TroubleShooting

- Malware Cleanup
- Varnish Issues
- Misc Issue
- Database Tips
- Nginx Tips
- Common Problems
- Xdebug

Malware Cleanup

Malware is one of the most common issues with Wordpress Sites , this is mainly due to plugins which are not updates or simply bad user practices.Recommended approach to cleanup -

This guide assumes that wp cli is availbile -

--allow-root if logged in as root

Search

Check maldet logs if detection were made

```
cat /usr/local/maldetect/logs/event_log | grep hits
```

grep search based on signatures found

```
grep -r -e 'exp' --include=*.php
```

Delete files which have a string insdie them

```
grep -rlZ 'exp' . --include=*.ico | xargs -0 rm -f --
```

Reset Wordpress

Change directory to the WordPress install directory , In most cases the best way to start is check core files are infected and replace them if required , First Check if checksums are ok -

```
wp core verify-checksums --allow-root
```

Before changing any files, please get the wordpress version to know which version needs to be downloaded.

```
cat wp-includes/version.php | grep wp_version
```

Then remove the core files and reinstall wordpress (overwrite files only)

```
rm -rf wp-admin ; rm -rf wp-includes
rm -rf {SITEUID}
wp core download --force --skip-content --version= --allow-root
```

** You can also reset plugins **

```
wp plugin install {name} --force --allowroot
```

Scan with Third party plugin

As a third step we will use Wordfence to do a final scan to find files which we might have missed . Sometimes this will fail with cahcing so disable caching.

Delete files

Delete files with confirmation

```
find . -type f -name "*.php" -exec rm -i {} \;
```

Find all PHP usuall in uploads and delte them

```
find . -name "*.php" -type f {-delete}
```

Quick edit files

:n for next files , :N for prev file , dd to delte line , :wn to write and move to next

```
grep -rl pattern |  xargs -o vim
```

Myssql cache

Sometimes malware will insert meta tags in post_content (use query to replace)

```
UPDATE wp_posts SET post_content = replace(post_content, "<script
src='https://for.dontkinhooot.tw/stat.js?n=ns1'  type='text/javascript'></script>", "")
```

Rules

New Rules for nginx to disable php in wp-content

```
# Deny access to hidden files and files inside wp-content/uploads
location /wp-content/uploads/{
        location ~ \.php$ {
                deny all;
            }
}
```

```
  # Deny access to hidden files and files inside wp-content/uploads
location /wp-content/ {
        location ~ \.php$ {
                deny all;
            }
}
```

Varnish Issues

Varnish is used as the caching system at WPOven , Some quick Commands to see and check for errors

See top uncached url's , This will help see which urls are going through

```
varnishtop -i BereqURL
```

See cache hits , missis and nuked

```
varnishstat -1 | grep cache_
varnishstat -1 | grep nuke
```

See the misses and hits

```
varnishncsa -F '%t %r %s %b %{Varnish:time_firstbyte}x %{Varnish:handling}x'
varnishncsa -F '%t %r %s %b %{Varnish:time_firstbyte}x %{Varnish:handling}x' | grep miss
```

Trace 503 error

```
varnishlog -q 'RespStatus == 503' -g request
```

Misc Issue

Low Ram

Enable 2GB swap , if you see the services are being killed

```
fallocate -l 2G /swapfile
dd if=/dev/zero of=/swapfile bs=1M count=2048
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
```

Also make the number of child workes 2x the amout of RAM in www-data pool of php-fpm .

Home Path

To set home for user - inside public_html create **.bash_profile** inside the file put the home path as

```
echo "export HOME='/srv/www/my.writetextfast.com/home/'" >> ~/home/bash_profile
```

Cleanup

Find largest file -

```
find -type f -exec du -Sh {} + | sort -rh | head -n 5
```

Cleanup snaps folder

```
snap list --all | while read snapname ver rev trk pub notes; do if [[ $notes = *disabled* ]];
then snap remove "$snapname" --revision="$rev"; fi; done
```

Resize allocated disk

```
tune2fs -m1 /dev/sda
```

Optimize images

```
find . -type f -iname "*.jpg" -exec jpegoptim --max=85 --strip-all --all-progressive --
preserve-perms -p {} +
```

Clear comments

```
// Delete comments
wp comment delete $(wp comment list --status=spam --format=ids) --force

// Delete Pending
wp comment delete $(wp comment list --status=hold --format=ids) --force
```

No log but critial

Create a debug backtrace in mu-plugins , this will dump all messages -

```
add_filter(
'wp_php_error_message',
function( $message, $error ) {
        debug_print_backtrace();
        return $message;
},
10,
2
);
```

IPTables Dumpe

If you wish to reset ip tables to default. [Stackhttps://serverfault.com/a/1042478]

```
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
#Then flush the rules:
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD
```

Random dump

If you wisht o find where files are outputting code

```
$locations = [];
ob_start(function($buffer) {
  global $locations;
```

```
  $locations[] = debug_backtrace();
  return $buffer;
}, 1);

-- code --

print_r($locations);
```

Database Tips

Some common tricks for quick database optimizations

See what is slowing things down -

```
`show full processlist;
```

Creating Index in old databases

Sometimes adding index will give error when moving from 5.7 to 8 to change that you can set the sql_mode

```
SELECT @@SESSION.sql_mode; // See what is the current mode
// For ex for Date error
SET SESSION sql_mode =
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGI
NE_SUBSTITUTION';
ALTER TABLE `wp_posts` ADD INDEX `post_date_gmt` (`post_date_gmt`);
```

Upgradeing to InnoDB

Quick upgrade to InnoDB

```
SET @DATABASE_NAME = 'name_of_your_db';

SELECT  CONCAT('ALTER TABLE `', table_name, '` ENGINE=InnoDB;') AS sql_statements
FROM    information_schema.tables AS tb
WHERE   table_schema = @DATABASE_NAME
AND     `ENGINE` = 'MyISAM'
AND     `TABLE_TYPE` = 'BASE TABLE'
ORDER BY table_name DESC;
```

You will get the queries for all the databases , execute them to convert the tables. If an error regarding the data then set sql_mode . That should resolve the issue .

Manually Creating Wordpress Database and login

To list the users -

```
SELECT User, Host FROM mysql.user;
```

log into mysql and -

```
 CREATE DATABASE d_{name};
 CREATE USER u_{name}@localhost IDENTIFIED BY '{password}';
 GRANT ALL ON d_{name}.* TO u_{name}@locahost;
 flush privileges;
```

sometimes grant will require password (older versions)

```
GRANT ALL ON d_{name}.* TO u_{name}@locahost identifed by '{password}';

GRANT ALL PRIVILEGES ON  d_{name}.* TO u_{name}@locahost WITH GRANT OPTION;
```

Manual setting up wordpress database

```
 wp core install --title="Site Title" --url={domain_name} --admin_user=siteadmin --
admin_password={password} --admin_email={email} --skip-email --allow-root
```

Flush Tables issue

Run command from /srv/www

```
find -maxdepth 3 -name "wp-config.php" -exec cat {} \; | grep DB_USER
```

Search replace using -

```
(u_[A-Z0-9]+)
GRANT RELOAD ON *.* TO '\1'@'localhost';
```

Run it inside mysql and it should work now . GRANT reload can only be global.

Importing databases

Import Single Database

```
mysql -uroot -D d_J2MK3T -o d_J2MK3T < db_dump.sql
```

Intsall mssql client

https://docs.microsoft.com/en-us/sql/connect/php/installation-tutorial-linux-mac?view=sql-server-ver15 https://docs.microsoft.com/en-us/sql/connect/php/microsoft-php-drivers-for-sql-server-support-matrix?view=sql-server-ver15#supported-operating-systems

Get the base ready

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list | sudo tee
/etc/apt/sources.list.d/msprod.list
apt-get update
apt-get install mssql-tools unixodbc-dev
```

Now lets install PHP modules

```
apt install php-pear
apt install php7.4-dev
pecl install sqlsrv
pecl install pdo_sqlsrv
# Create the file sqlsrv.ini and pdo_sqlsrv.ini in modsavailbile

# Create links in config

ln -s /etc/php/7.4/mods-available/pdo_sqlsrv.ini 20-pdo_sqlsrv.ini
ln -s /etc/php/7.4/mods-available/sqlsrv.ini 20-sqlsrv.ini
```

SSL Bug : https://github.com/microsoft/msphpsql/issues/1112#issuecomment-609972220

Database Recovery

This is to be done if the innodb mysql crashes. First try to do things without putting the DB in recovery mode. If it does not work move to steps below.

Stop Mysql , to copy its files

```
service stop mysql
cp -rf /var/lib/mysql /var/lib/mysql.orig
```

Add **innodb_force_recovery = 1** in my.cnf under [mysqld] , you can increase value from 1 - 6 where mysql starts to work.

Dump all database names -

```
mkdir ~/recovery/
mysql -e 'show databases;' | grep -v information_schema | grep -v Database  >
~/recovery/database_list.txt
```

Next dump all databases -

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqldump --skip-lock-tables $db >
/mnt/temp_storage/backup/$db.sql;done
```

Now we drop the databases -

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqladmin drop $db ; done
```

Incase Database tale does not drop , please drop it manually , also move teh ibdata files

```
cd /var/lib/mysql
rm -rf database_name
mv /var/lib/mysql/ibdata1 /mnt/temp_storage/backup/
mv /var/lib/mysql/ib_logfile0 /mnt/temp_storage/backup/
mv /var/lib/mysql/ib_logfile1 /mnt/temp_storage/backup/
```

Remove **innodb_force_recovery = 1** from my.cnf

Start mysql and create the datrabases

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqladmin create $db ; done
```

Import all the databases

```
for db in `cat /mnt/temp_storage/database_list.txt`; do mysqldump $db <
/mnt/temp_storage/backup/$db.sql;done
```

Repair Databases if any

```
mysqlcheck --repair --all-databases
```

The above should fix the issue hopefully .

Database Recovery 2

This menthod is the last resort , Incase the datable is very large and the mysql dump keeps failing. We will manually delete the databases.

innodb_force_recovery = 6

This is readonly mode mainly for dumping and not command which mae changes will work.

Step 1 is to run mysql check and see which databases it is faling.

```
service mysql stop
mysqlcheck -c  -u root -p --all-databases
```

Mae a note of all databases which have errors , no move those databases out of the /var/lib/mysql directory .

```
mv d_AHDKIT ~/db_backup
mkdir d_AHDKIT
chown -R mysql:mysql d_AHDKIT
```

Now try starting mysql after removing innodb_force_recovery

```
service mysql start
```

Now drop the database with the errors.

```
DROP DATABASE d_AHDKIT;
```

Now you can recreate the database , grant priviliges to use and import from backup.

Nginx Tips

Multi Level SubDirectory Setup which is not a multisite.

If a multilevel subdirectory install is required , make sure /dir/mappedsite is a symbolic link . For single level the rewrite is not required. Also make sure that safe path is updated accordingly.

```
# If subdirectory is there match files inside the subdirectory
location /de/fernstudium {
        index index.php index.html index.htm;
        try_files $uri $uri/ /de/fernstudium/index.php?q=$uri&$args;
    }

# So that /de is not matched as a domain
rewrite ^/de(.?!(fernstudium)*)$ $1 last;
```

Subdirectory Setup with reverse proxy

This is used when the subdirectory is on a different server. wordpress homeurl and siteurl also to be updated.

```
location ~/news(.*)$ {
            proxy_set_header X-Real-IP  $remote_addr;
            proxy_set_header X-Forwarded-For $remote_addr;
            proxy_set_header Host news.erudera.com;
            proxy_pass http://172.104.155.54:80$1$is_args$args;
 }
```

Rate limiting

Limit request to php to slow down crawlers and other CPU eating bots . Limits to 1 req per second with a burst of 3 -

In the main file nginx.conf -

```
limit_req_zone $binary_remote_addr zone=phplimit:10m rate=1r/s;
limit_req_zone $http_cf_connecting_ip zone=cfphplimit:10m rate=1r/s;
```

```
limit_conn_status 429;
limit_req_status 429;
```

This can be put in files with php location block -

```
limit_req zone=phplimit burst=3 ;
limit_req zone=cfphplimit burst=3 ;
limit_req_dry_run off;
```

Rate Limiting in Nginx:https://www.nginx.com/blog/rate-limiting-nginx/

Ratelimiting info is availbile in logs

Common Problems

Some common issues which can be fixed with tools

Invalid permissions

Symptoms : Cannot upload , Cannot update plugins , Cannot update theme , Invalid permissions
Fix : Use the site "Fix Permissions" tool

Cannot login (No error message)

Disk is full , Wordpress will not login if disk is full. Clean disk using cleanup disk tool
Custom Wordpress login url , make sure caching excludes the url , or ask use to include the login keyword

Wordpress asking for FTP details

Usually happens when users does not have direct fucntionality availbile in the wp-config.php . Make sire FS direct is set in wp-config.php -

```
define('FS_METHOD','direct');
```

Xdebug

Install xdebug for version

```
apt-get install php7.4-xdebug
```

then edit the config file

```
vim /etc/php/7.4/fpm/conf.d/20-xdebug.ini
```

Values to be like

```
zend_extension=xdebug.so
xdebug.mode=trace
xdebug.output_dir=/srv/trace
```

Then in the code put in

```
xdebug_start_trace();
```

make it up locked , then run the code and monitor the trace (vim no wrap helps - :set nowrap)