

Sri Lanka Institute of Information Technology



WEB SECURITY (IE2062)

Y2.S2.WE.CS



Bug Bounty Journey

IT23241978

S.D.W.Gunaratne

Briefly How I started

- Watching YouTube videos made my interest to cyber security. Here are some my favourite YouTube channels.

<https://www.youtube.com/@davidbombal>

<https://www.youtube.com/@0dayCTF>

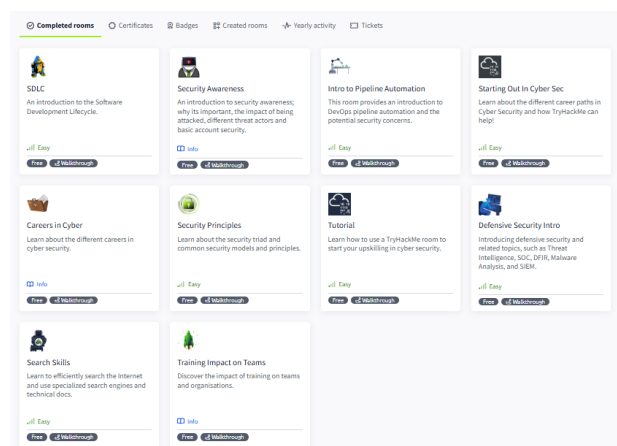
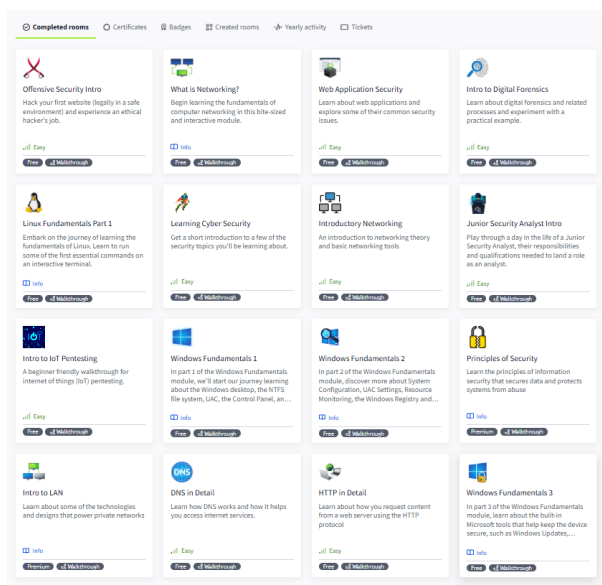
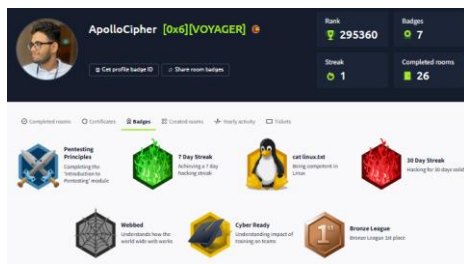
https://www.youtube.com/@_JohnHammond

<https://www.youtube.com/@NetworkChuck>

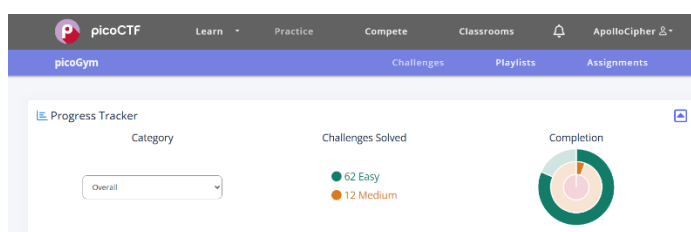
<https://www.youtube.com/@NahamSec>

- Follows walkthroughs and labs in famous plat forms like TryHackMe, Port swingger and PicoCTF to get hands on experience of the cyber security field.

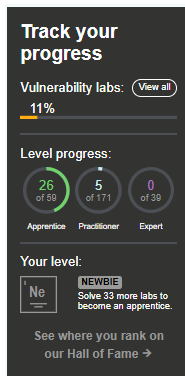
TryHackMe



PicoCTF



Port swinger



With learning these skills set, I aimed to solve real-world problems in web application.

Tools I Use

- Burp Suite (working with web applications)
- Nmap (To find open ports)
- Sublist3r (To find subdomains)
- WAFW00F (To identify web application firewalls)

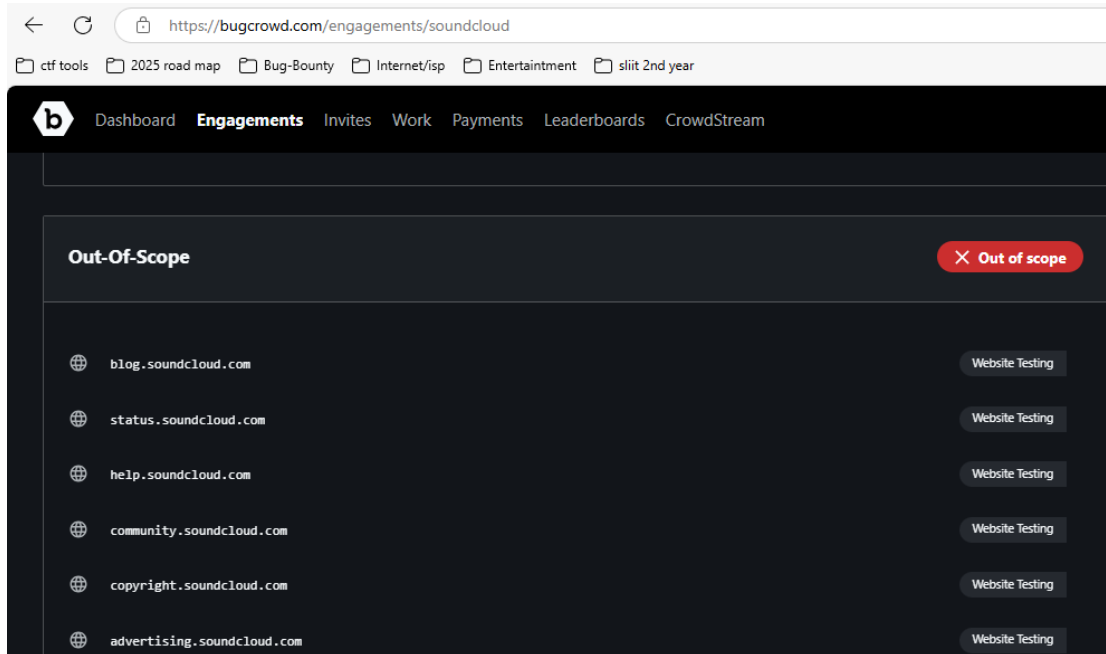
Platforms I Use

- HackerOne
- Bugcrowd



Challenges I Faced

- Owners limit their scope limitations. (can't use entire domain)





- Lack of resources (No free tools or limitation of it)
Example: - Burp suit community edition vs premium.
- Learn New technologies and methods. (cyber security is dynamically change)
- Submit my first bug bounty report.

Dashboard
Engagements
Invites
Work
Payments
Leaderboards
CrowdStream


Submissions
Workflows


Go back


Look! Your site can be shown inside another site — that's not safe.
Submitted 6 days ago · Last activity 6 days ago

ID	21974992-356f-4a5f-8b30-f0e1bd72423c
Submitted	29 Apr 2025 14:57:41 UTC
Target Location	www.underarmour.com
Target category	Web App
VRT	Server Security Misconfiguration > Lack of Security Headers > X-Frame-Options
Priority	P5
Bug URL	https://prod-na04.vercel.com.underarmour.com/en-us
Description	<p>This report describes a Missing Anti-clickjacking Header vulnerability. Full technical details and PoC are included in the attached PDF. All information includes in pdf file.</p> <p>The real reason to do this bug bounty program is gaining new knowledge about web security as a beginner.</p>
Files attached	 Under%20Armour%20-%20Bug%20bounty.pdf (418 KB)

Activity


ApolloCipher created the submission
6 days ago (29 Apr 2025 14:57:42 UTC)


teapot_bugcrowd sent a message
6 days ago (29 Apr 2025 15:08:38 UTC)

Hi


Thank you for your submission, however, this issue is considered to be a P5 (Informational) finding as per [Bugcrowd's Vulnerability Rating Taxonomy \(VRT\)](#). Therefore, this finding typically does not qualify for a reward.

Typically, this is the case when an issue lacks a demonstrated risk and is considered security best practice. While that will apply regularly when we assign a submission a P5 rating, if you are able to exploit this finding further (or chain it with other findings) to meet the definition of another item within the VRT, please do submit a new report. We look forward to reading it!

As you progress with bug bounties it's important to consider not just the vulnerability but also the impact that this vulnerability has, so we encourage you to always explore any finding to better understand the impact it may have. Each submission should aim to answer the question as an attacker I could...

If you're unsure of the next steps to take this with submission, we recommend the [Bugcrowd University](#) as a starting point for learning how you can escalate bugs from a P5, into P4s or even P3 findings!

Best regards,
-Bugcrowd Security Operations Team


teapot_bugcrowd changed the state to **Informational**
6 days ago (29 Apr 2025 15:08:40 UTC)

Benefits & Motivations

- Experience in Ethical hacking.
- Enhanced soft and hard skills.
- Personal development and increased self-confidence.
- Motivate with my achievements.

Dashboard Engagements Invites Work Payments Leaderboards CrowdStream

[?](#)
[Feedback](#)

N/A

Accuracy
100.0%

performance stats to meet the criteria to join private programs. Learn more about private programs

Vulnerabilities
1

Accuracy
100.00%

Priority percentiles

The priority percentile against other researchers based on valid reported vulnerabilities.

P1 - 0th

P2 - 0th

P3 - 0th

P4 - 0th

P5 - 47th

Reported vulnerabilities

Vulnerabilities scaled by technical severity in this period.

Critical (x40)
Severe (x20)
Moderate (x10)
Low (x5)

Submission type and severity

A look into the type and severity of vulnerabilities in this period.

VRT category	Count
Server Security Misconfiguration	1

Technical severity breakdown

Target type	Count
Web App	1

0

Critical High submissions Moderate Low

Achievements

View all achievements

Submission Shogun Level 1

Bounty Bee Level 1

Leaderboard rankings

View leaderboard

Submission Shogun

23976^m

1 submission

Bounty Bee

20655^m

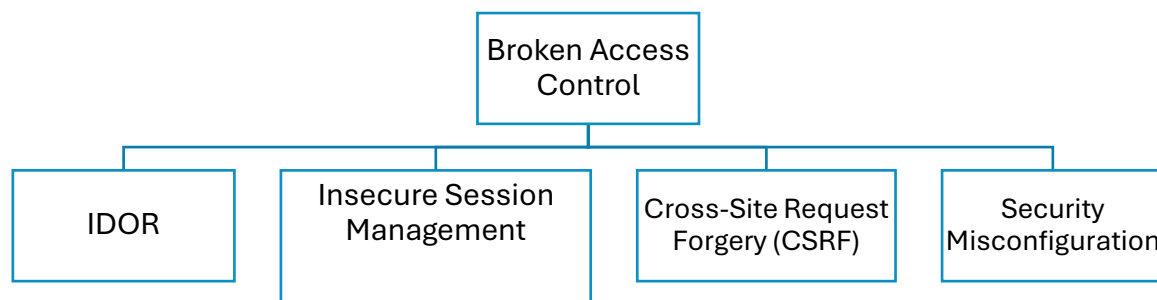
1 engagement

OWASP Top 10 Vulnerabilities

A01:2021-Broken Access Control moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.

Broken Access Control

The reason why we are using Access Control are to decide who can access, change, delete resources in the system. If this not properly implement their can be happen bad exploits by the adversary. To understand Broken access control easily we can divide to child classes like this.

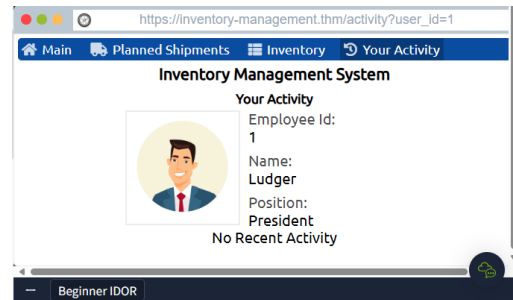
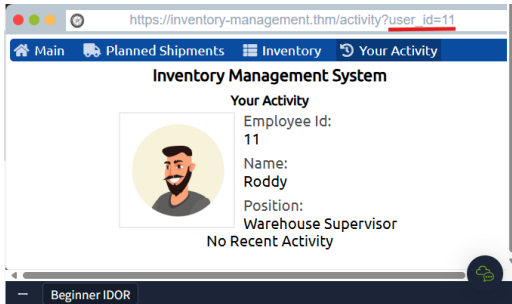


i. Insecure Direct Object References (IDOR)

This happens without having proper authorization checks, if app allows direct access to files, database records or other system resources. So that attackers can changing id and URL and get access, modify or terminate data. This can lead informational to business-critical risk level.

Example scenario:

If user(attacker) accesses his page and would try other possible ids and can access another user's sensitive data.

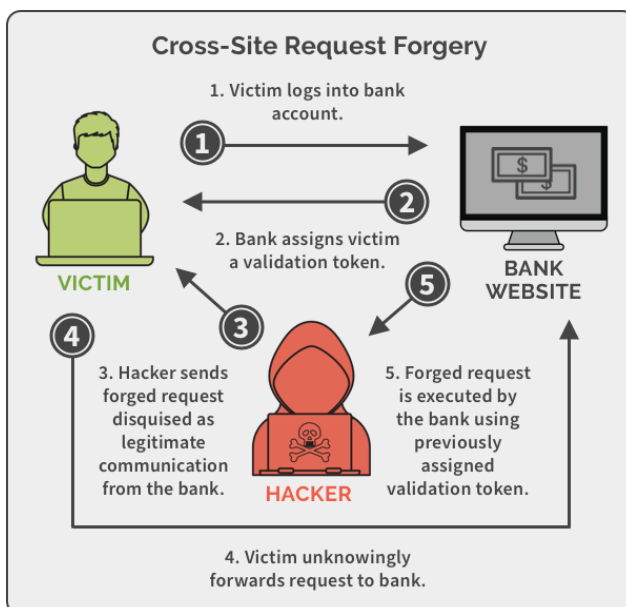


ii. Insecure Session Management

Usually, every session has unique session token to identify it uniquely. Insecure Session Management happens, because of weak session token generation, lack of session timeouts, effective session termination and tokens are not properly storing and sharing. Because of this kind of scenario gains many problems, such as session hijack, unauthorized access and security breaches. To mitigate this, we can do use https, use secure session token algorithms for token generation, implementing secure cookie attributes, and frequently validating and expiring session token.

iii. Cross-Site Request Forgery (CSRF)

Hacker tricks a user into performing an unwanted activity on website they are already logging.

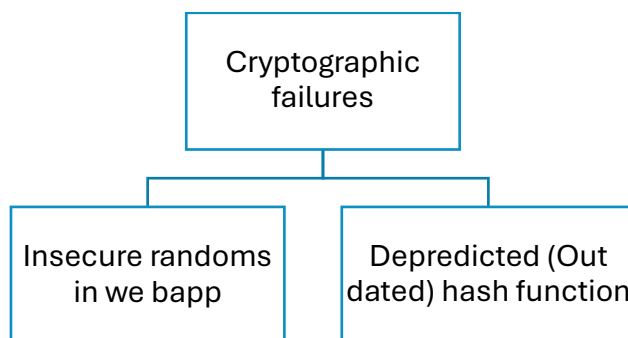


iv. Security Misconfiguration

Security misconfiguration happens when security settings are not configured correctly. Reason for happening these are outdated software; unused features being enabled.

Cryptographic Failures

When the applications don't manage encryption correctly, cryptographic failures happen. This is example for misuse or poorly manage encryptions means, use an outdated encryption algorithm, weak random numbers, short key length and improper keys storing.



i. Insecure randoms in webapp

Random numbers are used for cryptographic functions, session identifiers, token generations and password generations. These are the security issues regarding to insecure randoms.

- Brute force attacks
- Session hijack
- Predict token

ii. Depredated (Outdated) hash function

With highest computational power, older hash function no longer secure. Because of these old hash function leads many vulnerabilities in web applications. Security problems are mentioned below.

Bruteforce attack – Use reverse engineering techniques and cracked the hash data.

Collision attack – If two different plaintexts produce the same hash value.

Ex: attacker makes a fake word file with same hash as the original file. So, system don't understand the difference.

These are the example for out dates hash functions:

- MD5
- SHA-1

These are the example for stronger and secured hash functions:

- SHA-3 (File integrity /Block chain and cryptocurrencies)
- SHA-256 (digital signatures/SSL and TLS certificates /Block chain/data integrity/password hashing)
- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.

Insecure Design

Insecure design happens, when the software architecture isn't planning properly, that means developers build application with poor security aspect. This could lead data breaches and system exploitations. For addressing this insecure design there are some threat modelling approaches. Example like Application security verification standard (ASVS). Use Requirement Traceability Matrix (RTM) also help to reduce overlooked vulnerabilities and improving security.

Requirement Traceability matrix

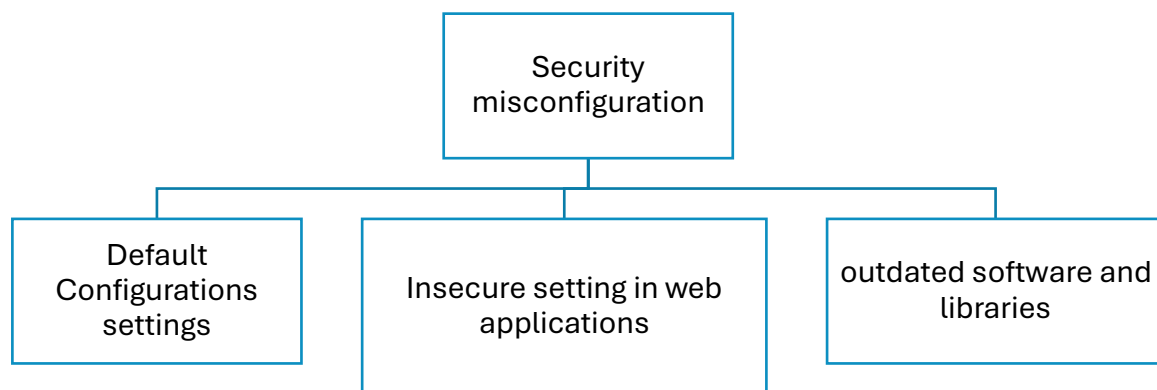
Use Case	Abuse Case	Threat	Mitigation
----------	------------	--------	------------

1.Login Functionality	<ul style="list-style-type: none"> • Bruteforce attacks • Password cracking 	<ul style="list-style-type: none"> • Weak password policy • Credential theft 	<ul style="list-style-type: none"> • Implement account lock • Strong passwords
--------------------------	---	--	--

A04:2021-Insecure Design is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modelling, secure design patterns and principles, and reference architectures.

Security Misconfiguration

Security misconfiguration happens when settings of application’s security are not properly configured. These are the types of Security misconfiguration attacks commonly use.



i. Default Configurations settings

Default configurations are essential for systems, because it helps users to setup easily and fast. As an example, web servers. When default setting can be harmful if users do not change it. example scenarios are:

- Default admin password is usually come up with default credential like admin/admin. This can be guessable for attacker.
- By default, files can be granted more access to files, and this could be vulnerable to attacks.
- Software can be enabling unwanted network port by default, and this could be vulnerable to attacks.

To mitigate these issues, we can implement password policies, disable unwanted features and user awareness and trainings.

ii. Insecure setting in web applications

Poor configuration, Insufficient security controls and lack of security knowledge make massive problems in web applications. These are the common insecure settings in web applications:

- Storing sensitive data without secured environments (like API keys and plaintext password)
- Using vulnerable or outdated libraries and software.
- Unsafely configure headers and cookies.

These are leads to sensitive data exposure, unauthorized access and session hijacking. To prevent this user can do check and update on time, secured password and proper configuration.

A05:2021-Security Misconfiguration moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it's not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.

A06:2021-Vulnerable and Outdated Components was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

A07:2021-Identification and Authentication Failures was previously Broken Authentication and is sliding down from the second position and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.

A08:2021-Software and Data Integrity Failures is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

A09:2021-Security Logging and Monitoring Failures was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.

A10:2021-Server-Side Request Forgery is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.



***** The End. *****