Web hosting policies and guidelines

Objective:- To formulate an approach to provide a trusted, reliable, and secured web presence to the public and help affiliates use the web hosting and network access in a secure manner, preserving their privacy and data confidentiality.

Web and network access policies

Data collection and privacy: Capurting network and system data pertinent to individual users (e.g. IP addresses, network traffic, credentials, credit card information, DNS queries etc.) is not permitted. Access to such data for research and training purposes is subject to approval from appropriate competent authorities like Institutional Review Board (IRB).

Impersonation:- Attempting to impersonate anyone using stolen access credentials, hosting TLS certificates, authenticator nonces etc. is prohibited not only by the IIITD network administrator but is a punishable offence under section 66D of the IT act of the Government of India.

TLS:- All services hosted MUST use valid TLS X.509 certificates. In case of hosting using subdomains of iiitd.ac.in, the IIITD's certificate MUST be used. For subdomains of iiitd.edu.in, you may use any valid certificate (e.g. those issued by LetsEncrypt).

Adverse network activity:- IIITD discourages the use of network or system activities that adversely affect the ability of other people to use the Intranet services or access the Internet. Activities for research purposes, that could potentially hinder Intranet services or access to the Internet must be approved by competent authorities like the Instutional Review Board (IRB). In such cases, such activities must be thoroughly discussed with the IT team to ensure that all appropriate precautions are taken to prevent disruption of normal services.

General:- It may be necessary for IT staff to examine system accounting logs and other records to resolve system problems. The IT reserves the right to access any account's directories to investigate and/or resolve system problems. In addition, the IIITD will cooperate with the appropriate legal authorities in investigating claims if illegal activity including but not limited to illegal transfer or use of copyrighted material, postings, or other illegal activity.
The IT does not monitor the activity of accounts except for measurements of system utilization and other Internet statistics, such as directory size, system hits, etc. However, in our efforts to promote good citizenship within the Internet community, if we become aware of illegal use of the IT services, we will respond. Our preferred course of action would be to advise the account owner of the inappropriate behavior and corrective action necessary. However, flagrant violations will be reported to higher authorities and could result in immediate termination of service.

Technical guidelines:-

KISS principle (Keep it Simple and Stupid) – At IIITD, we encourage everyone to use simple as possible hosting infrastructure and setups. For any kind of hosting the hoster must use one module/VM/program should be doing exactly one thing -- VMs hosting services should not be used for any other purpose, do not run tons of unnecessary services on server VMs (only the essential), etc. E.g. in case of web applications, a VM/host should be used for web server hosting alone. A separate VM must be used for hosting backend databases that MUST not be accessible via public IP addresses.

Hardware/OS:-

Hardware – VM - 4GB RAM , 2 Cores. [Preferably VMs] (frontend).

Backend: Could be VM/physical machine

- OS – Devuan (using SysVInit) , FreeBSD
- Non-systemd, running sysvinit, repo compatible with Ubuntu/Debian.

[End users can use whichever OS they prefer. Nothing changes]

Guidelines for frontend hosting:-

Infrastructure and access:

- **Web server**: Apache, Nginix.
- **Open ports:** 443 (for hosting) , 22 (for server management)
- **Precautions:**
  - Ensure only relevant HTTP modules are loaded at startup.
  - HTTP directory should have no other files than those required for web site rendering.
  - HTTPD must not run as root. Can run as other users like www/www-user.
  - HTTP directory must be owned by www/www-user.
  - Ensure that www/www-user cannot login.
- **Traffic firewalling:** DROP (not DENY/REJECT) traffic to any other port. You may use linux iptables or FreeBSD ipfw for the firewalling.
- Enable TCP SYN cookies to prevent SYN floods.
- **SSH for server management:**
  Should be only accessible from within IIITD network (local firewalling – ipfw/iptables)
- *Prefer public-key authentication over passwords.*

Where and how to host:

- Sites on iiitd.ac.in domain – MUST be hosted inside IIITD, with procured wildcard certificate of IIITD (contact the IT team for the same).
- As mentioned above, TLS certificate is a must. For those on iiitd.ac.in subdomain, one MUST use IIITD's official certificate.

User authentication, authorization, access controls and input validation:

- Input validation – For user portal hostings, please take care of the following for authentication of usernames and passwords. These fields must be of fixed lengths, and the program must check if the input exceeds the chosen lengths or not. The password field must involve hashing the password before being sent to the site. Additionally, CAPTCHA may be used to prevent brute force attempts using scripts.

  Additionally, if the front end site connects to a back end database, then it needs to ensure the following:

  1. Inputs must be bounded in their lengths.
  2. Inputs must not have characters like quote marks, apostrophe and double hyphen characters or special keywords like "UNION/union". These are vectors of SQL injection attacks.
  3. Ideally for every input special fixed length queries (predetermined/prestructured) must be passed to the back end database.

- Delays between failed login attempts: A deliberate delay must be added between subsequent incorrect login attempts, regardless of the username being bruteforced. This increases the time for the attacker to be able to successfully guess the login credentials.
  - Additionally, a transient account lockout may be implemented after three consecutive failed attempts for a short duration (e.g. 5 mins, but may be configurable by the site administrator).

- **User credential authentication (especially for portals)**: Users logging into portals are supposed to be authenticated using domain credentials. The portals need to be integrated with the domain credential authentication. Do contact the IT team for assistance regarding the same.
  - Portals, in addition to user credentials (username and passwords), MUST use google authenticator (Time based OTP) to authenticate users to specific portals. The hoster must validate user inputs (as described above) for such

scenarios as well. Do contact the IT team for assistance regarding the same.
    - **DO NOT use OAuth based authentication. It provides little to no user authentication.**
    - User sessions must be timed out if idle for more than some fixed durations (e.g. 15 mins).

Backend database hosting/management guidelines:

DB infrastructure and access:

Hardware/OS:-

Hardware – VM - 4GB RAM , 2 Cores. [Preferably VMs] (frontend).
Backend: Could be VM/physical machine
• OS – Devuan (using SysVInit) , FreeBSD
• Non-systemd, running sysvinit, repo compatible with Ubuntu/Debian.
[End users can use whichever OS they prefer. Nothing changes]

- **VM/Host:** The backend database MUST be hosted on a separate VM/host from the one hosting the front end.

- **DB Server:**Mysql, PostgreSQL, MongoDB, SQLite, Cassandra etc.
- **Open ports:** SQL server appropriate port (e.g. 3306 for Mysql) (for hosting), 22 (for server management).
- **Precautions:**
    - SQL server must NEVER have a publicly reachable IP address.
    - SQL server MUST NOT be run as root. Can run as other users like mysql.
    - SQL server should ONLY accept connections from localhost (same machine) and from the webserver, and no other process.
    - Ensure that SQL user cannot login.
    - SQL server should ONLY accept connections from localhost (same machine) and from the web server, and no other host/process.

- **SSH for server management:**
    - Should be only accessible from within IIITD network (local firewalling – ipfw/iptables)
    - *Prefer public-key authentication over passwords.*

- **Traffic firewalling:** DROP (not DENY/REJECT) traffic to any other port. You may use linux iptables or FreeBSD ipfw for the firewalling.
- Enable TCP SYN cookies to prevent SYN floods.

-

Final checklist

Parameters	**Compliance (Yes/No)**
Hosting Machine:- 1. **For institute hosted servers:** Must be a VM with 4 core and about 4GB RAM unless there is a need for much more capacity. 2. Frontend HTTP server and backend databases must be hosted on separate hosts/VMs. Compromise of one should not trivially.	
Services/OS:- 1. **OS:-** Devuan (not Ubuntu, Debian, Arch, Suse, Fedora, Centos) 2. **Web:-** Apache or nginix 3. **Database server:** Mysql, PostgreSQL, MongoDB, SQLite, Cassandra etc. 4. **Web server:** Ensure that ONLY those modules are loaded that are required, no more no less. a. The HTTP directory should have NO files or directories that are not required for the service to run. Only the bare bones files. b. Run HTTP server NOT as root but as www/www-user user. The HTTP directory should be owned by www/www-user and allow only read and write access to user and group. No access to other users should be allowed. c. Only port 443 should be enabled. No other service running on any other port, except SSH (TCP Port 22) for server management. No other services to be enabled unless required. d. Domain certificates: For sites on iiitd.ac.in, use the official IIITD domain wildcard certificate. For others, those on iiitd.edu.in, you may use other certificates like LetsEncrypt. Certificates are however mandatory.	

5. **SQL database server:**
    a. SQL server must NEVER have a publicly reachable IP address.
    b. SQL server MUST NOT be run as root. Can run as other users like mysql.
    c. SQL server should ONLY accept connections from localhost (same machine) and from the webserver, and no other process.
    d. Ensure that SQL user cannot login.
    e. SQL server should ONLY accept connections from localhost (same machine) and from the web server, and no other host/process.
    f. No other services to be enabled unless required.
    g. Use IPTables to allow traffic to only 80/443 (for HTTP/HTTPS) and 3306 (or any other appropriate port for the SQL server) and 22.
    h. Connections to the latter (*i.e.* SQL server port must be allowed only from the web server, using a private IP address. The SQL server MUST listen on ONLY a private IP address and it must be connected to from only the web server.
    i. Like the HTTPD server must also run with specific user, and NOT ROOT
    j. DROP everything else (no DENY/REJECT but DROP).

6. **SSH (For management):**
    a. Use IPtables firewall rules to allow only specific machines to access and administer.

    b. Use a public-private key pair to access and administer it and avoid using passwords for user authentication.

<u>Service management/scripts/applications:-</u>

If using backend data, the front end HTTP server could use the following strategies to avoid threats like SQL injection, directory enumeration etc.

1. The front end scripts should construct queries and only add specific arguments from user inputs.
2. The front end must validate the input – *e.g.* ensure that the username is never the administrator for the queries that involve user names/IDs, the inputs should be of fixed lengths and there should be no apostrophe and double hyphen symbols in the inputs.
3. The connection to the DB (HTTPD ←→ DB server) must be encrypted/protected using TLS. The DB server may use self-signed certificates for the same.

<u>User authentication for portals:-</u> 1. User portals must not be using OAuth authentication. 2. The portals must take input usernames and passwords from users and pass it on to the backend LDAP authentication framework managed by IT. 3. Additionally, the user portals need to also designed to accept Time based OTP (TOTP), associated with google 2FA authenticator that needs to be passed to the the backend authentication framework. 4. User passwords must be NOT be sent in plaintext, but must be hashed before sending. 5. Handling sessions: a. Session idle-timeouts must be implemented for user portals (*e.g.* one may choose a 10 or 15 minute idle-timeout to terminate sessions). b. Everytime a login attempt fails, the portal could add a certain delay so as to make bruteforce harder. c. Multiple failed login attempts should lead to transient account lockout.	

Final Notes:-

This document is subject to ongoing review and modifications.