Tudorache Alexandru-Theodor

Grupa 342

Laborator 6

Securitatea Sistemelor Informatice

1.

Candidate 1 -> prin operatia *seed = seed^seed*, dupa prima rulare seed-ul va ajunge la 0, ceea ce il va face inutil.

Candidate 2 -> din cauza operatiei *seed = seed + seed / 2*, seed-ul poate fi usor prezis la fiecare pas, ceea ce il face inutil.

Candidate 3 -> prin shiftarea la dreapta cu 2 biti, dupa un anumit numar de astfel de operatii seed-ul va ajunge sa fie 0, ceea ce il va face inutil.

2.

a) Parola: O functionalitate ca aceasta poate fi folosita intr-o aplicatie informatica pentru a-i sugera utilizatorului o parola puternica, in loc sa existe posibilitatea de a-si alege una slaba.

Un scenariu de utilizare:
Un utilizator doreste sa-si creeze un cont pe o anumita platforma. Platforma respectiva ii va sugera o parola puternica generata folosind o functionalitate precum cea prezentata.

b) URL-safe String: un astfel de String poate fi folosit pentru generarea unui URL.
Un scenariu de utilizare:

Generarea id-ului unui user, care va fi accesat, de exemplu, folosind url-ul website.com/users/id.

c) Hex Token: un asfel de token poate fi folosit pentru generarea unei chei de criptare.

Un scenariu de utilizare:

Vrem sa criptam un mesaj folosind OTP. Pentru criptarea acestuia vom avea nevoie de o cheie secreta. Aceasta poate fi reprezentata de tokenul nostru.

f) Am folosit biblioteca hashlib pentru a encripta parola ca in cazul in care un inamic patrunde in “baza de date” sa nu poata citi parola fara sa o decripteze.

Posibil output al programului:

```
a) Parola:
o3@.DQ7xjdHYW9

b) String URL-safe:
xR6jMxiag1xiRQJXzCmSmCdIz1cVebdAZA
Lungime URL: 34

c) Token hexazecimal:
b9d35d4e874c3c1db848814441b04b117d
Lungime Token: 34

d) Comparare 'secventa1' cu 'secventa2':
Sunt diferite
Comparare 'secventa1' cu 'secventa1':
Sunt egale

e) Cheie:
36d7f5422beeab1b395bf4bcd42b7a9991c016d725a43c99e3676166b6da2c9a3d233259f77748aef592501dda30503bd695
Lungime Cheie: 100

f) Stocare parola:
Password is correct
```

3.

a) Pentru generarea AccountID:

Deoarece se foloseste mereu acelasi seed, la fiecare rulare a codului va fi generat acelasi ID.

Pentru generarea SessionID:

Deoarece functia srand se foloseste de id-ul user-ului, pentru acelasi user va fi generat mereu acelasi sessionID. (Adica fiecare user va avea un singur session id, diferit de cele ale celorlalti useri)

b)

CWE-336: Same Seed in Pseudo-Random Number Generator (PRNG)

Weakness ID: 336
Abstraction: Variant
Structure: Simple

Presentation Filter: Complete

Description

A Pseudo-Random Number Generator (PRNG) uses the same seed each time the product is initialized.

c)

CWE-339: Small Seed Space in PRNG

Weakness ID: 339
Abstraction: Variant
Structure: Simple

Presentation Filter: Complete

Description

A Pseudo-Random Number Generator (PRNG) uses a relatively small seed space, which makes it more susceptible to brute force attacks.

e)

CWE-332: Insufficient Entropy in PRNG

Weakness ID: 332
Abstraction: Variant
Structure: Simple

Presentation Filter: Complete

Description

The lack of entropy available for, or used by, a Pseudo-Random Number Generator (PRNG) can be a stability and security threat.

CVE:

Observed Examples

Reference	Description
CVE-2019-1715	security product has insufficient entropy in the DRBG, allowing collisions and private key discovery

CWE-335: Incorrect Usage of Seeds in Pseudo-Random Number Generator (PRNG)

Weakness ID: 335
Abstraction: Base
Structure: Simple

Presentation Filter: Complete

Description

The software uses a Pseudo-Random Number Generator (PRNG) but does not correctly manage seeds.

CVE:

Observed Examples

Reference	Description
CVE-2019-11495	server uses erlang:now() to seed the PRNG, which results in a small search space for potential random seeds
CVE-2018-12520	Product's PRNG is not seeded for the generation of session IDs
CVE-2016-10180	Router's PIN generation is based on rand(time(0)) seeding.

CWE-337: Predictable Seed in Pseudo-Random Number Generator (PRNG)

Weakness ID: 337
Abstraction: Variant
Structure: Simple

Presentation Filter: Complete

Description

A Pseudo-Random Number Generator (PRNG) is initialized from a predictable seed, such as the process ID or system time.

CVE:

Observed Examples

Reference	Description
CVE-2019-11495	server uses erlang:now() to seed the PRNG, which results in a small search space for potential random seeds
CVE-2008-0166	The removal of a couple lines of code caused Debian's OpenSSL Package to only use the current process ID for seeding a PRNG
CVE-2016-10180	Router's PIN generation is based on rand(time(0)) seeding.
CVE-2018-9057	cloud provider product uses a non-cryptographically secure PRNG and seeds it with the current time

CWE-338: Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)

Weakness ID: 338
Abstraction: Base
Structure: Simple

Presentation Filter: Complete

Description

The product uses a Pseudo-Random Number Generator (PRNG) in a security context, but the PRNG's algorithm is not cryptographically strong.

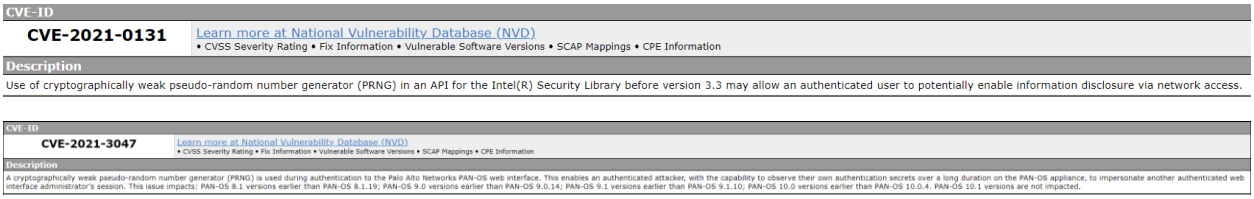
CVE:

Observed Examples

Reference	Description
CVE-2009-3278	Crypto product uses rand() library function to generate a recovery key, making it easier to conduct brute force attacks.
CVE-2009-3238	Random number generator can repeatedly generate the same value.
CVE-2009-2367	Web application generates predictable session IDs, allowing session hijacking.
CVE-2008-0166	SSL library uses a weak random number generator that only generates 65,536 unique keys.

f)

Am identificat 4 inregistrati CVE din 2021 care au legatura cu PRNG:



CVE-ID

CVE-2021-0131

Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Use of cryptographically weak pseudo-random number generator (PRNG) in an API for the Intel(R) Security Library before version 3.3 may allow an authenticated user to potentially enable information disclosure via network access.

CVE-ID

CVE-2021-3047

Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

A cryptographically weak pseudo-random number generator (PRNG) is used during authentication to the Palo Alto Networks PAN-OS web interface. This enables an authenticated attacker, with the capability to observe their own authentication secrets over a long duration on the PAN-OS appliance, to impersonate another authenticated web interface administrator's session. This issue impacts: PAN-OS 8.1 versions earlier than PAN-OS 8.1.19; PAN-OS 9.0 versions earlier than PAN-OS 9.0.14; PAN-OS 9.1 versions earlier than PAN-OS 9.1.10; PAN-OS 10.0 versions earlier than PAN-OS 10.0.4. PAN-OS 10.1 versions are not impacted.

CVE-ID

CVE-2021-3678

Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

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

CVE-ID

CVE-2021-37553

Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

In JetBrains YouTrack before 2021.2.16363, an insecure PRNG was used.