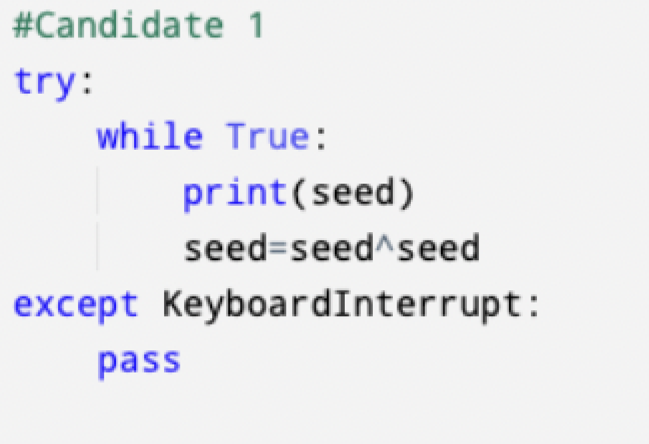
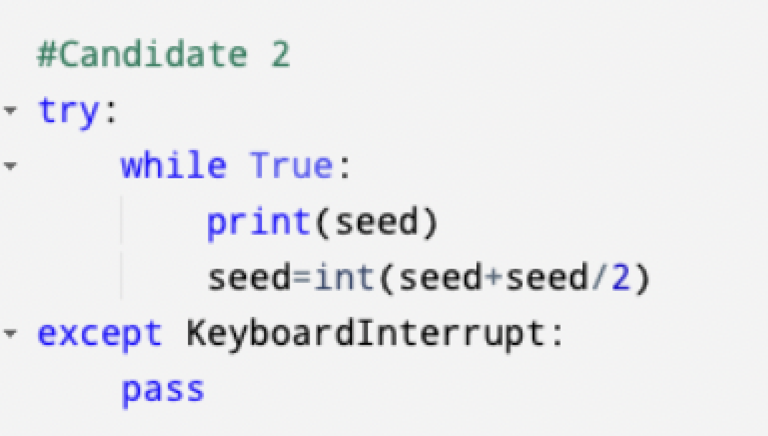
1. pentru toti candidatii, nu este simulata o secventa de numere random.



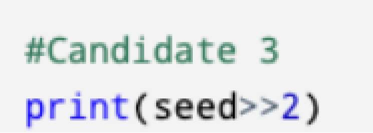
```
#Candidate 1
try:
    while True:
        print(seed)
        seed=seed^seed
except KeyboardInterrupt:
    pass
```

va fi mereu seed-ul urmat doar de 0



```
#Candidate 2
try:
    while True:
        print(seed)
        seed=int(seed+seed/2)
except KeyboardInterrupt:
    pass
```

crestere continua, usor de dedus



```
#Candidate 3
print(seed>>2)
```

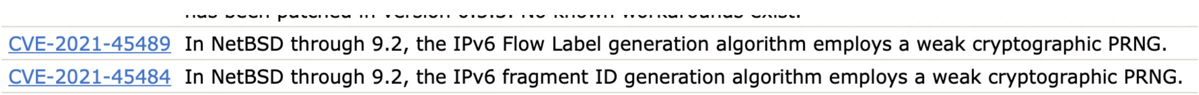
va fi catul impartirii seed-ului cu 4

3.
CWE ID: 336 https://cwe.mitre.org/data/definitions/336.html

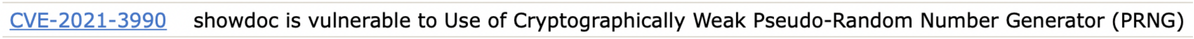
adversarul poate gasi seedul prin brute force
CWE ID: 339 https://cwe.mitre.org/data/definitions/339.html

brute force ID pe CAPEC: 112 https://capec.mitre.org/data/definitions/112.html
mentiuni despre seed: Periodicity, the need for seed values, or weaknesses in the generator all can result in a significantly smaller secret space. Assuming a finite secret space, a brute force attack will eventually succeed.

alte utilizari defectuoase: https://cwe.mitre.org/data/definitions/338.html

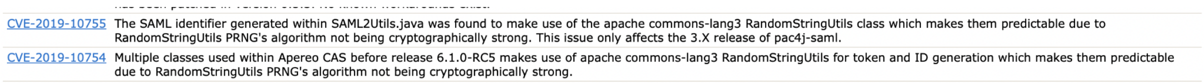


	has been patched in version 0.5.3. No known workarounds exist.
CVE-2021-45489	In NetBSD through 9.2, the IPv6 Flow Label generation algorithm employs a weak cryptographic PRNG.
CVE-2021-45484	In NetBSD through 9.2, the IPv6 fragment ID generation algorithm employs a weak cryptographic PRNG.

- 

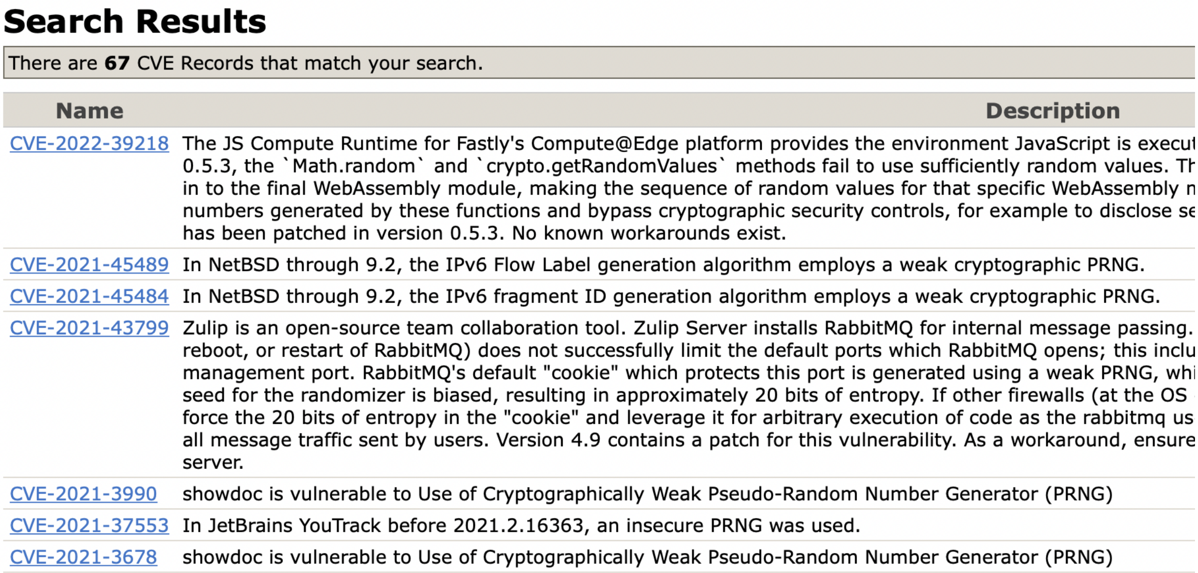
CVE-2021-3990	showdoc is vulnerable to Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)

https://cwe.mitre.org/data/definitions/337.html

- 

	has been patched in version 0.5.3. No known workarounds exist.
CVE-2019-10755	The SAML identifier generated within SAML2Utils.java was found to make use of the apache commons-lang3 RandomStringUtils class which makes them predictable due to RandomStringUtils PRNG's algorithm not being cryptographically strong. This issue only affects the 3.X release of pac4j-saml.
CVE-2019-10754	Multiple classes used within Apereo CAS before release 6.1.0-RC5 makes use of apache commons-lang3 RandomStringUtils for token and ID generation which makes them predictable due to RandomStringUtils PRNG's algorithm not being cryptographically strong.

67 inregistrari CVE cu referire la PRNG



Search Results

There are **67** CVE Records that match your search.

Name	Description
CVE-2022-39218	The JS Compute Runtime for Fastly's Compute@Edge platform provides the environment JavaScript is execut 0.5.3, the `Math.random` and `crypto.getRandomValues` methods fail to use sufficiently random values. Th in to the final WebAssembly module, making the sequence of random values for that specific WebAssembly n numbers generated by these functions and bypass cryptographic security controls, for example to disclose se has been patched in version 0.5.3. No known workarounds exist.
CVE-2021-45489	In NetBSD through 9.2, the IPv6 Flow Label generation algorithm employs a weak cryptographic PRNG.
CVE-2021-45484	In NetBSD through 9.2, the IPv6 fragment ID generation algorithm employs a weak cryptographic PRNG.
CVE-2021-43799	Zulip is an open-source team collaboration tool. Zulip Server installs RabbitMQ for internal message passing. reboot, or restart of RabbitMQ) does not successfully limit the default ports which RabbitMQ opens; this inclu management port. RabbitMQ's default "cookie" which protects this port is generated using a weak PRNG, whi seed for the randomizer is biased, resulting in approximately 20 bits of entropy. If other firewalls (at the OS force the 20 bits of entropy in the "cookie" and leverage it for arbitrary execution of code as the rabbitmq us all message traffic sent by users. Version 4.9 contains a patch for this vulnerability. As a workaround, ensure server.
CVE-2021-3990	showdoc is vulnerable to Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)
CVE-2021-37553	In JetBrains YouTrack before 2021.2.16363, an insecure PRNG was used.
CVE-2021-3678	showdoc is vulnerable to Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)