

السلام عليكم..... نحن مجموعة من طلاب الامن السيبراني مستوى ثالث جامعة السعيدة قمنا بعمل ادائه اسمها Gemini نقوم بأكثر من عمل سوف نتكلم عنها فيما بعد.

- المهندسين الذين قام بعمل هذي الأداء:

١- م/محمد سعيد يسر

٢- م/مساعِد عبد الحكيم المخلافي

٣- م/الياس منصور مأمون

٤- م/ايمن غالب الفضل

تحت اشراف الدكتور الفاضل / عادل معوضه

اداءة Gemini هي بلغة python تقوم بالعمل في نظام kali

صور توضيحية ل كود الأداء

The screenshot displays a Kali Linux terminal window with a dark background. A Python script named 'GEMINI.py' is being executed. The script's output is visible in the terminal, showing various stages of a penetration test. The script starts by importing necessary modules like 'socket', 'os', 'sys', 'time', 'urllib', 'requests', 'ssl', 'urllib3', 'httpx', 'nmap', 'nmap2', 'nmap3', 'nmap4', 'nmap5', 'nmap6', 'nmap7', 'nmap8', 'nmap9', 'nmap10', 'nmap11', 'nmap12', 'nmap13', 'nmap14', 'nmap15', 'nmap16', 'nmap17', 'nmap18', 'nmap19', 'nmap20', 'nmap21', 'nmap22', 'nmap23', 'nmap24', 'nmap25', 'nmap26', 'nmap27', 'nmap28', 'nmap29', 'nmap30', 'nmap31', 'nmap32', 'nmap33', 'nmap34', 'nmap35', 'nmap36', 'nmap37', 'nmap38', 'nmap39', 'nmap40', 'nmap41', 'nmap42', 'nmap43', 'nmap44', 'nmap45', 'nmap46', 'nmap47', 'nmap48', 'nmap49', 'nmap50', 'nmap51', 'nmap52', 'nmap53', 'nmap54', 'nmap55', 'nmap56', 'nmap57', 'nmap58', 'nmap59', 'nmap60', 'nmap61', 'nmap62', 'nmap63', 'nmap64', 'nmap65', 'nmap66', 'nmap67', 'nmap68', 'nmap69', 'nmap70', 'nmap71', 'nmap72', 'nmap73', 'nmap74', 'nmap75', 'nmap76', 'nmap77', 'nmap78', 'nmap79', 'nmap80', 'nmap81', 'nmap82', 'nmap83', 'nmap84', 'nmap85', 'nmap86', 'nmap87', 'nmap88', 'nmap89', 'nmap90', 'nmap91', 'nmap92', 'nmap93', 'nmap94', 'nmap95', 'nmap96', 'nmap97', 'nmap98', 'nmap99', 'nmap100'. The script then proceeds to perform a series of checks on the target IP address, including host discovery, port scanning, and service identification. The output shows the script's progress, including the discovery of open ports and the identification of services running on those ports. The script also performs a series of checks on the target IP address, including host discovery, port scanning, and service identification. The output shows the script's progress, including the discovery of open ports and the identification of services running on those ports. The script also performs a series of checks on the target IP address, including host discovery, port scanning, and service identification. The output shows the script's progress, including the discovery of open ports and the identification of services running on those ports.

```
GNU nano 2.9.6 my@kali: /usr/local/bin/gemin

with socket.create_connection(host, 443), (timeout=5) as sock:
    with context.wrap_socket(sock, server_hostname=host) as ssock:
        cipher = ssock.cipher()
        print(f"{Fore.GREEN}[A] البتة: {ssock.version()} | البتة: {cipher[2]} bits")
    except: print(f"{Fore.RED}[A] البتة: SNI.")

def check_malicious_link():
    link = input(f"{Fore.BLUE} أدخل الرابط للتحليل: ")
    suspicious = ["https", "http", "ftp", "mailto", "telnet"]
    is_bad = any(word in link.lower() for word in suspicious)
    try:
        res = requests.get(link, timeout=5, allow_redirects=True)
        if is_bad or len(res.history) > 1:
            print(f"{Fore.RED}[A] الرابط مشكوك فيه ")
        else: print(f"{Fore.GREEN}[A] الرابط يبدو آمن ")
    except: print(f"{Fore.RED}[A] الرابط مشكوك فيه ")

def network_scan():
    from scapy.all import ARP, Ether, sdp
    ip_range = input(f"{Fore.BLUE}[B] (192.168.1.1/24) الشبكة: ")
    print(f"{Fore.YELLOW} جارٍ فحص الشبكة {ip_range} ")
    try:
        result = sdp([Ether(dst="ff:ff:ff:ff:ff:ff")/sdp(ip_range), timeout=2, verbose=0)]
        for .. fcvd in result: print(f"{Fore.GREEN}[B] {fcvdst.ip} | MAC: {fcvdst.hsrc}")
    except: print(f"{Fore.RED}[A] يرجى التحقق من IP ")

def phone_scan():
    ip = input(f"{Fore.BLUE} أدخل IP الهاتف: ")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(2)
    if s.connect_ex((ip, 5555)) == 0:
        print(f"{Fore.RED}[A] الهاتف غير متاح ")
    else: print(f"{Fore.GREEN}[B] ADB: {ip} ")
    s.close()

def password_check():
    print(f"{Fore.YELLOW} (re باستخدام) اختبار قوة كلمة المرور ")
    pwd = getpass.getpass("أدخل كلمة المرور للاختبار: ")
    score = 0
    if len(pwd) > 8: score += 1
    if re.search(r"[a-z]", pwd): score += 1
    if re.search(r"[0-9]", pwd): score += 1
    if re.search(r"[!@#$%^&*]", pwd): score += 1
    levels = ["أقوية جد", "قوية", "متوسطة", "ضعيفة", "أضعف جد"]
    print(f"{Fore.CYAN} القوة التقديرية: {levels[score]} ({score}/4)")
    input("لمتابعة اضغط Enter ")

def main():
    if not login_screen():
        return
    os.system("clear")
    print(BANNER)
    speak("Access Granted. Welcome back commander. How can I help you today?")
    print(f"{Fore.GREEN} أريد من إداة التقييم المساعدة ")
    print(f"{Fore.WHITE} أنا مساعد الآمن، جاهز لتقديم المهام المطلوبة ")
    while True:
        print(f"{Fore.CYAN} القائمة الرئيسية لـ GEMINI: ")
        print(f"{Fore.WHITE} 1 {Fore.MAGENTA} (Web) فحص ثغرات المواقع ")
        print(f"{Fore.MAGENTA} 2 {Fore.WHITE} (SSL) فحص قوة تشفير المواقع ")
        print(f"{Fore.MAGENTA} 3 {Fore.WHITE} (فحص الروابط) (كشف التلقيم) ")
        print(f"{Fore.MAGENTA} 4 {Fore.WHITE} (فحص الشبكة) (تحليل الأجهزة) ")
        print(f"{Fore.MAGENTA} 5 {Fore.WHITE} (فحص المواقع) (فحص ADB) ")
        print(f"{Fore.MAGENTA} 6 {Fore.WHITE} (اختبار قوة كلمة المرور) ")
        print(f"{Fore.RED} 0 {Fore.WHITE} (خروج) ")
        choice = input(f"{Fore.CYAN} ما نختار أمر: (Style:RESET_ALL)")
        if choice == '1': web_scan()
        elif choice == '2': check_ssl()
        elif choice == '3': check_malicious_link()
        elif choice == '4': network_scan()
        elif choice == '5': phone_scan()
        elif choice == '6': password_check()
        elif choice == '0':
            speak("Goodbye and stay safe")
            print(f"{Fore.YELLOW} أنا آمن في تطبيق الخروج. ")
            break
        else:
            print(f"{Fore.RED}[A] خيار غير صحيح ")

if __name__ == "__main__":
    main()
```

الكود يشمل مكاتب ودوال لكل ادائه واجهة الأداء



نقوم بكتابة كلمة المرور ل الدخول.



```
GEMINI
Security Intelligent Tool V4.0 (PRO)

المساعد الذكي للأمن السيبراني - كالي لينكس

! بك في أداة الفحص الذكية أرحب
. أنا مساعدك الأمني، جاهز لتنفيذ المهام المطلوبة

القائمة الرئيسية للمهام:

1>> فحص ثغرات المواقع (Web)
2>> فحص قوة تشفير المواقع (SSL)
3>> فحص الروابط (كشف التلغيم)
4>> فحص الشبكة (اكتشاف الأجهزة)
5>> فحص الثغرة (ADB)
6>> اختبار قوة كلمة المرور
0>> إغلاق الأداة

>> بانتظار أمرك
ALSA lib pcm.c:2722:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2722:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2722:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:878:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:878:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:878:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:878:(find_matching_chmap) Found no matching channel map
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
```

اول ما تعرض لك الاداءة تقوم بالترحيب بك صوتياً
وتعرض لك المهام التي تقوم بها وكل اداءه مربوطة برقم

```
# ترحيب صوتي عند الدخول الناجح
speak("Access granted. Welcome back commander. How can I help you today?")

print(f"{Fore.GREEN}!مرحباً بك في أداة الفحص الذكية")
print(f"{Fore.WHITE}. أنا مساعدك الأمني، جاهز لتنفيذ المهام المطلوبة\n")
```

اداءه Gemini تجمع بين ٦ أدوات تعمل كل أداة بعمل خاص بها
سوف اشرح كل أداء على حدا.

١-

```
1
🔗 أدخل رابط الموقع (مثال: google.com):
```

هذي الاداءه نقوم بإدخال رابط الموقع وهي توم بفحصه
ان الموقع مومان ولا يوجد به ثغرات.

```
def web_scan():
    url = input(f"{Fore.BLUE}🔗 أدخل رابط الموقع (مثال: google.com): ")
    if not url.startswith('http'): url = 'https://' + url
    print(f"{Fore.YELLOW}🔍 جاري فحص الرؤوس الأمنية لـ {url}...")
    try:
        r = requests.get(url, timeout=5)
        headers = {"X-Frame-Options": "Clickjacking", "Content-Security-Policy": "XSS", "X-Content-Type-Options": "Sniffing"}
        for h, desc in headers.items():
            status = f"{Fore.GREEN}[✅] مؤمن" if h in r.headers else f"{Fore.RED}[❌] معرض لثغرة {desc}"
            print(f"{h}: {status}")
    except: print(f"{Fore.RED}[!] تعذر الوصول للموقع")
```

```
2
>>> بانتظار امر
🔗 أدخل الدومين لفحص التشفير: https://www.kali.org/docs/
[✅] البروتوكول: TLSv1.3 | 25256 bits
القائمة الرئيسية للمهاجم:
```

٢-

نقوم بإدخال رابط الموقع من اجل تحليل نوع وقوة التشفير
المستخدم في الموقع.

```
def check_ssl():
    host = input(f"{Fore.BLUE}🔒 أدخل الدومين لفحص التشفير: ")
    if "://" in host: host = urlparse(host).netloc
    try:
        context = ssl.create_default_context()
        with socket.create_connection((host, 443), timeout=5) as sock:
            with context.wrap_socket(sock, server_hostname=host) as ssock:
                cipher = ssock.cipher()
                print(f"{Fore.GREEN}[✅] البروتوكول: {ssock.version()} | القوة: {cipher[2]} bits")
    except: print(f"{Fore.RED}[!] فشل فحص SSL.")
```

```
3 >>> انتظر أمر
أدخل الرابط لتحليله: https://www.kali.org/docs/
الرابط يبدو آمن [✓]
```

-٣

هذه الأداة تقوم بفحص الروابط هل هي محمية ولا ملغمة بكود خبيث.

```
def check_malicious_link():
    link = input(f"{Fore.BLUE}🔗 أدخل الرابط لتحليله: ")
    suspicious = ["login", "free", "gift", "verify", "update"]
    is_bad = any(word in link.lower() for word in suspicious)
    try:
        res = requests.get(link, timeout=5, allow_redirects=True)
        if is_bad or len(res.history) > 1:
            print(f"{Fore.RED}[⚠️] !تحذير: الرابط مشبوه أو يحتوي تحويلات مخفية")
        else: print(f"{Fore.GREEN}[✓] الرابط يبدو آمناً")
    except: print(f"{Fore.RED}[!] تعذر تحليل الرابط.")
```

```
4 >>> انتظر أمر
نطاق الشبكة 192.168.1.1 : (192.168.1.1/24)
جارى فحص الأجهزة المتصلة
```

-٤

تقوم بفحص الأجهزة المتصلة بهذه الشبكة.

```
def network_scan():
    from scapy.all import ARP, Ether, srp
    ip_range = input(f"{Fore.BLUE}🌐 (192.168.1.1/24) نطاق الشبكة: ")
    print(f"{Fore.YELLOW}🔍 جارى فحص الأجهزة المتصلة...")
    try:
        result = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/ARP(pdst=ip_range), timeout=2, verbose=0)[0]
        for _, rcved in result: print(f"{Fore.GREEN}IP: {rcved.psrc} | MAC: {rcved.hwsrc}")
    except: print(f"{Fore.RED}[!] sudo يرجى التشغيل بـ")
```

```
5 >>> انتظر أمر
الهاتف: 192.168.2.2 IP أدخل
ADB الهاتف مؤمن من منفذ [✓]
```

-٥

هذه الأداة تقوم بفحص الهاتف هل مؤمن ولا قد تم اختراقه.

```
def phone_scan():
    ip = input(f"{Fore.BLUE}📱 الهاتف IP أدخل: ")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(2)
    if s.connect_ex((ip, 5555)) == 0:
        print(f"{Fore.RED}[⚠️] !مفتوح! الهاتف معرض للاختراق ADB خطر: منفذ")
    else: print(f"{Fore.GREEN}[✓] ADB الهاتف مؤمن من منفذ")
    s.close()
```

```
6 >> بانتظار امرك
... (re باستخدام) اختبار قوة كلمة المرور [*]
ادخل كلمة المرور للاختبار:
القوة التقديرية: متوسطة (2/4)
```

-٦

هذه الاداءة تقوم بفحص كلمة المرور هل هي قوية ولا
ضعيفة .

```
def password_check():
    print(Fore.YELLOW + "\n[*] اختبار قوة كلمة المرور ...")
    pwd = getpass.getpass("ادخل كلمة المرور للاختبار: ")

    # حساب القوة يدوياً
    score = 0
    if len(pwd) >= 8: score += 1
    if re.search(r"[A-Z]", pwd): score += 1
    if re.search(r"\d", pwd): score += 1
    if re.search(r"[@#$$%^&+=]", pwd): score += 1

    levels = ["قوية جداً", "قوية", "متوسطة", "ضعيفة", "ضعيفة جداً"]
    print(f"{Fore.CYAN} القوة التقديرية: {levels[score]} ({score}/4)")
    input("\nمغطى Enter للعودة للقائمة...")
```