

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Insecure Randomness: TryHackMe Writeup



Ansul Kotadia · [Follow](#)

6 min read · Jan 17, 2025

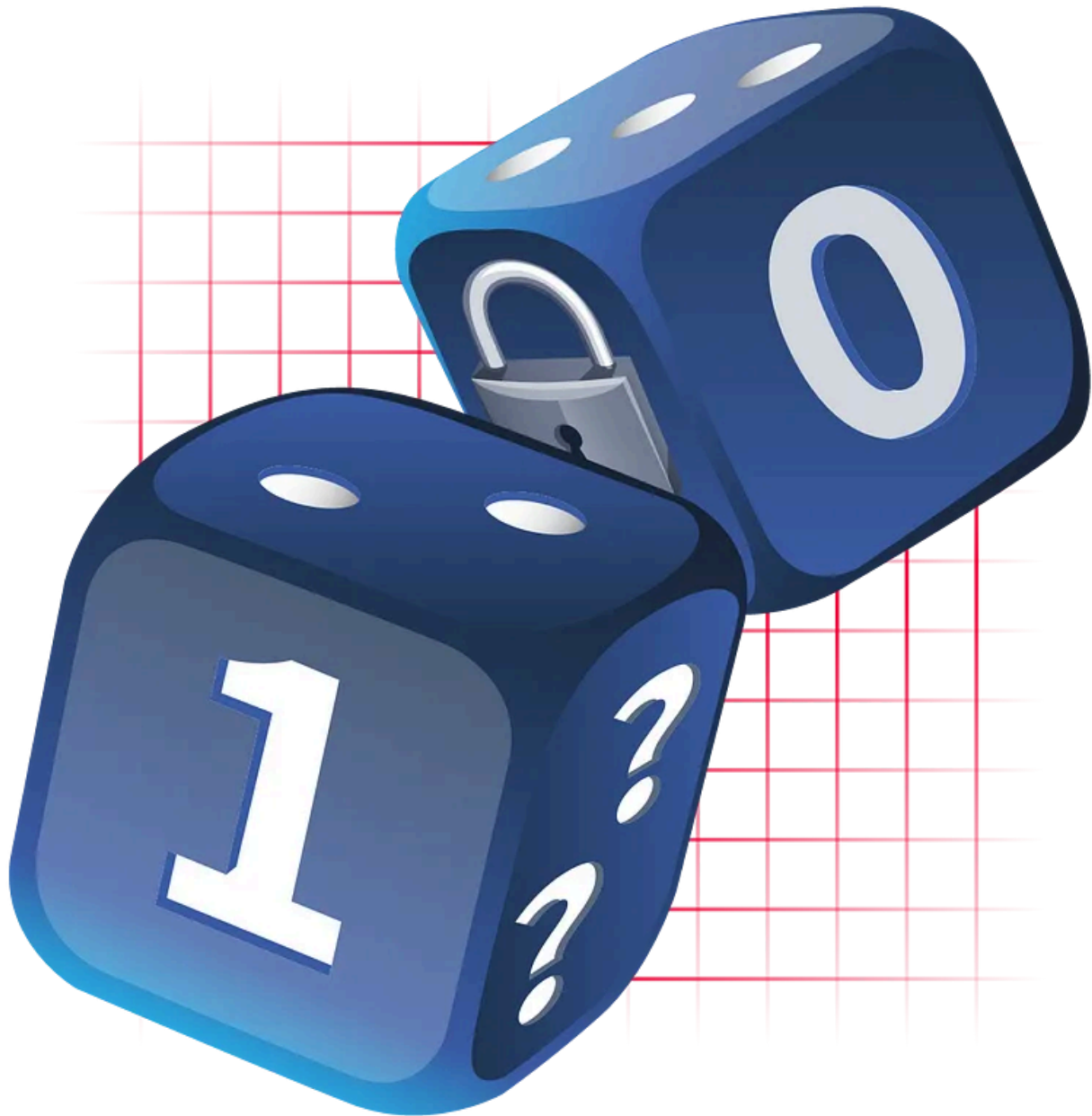


Listen



Share

... More



Insecure Randomness THM

Task 1: Introduction

Insecure randomness occurs when web applications use predictable or poorly generated random values, making them vulnerable to attacks. While randomness is essential for securing tokens, session IDs, and cryptographic keys, insecure implementation can allow attackers to exploit these predictable values to bypass authentication, hijack sessions, or even decrypt sensitive data.

Learning Objectives

Throughout this room, you will gain a comprehensive understanding of the following key concepts:

- Understanding insecure randomness
- Type of random number generators
- Weak or Insufficient Entropy
- Predictable seeds during token generation

No answer needed.

Task 2: Few Important Concepts

In this section, we will understand the fundamental concepts surrounding randomness and its role in security:

Randomness

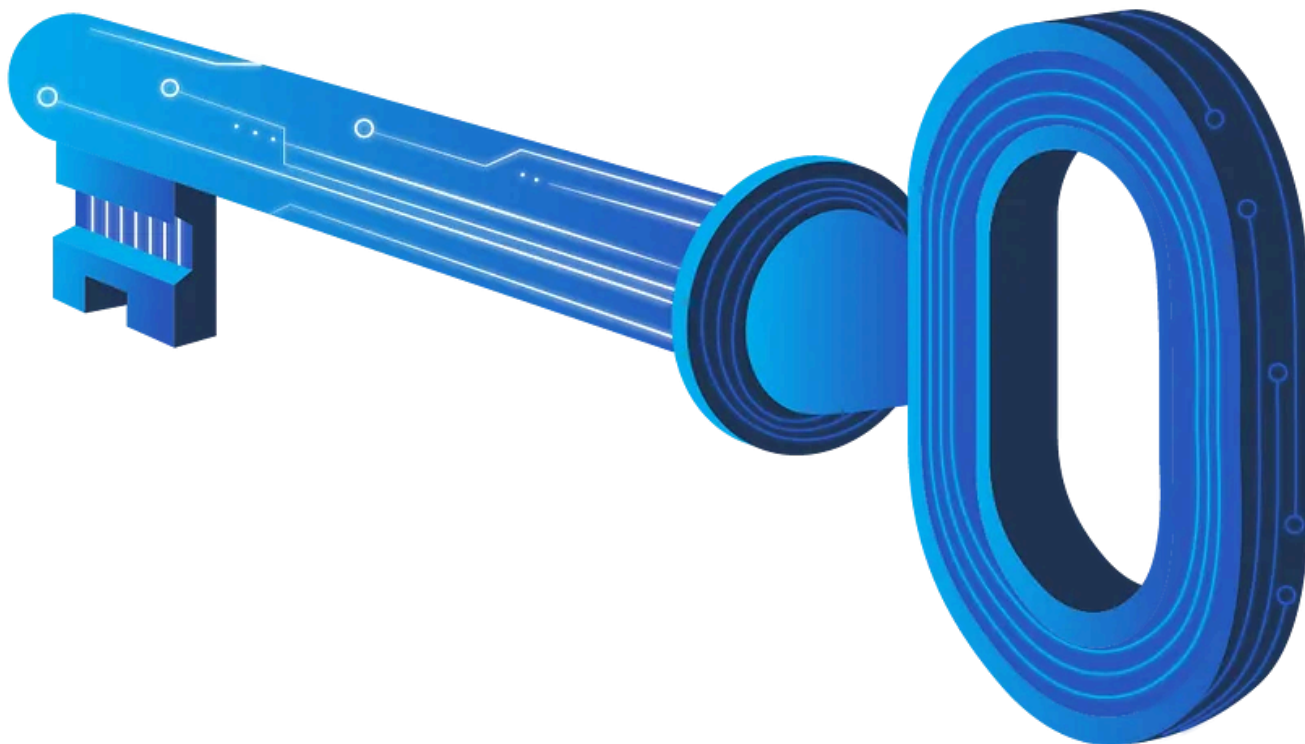
Randomness refers to the lack of pattern or predictability in data, making it an essential component in secure systems. In cryptography, true randomness ensures an attacker cannot predict values such as keys, tokens, and nonces. We will explore how randomness is generated and the distinction between True Random Number Generators (TRNG) and Pseudorandom Number Generators (PRNG).

Entropy

Entropy represents the amount of randomness or unpredictability in a system and is often used to assess the security of cryptographic keys, tokens, or random values. Higher entropy indicates greater uncertainty, making it more difficult for attackers to predict or guess the values, which is essential for secure cryptographic operations. Low entropy can lead to weak security, increasing the risk of attacks like brute-forcing or token prediction.

Cryptographic Keys

Cryptographic keys are secret values used in algorithms to encrypt and decrypt data, ensuring confidentiality, integrity, and authentication. They are critical components in symmetric and asymmetric encryption methods and must be securely generated and managed to prevent unauthorised access. The strength of a cryptographic key depends on its length and randomness.

**Questions:**

#2.1 What measures the amount of randomness or unpredictability in a system?

Answer: Entropy

#2.2 Is it a good practice to keep the same seed value for all cryptographic functions?
(yea/nay)

Answer: nay

Task 3: Types of Random Number Generators

True Random Number Generator (TRNG):

TRNGs generate randomness by relying on unpredictable physical phenomena like thermal noise or radioactive decay. Since these generators stem from natural events, they produce inherently random values. TRNGs are commonly used in highly sensitive cryptographic operations, such as generating the keys for algorithms like RSA or ECC. These keys are then used in tasks like encryption, digital signatures, and certificate creation, where unpredictability is crucial for security. However, TRNGs require specialised hardware and can be slower than other RNGs, making them less suitable for tasks requiring rapid number generation.

Pseudorandom Number Generator (PRNG):

PRNGs, unlike TRNGs, generate random numbers algorithmically based on an initial seed value. While they may appear random, they are deterministic, meaning the same seed will always produce the same sequence of numbers. PRNGs are faster and more efficient than TRNGs and are suitable for applications that quickly need large quantities of random numbers, like simulations or gaming. However, since they are algorithmic, predictability becomes a risk if an attacker can deduce the seed or its generation method.

Question:

#3.1 You prepare a game involving immediate interaction and random event simulation but with no critical security requirements. Which type of RNG would be most appropriate for this purpose? Write the correct option only.

- a) TRNG
- b) Statistical PRNG
- c) We should not use randomness in games
- d) None of the above

Answer: B

Task 4: Weak or Insufficient Entropy

Open in app ↗

Medium

🔍 Search



unpredictability or randomness in a system, often derived from sources like environmental factors (e.g., hardware noise or user interactions). When these entropy sources are weak or insufficient, the generated random values are not truly random and become vulnerable to attacks.

For example, if an encryption key is generated using low-entropy data, such as a timestamp, an attacker could use this predictable information to reduce the complexity of finding the key. Similarly, poor entropy sources, like system clocks or predictable user inputs, can lead to weak randomness in applications.

Questions:

#4.1 What is the flag value after logging in as the victim user?

Answer: THM{VICTIM_SIGNED_IN}

#4.2 What is the flag value after logging in as the master user?

Answer: THM{ADMIN_SIGNED_IN007}

#4.3 What is the PHP function used to create the token variable in the code above?

Answer: time()

Task 5: Predictable Seed in PRNGs

In this task, the focus shifts to cases where a **predictable seed** is used to initialise PRNGs. If the seed is weak or predictable, an attacker can reproduce the entire sequence of random numbers, leading to severe vulnerabilities in systems that rely on these **random** values.

An example of the impact of predictable seeding is in CAPTCHA systems, where the random value determining the CAPTCHA challenge will be generated to detect a bot activity. If the seed used to initialise the PRNG is predictable, an attacker could predict the CAPTCHA values ahead of time, allowing them to bypass the CAPTCHA and access restricted areas of the application without solving it.

This issue also manifests in systems like lottery or game applications, where PRNGs determine the outcome of random draws. When these generators are seeded with predictable values, such as timestamps, attackers can manipulate the system by predicting the outcome, ensuring they win consistently. By exploiting the predictable PRNG seed, the attacker can reverse-engineer or replicate the same random sequence, breaking the system's fairness.

Questions:

#5.1 What is the flag value after logging in as magic@mail.random.thm

Answer: THM{MAGIC_SIGNED_IN11010}

#5.2 What is the flag value after logging in as hr@mail.random.thm?

🌟 **For this answer read the same story at:** [Insecure Randomness: TryHackMe Writeup | by Ansul Kotadia | Jan, 2025 | Medium](https://ansul71098.medium.com/insecure-randomness-tryhackme-writeup-8c39dbe5e6f8)

#5.3 What is the PHP function used to seed the RNG in the code above

Answer: mt_srand

Task 6: Mitigation Measures

When discussing best practices for identifying and mitigating insecure randomness, it's important to address both pentesters and secure coders, as their perspectives and responsibilities differ. Here's a breakdown of the best practices for each:

Pentesters:

- **Identify Weak Randomness in Code:** During code reviews or application assessments, look for the use of weak random number generators like `mt_rand()` or `rand()`, especially when they generate security-sensitive values like session tokens or password reset links.
- **Reverse Engineer Predictable Tokens:** Attempt to exploit predictable randomness by reverse-engineering the seed used in PRNGs. Tools like `php_mt_seed` can help pentesters demonstrate how predictable tokens (e.g., magic links) can be recreated. Test for weak or predictable seeds like timestamps, IP addresses, or user-specific values.
- **Test Token Exhaustion:** If Cryptographically Secure Pseudorandom Number Generators (CSPRNGs) are not used, run brute-force or replay attacks against generated tokens, session IDs, or other randomness-dependent features. Ensure that tokens are not guessable or predictable.

Secure Code Developers:

- **Use Cryptographically Secure PRNGs:** Always use CSPRNGs, such as `random_bytes()` or `openssl_random_pseudo_bytes()` in PHP or `java.security.SecureRandom` in Java. These CSPRNGs are designed to generate unpredictable values suitable for security-critical applications like session tokens, API keys, or password reset tokens.
- **Avoid Predictable Seed Values:** Never use predictable values like the current timestamp, IP address, or process ID for seeding random number generators.

These values can be easily guessed or reverse-engineered by attackers. Instead, use entropy from cryptographic sources or system-provided randomness (e.g., `/dev/urandom` in Linux).

- **Regenerate Randomness for Every Critical Operation:** Avoid reusing random values or seeds across multiple requests or users. Regenerate fresh randomness for each operation that requires secure tokens, such as session management, password resets, or magic links.
- **Use Strong Algorithms for Key Generation:** When generating cryptographic keys, always use secure key generation functions that derive keys from strong sources of entropy. For example, in PHP, you can use `openssl_pkey_new()` for RSA key generation, which relies on secure randomness.

Through thorough testing techniques and secure coding practices, both pentesters and secure developers can ensure the elimination of vulnerabilities related to insecure randomness.

Questions:

#6.1 Which of the following can be considered as a weak seed value? Write the correct letter only.

- a) Timestamp
- b) IP Address
- c) 6-digit constant value
- d) All of the above

Answer: D

Task 7: Conclusion

This room explored the key aspects of insecure randomness, starting with an overview of its fundamental concepts. Later explored the differences between TRNG and PRNG, highlighting their importance in secure systems. Through practical exercises, demonstrated how weak entropy and predictable seed values can lead to severe vulnerabilities, such as account takeovers. Lastly, the room examined the

best practices for secure coders and pentesters, ensuring defence against insecure randomness.

No answer needed.

Thank you!

Tryhackme

Tryhackme Walkthrough

Tryhackme Writeup

Insecure Randomness

Cybersecurity



Follow

Written by Ansul Kotadia

62 Followers · 85 Following

Responses (1)



What are your thoughts?

Respond



Lopoxi

6 days ago



Your writeups are detailed!



[Reply](#)

More from Ansul Kotadia



In T3CH by Ansul Kotadia

Silver Platter: TryHackMe Writeup

Introduction: Exploring TryHackMe's SilverPlatter Room



Jan 12




75



4






 Ansul Kotadia

Light: TryHackMe Answers

Room Introduction: "I am working on a database application called Light! Would you like to try it out? If so, the application is running on..."

✦ 6d ago 🖱 4




 Ansul Kotadia

Pyrat: TryHackMe Writeup

Pyrat THM

★ Oct 4, 2024 🖱 3



 Ansul Kotadia

The Sticker Shop: TryHackMe Writeup.

Hey there, fellow hackers! 🙌 Let's dive into a fun and easy TryHackMe room called The Sticker Shop. This room challenges us to exploit a...

★ Dec 2, 2024 🖱 40 💬 1



See all from Ansul Kotadia

Recommended from Medium



In T3CH by Axoloth

TryHackMe | Training Impact on Teams | WriteUp

Discover the impact of training on teams and organisations

★ Nov 5, 2024 🖱 60



Ansul Kotadia

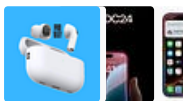
Lo-Fi: TryHackMe Writeup.

Tackling the Lo-Fi TryHackMe room turned out to be a fascinating adventure! With a mix of curiosity and determination, I jumped right into...

Jan 18 🖱 9 💬 1



Lists



Tech & Tools

22 stories · 388 saves



Medium's Huge List of Publications Accepting Submissions

414 stories · 4423 saves



Staff picks

804 stories · 1587 saves



Natural Language Processing

1894 stories · 1555 saves



In InfoSec Write-ups by OverloOked

THM Lo-Fi walkthrough

MODE : Easy

Jan 18 🖱 54 💬 2





 TheHiker

Silver-Platter , TryHackMe Walkthrough | TheHiker

Hello everyone, today I'll be covering the "Silver-Platter" room on TryHackMe. I think that this room is great for intermediate students...

Jan 12  27



 IritT

CyberChef: The Basics — Crypto 101 — Defensive Security Tooling- Cryptography-TryHackMe Walkthrough

This room is an introduction to CyberChef, the Swiss Army knife for cyber security professionals.

Nov 2, 2024



K9ine95

Block ~ Tryhackme ~ walkthrough

One of your junior system administrators forgot to deactivate two accounts from a pair of recently fired employees. We believe these...

Aug 12, 2024



2



See more recommendations