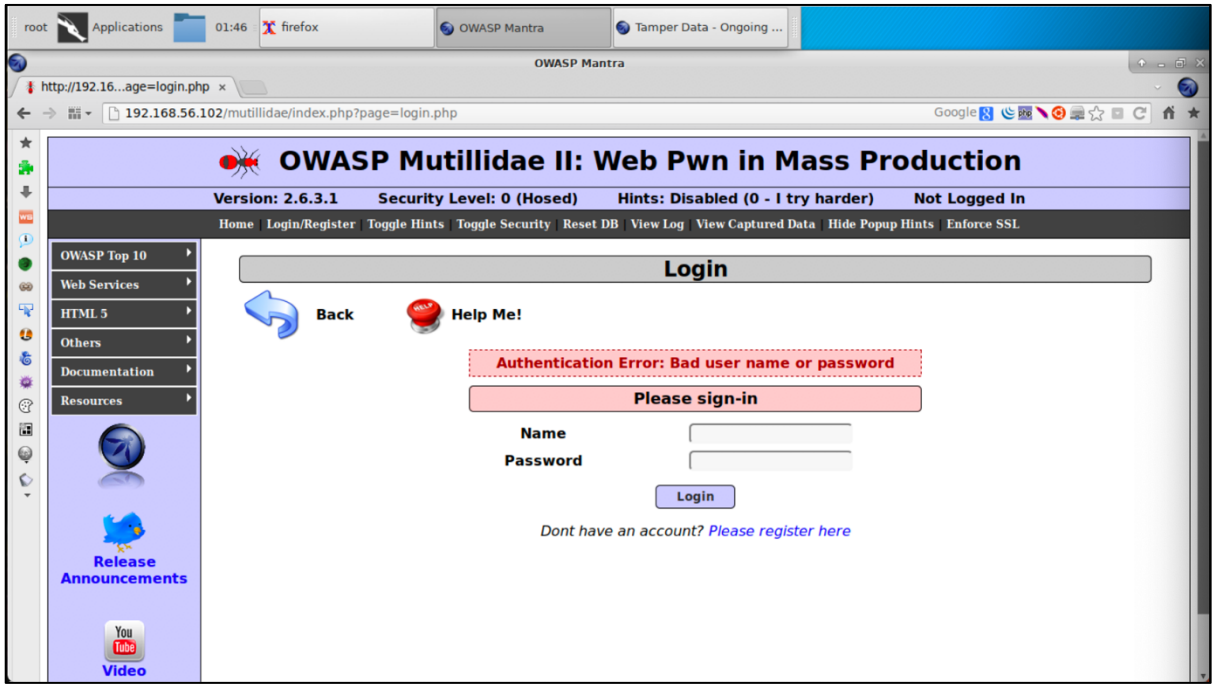

Figure 6: Successful data tampering and denial of service

Question B - 2

The application is found to be vulnerable to SQL injection by submitting username followed by ' to the field (see Figure 7). This returns an error where we can see the way the SQL query is built internally. Here we see that input data isn't sanitized and is susceptible to SQL injection. To further test this, ' -- was submitted, in which we are submitting a blank username and closing the string followed by a comment symbol (--) which comments out the password input. As expected, the system does not issue an error and instead treats it as an invalid input credential (see Figure 8). To exploit this vulnerability we issue a statement using an OR clause, *jeremy' or 1 = 1* --. This translates to the system that it should retrieve all data if the username is *jeremy* or if 1 equals 1 (which is always true). This displays all the user data to the attacker and successfully accomplishes an SQL injection attack (see Figure 9).

This is quite an unforgiving vulnerability to have and one of the most common forms of malware injection (Chou, 2013), as attackers can easily retrieve credentials for not only user accounts, but admin accounts as well, which give them administrative access to the collaboration tool from which they could steal and alter delivery routes of prototypes, invalidate contracts and also access and corrupt/sell innovative technologies from ongoing researches for financial gain.

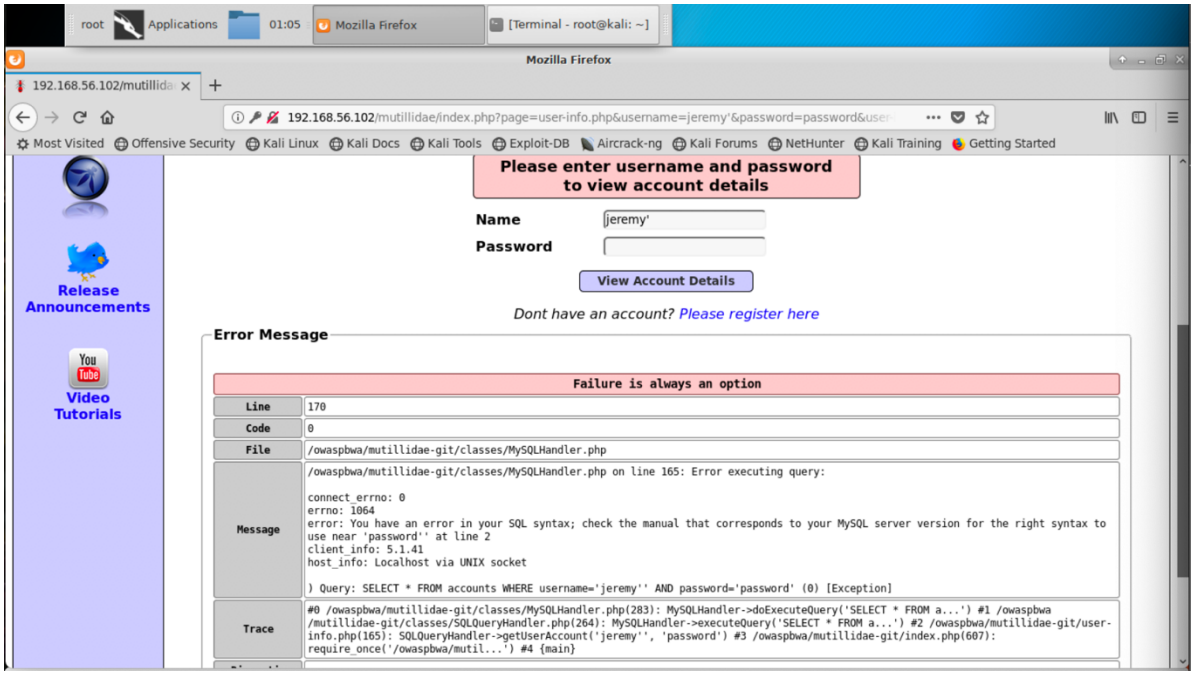

Figure 7: Verifying if the application is vulnerable to SQL injection

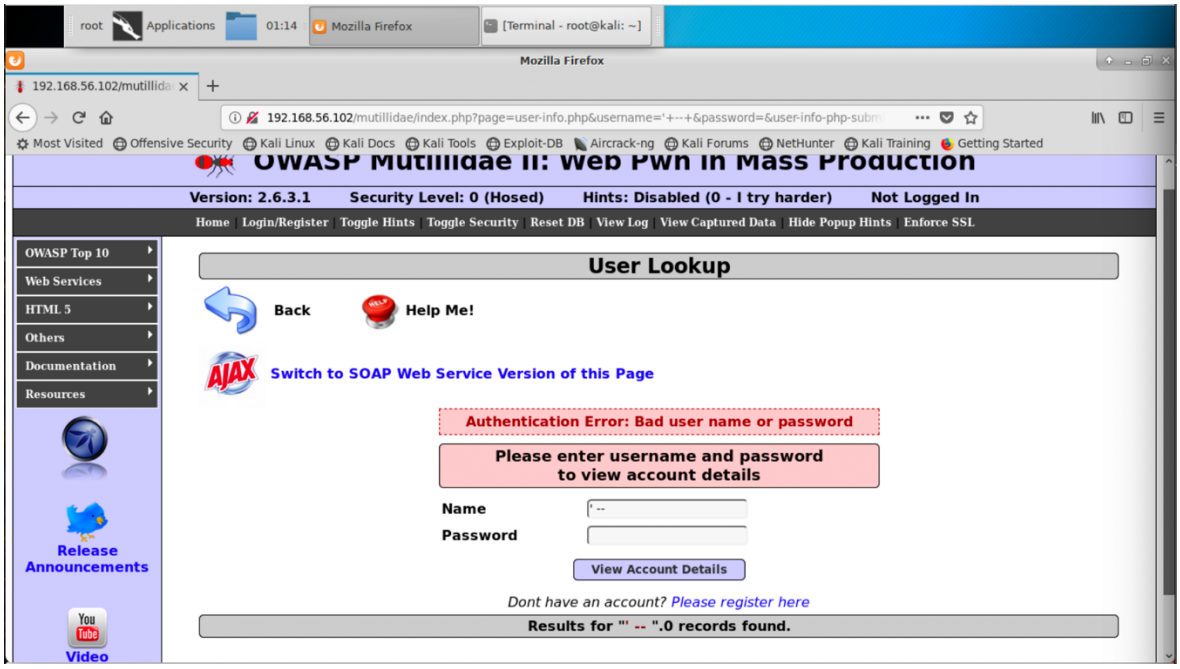

Figure 8: Testing unsanitized input

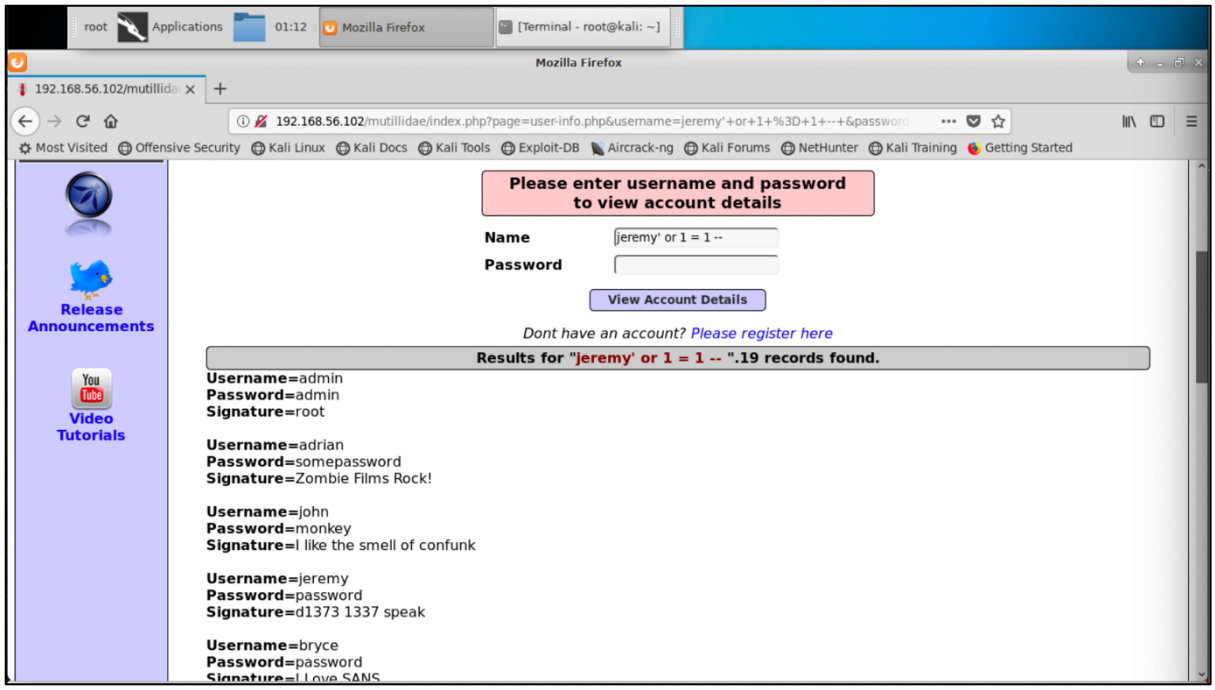

Figure 9: Successful SQL Injection attack

Question B - 3

The application is found to be vulnerable to cross-site scripting (XSS) attacks due to it not validating and encoding special characters (which can be used for HTML syntax as well) from user input before displaying the feedback (see Figure 10 and 11). Therefore, it executes any valid HTML syntax submitted as input, as evidenced in Figure 12 with the displaying of the document cookie. This vulnerability is dangerous due to the fact that the attacker can include unsecure links and views to the user interface to trick users to unknowingly input credentials, or share classified documents with the attacker (Chou, 2013). The attacker can also run background scripts (e.g. keyloggers) to corrupt government research and steal credentials.

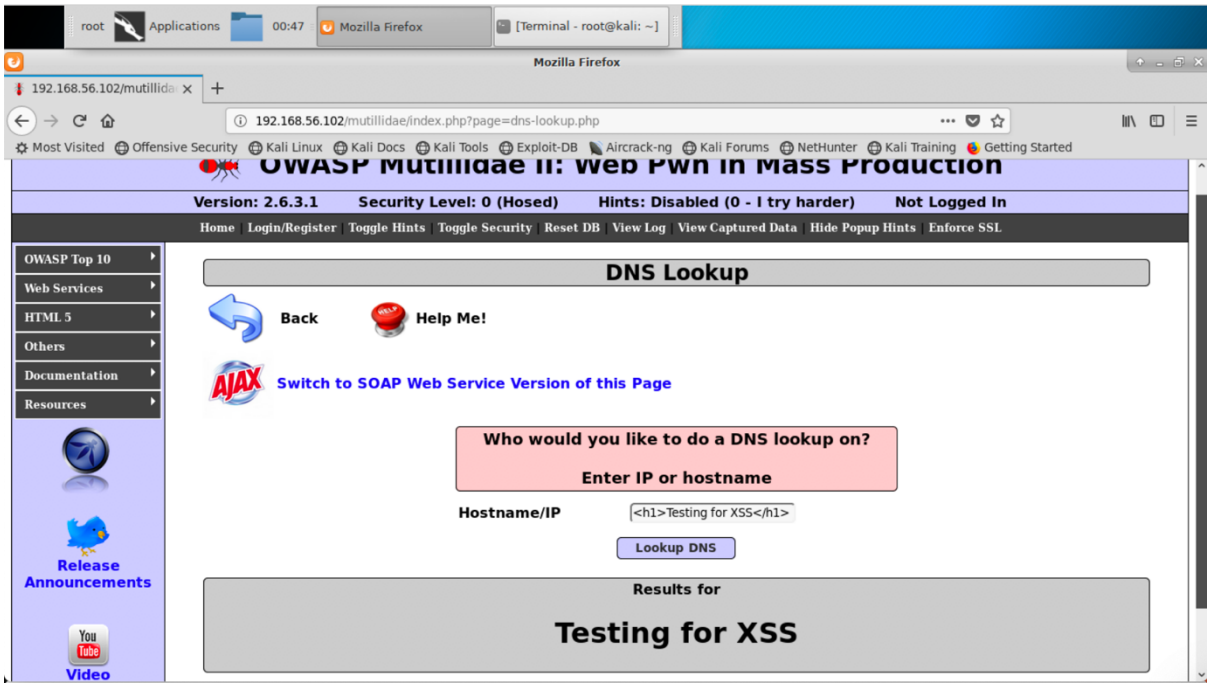

Figure 10: Testing for XSS vulnerability

Figure 11: Unencoded input in page source

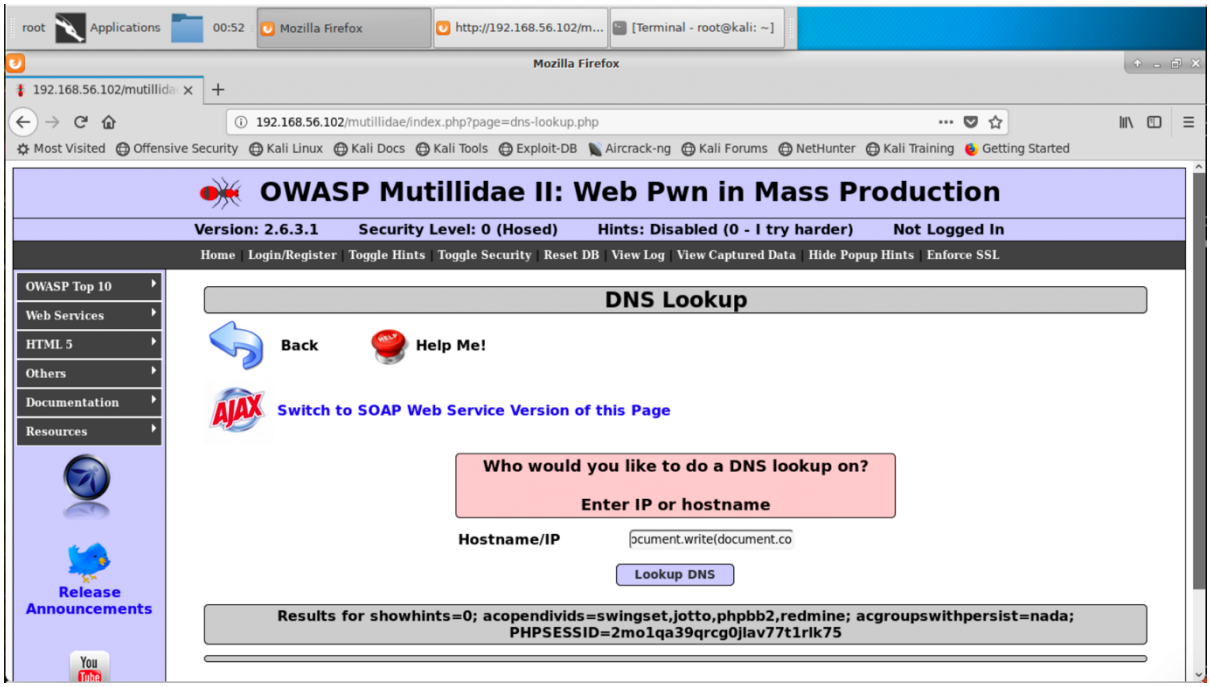

Figure 12: Result of a sample XSS attack

Question B - 4

It has also been identified that the application is vulnerable to Cross-Site Request Forgery (CSRF) attacks. Here attackers can socially engineer users to execute malicious activities without the user's knowledge by manipulating scripts for state changing actions (NetSparker Team, 2020). For example, here the attacker creates a malicious password change script by copying the original form from the website (see Figure 13 and 14) and possibly emails the user to warn him/her to change their password to avoid being hacked. The user trusts the email and opens the link which leads the user to a intermediary page which prompts the user to change the password (see Figure 16). But according to the CSRF markdown the attacker has already defined a password for the user account in hidden fields (see Figure 15) and when the user clicks the button, the user is alerted that the password has already been changed (see Figure 17).

This is dangerous to the collaboration tool as users will not be able to access the tool to share or access time-sensitive research. If the admin account has been attacked, time-sensitive research will not be able to be shared on time. Additionally, these type of Denial of Service attacks gives attackers more time to browse the classified documents and corrupt/steal selectively.

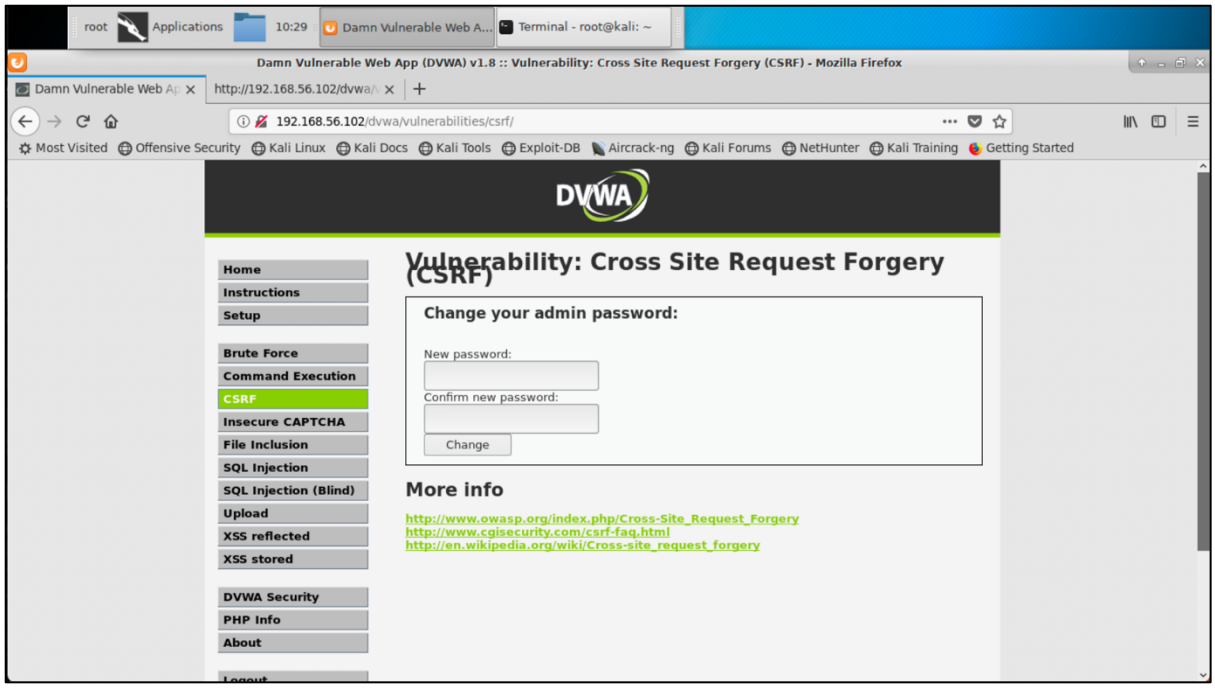

Figure 13: Regular password change screen

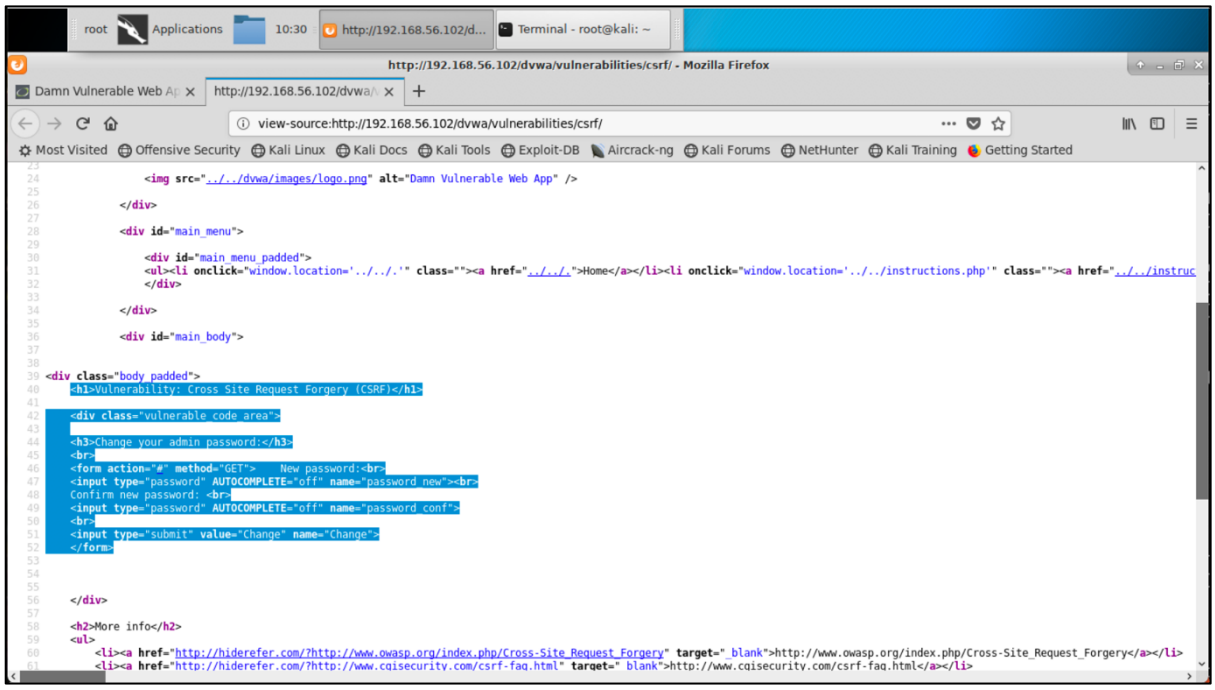

Figure 14: Attacker gets password change markdown from page source

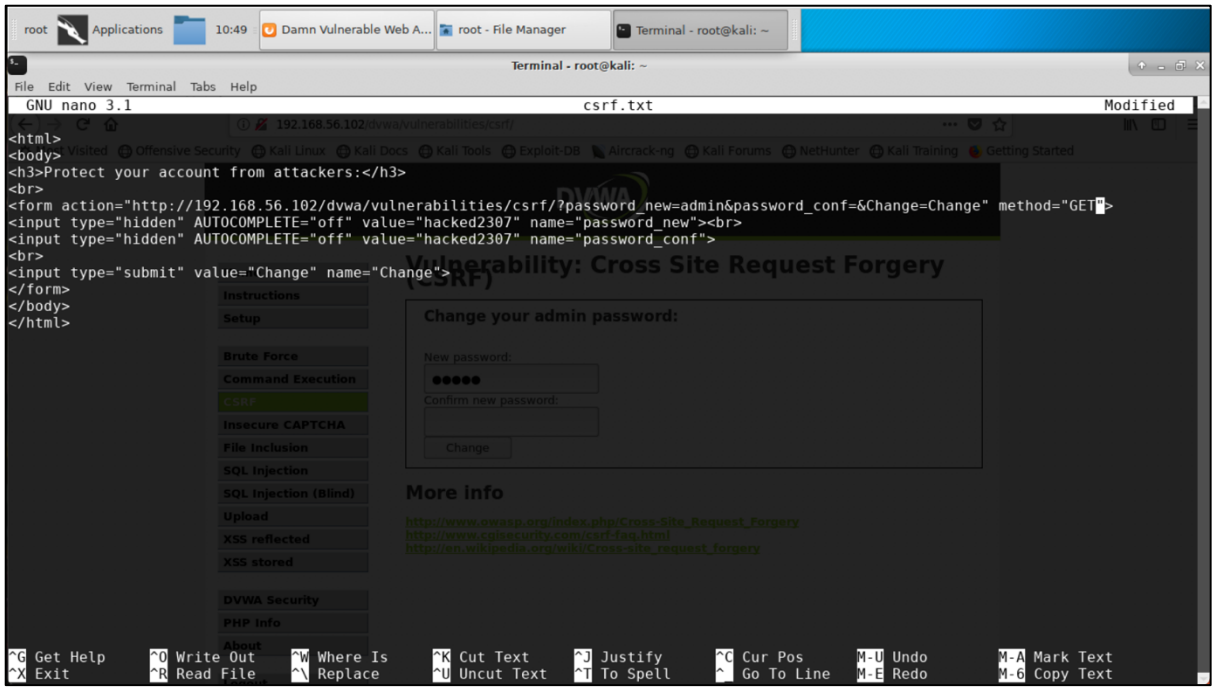

Figure 15: Attacker alters password change form with hidden fields

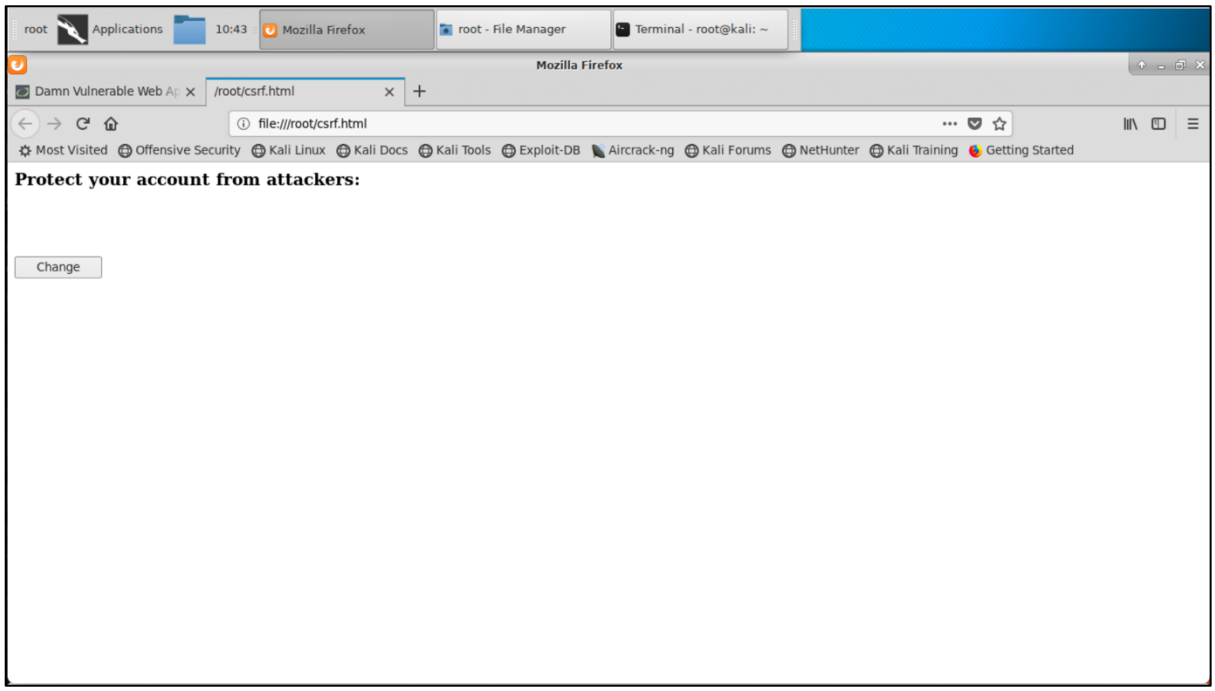

Figure 16: User is led to malicious page

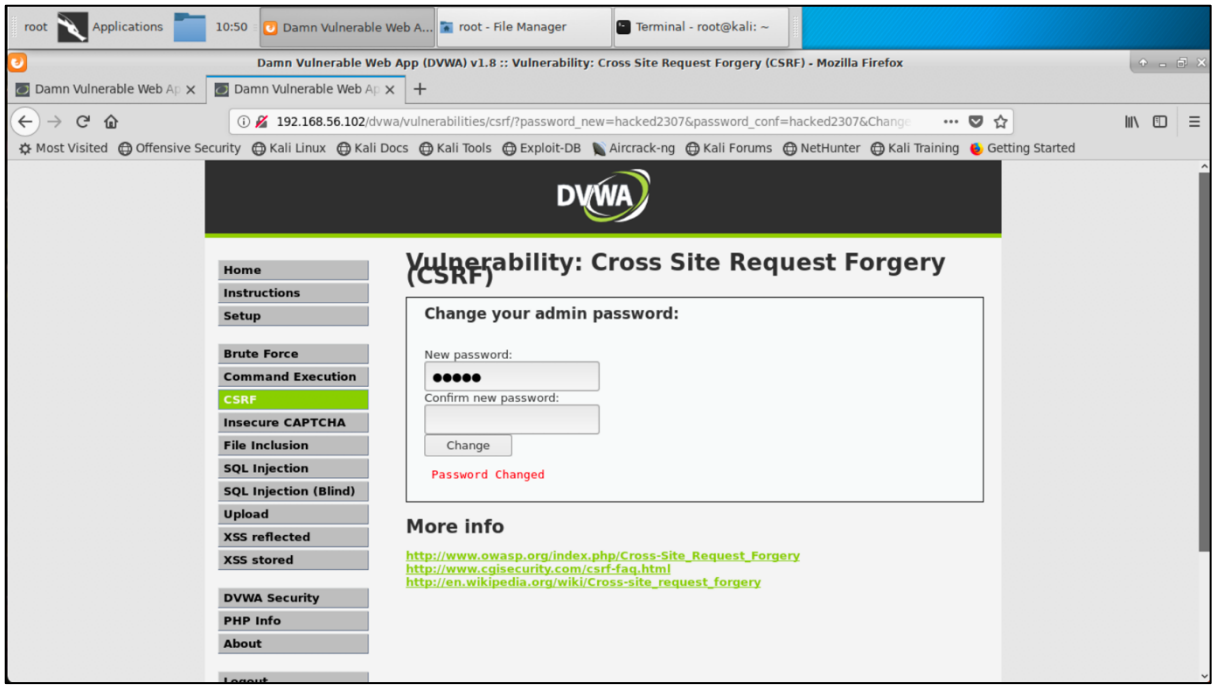

Figure 17: Password successfully changed by attacker

C Man-in-the-Middle Attacks and Social Engineering

Question C - 1

Through ARP spoofing it has been identified that the collaboration tool is vulnerable to MitM attacks. Ettercap was used to perform ARP poisoning on the collaboration tool's server and a connected client. When the client logs into the server, the ARP spoofing allowed the tester to view the login credentials (see Figure 18). Using such an attack, attackers can easily steal credentials and access the collaboration tool to steal research or documents.

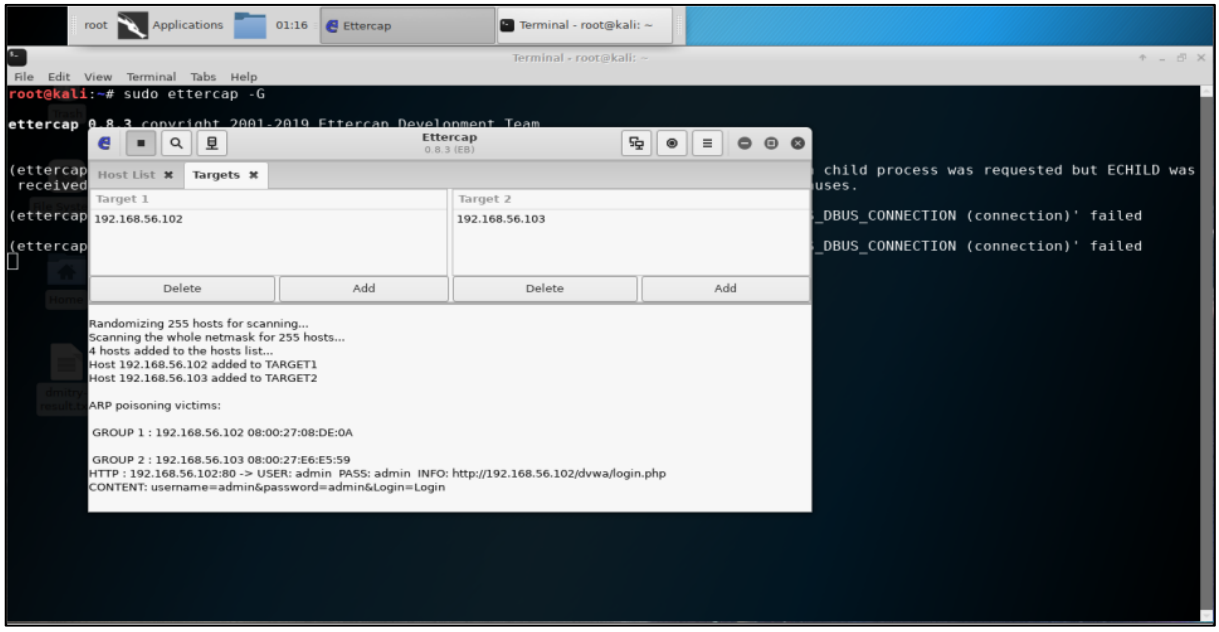

Figure 18: Getting client credentials by ARP poisoning

Further, using Wireshark we were able to monitor all network traffic in real time between the client and the server and use filters to locate specific requests such as form inputs (see Figure 19 and 20). This very dangerous because attackers can access and modify or corrupt research prototypes and/or documents that are being freshly uploaded to the collaboration tool even before the government organization can view them. Therefore, it can be very hard to find if the resource was accessed illegally and manipulated due to there being no original copy on the collaboration tool (Yadav *et al.*, 2017).

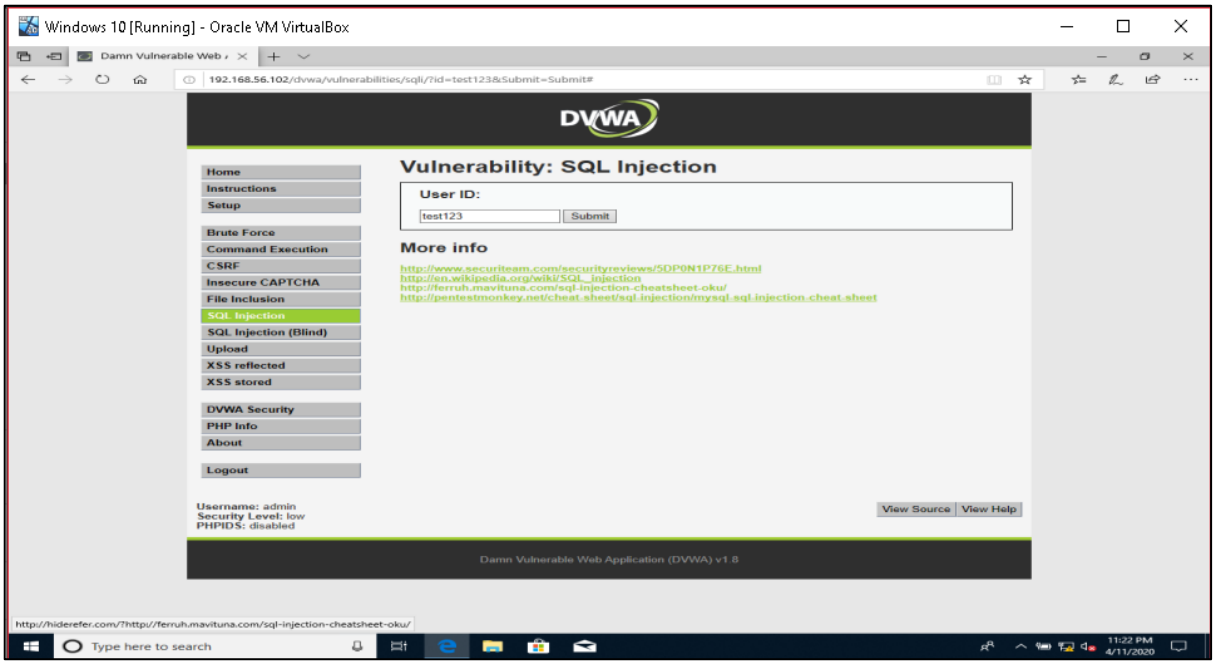

Figure 19: Client submitting data to the form field

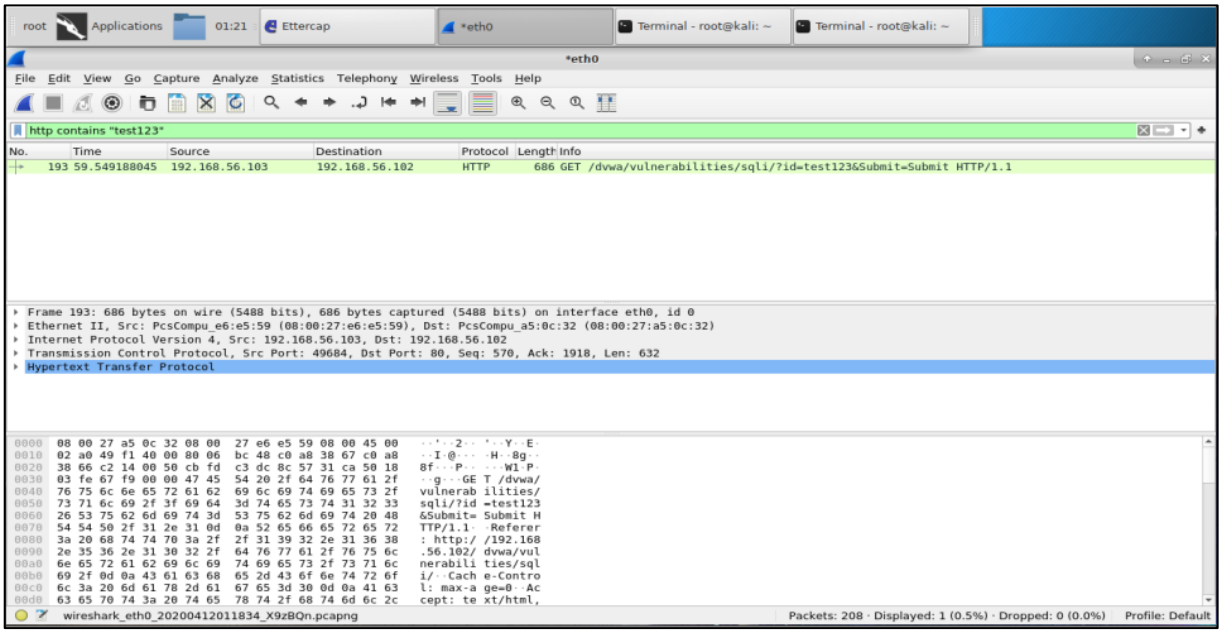

Figure 20: Monitoring form input from client to server using Wireshark

In addition to receiving information , the attacker can also modify client inputs by intercepting the request by writing an Ettercap regex filter to capture and modify the username credential during the interception of the login request (see Figures 21 and 22). There is a huge risk that attackers can write such filters to modify contact information or physical addresses that users share with the government organizations, to steal identities and receive illegal delivery of classified hardware directly to the attacker's address.

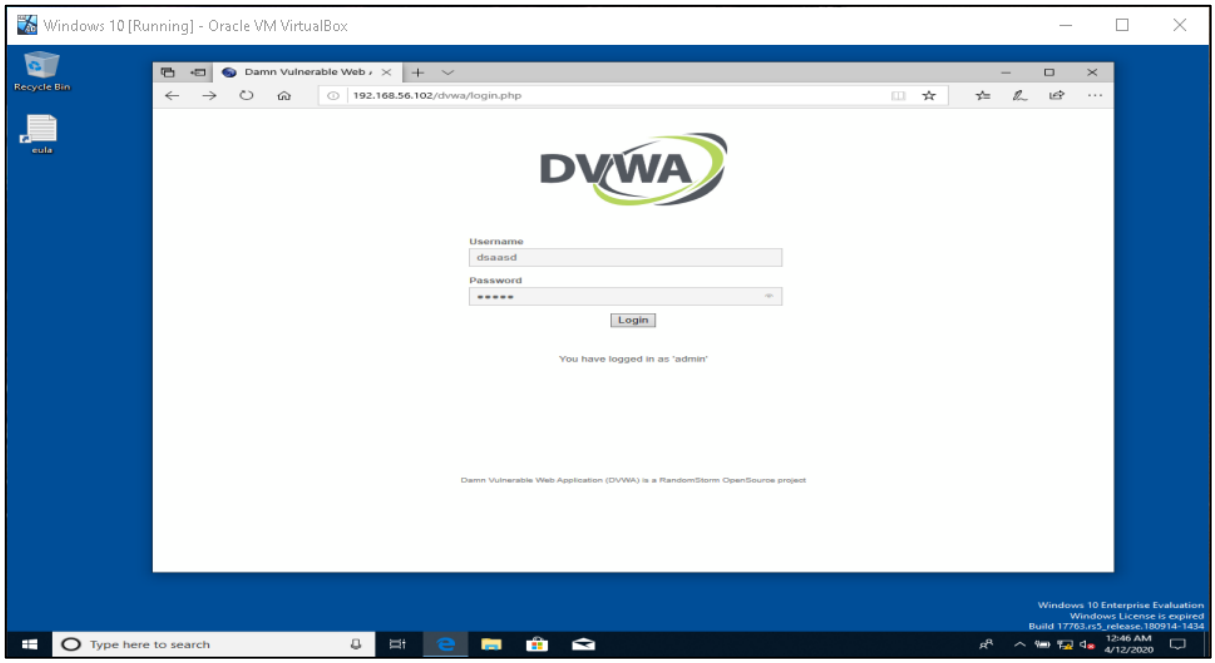

Figure 21: Client entering invalid username

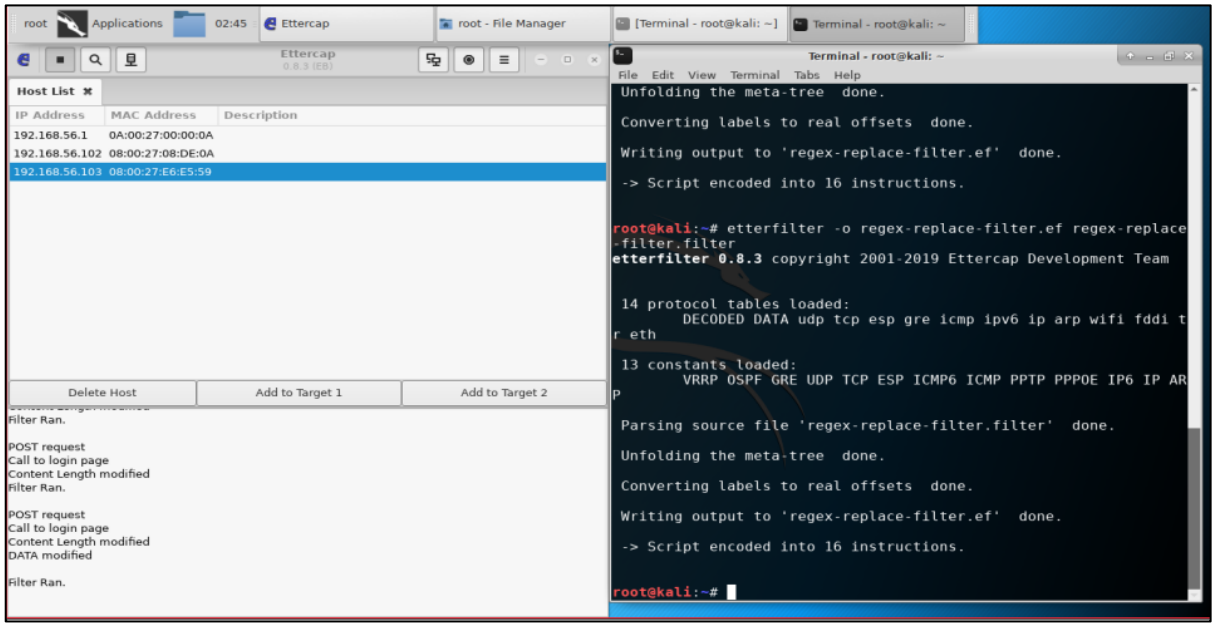

Figure 22: Using Ettercap regex filter to modify username with correct credential

Question C - 2

Attackers can clone the entire collaboration tool using the *wget* command and create a phishing site on the attacker's server. From here the attacker can edit the login form (see Figure 23) such that authentication is redirected to a custom script on the attackers machine that harvests credentials and redirects the user back to the authentic website (see Figure 24) without the user's knowledge.

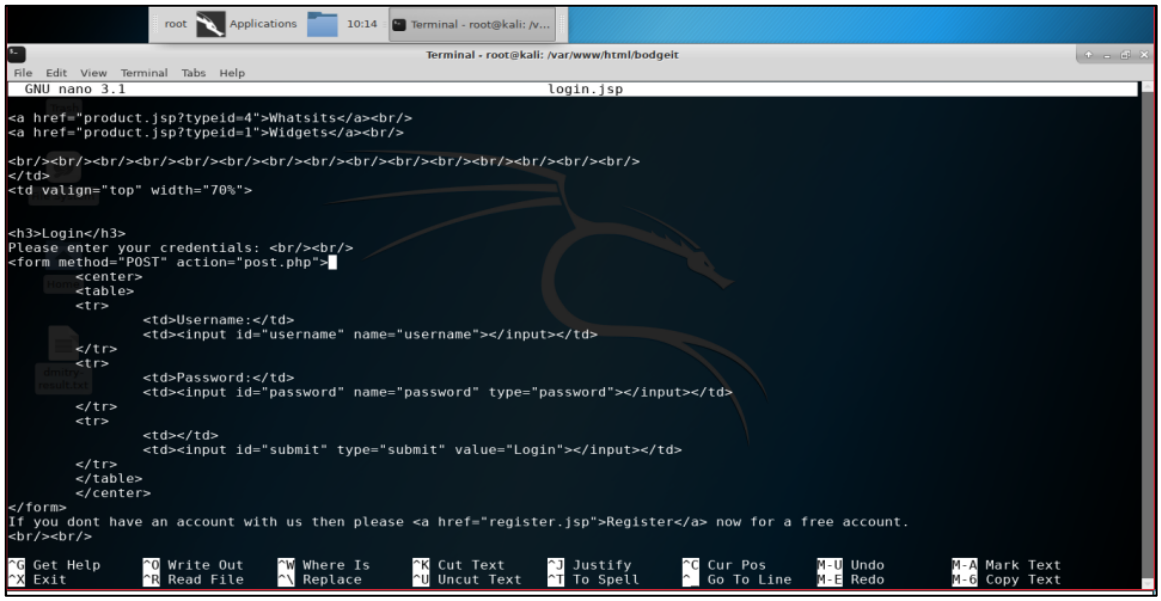

Figure 23: Edited markdown on login page

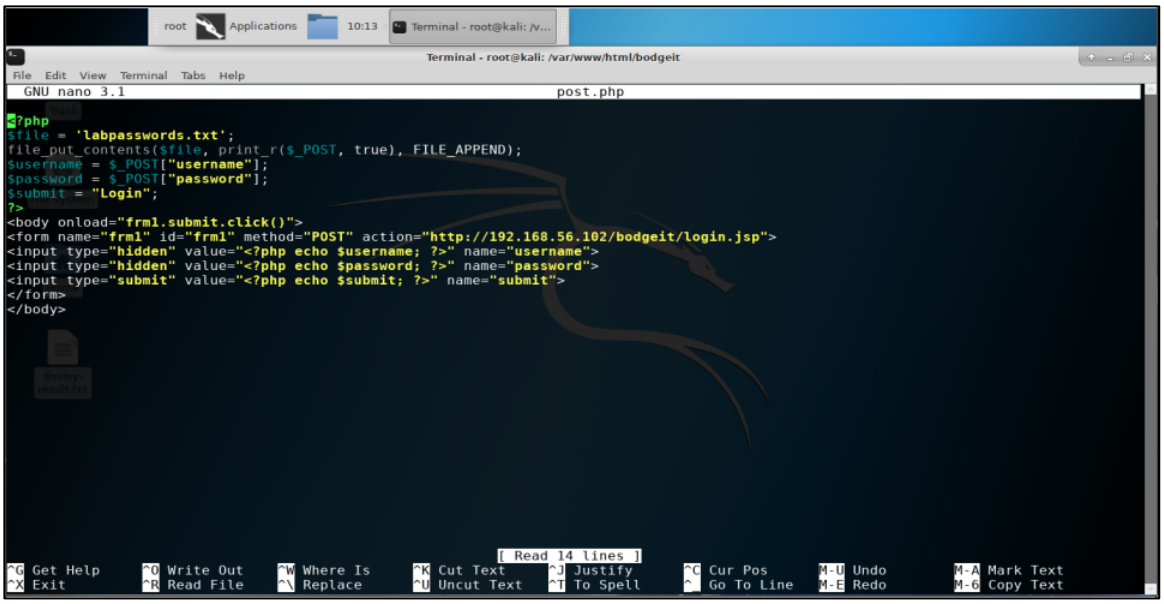

Figure 24: Attacker's custom PHP script to get and store the user's credentials

Using social engineering, the attacker can make the user believe that the domain of the cloned website is the legitimate URL of the collaboration tool (Krombholz *et al.*, 2013), and convince the user to create an account (see Figure 25) and login to the application (see Figure 26), whereas during login the attacker can actually steal the login credentials and store them to a file (see Figure 27) for later use. The user will not suspect anything as he/she has been redirected to the original website after login. This is quite a dangerous vulnerability as attackers depend on the ignorance of employees to steal credentials. Therefore the employees have to be made aware to check the URL of the application when logging onto the collaboration tool. Not

only can these attackers steal sensitive research, but they can get innocent employees terminated for stealing data by using their credentials.

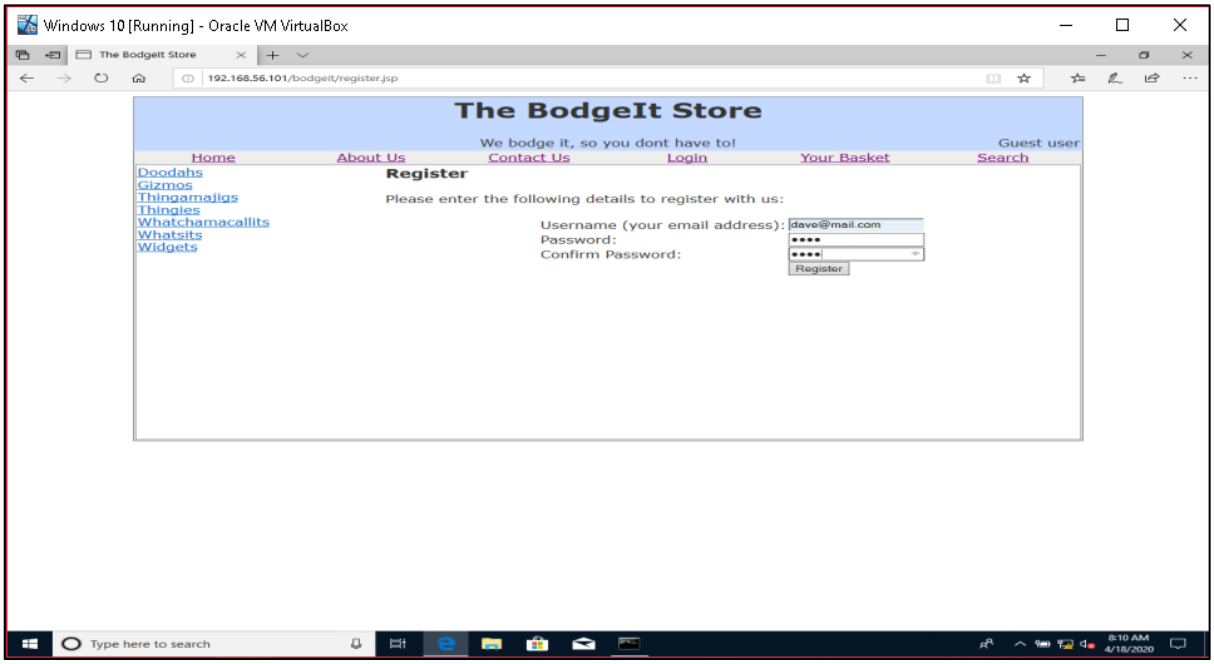

Figure 25: User creating a new account on the cloned site

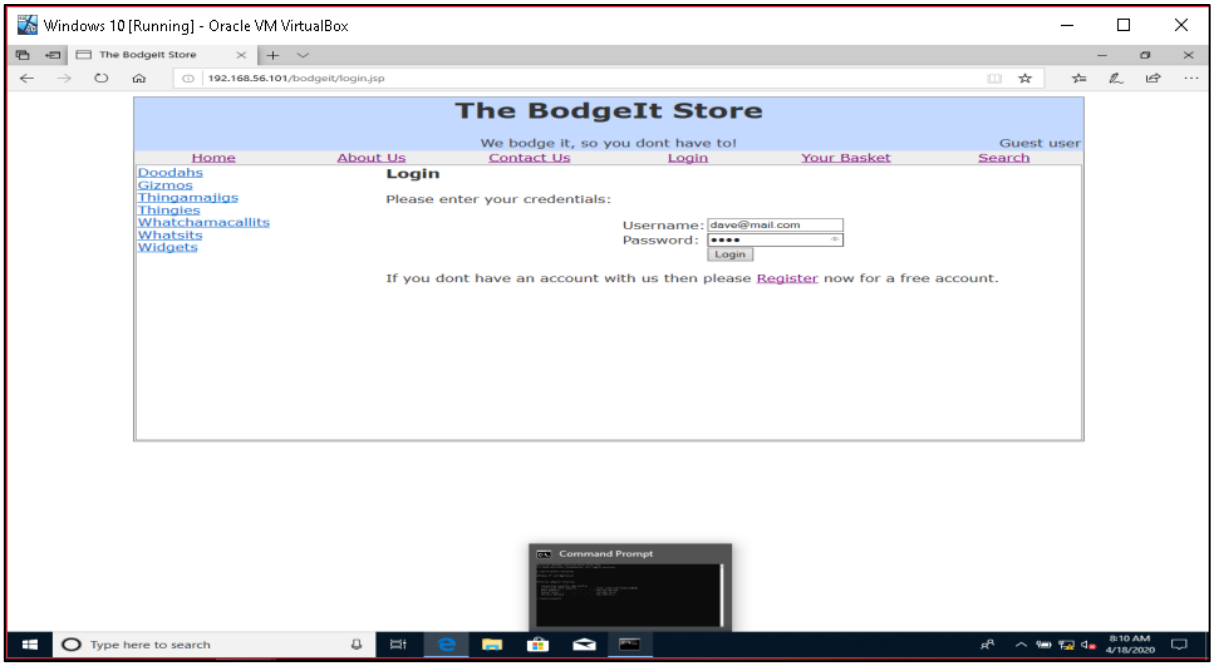

Figure 26: User logging into the cloned site

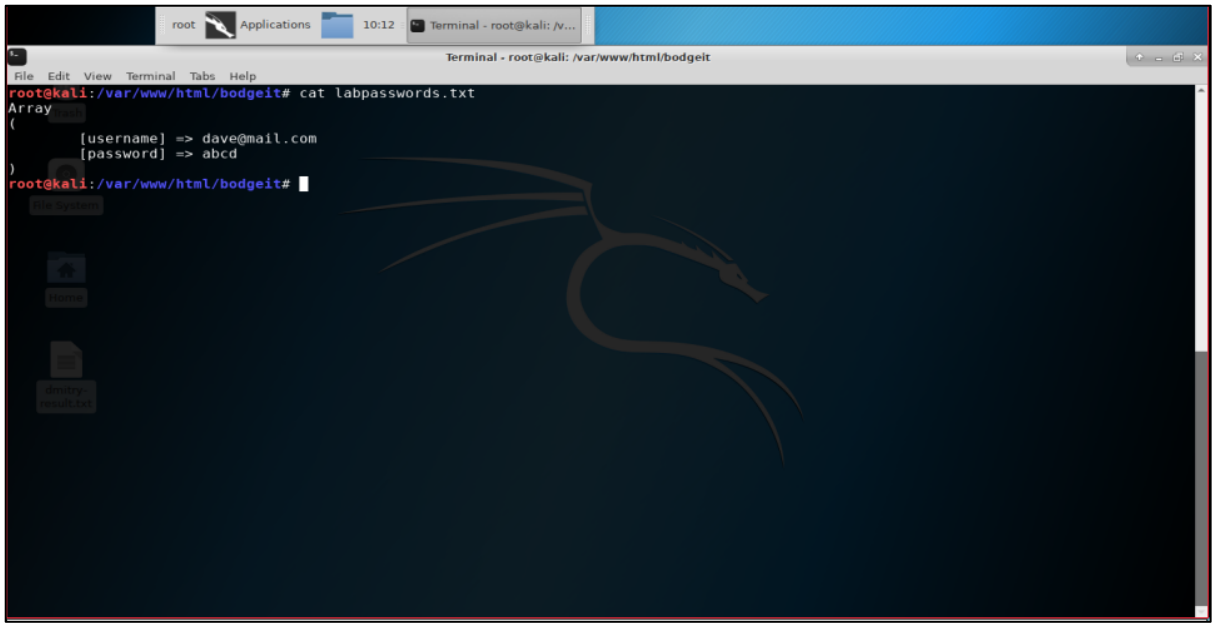

Figure 27: Harvested credentials on the attacker's machine

Question C - 3

It has been found that attackers can utilize reverse shells to gain access to client systems. First the attacker will use Metasploit to generate a reverse meterpreter shell (see Figure 28). Reverse shells, when executed by clients can provide access of the clients machine to the attacker. Next the attacker will create a listener on their machine to listen for any TCP sessions on the targeted client machine (see Figure 29). Using socio-technical engineering such as *baiting* (Krombholz *et al.*, 2013), the attacker can lead the user to believe that the reverse shell file on the attacker's server is a harmless software such as a browser extension, funny image etc. and download it (see Figure 30). Once the client downloads the file and runs it (see Figure 31), the attacker will gain direct access to the client's machine. The attacker can then even run shell commands directly on the client's machine (see Figure 32 and 33).

In terms of the collaboration tool, this allows attackers to gain direct access to the client machines of employees. This can be disastrous because attackers can steal fresh research documents and codebases and edit/corrupt the existing resources before employees share them on the collaboration tool. This will cause the government organizations to point blame at the R&D firm and will result in business losses and even termination of contracts.

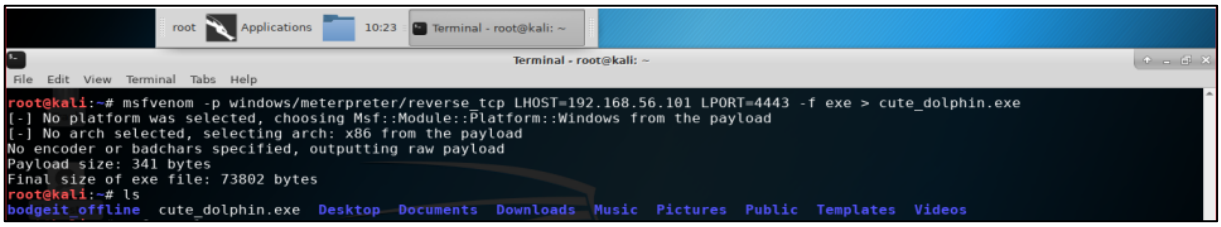

Figure 28: Creating a reverse meterpreter shell

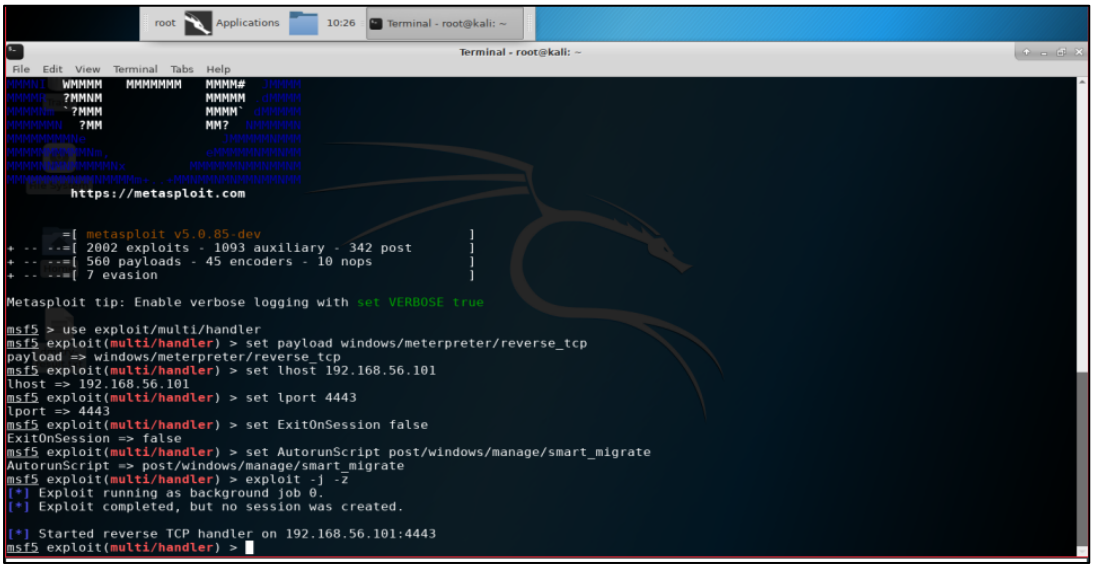

Figure 29: Creating a new listener with metasploit

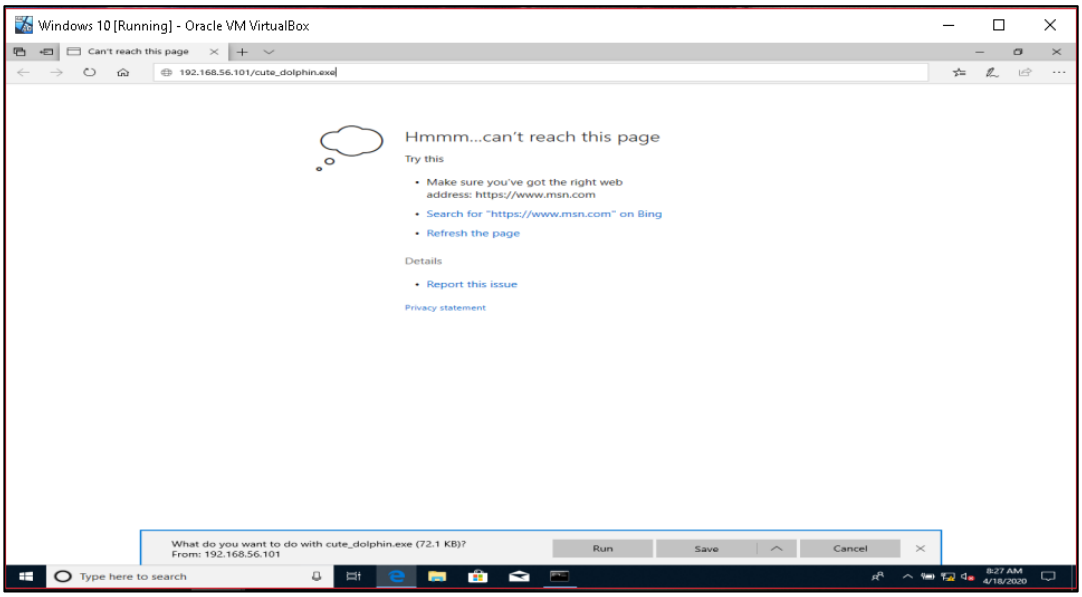

Figure 30: Convincing the user to download the shell file

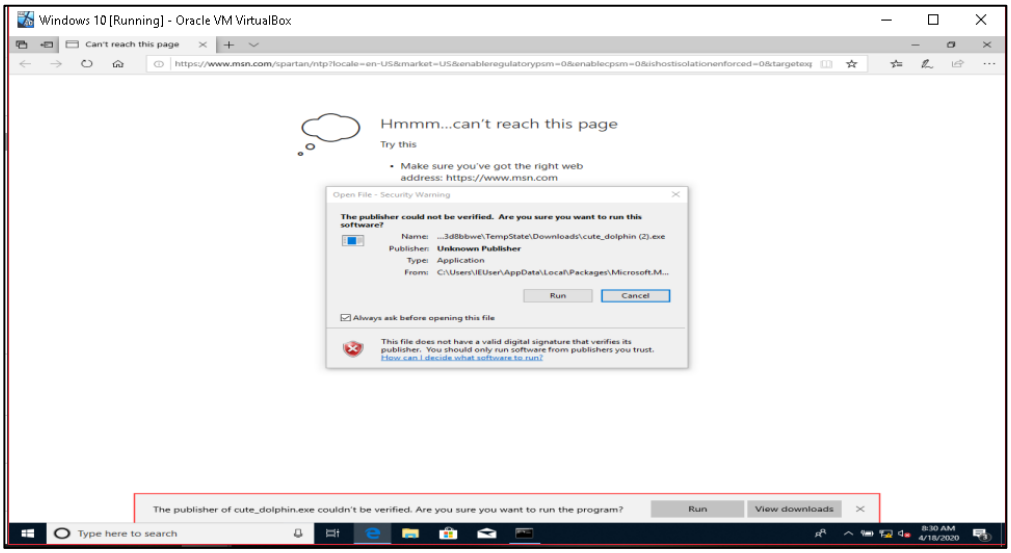

Figure 31: User running the reverse shell on the client

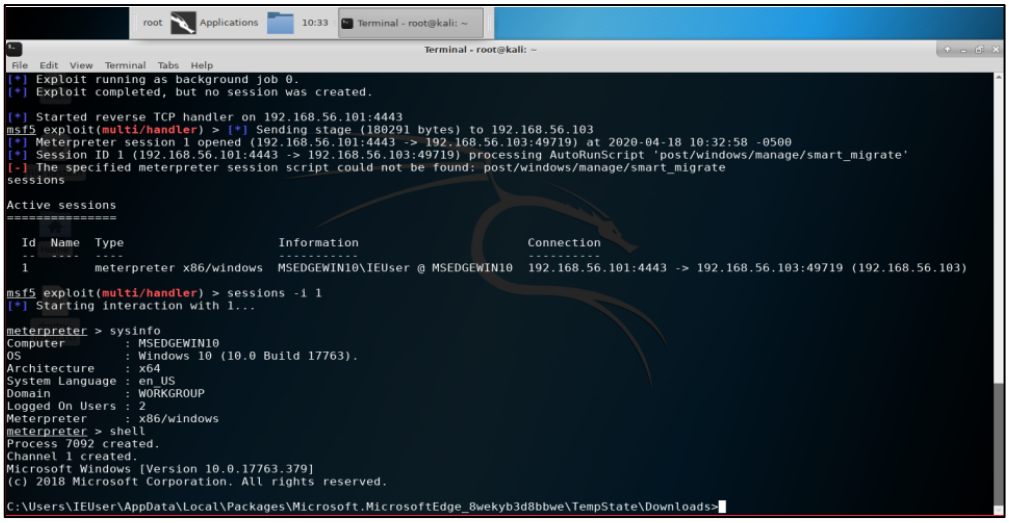

Figure 32: Attacker gaining access to the client's shell

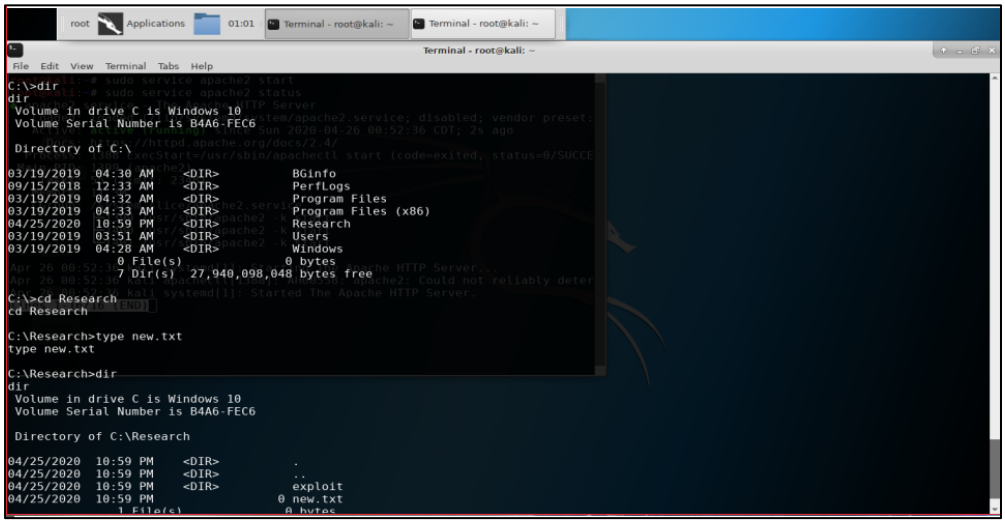

Figure 33: Attacker accessing classified research directories

D Protecting your Server

Question D - 1

Port knocking is an authentication method that works through closed ports, where the server closes a series of vulnerable ports and creates a knock sequence such that only the users that attempt to connect to that sequence of ports in order is allowed to access the required port (Krzywinski, 2003) (see Figure 34).

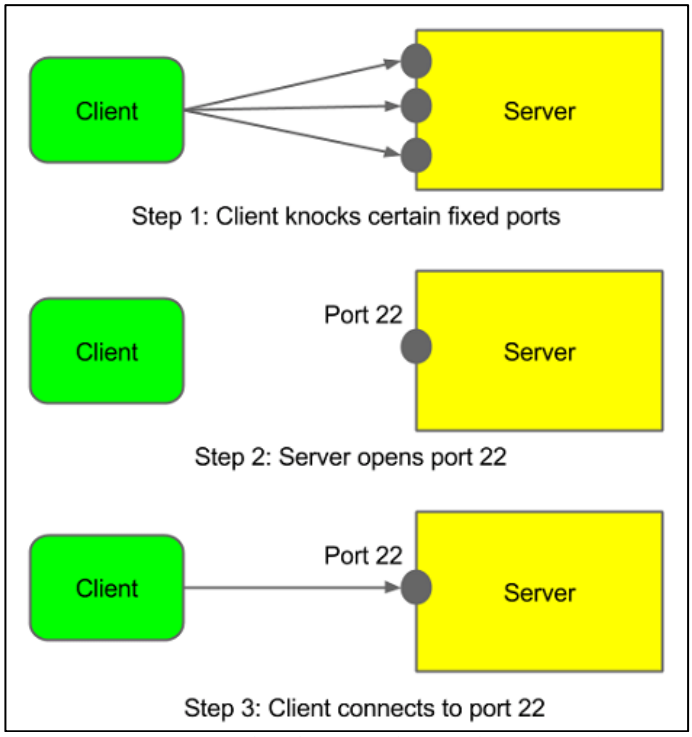

Figure 34: Port knocking visualization (Arora, 2013)

Not only is port knocking a powerful system, it can also be easily integrated to existing applications with minimal changes (Arora, 2013). Therefore, it is very suitable for our collaboration tool as it is a production application and frequent downtime isn't feasible. If a brute-force attack is launched to figure out the sequence and access the SSH daemon for example, the server administrator can flag the source as malicious and block all incoming traffic from that source.

Question D - 2

A false positive occurs when legitimate traffic triggers an NIDS alert. False negatives on the other hand occur when malicious traffic *does not* trigger an alert. For an NIDS system to be

accurate, both these types of alerts have to be minimized. For this, threat databases and anomaly baselines have to be regularly updated (Attacks, 2003). If the NIDS system implemented on the collaboration tool's server frequently generates alerts for false positives and false negatives, the application has to be constantly redeployed following inspections, which halts productivity and destroys the shelf life of time-sensitive research collaboration. Further, the R&D company will be adversely affected in terms of funding for time-sensitive deliveries of research updates.

Question D - 3

IDS are detection and monitoring tools so they do not take action against potential threats. A human element is needed to analyze the results and do the needful, whereas IPS is a control system that can catch and drop malicious network packets based on a defined ruleset (Petters, 2020) (see Figure 35).

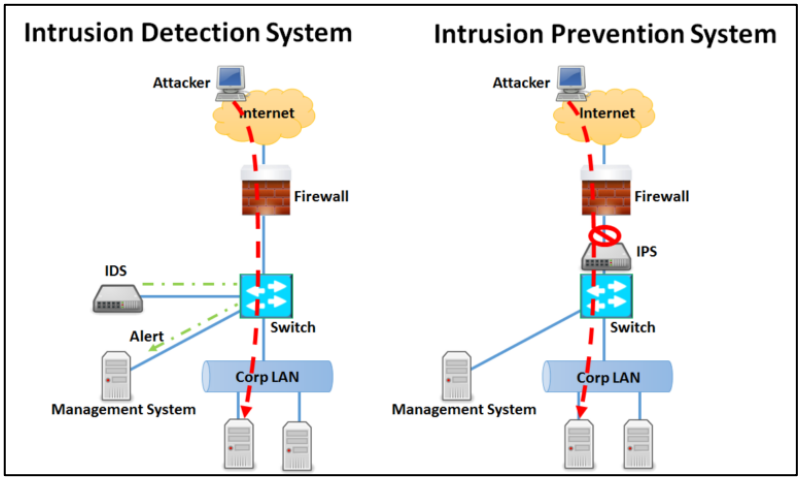

Figure 35: IDS vs IPS visualization (IPWITHEASE, 2017)

For the scenario of the collaboration tool, it is highly suggested to implement an IPS system due to the fact that is passive and does not require a dedicated security engineer to take action against threats, while also being the more popular choice currently. This can benefit the R&D company financially as well as increase the collaboration tool's uptime such that time-sensitive research collaborations do not get delayed due to potential malicious activity. But IPS systems can also cause the loss of data if any false positives trigger the system, therefore, the threat database has to be constantly updated to prevent this (*IDS vs. IPS: What's the Difference?*, 2019).

Question D - 4

Firewall

A firewall is a collection of rules that monitors and controls network packets. UncomplicatedFirewall (ufw) is an example for a firewall and is actually an abstracted interface for iptables in Linux, with much simpler syntax and user-friendly controls. But, ufw is not optimized for every Linux kernel and user only has very high-level control over network traffic (Hainic, 2017).

Snort

Snort is a lightweight network intrusion detection system (NIDS). It is free and syntax is extremely user friendly. Nevertheless, Snort also lacks most innovative features existing in commercial NIDS and the administrator has to develop logging methods for the system manually (McIntyre, 2001). Packet processing is also slower on Snort as compared to other NIDS (UpGuard, 2020).

iptables

iptables is a firewall system that is built into the Linux kernel. It allows much deeper and low-level control into monitoring and filtering network packets. It is generally considered to be complicated and requires knowledge of advanced protocols to utilize the full functionality effectively. Regardless, it is more robust and reliable than an abstracted firewall interface, while also being unforgiving in terms of performance if not configured properly (Rutledge, 2019).

For the collaboration tool, the best choice would be to utilize a cascaded security solution with a combination of iptables and Snort. The main reason for this is that the iptables firewall sits at the perimeter of the network layer, while NIDS such as Snort sit inside the network layer and evaluates intrusions after they happen (Yadav, 2018). iptables offer more low-level customizability and robustness than an abstracted interface. Since this collaboration tool deals with classified and sensitive research for the government, low-level security configurations are needed to ensure protection against malicious attackers from different countries.

Question D - 5

- It is evident by the social engineering vulnerabilities identified, that employees have to be made aware of security risks more prominently. According to (Chickowski, 2018), insider risks are a large vulnerability when it comes to online collaboration tools. Research shows 76% of communications between employees on collaboration tools tend to contain passwords. Such employees may inadvertently be preyed on by malicious attackers to reveal passwords or information on sensitive research being done on our application. To mitigate this the company as well as the government organizations need to enforce active monitoring systems on the application as well as end to end encryption on messages.

- According to Cisco's Collaboration Security report (*Collaboration Security*, 2013), malware such as viruses, worms, phishing schemes can be attached to shared documents (i.e. research documents such as developer manuals and design documents). Therefore, shared resources have to always be sanitized and filtered before being downloaded to local machines. Source files should be shared as compressed encrypted files or access-controlled repository links, to mitigate the risk of code tampering and corruption of research.

References

Arora, H. (2013) *How Port Knocking Can Add Extra Layer of Server Security*, *The Geek Stuff*. Available at: https://www.thegeekstuff.com/2013/10/port-knocking/ (Accessed: 16 April 2020).

Attacks, N. (2003) 'Introduction to Intrusion Detection Systems', *Cisco Security Professional's Guide to Secure Intrusion Detection Systems*, pp. 1–38. doi: 10.1016/b978-193226669-6/50021-5.

Chandel, R. (2017) *NetBIOS and SMB Penetration Testing on Windows*, *Hacking Articles*. Available at: https://www.hackingarticles.in/netbios-and-smb-penetration-testing-on-windows/ (Accessed: 11 April 2020).

Chickowski, E. (2018) *Insider Dangers Are Hiding in Collaboration Tools*, *Dark Reading*.

Chou, T.-S. (2013) 'Security Threats on Cloud Computing Vulnerabilities', *International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 3, June 2013*, 5(3), pp. 79–88. doi: 10.5121/ijcsit.2013.5306.

Collaboration Security (2013) *Cisco*. Available at: https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cucm/srnd/collab09/clb09/security.html (Accessed: 17 April 2020).

Cotten, T. (2016) *Why is Apache Vulnerable by Default?*, *Medium*. Available at: https://blog.cotten.io/why-is-apache-vulnerable-by-default-743eec222013 (Accessed: 19 April 2020).

Geer, D. (2017) *Securing risky network ports*. Available at: https://www.csoonline.com/article/3191531/securing-risky-network-ports.html (Accessed: 9 April 2020).

Hainic, C. (2017) *USING IPTABLES INSTEAD OF UFW IN A BASIC SERVER SETUP &*

FOR DOCKER, *Spyhce* Available at: https://spyhce.com/blog/using-iptables-instead-ufw-basic-server-setup (Accessed: 17 April 2020).

IDS vs. IPS: What's the Difference? (2019) *DNS Stuff*. Available at: https://www.varonis.com/blog/ids-vs-ips/ (Accessed: 16 April 2020)

Krombholz, K. *et al.* (2013) 'Social engineering attacks on the knowledge worker', *SIN 2013 - Proceedings of the 6th International Conference on Security of Information and Networks*, pp. 28–35. doi: 10.1145/2523514.2523596.

Krzywinski, M. (2003) *Port Knocking*, *Linux Journal*. Available at: https://www.linuxjournal.com/article/6811 (Accessed: 16 April 2020).

McIntyre, J. (2001) *Using Snort for intrusion detection*, *TechRepublic*. Available at: https://www.techrepublic.com/article/using-snort-for-intrusion-detection/ (Accessed: 17 April 2020).

Nidecki, T. A. (2019) *What is a Buffer Overflow*, *Acunetix*. Available at: https://www.acunetix.com/blog/web-security-zone/what-is-buffer-overflow/ (Accessed: 11 April 2020).

Olzak, T. (2007) *The problem with NetBIOS*, *TechRepublic*. Available at: https://www.thegeekstuff.com/2013/10/port-knocking/ (Accessed: 16 April 2020).

Petters, J. (2020) *IDS vs. IPS: What is the Difference?*, *Varonis*. Available at: https://www.varonis.com/blog/ids-vs-ips/ (Accessed: 16 April 2020).

Port 143 Details (2009) *Speed Guide*. Available at: https://www.speedguide.net/port.php?port=143 (Accessed: 19 April 2020).

Rutledge, M. (2019) *What are the disadvantages of iptables?* Available at: https://www.thegeekstuff.com/2013/10/port-knocking/ (Accessed: 16 April 2020)

Samba 'username map script' Command Execution (2018). Available at: https://www.rapid7.com/db/modules/exploit/multi/samba/usermap_script (Accessed: 12 April 2020).

Team, S. (2019) *Top 20 and 200 most scanned ports in the cybersecurity industry*. Available at: https://securitytrails.com/blog/top-scanned-ports (Accessed: 12 April 2020).

Turner, P. (2019) *Best Practices for Securing SSH: What Are Your SSH Security Risks?*, *Venafi*. Available at: https://www.venafi.com/blog/best-practices-ssh-key-management-what-are-your-ssh-security-risks (Accessed: 9 April 2020).

UpGuard (2020) *Top 6 Free Network Intrusion Detection Systems (NIDS) Software in 2020*, *UpGuard*. Available at: https://www.upguard.com/articles/top-free-network-based-intrusion-detection-systems-ids-for-the-enterprise (Accessed: 16 April 2020)

Vulnerability Details : CVE-2016-0777 (2016). Available at: https://www.cvedetails.com/cve/CVE-2016-0777/ (Accessed: 12 April 2020).

Vulnerability Details : CVE-2016-10708 (2018). Available at: https://www.cvedetails.com/cve/CVE-2016-10708/ (Accessed: 12 April 2020)

Vulnerability Details : CVE-2016-2112 (2016). Available at: https://www.cvedetails.com/cve/CVE-2016-2112/ (Accessed: 12 April 2020).

Vulnerability Details : CVE-2017-15906 (2017). Available at: https://www.cvedetails.com/cve/CVE-2017-15906/ (Accessed: 12 April 2020).

Vulnerability Details : CVE-2017-7494 (2017). Available at: https://www.cvedetails.com/cve/CVE-2017-7494/ (Accessed: 12 April 2020).

Yadav, A. (2018) *Network Design: Firewall, IDS/IPS*, *Infosec Institute*. Available at: https://resources.infosecinstitute.com/network-design-firewall-idsips/#gref (Accessed: 17 April 2020).

Yadav *et al.* (2017) 'Prevention of MITM Attacks in Cloud Computing by Lock Box Approach Using Digital Signature', *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(5), pp. 567–572. doi: 10.23956/ijarcsse/sv7i5/0276.