







Knowledgebase (KB)






NeoSOFT®
TECHNOLOGIES

All Categories

-  HR Policies (40)
-  Employee Performance Management System (0)
-  Training (16)
-  Utilities (2)
-  Knowledge Sharing (15)
 -  Mobile (0)
 -  Others / Misc (6)
 -  Web (3)
 -  Database Guidelines
 -  Hardware - Software Performance Guidelines
 -  Hosting Guidelines
 -  HTML-CSS-JS Guidelines
 -  Neosoft Secure Coding Practices
 -  Security Guidelines

There are no categories to display in knowledge base.

Recently Viewed

-  Neosoft Secure Coding Practices
-  HTML-CSS-JS Guidelines
-  Hosting Guidelines
-  Hardware - Software Performance Guidelines
-  Database Guidelines

[Home](#) » [Group Categories](#) » [Knowledge Sharing](#)

Security Guidelines

Article Number: 124 | Rating: 4/5 from 2 votes | Last Updated: Wed, Mar 30, 2016 at 1:16 PM

Network Security

- Firewalls : A firewall is a network security system, either hardware- or software-based, that controls incoming and outgoing network traffic based on a set of rules. Acting as a barrier between a trusted network and other untrusted networks -- such as the Internet -- or less-trusted networks -- such as a retail merchant's network outside of a cardholder data environment -- a firewall controls access to the resources of a network through a positive control model. This means that the only traffic allowed onto the network defined in the firewall policy is; all other traffic is denied.
 - Always enable firewall ("iptables")
 - By Default the 'iptables' enabled in LINUX with all port blocked.
 - Block all unused ports on both TCP and UDP protocols
 - By Default done by 'iptables'
 - Port / Service for open access CMD : 'iptables -A INPUT -m state --state NEW -m tcp --dport 80 -j ACCEPT'
 - Recommended Firewall:
 - LINUX WHM. ConfigServer&Firewall(CSF)
 - CentOS < 7. IPTABLES
 - CentOS 7.x. FirewallD or IPTABLES
 - Do not use default ports for critical services such as SSH, FTP
 - Default Ports :
 - FTP = 21
 - SSH / SFTP = 22
 - MySQL / MariaDB= 3306
 - Telnet = 23
 - SMTP = 25
 - POP3 = 110
 - IMAP =143
- Use key based authentication for SSH instead of simple passphrase :

SSH server can authenticate clients using a variety of different methods. The most basic of these is password authentication, which is easy to use, but not the most secure. Although passwords are sent to the server in a secure manner, they are generally not complex or long enough to be resistant to repeated, persistent attackers. Modern processing power combined with automated scripts make brute forcing a password-protected account very possible. Although there are other methods of adding additional security (fail2ban, etc.), SSH keys prove to be a reliable and secure alternative. SSH key pairs are two cryptographically secure keys that can be used to authenticate a client to an SSH server. Each key pair consists of a public key and a private key. The private key is retained by the client and should be kept absolutely secret. Any compromise of the private key will allow the attacker to log into servers that are configured with the associated public key without additional authentication. As an additional precaution, the key can be encrypted on disk with a passphrase. The associated public key can be shared freely without any negative consequences. The public key can be used to encrypt messages that only the private key can decrypt. This property is employed as a way of authenticating using the key pair.

- **neo_dev@localhost\$ [Note: You are on local-host here]**
- neo_dev@localhost\$ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/jsmith/.ssh/id_rsa):[Enter key]

Enter passphrase (empty for no passphrase): [Press enter key]

Enter same passphrase again: [Press enter key]

Your identification has been saved in /home/jsmith/.ssh/id_rsa.

Your public key has been saved in /home/jsmith/.ssh/id_rsa.pub.

The key fingerprint is:

33:b3:fe:af:95:95:18:11:31:d5:de:96:2f:f2:35:f9 neo_dev@localhost

- **neo_dev@localhost\$ ssh-copy-id -i ~/.ssh/id_rsa.pub remote-host**

neo_dev@localhost's password:

Now try logging into the machine, with "ssh 'remote-host'", and check in: .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

- **neo_dev@localhost\$ ssh-copy-id -i remote-host**
- Recommend ping monitoring tools to end client such as Pingdom.
- Monitor the network traffic and data flood by automated tool.
 - Setup with PRTG, Cacti or MRTG but only possible with dedicated hosting server and set up by hosting service provider.
- 27 X 7 Monitoring and reporting the network activities

Server Level Security

- Follow checklist for the periodic audits
 - Check for [zombie processes](#)
 - **kill \$(ps -A -ostat,ppid | awk '/[zZ]/{print \$2}')**
 - Check the application log level that all should be on Critical or Error. NOT DEBUG.
 - Setup by each individual ".conf" files.
 - Use log archive with tools -**Logrotate** : logrotate is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large. Normally, logrotate is run as a daily cron job. It will not modify a log multiple times in one day unless the criterium for that log is based on the log's size and logrotate is being run multiple times each day, or unless the -f or -force option is used. Any number of conf files may be given on the command line. Later conf files may override

multiple times each day, or once an hour. The first option is usually better because it may be given on the command line and each config file may contain the options given in earlier files, so the order in which the logrotate config files are listed in is important. Normally, a single config file which includes any other config files which are needed should be used. See below for more information on how to use the include directive to accomplish this. If a directory is given on the command line, every file in that directory is used as a config file. If no command line arguments are given, logrotate will print version and copyright information, along with a short usage summary. If any errors occur while rotating logs, logrotate will exit with nonzero status.

- Let's say that we are running a service called "linuxserver" that is creating logfiles called "linux.log" within the /var/log/linuxserver directory. To include "linuxserver" log files in the log rotation we need to first create a logrotate configuration file and then copy it into the /etc/logrotate.d directory.

```
/var/log/linuxserver/linux.log {
rotate 7
daily
compress
missingok
notifempty
create 660 linuxuser linuxuser }
```

- Once config file is ready just simply copy it into logrotate directory and change owner and permissions:
 - cp linuxserver /etc/logrotate.d/
 - chmod 644 /etc/logrotate.d/linuxserver
 - chown root.root /etc/logrotate.d/linuxserver
 - logrotate -f /etc/logrotate.d/linuxserver
- 'Log' & 'tmp' mount point always separate from rest system.
- All Application & System must generate the logs at 'Log' Mount point only.
- Check logs, especially error logs for security issues.
- Prevent the various attack (network & server) by identified their logs. automated tool 'fail2ban' work with major types of firewall (software).
 - All config made by itself after Installing in System.
 - no need too much look after in 'Fail2Ban' conf.
- Use maldet (or similar tool) to scan for malicious scripts - directories, alerts, schedule
- Backups
 - Server Snapshot if possible
 - Database point-in-time backup or incremental backups
 - User file backup
 - Configuration backup
 - Done by manually
 - Apache = /etc/httpd/
 - mysql = /etc/my.cnf
 - All application's Conf files in "/etc"
- File System
 - Ownership of file system should be limited to correct user, root user should never be the owner of folders exposed on web server
 - File/Folder permissions should never be 777
 - Use following cron job to detect 777 file permissions and fix (PAI)
- Tools
 - maldetect
 - Chkrootkit
 - Fail2Ban
 - /etc/shadow/ & /etc/passwd should have 600 permissions
 - Block access to /lib/www/perl via HTACCESS
 - PHP Handler is always fastcgi or suPHP
 - Automated scanning tools: gollismero , ZED Attack Proxy ,prss
- Apache (Sharad)
 - Pre virtual host and post virtual host

Database Security

- Disable network access access e.g. Use Skip networking for MySQL
- Any server side database client such as **PhpMyAdmin** or **Adminer** shouldn't be accessible to general public over internet
- Grant Least Privileges: Use proper access control, grant only necessary privileges for each user
- If architecture permits, database server should be on separate server
- Use a profiler or monitoring tools such as NewRelic or performance sensitive application
- References (In additions to measures listed here, also follow these):
 - <https://www.giac.org/paper/gsna/128/auditing-mysql-database-server-independent-auditors-perspective/105994>
 - <http://www.kitebird.com/articles/ins-sec.html>

Application Security

- Front End Security
 - Validate for length, data type, format and range
 - Display error codes to users instead of actual errors
 - Use online tools for XSS
 - Pre virtual host and post virtual host
 - User XFrame headers to disable your website being embedded by others
 - Do not use /admin for dashboard/backend section of application, including frameworks, CMS, shopping carts.
- Server
 - Disable directory listing on server
 - Apache: Options -Indexes
 - All directories should have index.php or should be handled through index.php
 - Document root should be /home
 - Server signature should be off
 - FTP should be off, enable SFTP, disable shell if necessary

- Always use the latest version available of the server OS
- Ensure Security Testing against OWASP Top 10
 - Injection
 - Avoid use of “eval” function, do not use a string as a function call without validation
 - Use prepared statements or properly escaped strings for any SQL query
 - Broken Authentication and Session Management
 - Cross Site Scripting (XSS)
 - Insecure Direct Object Reference
 - Security Misconfiguration
 - Sensitive Data Exposure
 - Missing Function Level Access Control
 - Cross Site Request Forgery (CSRF)
 - Using Components with Known Vulnerabilities
 - Unvalidated Redirects and Forwards
- CAPTCHA required for admin user authentication
- Implement throttling for protection against DDoS , Dictionary or Brute Force attack
- PHP
 - phpinfo() should be disabled on production server
 - Do not use actual column names of table as GET or POST parameters
 - Use variables through \$_GET, \$_POST library function, Do not use \$_REQUEST. Preferably use request mechanism provided by the framework
 - Preferably use request mechanism provided by the framework
 - Never accept PHP code as an input from user, use sanitization function of different frameworks.
 - Missing Function Level Access Control
 - Never save sensitive data in Security Guidelines such as password, transaction information etc.
 - Verify file uploads
 - For images, generate image from uploaded image to verify validity
 - Check if MIME type matches with file extension
 - Ban sensitive functions on new projects. Never ban functions on live(created by someone else) sites, and ensure banned functions are not referenced in live code.
 - Use CAPTCHA whenever appropriate
 - Do not create users from commonly used name such as sales, or from the the emails used on the websites such as ‘sales’, ‘support’

Content Security Guidelines

- Do not store credit card or banking information of the user without PCI DSS compliant server and process
- We strongly recommend using HTTPS and strong cipher suite for any sensitive data

General Guidelines

- Use strong, randomly generated passwords in all layers of the system
- Change them often
- Perform quarterly security audit from third party
- Maintain logs of privileged users activities with timestamps

Posted by: PHP - Administrator - Wed, Mar 30, 2016 at 1:16 PM This article has been viewed 621 times.
Filed Under: Knowledge Sharing

Article Rating (2 Votes)



You've Already Voted.

Bookmark Article (CTRL-D)

Attachments

There are no attachments for this article.

Related Articles

- [Hosting Guidelines](#)
Viewed 249 times since Wed, Mar 30, 2016
- [Neosoft Secure Coding Practices](#)
Viewed 909 times since Wed, Mar 30, 2016
- [Hardware - Software Performance Guidelines](#)
Viewed 317 times since Wed, Mar 30, 2016
- [Database Guidelines](#)
Viewed 448 times since Wed, Mar 30, 2016
- [HTML-CSS-JS Guidelines](#)
Viewed 446 times since Wed, Mar 30, 2016