# Module 2: Essential Problem-Solving Skills for Web Hosting

This module covers essential problem-solving skills for web hosting professionals. It includes logical thinking in troubleshooting, debugging skills, analytical and critical thinking, and decision making in web hosting scenarios.

DK **by Dan K**

# 2.1 Logical Thinking in Troubleshooting

### Identifying the issue

What exactly is happening?

### Breaking it into subproblems

Is it server-related, network-related, or application-related?

### Examining dependencies

Is the issue affecting a specific website, email, or database?

### Narrowing down the scope

Is it occurring for all users or only some?

**Breaking Down Problems into Smaller Components**

Logical thinking involves analyzing problems step by step rather than tackling everything at once. The key approach includes:

**Example:**

If a website is down:

1. Check if the server is running.
2. Verify web server logs for errors.
3. Test database connections.
4. Check DNS resolution.
5. Is the service Active?

**Structuring Solutions Systematically**

A structured troubleshooting approach follows a repeatable process:

1. Gather information.
2. Identify possible causes.
3. Test solutions one by one.
4. Monitor the impact of each fix.

# 2.2 Debugging Skills for Web Hosting

### Apache Logs

/var/log/apache2/error.log

### Nginx Logs

/var/log/nginx/error.log

### PHP Error Logs

/var/log/php_errors.log

### MySQL Error Logs

/var/log/mysqld.log

**Using Error Logs (Apache, Nginx, PHP, MySQL)**

Log files are a primary source of information when troubleshooting hosting problems.

**Example:**

If a WordPress site is showing a **500 Internal Server Error**, checking */var/log/apache2/error.log* may reveal:

- PHP Fatal error: Allowed memory size of 134217728 bytes exhausted

**Solution:** Increase PHP memory limit in *php.ini* or wp-config.php.

# Browser Developer Tools for Frontend Issues

## Chrome DevTools

(F12 or right-click → Inspect)

**What to Check?**

- Console Errors (JavaScript issues)
- Network Tab (Check if resources are loading properly)
- Elements Tab (See how the page is structured)

**Example:**

If images are not loading:

- Open **Network Tab** → Check for **404 errors**.
- If missing, check file paths and permissions.

**Command-Line Utilities for Troubleshooting**

- **ping** – Checks network connectivity
- **dig / nslookup** – Checks DNS resolution
- **traceroute** – Identifies network routing issues

**Example:**

To check if a domain resolves correctly:

- *dig* **example.com**

If it returns an incorrect IP, check DNS settings.

## Firefox Developer Tools

# 2.3 Analytical & Critical Thinking

1. **Evaluate Possible Causes**
   Analyze logs, error messages, and past incidents
2. **Identify Potential Solutions**
   Based on the evaluated causes
3. **Test Solutions**
   Apply potential fixes and monitor results
4. **Implement Best Solution**
   Choose the most effective and sustainable fix

**Evaluating Possible Causes of Failures**

Instead of assuming, analyze logs, error messages, and past incidents.

**Example:**

If email is not being delivered, possible causes:

- SMTP service is down or Misconfigurations
- Wrong DNS (MX) records.
- Blacklisted IP.

**Solution:** Check mail logs, verify MX records, and test sending emails manually.

# Prioritizing Solutions Based on Impact and Urgency


### High Impact, High Urgency

Immediate action required

### High Impact, Low Urgency

Plan for near-term resolution

### Low Impact, High Urgency

Quick fixes if possible

### Low Impact, Low Urgency

Address in regular maintenance

**High-impact, quick fixes first.**

**Preventative solutions for recurring issues.**

Example:

- A critical website outage needs **immediate** fixing.
- A slow website may be optimized **later**.

# 2.4 Decision Making in Web Hosting


**Choosing the Best Approach for Resolving Issues**

Factors to consider:

- **Speed** – Quick fixes vs. long-term solutions.
- **Risk** – Avoid solutions that might break other services.
- **Sustainability** – Implement solutions that prevent future problems.

**Example:**

If a site is slow due to a heavy database query:

- **Quick fix:** Increase server resources.
- **Best solution:** Optimize database queries and caching.

# Risk Assessment & Mitigation Strategies

Before applying changes, assess risks:

- Backup data before making major changes.
- Test solutions in a staging environment first.
- Monitor after implementation.

DK **by Dan K**

# Research & Brainstorming Techniques

### How to Find Solutions Using Online Resources

Search for specific error messages.

Use documentation from hosting control panels.

Check forums like StackOverflow and cPanel Community.

## Example:

If an SSL certificate is not working:

- Search *"cPanel AutoSSL not working"* in Google.
- Read discussions on cPanel forums.

# Collaboration in Troubleshooting

### Internal team brainstorming

Discussing with colleagues.

### Forums & communities

Posting detailed questions.

### Vendor support

Contacting hosting providers for assistance.

# Teamwork & Communication

## Working with Teams to Resolve Technical Issues

- Assign tasks (one person checks logs, another tests solutions).
- Use documentation to ensure continuity.

## Explaining Complex Problems in Simple Terms

When communicating with non-technical users:

- Avoid jargon.
- Use analogies (e.g., "Think of DNS like a phonebook").

# Emotional Intelligence & Adaptability

## Staying Calm Under Pressure

- Break down the problem logically.
- Avoid panic-driven solutions.

## Adapting to Unexpected Hosting Challenges

- Be ready to learn new technologies.
- Adjust approaches based on changing circumstances.

# Hands-On Troubleshooting Scenarios

## Scenario 1: Website Downtime Issue

**Problem:** A client reports that their website is down.

### Troubleshooting Steps:

Check if the server is running:

*You can ping the server to confirm it is online*

*ping unknown link or ping 11.22.33.44*

Check if the website resolves correctly:

*dig **example.com***

Check web server logs for errors:

*tail -f /var/log/apache2/error.log*

Fix identified issue (e.g., restart Litespeed if it's down):

*systemctl restart lsws*

**Solution:** If logs show a memory issue, increase memory allocation.

# Scenario 2: Email Not Sending from Server

**Problem:** A user cannot send emails from their cPanel hosting account.

## Troubleshooting Steps:

1. Check email queue for stuck messages:

   *exim -bp*

2. Verify mail logs for errors:

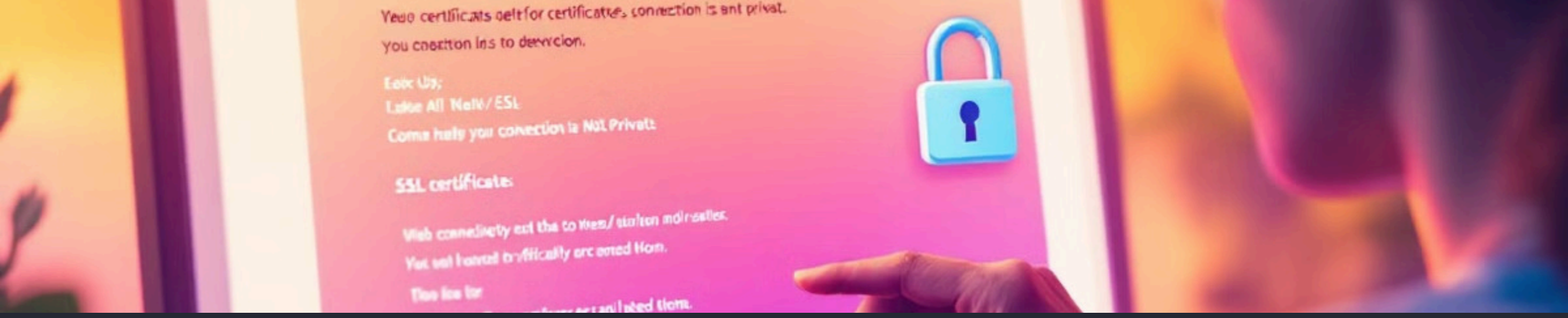
   tail -f /var/log/exim_mainlog

3. Check DNS MX records to ensure correct mail routing:

   *dig **example.com** MX*

4. Fix identified issue (e.g., unblock port 25 if blocked):

**Solution:** If the server is blacklisted, request delisting and set up SPF/DKIM/DMARC records.



# Scenario 3: SSL Certificate Error ("Your Connection is Not Private")

**Problem:** A website shows an SSL error after renewal.

## Troubleshooting Steps:

1. Check if the certificate is valid:

   *openssl s_client -connect **example.com:443***

2. Verify SSL installation in cPanel/Plesk or Panel in question.

3. Check Apache/Nginx configuration:

   *cat /etc/apache2/sites-enabled/example.conf*

4. Restart web server if changes were made:

   *systemctl restart apache2*

**Solution:** If the certificate expired, renew it using Let's Encrypt: