#thehacktivists

Bug Bounty Testing Essential Guideline for Intermediate Hackers

These types of weaknesses can allow an attacker to either capture or bypass the authentication methods that are used by a web application.

- User authentication credentials are not protected when stored.
- Predictable login credentials.
- Session IDs are exposed in the URL (e.g., URL rewriting).
- Session IDs are vulnerable to session fixation attacks.
- Session value does not timeout or does not get invalidated after logout.
- Session IDs are not rotated after successful login.
- Passwords, session IDs, and other credentials are sent over unencrypted connections.

The goal of an attack is to take over one or more accounts and for the attacker to get the same privileges as the attacked user.

Source : https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management

Privilege Escalation

Privilege escalation vulnerability allows malicious user to obtain privileges of another user they are not entitled to. Privilege escalation occurs in two forms: Vertical privilege escalation – Occurs when user can access resources, features or functionalities related to more privileged accounts.

Hackerone Reports :

- https://hackerone.com/reports/246419
- https://hackerone.com/reports/13959
- https://hackerone.com/reports/29420

https://hackerone.com/reports/244567
https://hackerone.com/reports/272570

Source : https://www.owasp.org/index.php/Testing_for_Privilege_escalation_(OTG-AUTHZ-003)

Authentication Bypass

Note : Authenticating a user involves establishing that the user is in fact who he claims to be. Without this facility, the application would need to treat all users as anonymous the lowest possible level of trust.

A flaw in the application that allows users to access application resources without authentication is referred as' Authentication Bypass. Authentication bypass vulnerability is generally caused when it is assumed that users will behave in a certain way and failing to foresee the consequences of users doing the unexpected.

we are going to focus on four key areas which should be examined when testing authentication:

- Forced Browsing
- Parameter Modification
- Session Identifier Prediction
- SQL Injection within Login Forms

Hackerone Reports :

- https://hackerone.com/reports/124845 https://hackerone.com/reports/270981

Source : https://zseano.com/tutorials/3.html
Source : https://www.bugcrowd.com/authentication-bypass/
https://www.owasp.org/index.php/Testing_for_Bypassing_Authentication_Schema_(OTG-AUTHN-004)

Weak Login Function

Depending on the nature of the password-protected resource, an attacker can mount one or more of the following types of attacks:

- Access the contents of the password-protected resources.
- Access password-protected administrative mechanisms such as "dashboard", "management console" and "admin panel," potentially progressing to gain full control of the application.

Source : https://www.owasp.org/index.php/Testing_for_Weaker_authentication_in_alternative_channel_(OTG-AUTHN-010)

Session Fixation

session fixation attacks attempt to exploit the vulnerability of a system that allows one person to fixate another person's session identifier. Most session fixation attacks are web based, and most rely on session identifiers being accepted from URLs or POST data.

Hackerone Reports :

- https://hackerone.com/reports/18501 https://hackerone.com/reports/135797

Source : https://www.owasp.org/index.php/Session_fixation
Source : http://projects.webappsec.org/w/page/13246960/Session%20Fixation

Missing Secure or HTTPOnly Cookie Flag

HttpOnly is an additional flag included in a Set-Cookie HTTP response header. Using the HttpOnly flag when generating a cookie helps mitigate the risk of client side script accessing the protected cookie.

Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted tunnel. For example, after logging into an application and a session token is set using a cookie, then verify it is tagged using the ";secure" flag. If it is not, then the browser believes it safe to pass via an unencrypted channel such as using HTTP.

Hackerone Reports :

- https://hackerone.com/reports/75357 https://hackerone.com/reports/58679
-

Source : https://www.owasp.org/index.php/HttpOnly

Failure to Invalidate Session

Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.

On Logout (Client and Server-Side)

Failure to invalidate the session on the server when the user chooses to logout. The act of logging out should invalidate the session identifier cookie on the client browser as well as invalidated the session object on the server. Failure to do so will allow the session to be re-animated after logout.

Hackerone Reports :   https://hackerone.com/reports/193556

Long Timeout

Failure to invalidate the session when the user closes the browser without logging out. Failure to do so will allow the session to be re-animated in a new browser session.

Failure to automatically terminate the session on the server after some predefined period of inactivity. Failure to do so means the application has no protection from an attacker resuming an abandoned user session.

Hackerone Reports :   https://hackerone.com/reports/244875

On Password Reset and/or Change

User not always get a notification about password change. When a user changes his password via password reset link then such a notification is not sent to the user.

Web app did not verify the email addresses of user accounts before sending an email to them. An attacker can use this functionality and send faulty password reset links to legitimate users.

Hackerone Reports :

- https://hackerone.com/reports/15785   https://hackerone.com/reports/145488
- https://hackerone.com/reports/92251   https://hackerone.com/reports/315512

Concurrent Sessions On Logout

When login to webapp using two different computers I can easily browse the session concurrently . This means that if an attacker somehow knows the password of the user by any means he can login using that info and the main user will not get notified.

Hackerone Reports :

- https://hackerone.com/reports/20122   https://hackerone.com/reports/347748

On Email Change

The e-mail change functionality does not require the current user password to be completed. Since the e-mail could be used to reset the password of the account, could associate a new e-mail to the account, change the primary e-mail associated with the new one and then use the "forgot password" functionality to reset it.

Hackerone Reports : https://hackerone.com/reports/223461

Source : https://affinity-it-security.com/what-is-a-session-management-vulnerability/
https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management

Concurrent Logins

When a user logs in more than the specified number of times, the new user session is allowed access, but the oldest user session is locked out from further access. By locking out the oldest session instead of the newest session.

Sensitive Data Exposure - Essential Guideline

Sensitive Data Exposure occurs when an application does not adequately protect sensitive information. This vulnerability allows an attacker to access sensitive data such as credit cards, tax IDs, authentication credentials, etc to conduct credit card fraud, identity theft, or other crimes. Losing such data can cause severe business impact and damage to the reputation.

Source : https://www.owasp.org/index.php/Top_10-2017_A3-Sensitive_Data_Exposure
https://www.kiuwan.com/blog/owasp-top-10-2017-a3-sensitive-data-exposure-identify-your-weaknesses/

Critically Sensitive Data

Password Disclosure

- https://hackerone.com/reports/288638 https://hackerone.com/reports/322988
- https://hackerone.com/reports/738

Private API Keys

An API Key and Secret Key are secret tokens used to authenticate you with servers. API Key and Secret Key are only known to you and the server. It is important to keep both keys confidential to protect your account.

- https://hackerone.com/reports/124100 https://hackerone.com/reports/196655

User Enumeration

User enumeration is when a malicious actor can use brute-force to either guess or confirm valid users in a system. User enumeration is often a web application vulnerability, though it can also be found in any system that requires user authentication. Two of the most common areas where user enumeration occurs are in a site's login page and its 'Forgot Password' functionality.

- https://hackerone.com/reports/43269 https://hackerone.com/reports/250457
- https://hackerone.com/reports/223531 https://hackerone.com/reports/335427

Source : https://blog.rapid7.com/2017/06/15/about-user-enumeration/
https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002)

Visible Detailed Error/Debug Page

Detailed Server Configuration

You can view detailed configuration information for the content server that a backed-up Web site was located on at the time of the backup. This information includes the custom metadata fields, content types, security groups, configuration and environment settings, installed components, defined providers, and so on. This information is especially useful when you must restore a site and want to configure the content server so that it matches the server where the site originated.

- https://hackerone.com/reports/297339
- https://hackerone.com/reports/183548
- https://hackerone.com/reports/318603

https://hackerone.com/reports/201901
https://hackerone.com/reports/318603
https://hackerone.com/reports/287837

Source : https://www.owasp.org/index.php/Insecure_Configuration_Management

Full Path Disclosure

Full Path Disclosure refers to being able to see the full path of something hosted on the server. It's often, but not always, the location of the script itself.
Hackerone Reports :

- https://hackerone.com/reports/26825
- https://hackerone.com/reports/159481
- https://hackerone.com/reports/230098

https://hackerone.com/reports/210572
https://hackerone.com/reports/87505

Source : https://www.owasp.org/index.php/Full_Path_Disclosure
Source : https://support.detectify.com/customer/en/portal/articles/2243201-full-path-disclosure

Descriptive Stack Trace

A descriptive stack trace describes the sequence of functions and events that led to a certain point in the execution of a program. To put it even more simply, when web app explodes, it tells you where it exploded and what steps it followed that led to the explosion. You can effectively use it as a map to debug code.

Hackerone Reports :

- https://hackerone.com/reports/41469
- https://hackerone.com/reports/46366

https://hackerone.com/reports/222108
https://hackerone.com/reports/221833

Source : https://medium.com/@jgefroh/the-dreaded-stack-trace-2692c053b28e
Source : https://www.owasp.org/index.php/Testing_for_Stack_Traces_(OTG-ERR-002)

Token Leakage via Referer

The reset password token is leaking through the HTTP referer header ... This happens when user clicks at the link sent to their email and when the page is rendered with the token at the URL.

There are currently two priority levels for Token Leakage via Referer, the first is P4 when the token is being sent over HTTP, the second is a P5 when the token is sent over HTTPS. I think these should be consolidated into a P4 as the risk here is that the token is going to end up in the logs of the destination server.
Hackerone Reports :

- https://hackerone.com/reports/738
- https://hackerone.com/reports/47140
- https://hackerone.com/reports/272379

https://hackerone.com/reports/5691
https://hackerone.com/reports/66626
https://hackerone.com/reports/265740

Source : https://robots.thoughtbot.com/is-your-site-leaking-password-reset-links
Source : https://portswigger.net/kb/issues/00500400_cross-domain-referer-leakage

Sensitive Token in URL

Information exposure through query strings in URL is when sensitive data is passed to parameters in the URL. This allows attackers to obtain sensitive data such as usernames, passwords, tokens (authX), database details, and any other potentially sensitive data.

Source : https://portswigger.net/kb/issues/00500700_session-token-in-url
Source : https://www.owasp.org/index.php/Information_exposure_through_query_strings_in_url

Hackerone Reports :

- https://hackerone.com/reports/83667
- https://hackerone.com/reports/341372

https://hackerone.com/reports/295461
https://hackerone.com/reports/151058

Weak Password Reset Implementation

The web app contains a mechanism for users to recover or change their passwords without knowing the original password, but the mechanism is weak. It is common for an application to have a mechanism that provides a means for a user to gain access to their account in the event they forget their password.

Source : https://www.owasp.org/index.php/Forgot_Password_Cheat_Sheet
https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)

Password Reset Token Sent Over HTTP

Hackerone Reports :

- https://hackerone.com/reports/15412
- https://hackerone.com/reports/214571

https://hackerone.com/reports/158186

Cleartext Transmission of Sensitive Data

Hackerone Reports : https://hackerone.com/reports/173268

Directory Listing Enabled

A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers. Exposing the contents of a directory can lead to an attacker gaining access to source code or providing useful information for the attacker to devise exploits, such as creation times of files or any information that may be encoded in file names. The directory listing may also compromise private or confidential data.

Hackerone Reports :

- https://hackerone.com/reports/307666
- https://hackerone.com/reports/110655

https://hackerone.com/reports/175760

Disclosure of Known Public Information

Hackerone Reports : https://hackerone.com/reports/378122

Mixed Content (HTTPS Sourcing HTTP)

This means that the URL in question is loaded over a secure HTTPS protocol, but some other resources (such as images, videos, stylesheets, scripts) are loaded over an insecure HTTP connection.

Hackerone Reports :

- https://hackerone.com/reports/85541 https://hackerone.com/reports/108692
- https://hackerone.com/reports/146707

Source : https://resources.infosecinstitute.com/https-mixed-content-vulnerability/
Source : https://sitebulb.com/hints/security/mixed-content-loads-http-resources-on-https-url/

Sensitive Data Hardcoded

Web app contains hardcoded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

If hard-coded passwords are used, it is almost certain that malicious users will gain access to the account in question. This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code.

OAuth Secret

Hackerone Reports :

- https://hackerone.com/reports/5314 https://hackerone.com/reports/5786

File Paths

This one is particularly dangerous, because the filename and path are hardcoded in the webapp. An attacker can modify index page path for ./dir/dir/, open the file and see something like MD5 digests and can be brute forced (with enough time) or dictionary cracked by a malicious user, thus giving administrator access to the forum.

Hackerone Reports : https://hackerone.com/reports/291200

Source : https://cwe.mitre.org/data/definitions/798.html

Internal IP Disclosure

In some cases, web servers may be prone to sharing internal IP addresses in response to specially crafted queries. Discovering the private addresses used within an organization can help an attacker in carrying out network-layer attacks aiming to penetrate the organization's internal infrastructure.

Hackerone Reports :

- https://hackerone.com/reports/253429 https://hackerone.com/reports/329791
- https://hackerone.com/reports/330716 https://hackerone.com/reports/42780
-

Source : https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed

JSON Hijacking

JSON Hijacking is similar to CSRF(Cross Site Request Forgery) but there is just a little bit difference, In CSRF you trick the victim/user to do some malicious/unwanted activity but in JSON Hijacking you trick the user to access a crafted link which will read some data form victim account and pass it to attacker.

Hackerone Reports : https://hackerone.com/reports/54034

Source : https://www.thesecuritybuddy.com/vulnerabilities/what-is-json-hijacking-or-javascript-hijacking/

Cross Site Script Inclusion (XSSI)

Browsers prevent pages of one domain from reading pages in other domains. But they do not prevent pages of a domain from referencing resources in other domains. In particular, they allow images to be rendered from other domains and scripts to be executed from other domains.

An included script doesn't have its own security context. It runs in the security context of the page that included it. For example, if www.evil.example.com includes a script hosted on www.google.com then that script runs in the evil context not in the google context. So any user data in that script will "leak."

Hackerone Reports :

- https://hackerone.com/reports/118631 https://hackerone.com/reports/138270

Source : https://www.scip.ch/en/?labs.20160414

Insufficient Security Configurability - Essential Guideline

It occurs when a webapp permits an attacker to access sensitive content or functionality without having to properly authenticate. Strong authentication, encryption and security logging are not available. Weak passwords, a lack of multifactor authentication and application vulnerabilities create risks.

Source : https://www.owasp.org/index.php/Top_10_2014-I8_Insufficient_Security_Configurability

Weak Password Policy

The weakness occurs when the application does not check complexity or minimum length of the provided passwords. Entire security of application depends on its authentication mechanism. Weak password requirements allow users to create weak passwords, susceptible to a variety of attacks.

Hackerone Reports :

- https://hackerone.com/reports/17160 https://hackerone.com/reports/255668
- https://hackerone.com/reports/276123

Source : https://cwe.mitre.org/data/definitions/521.html
Source : https://www.owasp.org/index.php/Testing_for_Weak_password_policy_(OTG-AUTHN-007)

Weak Password Reset Implementation

The password change and reset function of an application is a self-service password change or reset mechanism for users. This self-service mechanism allows users to quickly change or reset their password without an administrator intervening.

Hackerone Reports :https://hackerone.com/reports/8082

Source : https://cwe.mitre.org/data/definitions/640.html
https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG-AUTHN-009)

Token is Not Invalidated After Use

- https://hackerone.com/reports/283550 https://hackerone.com/reports/265775

Token is Not Invalidated After Email Change

Hackerone Reports : https://hackerone.com/reports/223461

Token is Not Invalidated After Password Change

- https://hackerone.com/reports/220185 https://hackerone.com/reports/244642

Token is Not Invalidated After New Token is Requested

- https://hackerone.com/reports/176116 https://hackerone.com/reports/272839

Lack of Verification Email

Hackerone Reports : https://hackerone.com/reports/90643

Lack of Notification Email

Hackerone Reports :

- https://hackerone.com/reports/280519
- https://hackerone.com/reports/282572

https://hackerone.com/reports/42403

Weak Registration Implementation

Hackerone Reports :https://hackerone.com/reports/263846

Weak 2FA Implementation

- https://hackerone.com/reports/214433

https://hackerone.com/reports/121696

Unvalidated Redirects and Forwards - Essential Guideline

Web app frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Hackerone Reports :

- https://hackerone.com/reports/387007 https://hackerone.com/reports/190188
- https://hackerone.com/reports/179328

Source : https://hdivsecurity.com/owasp-unvalidated-redirects-and-forwards
Source : https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet

Tabnabbing

Reverse tabnabbing is an attack where a page linked from the target page is able to rewrite that page, for example to replace it with a phishing site. As the user was originally on the correct page they are less likely to notice that it has been changed to a phishing site, especially it the site looks the same as the target.

Hackerone Reports :

- https://hackerone.com/reports/220737 https://hackerone.com/reports/306414

Source : https://www.owasp.org/index.php/Reverse_Tabnabbing

Server Security Misconfiguration - Essential Guideline

Improper server or web application configuration leading to various flaws Debugging enabled, Incorrect folder permissions, Using default accounts or passwords, Setup/Configuration pages enabled. Current web app security architectures do not follow security by default. On the contrary, programmers must apply security measures to avoid access to private or confidential resources.

Source : https://www.owasp.org/index.php/Top_10-2017_A6-Security_Misconfiguration

Using Default Credentials

These default username and password combinations are widely known by penetration testers and malicious attackers, who can use them to gain access to various types of custom, open source, or commercial applications.

Hackerone Reports :

- https://hackerone.com/reports/192074 https://hackerone.com/reports/174883
- https://hackerone.com/reports/398797

Source : https://www.owasp.org/index.php/Testing_for_default_credentials_(OTG-AUTHN-002)
https://www.owasp.org/index.php/Testing_for_Default_or_Guessable_User_Account_(OWASP-AT-003)

Misconfigured DNS

Subdomain Takeover

Subdomain takeover vulnerabilities occur when a subdomain (subdomain.example.com) is pointing to a service (e.g. GitHub pages, Heroku, etc.) that has been removed or deleted. This allows an attacker to set up a page on the service that was being used and point their page to that subdomain.

For example, if subdomain.example.com was pointing to a GitHub page and the user decided to delete their GitHub page, an attacker can now create a GitHub page, add a CNAME file containing subdomain.example.com, and claim subdomain.example.com.

- https://hackerone.com/reports/294201 https://hackerone.com/reports/202767

Source : https://0xpatrik.com/takeover-proofs/
Source : https://0xpatrik.com/subdomain-takeover-basics/
Source : https://www.hackerone.com/blog/Guide-Subdomain-Takeovers

OAuth Misconfiguration

OAuth is an open standard for authorization, commonly used as a way for Internet users to log in to third party websites using their Microsoft, Google, Facebook, Twitter, One Network etc. accounts without exposing their password.

Account Takeover

Hackerone Reports :

- https://hackerone.com/reports/131202
- https://hackerone.com/reports/3930

Insecure Redirect URI

Hackerone Reports : https://hackerone.com/reports/270028

Mail Server Misconfiguration

Missing SPF records are a common and long-standing security issue that puts sensitive information at risk. We found that less than half of those domains have configured email authentication correctly to prevent spoofed emails being sent from their domains, which means that users are at risk of receiving false emails appearing to come from domains that they trust.

Source : https://blog.detectify.com/2016/06/20/misconfigured-email-servers-open-the-door-to-spoofed-emails-from-top-domains/

Missing SPF on Email Domain

- https://hackerone.com/reports/324372
- https://hackerone.com/reports/57736

Missing SPF on Non-Email Domain

- https://hackerone.com/reports/92740
- https://hackerone.com/reports/325734
- https://hackerone.com/reports/54779

Database Management System (DBMS) Misconfiguration

The database typically contains the crown jewels of any environment; it usually holds the most business sensitive information which is why it is a high priority target for any attacker.

Hackerone Reports :

- https://hackerone.com/reports/200818
- https://hackerone.com/reports/198292
- https://hackerone.com/reports/179751
- https://hackerone.com/reports/23098

Source : https://a-lign.com/common-database-vulnerabilities-and-misconfigurations
https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2014/july/top-10-common-database-security-issues/

Lack of Password Confirmation

Delete Account

Hackerone Reports : https://hackerone.com/reports/42403

Change Email Address

Hackerone Reports : https://hackerone.com/reports/245334

Change Password

Hackerone Reports : https://hackerone.com/reports/546

No Rate Limiting on Form

Registration Hackerone Reports : https://hackerone.com/reports/275186

Login https://hackerone.com/reports/280389 https://hackerone.com/reports/138863

Email-Triggering

- https://hackerone.com/reports/224927 https://hackerone.com/reports/280534
- https://hackerone.com/reports/170310

Directory Listing Enabled

The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.

Source : https://cwe.mitre.org/data/definitions/548.html

Sensitive Data Exposure

An attacker can see the files located in the directory and could potentially access files which disclose sensitive information. Like attacker access system enabled directory.

- https://hackerone.com/reports/110655 https://hackerone.com/reports/223384
- https://hackerone.com/reports/175760

Non-Sensitive Data Exposure Hackerone Reports : https://hackerone.com/reports/371464

Same-Site Scripting

Hello, I'd like to document what appears to be a common named misconfiguration that can result in a minor security issue with web applications.

It's a common and sensible practice to install records of the form "**localhost. IN A 127.0.0.1**" into nameserver configurations, bizarrely however, administrators often mistakenly drop the trailing dot, introducing an interesting variation of Cross-Site Scripting (XSS) I call Same-Site Scripting.

The missing dot indicates that the record is not fully qualified, and thus queries of the form "**localhost.example.com**" are resolved. While superficially this may appear to be harmless, it does in fact allow an attacker to cheat the RFC2109 (HTTP State Management Mechanism) same origin restrictions, and therefore hijack state management data.

Hackerone Reports :

- https://hackerone.com/reports/1509 https://hackerone.com/reports/7085
- https://hackerone.com/reports/7949

Source : https://www.securityfocus.com/archive/1/486606
Source : https://www.soom.cz/clanky/1158--Same-Site-Scripting-SSS

Unsafe File Upload

Arbitrary code execution is possible if an uploaded file is interpreted and executed as code by the recipient. This is especially true for .asp and .php extensions uploaded to web servers because these file types are often treated as automatically executable, even when file system permissions do not specify execution. For example, in Unix environments, programs typically cannot run unless the execute bit is set, but PHP programs may be executed by the web server without directly invoking them on the operating system.

Source : https://resources.infosecinstitute.com/file-upload-vulnerabilities/
Source : https://www.owasp.org/index.php/Test_Upload_of_Malicious_Files_(OTG-BUSLOGIC-009)

File Extension Filter Bypass | No Antivirus

Hackerone Reports :

- https://hackerone.com/reports/305237 https://hackerone.com/reports/27704
- https://hackerone.com/reports/713

No Size Limit

- https://hackerone.com/reports/390 https://hackerone.com/reports/263109
- https://hackerone.com/reports/127995

Exposed Admin Portal - To Internet

Do not enable access to your management interface from the internet or from other untrusted zones inside your enterprise security boundary. This applies whether you use the dedicated management port (MGT) or you configured a data port as your management interface.

Hackerone Reports : https://hackerone.com/reports/297339

Source : https://resources.infosecinstitute.com/dangers-web-management/
https://www.owasp.org/index.php/Enumerate_Infrastructure_and_Application_Admin_Interfaces_(OTG-CONFIG-005)

Lack of Security Headers

Whenever a browser requests a page from a web server, the server responds with the content along with HTTP Response Headers. Some of these headers contain content meta data such as the content-encoding, cache-control, status error codes, etc.

In many cases they are very easy to implement and only require a slight web server configuration change. HTTP security headers provide yet another layer of security by helping to mitigate attacks and security vulnerabilities.

Source : https://geekflare.com/http-header-implementation/
Source : https://www.keycdn.com/blog/http-security-headers
Source : https://www.thesslstore.com/blog/http-security-headers

-

Content-Security-Policy | X-Content-Security-Policy | X-Webkit-CSP

The content-security-policy HTTP header provides an additional layer of security. This policy helps prevent attacks such as Cross Site Scripting (XSS) and other code injection attacks by defining content sources which are approved and thus allowing the browser to load them.

Example : content-security-policy: script-src 'self' https://www.google-analytics.com

Source : https://www.keycdn.com/support/content-security-policy
Source : https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

X-XSS-Protection

The x-xss-protection header is designed to enable the cross-site scripting (XSS) filter built into modern web browsers. This is usually enabled by default, but using it will enforce it. It is supported by Internet Explorer 8+, Chrome, and Safari. Here is an example of what the header looks like.

Example : x-xss-protection: 1; mode=block

X-Frame-Options

The x-frame-options header provides clickjacking protection by not allowing iframes to load on your site. It is supported by IE 8+, Chrome 4.1+, Firefox 3.6.9+, Opera 10.5+, Safari 4+. Here is an example of what the header looks like.

Example : x-frame-options: SAMEORIGIN

Public-Key-Pins

The public-key-pins header tells the web browser to associate a public key with a certain web server to prevent MITM attacks using rogue and forged X.509 certificates. This protects users in case a certificate authority is compromised. Here is an example of what the header looks like.

Example : public-key-pins: pin-sha256="t/OMbKSZLWdYUDmhOyUzS+ptUbrdVgb6Tv2R+EMLxJM="; pin-sha256="PvQGL6PvKOp6Nk3Y9B7npcpeL40twdPwZ4kA2IiixqA="; pin-sha256="ZyZ2XrPkTuoiLk/BR5FseiIV/diN3eWnSewbAIUMcn8="; pin-sha256="0kDINA/6eVxlkns5z2zWv2/vHhxGne/W0Sau/ypt3HY="; pin-sha256="ktYQT9vxVN4834AQmuFcGlSysT1ZJAxg+8N1NkNG/N8="; pin-sha256="rwsQi0+82AErp+MzGE7UliKxbmJ54lR/oPheQFZURy8="; max-age=600; report-uri="https://www.keycdn.com"
Source : https://scotthelme.co.uk/hpkp-http-public-key-pinning/

X-Content-Type-Options

The x-content-type header prevents Internet Explorer and Google Chrome from sniffing a response away from the declared content-type. This helps reduce the danger of drive-by downloads and helps treat the content the right way. Here is an example of what the header looks like.

Example : x-content-type-options: nosniff

Strict-Transport-Security

The strict-transport-security header is a security enhancement that restricts web browsers to access web servers solely over HTTPS. This ensures the connection cannot be establish through an insecure HTTP connection which could be susceptible to attacks.

Example : strict-transport-security: max-age=31536000; includeSubDomains; preload

Source : https://caniuse.com/#search=hsts
Source : https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security

Cache-Control for a Non-Sensitive Page

Content-Security-Policy-Report-Only

Example : Content-Security-Policy-Report-Only: default-src https:; report-uri /csp-violation-report-endpoint/

Source : https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy-Report-Only

Path Traversal

SSL Attack (BREACH, POODLE etc.)

All systems and applications utilizing the Secure Socket Layer (SSL) 3.0 with cipher-block chaining (CBC) mode ciphers may be vulnerable. However, the POODLE (Padding Oracle On Downgraded Legacy Encryption) attack demonstrates this vulnerability using web browsers and web servers, which is one of the most likely exploitation scenarios. Some Transport Layer Security implementations are also vulnerable to the attack.

Source : https://www.gracefulsecurity.com/tls-ssl-vulnerabilities/
Source : https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/

Rate Limit Brute Forcing OTP Vulnerability

Source :
https://www.owasp.org/index.php/OWASP_Periodic_Table_of_Vulnerabilities_-_Brute_Force_(Generic)_/_Insufficient_Anti-automation

Logical Bugs | Vulnerabilities

Hackerone Reports :

- https://hackerone.com/reports/165727 https://hackerone.com/reports/165727

Source :
https://medium.com/bugbountywriteup/bugbounty-i-dont-need-your-current-password-to-login-into-your-account-how-could-i-e51a945b083d

Same Site Scripting

It's a common and sensible practice to install records of the form "localhost. IN A 127.0.0.1" into nameserver. administrators often mistakenly drop the trailing dot. The missing dot indicates that the record is not fully qualified, and thus queries of the form "localhost.example.com" are resolved.

While superficially this may appear to be harmless, it does in fact allow an attacker to cheat the RFC2109 (HTTP State Management Mechanism) same origin restrictions, and therefore hijack state management data.

Hackerone Reports : https://hackerone.com/reports/1509

Source : https://seclists.org/bugtraq/2008/Jan/270

Missing Certification Authority Authorization (CAA) Record

DNS Certification Authority Authorization (CAA) is an Internet security policy mechanism which allows domain name holders to indicate to certificate authorities whether they are authorized to issue digital certificates for a particular domain name. It does this by means of a new "CAA" Domain Name System (DNS) resource record.

Practical | How to test :

- https://caatest.co.uk
- https://dns.google.com

Hackerone Reports : https://hackerone.com/reports/261706

Source : https://www.ssl.com/article/certification-authority-authorization-caa
Source : https://www.globalsign.com/en-in/blog/what-is-certificate-authority-authorization-checking

Email Spoofing to Spam Folder

Misconfigured email servers open the door to spoofed emails from top domains

Hackerone Reports : https://hackerone.com/reports/263508

Source : https://support.google.com/a/answer/7490901?hl=en

Missing or Misconfigured SPF and/or DKIM

DMARC – Domain-based Message Authentication, Reporting and Conformance is a technology that uses the resources of DNS and email servers to help avoid email abuse—specifically, phishing. It is layered over two technologies (SPF and DKIM) that allow for the specification of policies for incoming email. Publishing a DMARC record is one of the Requirements;

SPF – Sender Policy Framework is a technology that allows an administrator to publish information about legitimate sending hosts in a specially formatted DNS resource record;

DKIM – DomainKeys Identified Mail is a technology that allows a mail receiver to check that incoming mail from a domain is authorized by that domain's administrators and that the email has not been changed as it has gone through the network.

Practical | How to test :

- https://www.dmarcanalyzer.com/spf/checker
- https://www.dmarcanalyzer.com/dkim/dkim-check
- https://www.dmarcanalyzer.com/dmarc/dmarc-record-check

Hackerone Reports : https://hackerone.com/reports/60260

Source : https://geekflare.com/fix-email-spoofing-missing-spf-record-vulnerability
Source : https://blog.detectify.com/2016/06/20/misconfigured-email-servers-open-the-door-to-spoofed-emails-from-top-domains/

Session Cookie Scoped to Parent Domain

This session cookie is scoped to the parent domain instead of a sub-domain. If a cookie is scoped to a parent domain, then this cookie will be accessible by the parent domain and also by any other sub-domains of the parent domain. This could lead to security problems.

Source : https://portswigger.net/kb/issues/00500300_cookie-scoped-to-parent-domain

Missing Secure or HTTPOnly Cookie Flag

If the HttpOnly attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Hackerone Reports :

- https://hackerone.com/reports/75357 https://hackerone.com/reports/58679

Source : https://resources.infosecinstitute.com/securing-cookies-httponly-secure-flags

CAPTCHA ByPass Vulnerability

CAPTCHA ("Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge-response test used by many web applications to ensure that the response is not generated by a computer. CAPTCHA implementations are often vulnerable to various kinds of attacks even if the generated CAPTCHA is unbreakable.

Hackerone Reports :

- https://hackerone.com/reports/210417 https://hackerone.com/reports/246801

Source : https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-008)
Source : https://www.owasp.org/index.php/Testing_for_Captcha_(OWASP-AT-012)

Exposed Admin Portal | IP Address Disclosure

Administrator interfaces may be present in the application or on the application server to allow certain users to undertake privileged activities on the site. Tests should be undertaken to reveal if and how this privileged functionality can be accessed by an unauthorized or standard user.

Hackerone Reports :

- https://hackerone.com/reports/128114 https://hackerone.com/reports/329791

Source : https://www.owasp.org/index.php/Enumerate_Infrastructure_and_Application_Admin_Interfaces_(OTG-CONFIG-005)

Missing DNSSEC

DNSSEC is a cryptographic security extension to the DNS protocol. The Domain Name System (DNS) translates domain names into IP addresses (and vice versa). If, say, you want to visit example.com, your computer (the client) has to ask a name server for the IP address of example.nl's web server, which will be something like 192.0.2.36 (IPv4) or 2001:db8::2:14 (IPv6).

E-mail and other internet protocols use the same system. With DNSSEC, a digital signature is attached to the DNS information (records) that the server sends to the client, so that the client can check their authenticity.

Practical | How to test :

- http://dnsviz.net
- https://en.internet.nl
- https://dnssec-analyzer.verisignlabs.com

Hackerone Reports : https://hackerone.com/reports/169704

Source : https://www.sidn.nl/faq/dnssec?language_id=2
Source : https://www.internetsociety.org/deploy360/dnssec/tools

Brute Force Username Enumeration

User enumeration is when a malicious actor can use brute-force to either guess or confirm valid users in a system. User enumeration is often a web application vulnerability, though it can also be found in any system that requires user authentication. Two of the most common areas where user enumeration occurs are in a site's login page and its 'Forgot Password' functionality.

- https://hackerone.com/reports/335427 https://hackerone.com/reports/250457

Source : https://blog.rapid7.com/2017/06/15/about-user-enumeration
Source : https://www.owasp.org/index.php/Testing_for_User_Enumeration_and_Guessable_User_Account_(OWASP-AT-002)

Potentially Unsafe HTTP Method Enabled - OPTIONS | TRACE

HTTP offers a number of methods that can be used to perform actions on the web server. Many of theses methods are designed to aid developers in deploying and testing HTTP applications. These HTTP methods can be used for nefarious purposes if the web server is misconfigured. Additionally, Cross Site Tracing (XST), a form of cross site scripting using the server's HTTP TRACE method, is examined.

Hackerone Reports :

- https://hackerone.com/reports/119860 https://hackerone.com/reports/191220

Source : https://www.owasp.org/index.php/Test_HTTP_Methods_(OTG-CONFIG-006)
Source : https://www.sans.org/reading-room/whitepapers/testing/penetration-testing-web-application-dangerous-http-methods-33945

Lack of Forward Secrecy

Perfect Forward Secrecy is a feature of specific key agreement protocols that gives assurances your session keys will not be compromised even if the private key of the server is compromised. By generating a unique session key for every session a user initiates, even the compromise of a single session key will not affect any data other than that exchanged in the specific session protected by that particular key. Perfect Forward Secrecy represents a huge step forwards in protecting data on the transport layer and following on from Heartbleed, everyone using SSL/TLS should be looking to implement it.

Forward secrecy further protects data on the transport layer of a network that uses common SSL/TLS protocols, including OpenSSL, which had previously been affected by the Heartbleed exploit. If forward secrecy is used, encrypted communications and sessions recorded in the past cannot be retrieved and decrypted should long-term secret keys or passwords be compromised in the future, even if the adversary actively interfered, for example via a man-in-the-middle attack.

- https://hackerone.com/reports/44294 https://hackerone.com/reports/32570

Practical | How to test : https://safeweb.norton.com/heartbleed

Source : https://scotthelme.co.uk/perfect-forward-secrecy

Insecure Cipher Suite

A cipher suite is a combination of authentication, encryption and message authentication code (MAC) algorithms. All of which are used during the negotiation of security settings for a TLS/SSL connection as well as for the secure transfer of data.

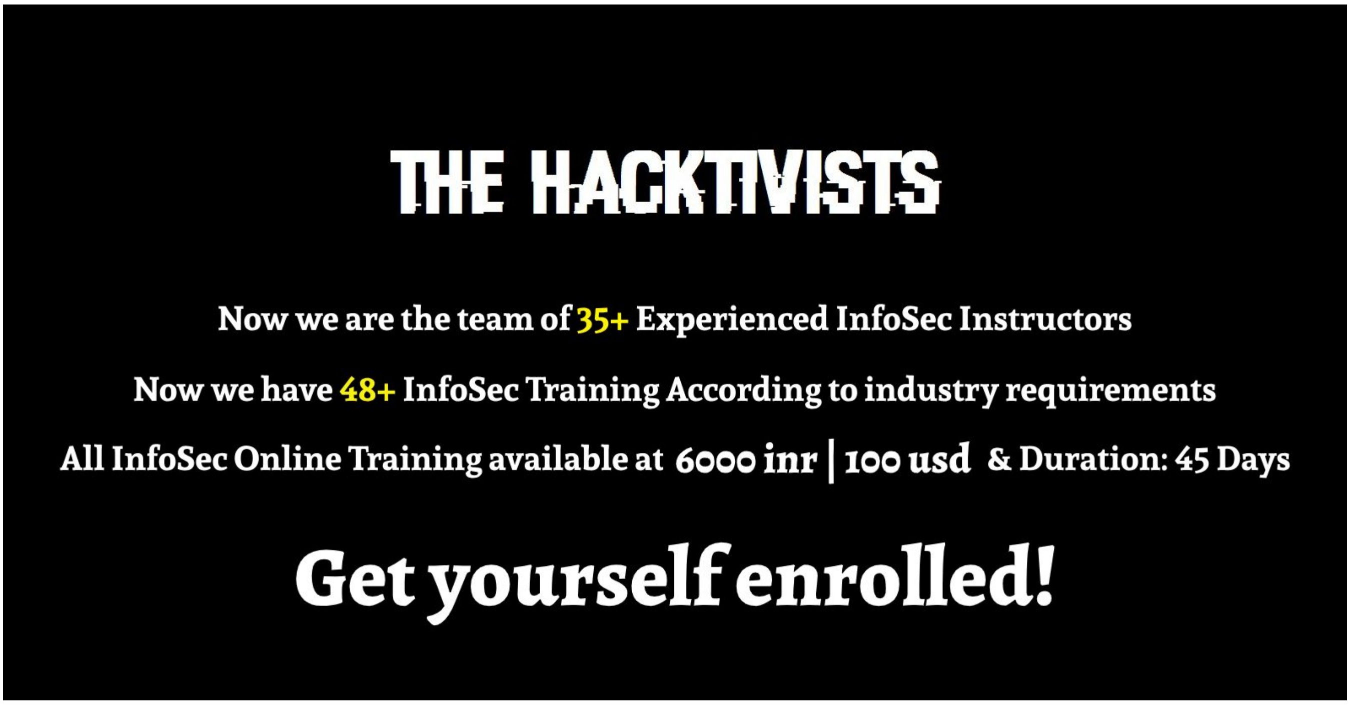
Practical | How to test :

-

Hackerone Reports :

- https://hackerone.com/reports/194761 https://hackerone.com/reports/99157

Source : https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening
Source : https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_(OTG-CRYPST-001)



THE HACKTIVISTS

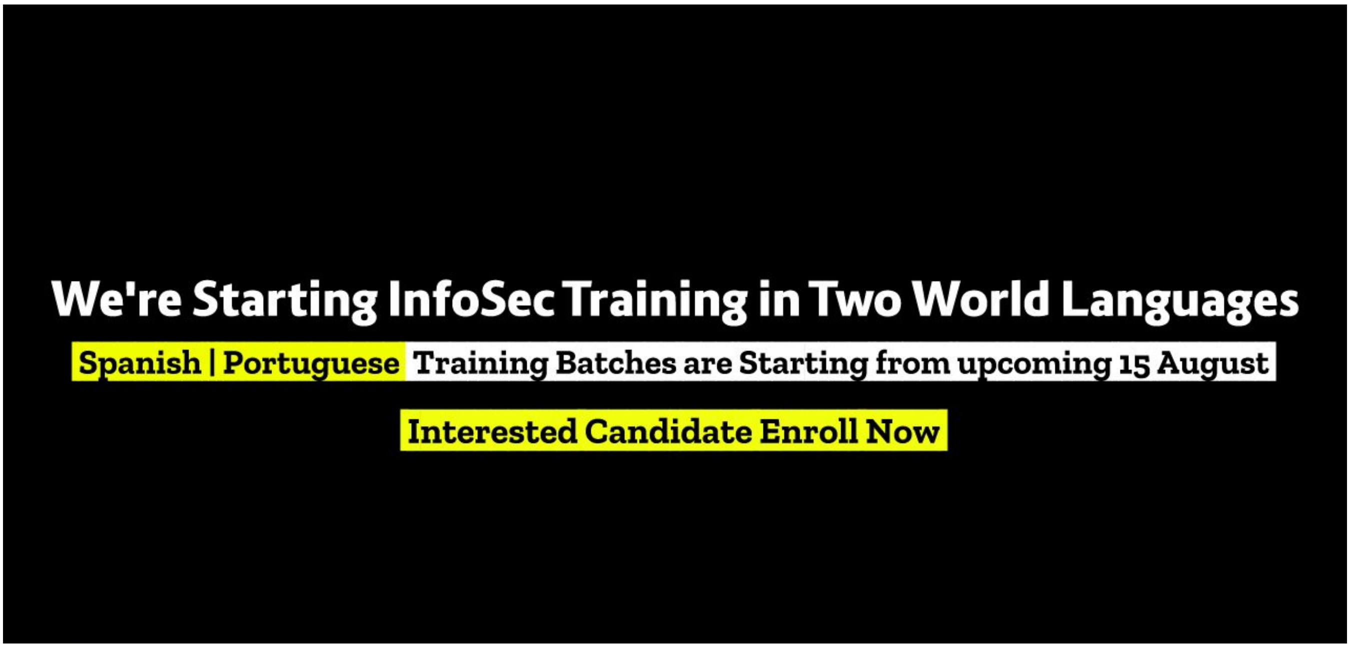
Now we are the team of 35+ Experienced InfoSec Instructors

Now we have 48+ InfoSec Training According to industry requirements

All InfoSec Online Training available at 6000 inr | 100 usd & Duration: 45 Days

Get yourself enrolled!

Call | WhatsApp : +91 96809 81337   Training #Fee : 6000 inr | 100 usd



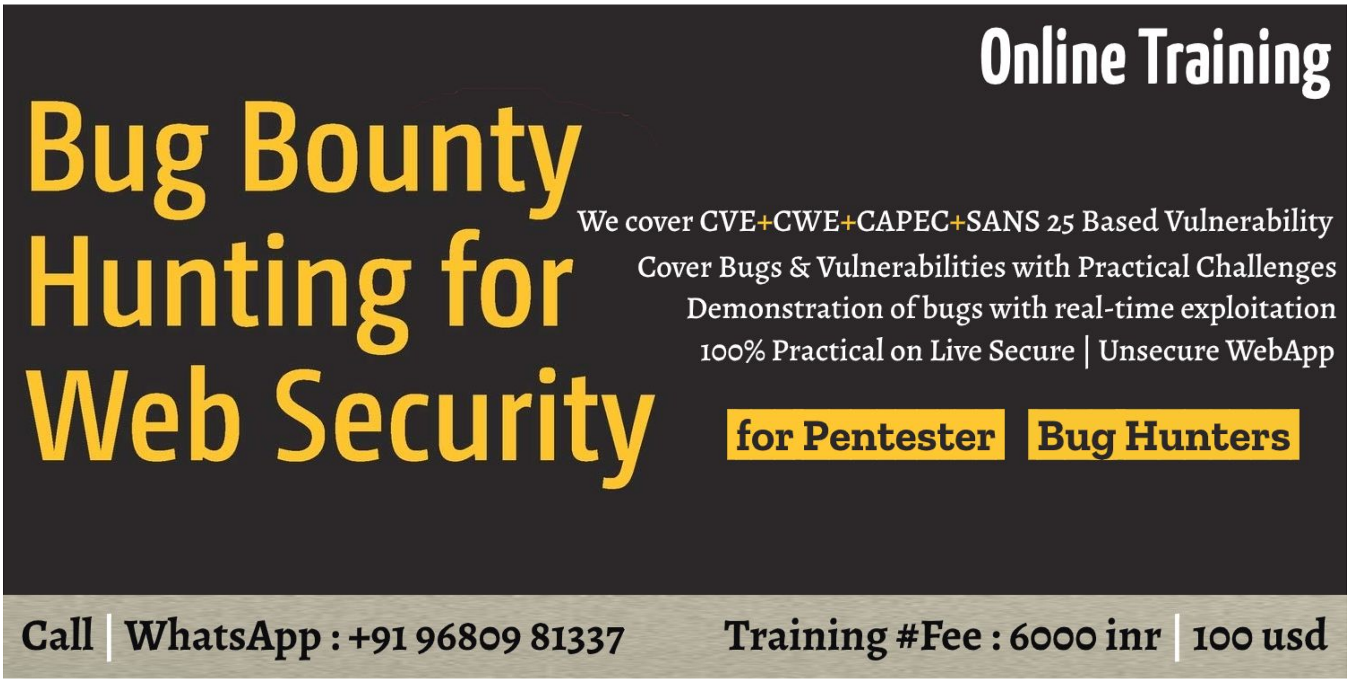
We're Starting InfoSec Training in Two World Languages

Spanish | Portuguese Training Batches are Starting from upcoming 15 August

Interested Candidate Enroll Now

Bug Bounty Hunting | WebApp Pentest Training

Live Websites Practice



Online Training

Bug Bounty Hunting for Web Security

We cover CVE+CWE+CAPEC+SANS 25 Based Vulnerability
Cover Bugs & Vulnerabilities with Practical Challenges
Demonstration of bugs with real-time exploitation
100% Practical on Live Secure | Unsecure WebApp

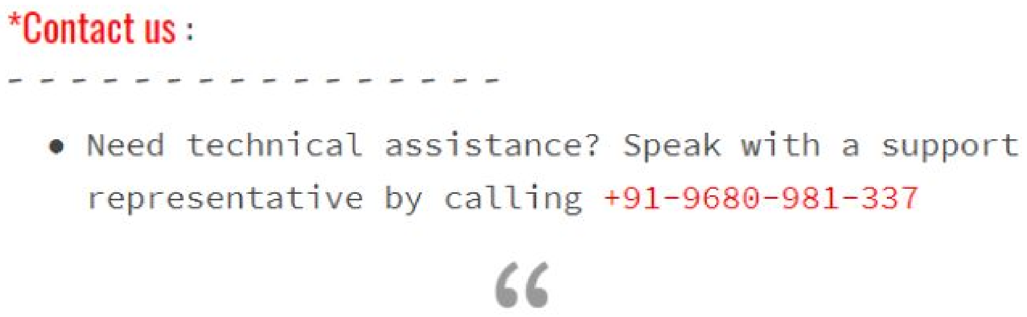
for Pentester | Bug Hunters

Call | WhatsApp : +91 96809 81337 Training #Fee : 6000 inr | 100 usd

Download All The Hacktivists™ InfoSec Training Modules

https://1drv.ms/f/s!Ah6mcJeP80hdgRcEE-HkUl45GL7_

http://www.mediafire.com/folder/xxq1ttcg29hk9/The_Hacktivists™_Training_Program



*Contact us :

- - - - - - - - - - - - - - - - -

- Need technical assistance? Speak with a support representative by calling +91-9680-981-337

“