Module IV:

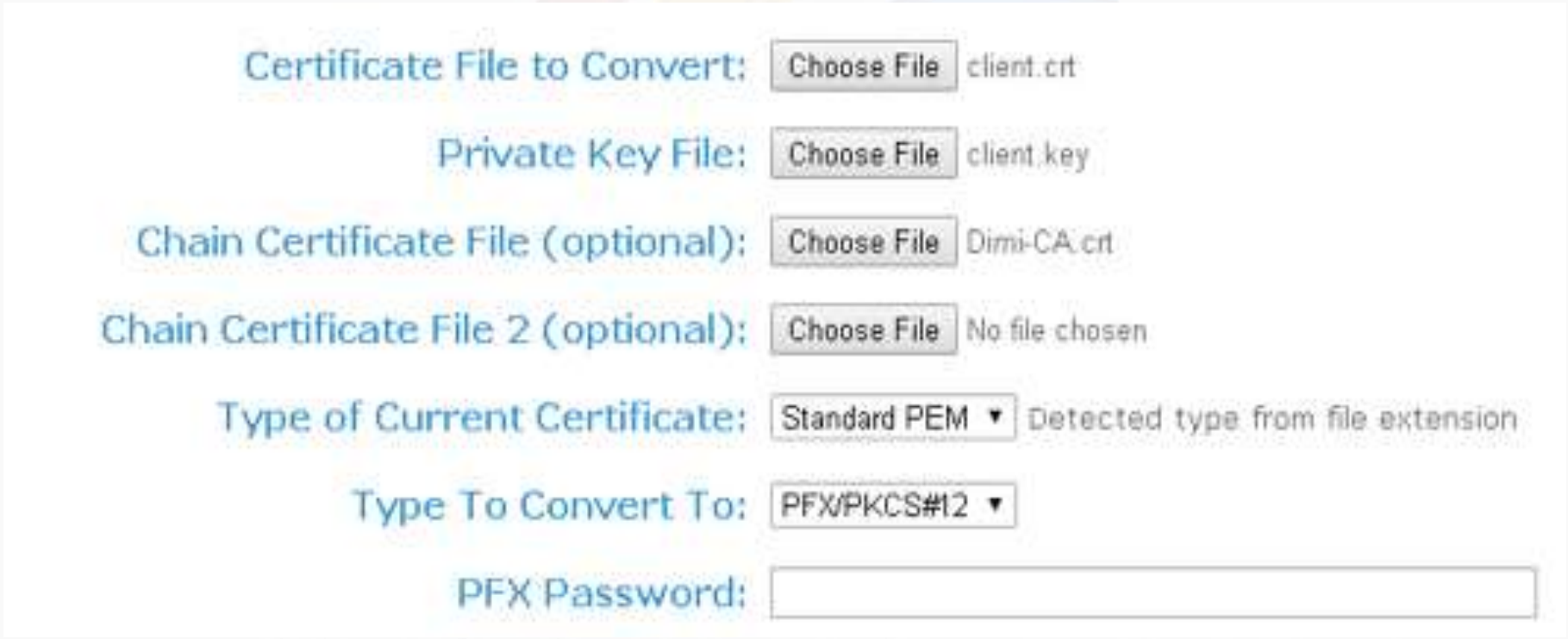
Protecting the Management Environment

SSL Client and Server Authentication:

Client Authentication

Clients access the Web application through HTTPS. HTTPS is terminated on the edge VIP and requests for client certificate.

1. Import the Web server certificate along with root CA.
2. Create the HTTPS application profile with the following parameters:
   - Select Type as **HTTPS** from the list.
   - Select the **Virtual Server Certificates** tab, and then select **CA Certificates** tab. CA is used to verify client certificate.
   - Select the service certificate configured in step 1.
   - Select Client Authentication as **Required** from the list.
3. Create a virtual server.
4. Import a client certificate signed by the root CA in the browser.
5. Go to the Web site https://www.sslshopper.com/ssl-converter.html.
6. Convert certificate and private key to the *pfx* file. For complete examples of certificate and private key, refer to the Example: Certificate and Private Key topic.
7. Import the *pfx* file in the browser.


Certificate File to Convert:
Choose File
client.crt
Private Key File:
Choose File
client.key
Chain Certificate File (optional):
Choose File
Dimi-CA.crt
Chain Certificate File 2 (optional):
Choose File
No file chosen
Type of Current Certificate:
Standard PEM
Detected type from file extension
Type To Convert To:
PFX/PKCS#12
PFX Password:

Server Authentication

Clients access the Web application through HTTPS. HTTPS is terminated on the edge VIP. The edge establish new HTTPS connections to the servers, it requests and verifies server certificate.

1. Import the Web server certificate and root CA certificate for server certificate authentication.
2. Create the HTTPS application profile with the following parameters:
    1. Select Type as **HTTPS** from the list.
    2. Select the **Enable Pool Side SSL** check box.
    3. Select the **Pool Certificates** tab, and then select **CA Certificates** tab. CA is used to verify client certificate from backend HTTPS server.
    4. Select the **Server Authentication** check box.
    5. Select the CA certificate configured in step 1.
    6. Select the required cipher from the **Ciphers** list.
3. Create a virtual server.

Authorization in vSphere:

you can allow some user to manage certain virtual environments without having access to the remaining virtual environments on the physical server and/or to the physical server itself or to complete only a restricted set of tasks in the virtual environment context (e.g. start, stop, and restart a virtual environment without having the right to back up this virtual environment or configure its resources).

To achieve this goal, a well-balanced user authentication and authorization strategy has been implemented. This strategy is based on the following main components:

- users;
- groups;
- permissions;
- roles;
- authentication databases.

Users are objects characterized by the *roles* delegated to them in a certain scope. *Users* can be members of *groups. Users* and *groups* can be retrieved either from local databases or from databases on external computers in your network. The information on these databases is stored on the physical server in the form of *authentication databases*. *Roles* are sets of abstract privileges that can be assigned to a *user* or a *group* to form a *permission*.

Permissions enable *users* or *groups* to perform certain operations in different scopes, which can be represented by one of the following entities:

- virtual environments;
- physical servers;
- logical units;
- server group.

Virtuozzo Automator allows you to manage any of the aforementioned components in the following way:

- View the users currently existing on the physical server, create a new user, edit its properties (e.g. add users to groups), and remove an existing user from the physical server.

- View the groups currently existing on the physical server, create a new group, edit its properties, and remove an existing group from the physical server.

- View the roles currently existing on the physical server, create a new role, edit its properties, and remove an existing role from the physical server.

- View the authentication databases currently existing on the physical server, create a new realm, set the default realm, and remove an existing realm from the physical server.

- Grant users permissions, i.e. define what rights the users will have within a physical server of virtual environment(s).

Managing Network Accounting and Shaping:

Apart from viewing the current state of affairs with the physical server traffic, the **Traffic** subtab allows you to do the following:

- Define the network classes for the physical server traffic by clicking the **Configure Accounting** button;
- Specify the bandwidth limit for the existing network interface cards by clicking the **Configure Interfaces** button;
- Set up the traffic shaping rules for each network interface card on the physical server by clicking the **Configure Rates** button;
- Enable traffic shaping for the physical server by clicking the **Enable Shaping** button;

1. Setting Up Network Classes
2. Configuring Network Adapters
3. Configuring Network Shaping

Infrastructure > physical server > **Network** tab > **Traffic** subtab

SSL Certificate:

SSL certificates are what enable websites to move from HTTP to HTTPS, which is more secure. An SSL certificate is a data file hosted in a website's origin server. SSL certificates make SSL/TLS encryption possible, and they contain the website's public key and the website's identity, along with related information. Devices attempting to communicate with the origin server will reference this file to obtain the public key and verify the server's identity. The private key is kept secret and secure.

SSL certificates include:

- The domain name that the certificate was issued for
- Which person, organization, or device it was issued to
- Which certificate authority issued it
- The certificate authority's digital signature
- Associated subdomains
- Issue date of the certificate
- Expiration date of the certificate
- The public key (the private key is kept secret)

Why do websites need an SSL certificate?

A website needs an SSL certificate in order to keep user data secure, verify ownership of the website, prevent attackers from creating a fake version of the site, and gain user trust.

Encryption: SSL/TLS encryption is possible because of the public-private key pairing that SSL certificates facilitate. Clients (such as web browsers) get the public key necessary to open a TLS connection from a server's SSL certificate.

Authentication: SSL certificates verify that a client is talking to the correct server that actually owns the domain. This helps prevent domain spoofing and other kinds of attacks.

HTTPS: Most crucially for businesses, an SSL certificate is necessary for an HTTPS web address. HTTPS is the secure form of HTTP, and HTTPS websites are websites that have their traffic encrypted by SSL/TLS.

Server Hardening:

Server Hardening is the process of enhancing server security through a variety of means which results in a much more secure server operating environment. This is due to the advanced security measures that are put in place during the server hardening process.

Server Hardening Tips & Tricks:

-Use Data Encryption for your Communications
- Avoid using insecure protocols that send your information or passwords in plain text.
- Minimize unnecessary software on your servers.
- Disable Unwanted SUID and SGID Binaries
- Keep your operating system up to date, especially security patches.
- Using security extensions is a plus.
-User Accounts should have very strong passwords
- Change passwords on a regular basis and do not reuse them
- Lock accounts after too many login failures. Often these login failures are illegitimate attempts to gain access to your system.
- Do not permit empty passwords.

Hardening the 4 main infrastructure layers:

Hardening activities can be classified into a few different layers:

- Server hardening
- Application hardening
- Operating System hardening
- Database hardening

Server hardening guidelines:

1. Implement a"least functionality" approach. for example: Do not install the IIS server on a domain controller
2. Install the appropriate post-Service Pack security hot fixes
3. Avoid installing applications on the server unless they are absolutely necessary to the server's function. For example, don't install e-mail clients, office productivity tools, or utilities that are not strictly required for the server to do its job
4. Use two different network interfaces in the server. One will be for the network and the other will be for the administrator
5. Create a secure remote administration for the server

6. Harden the OS and application layers
7. Consider using the server local firewall. Windows- Windows firewall, Linux-IPtables, AppArmor
8. Avoid the use of insecure protocols for processing requests, especially those that send information (i.e. passwords) in plain text
9. Keep a backup for all your data and files.
10. Secure separate partitions.
11. When hosting multiple applications, make sure that each has their own accounts separate from the others.
12. Never provide write access to web content directories.
13. Remove administrative shares if not needed.
14. Closely monitor failed login attempts. Lock accounts after a specified number of failures.
15. Rename the guest account even though it may be disabled.
16. Enable account lockout on the local administrator account
17. Rename the local Administrator account to something other than Administrator
18. Enforce strong account and password policies for the server.
19. Do not allow users and administrators to share accounts.
20. Disable FTP, SMTP , NNTP, Telnet services if they are not required.
21. Install and configure URLScan.

22. For non-public sites authentication methods should be put in place and for sites that are only to be accessible by internal users.
23. Web server logs should be reviewed routinely for suspicious activity. Any attempts to access unusual URLs on the web server typically indicate an attempt to exploit problems in outdated or unpatched web servers.
24. Domain Name Servers (DNS) provide the translation of human-friendly names for network destination (such as a web site URL) to the IP addresses understood by routers and other network devices.
25. Access to the server may be prevented by blocking port 53, or restricted by limiting access to the DNS server to one or more specified external systems.
26. Anonymous FTP accounts should be used with caution and monitored regularly.
27. In the case of authenticated FTP it is essential that Secure FTP be used so that login and password credentials are encrypted, rather than transmitted in plain text.

Application Hardening and Guidelines:

Application hardening is the process of securing applications against local and Internet-based attacks. Application hardening can be implemented by removing the functions or components that you don't require. We can restrict access and make sure the application is kept up-to-date with patches. Maintaining application security is very important because we need to make the application to be accessible to users. Most applications have problems of buffer overflows in the legitimate user input field so patching the application is the only way to secure it from attack. The following are some of the successfully proven application hardening guidelines:

1. Apply vendor provided patches in a timely manner for all 3rd party applications
2. For securing an IIS, the first step is to remove all simple files. To help the user in the setting of sample files, which can be used by the user to examine and as a reference when constructing their web sites. But these sample files are full of vulnerabilities and holes, so they should never be present on a production web server.
3. Sample files are stored in virtual and physical directories, so to remove IIS sample application, remove the virtual and physical directories. For example, IIS samples are present in the Virtual Directory of \IISS samples and its location is C:\Inetpub\IISsample.
4. The next step in securing IIS is to set up the appropriate permissions for the web server's file and directories this is possible using Access Control Lists (ACLs).

5. Avoid the use of insecure protocols for processing requests, especially those that send information (i.e. passwords) in plain text.
6. Never install IIS unless the server is to be a dedicated Web Server
7. Install SSL Architecture
8. Install and configure a web application firewall (WAF)
9. Avoid installing and do not run network device firmware versions that are no longer available from the manufacturer.
10. Closely monitor the security bulletins applicable to applications and other software used.
11. Use cryptographic and CHEKSUM controls wherever it is applicable.
12. Implement an Active directory which allows only single login to multiple applications, data sources, and system.

Database hardening guidelines:

Databases often store sensitive data. Incorrect data or loss of data could negatively affect business operations. Databases can be used as bases to attack other systems from. The following are some of the successfully proven database hardening guidelines:

1. Having a TNS Listener Password (encrypted) to prevent unauthorized administration of the Listener
2. Turning on Admin Restrictions to ensure certain commands cannot be called remotely
3. Turning on TCP Valid Node Checking allow certain hosts to connect to the database server and prevent others
4. Switching off XML Database if it is not used
5. Turning off External Procedures if not required
6. Encrypting Network Traffic using the Oracle Net Manager tool
7. Locking and Expiring Unused Accounts
8. Defining user account naming standards
9. Defining and Enforcing Password Policy
10. Role-based access control privileges

11. Periodic review and revoking of any unnecessary permissions
12. Enabling data protection for preventing users access sensitive tables
13. Ensuring usage of PL/SQL coding standard
14. Carrying out database security audits in a periodic manner
15. Disabling all the Null sessions (anonymous logons).
16. Rolling out all the necessary database patches as soon as released by the vendors.

Operating System hardening guidelines:

Operating System hardening is the process that helps in reducing the cyber-attack surface of information systems by disabling functionalities that are not required while maintaining the minimum functionality that is required. The following are some of the successfully proven operating system hardening guidelines:

1. Keep operating systems updated with the latest, most robust versions. Also, make sure that security patches and hotfixes are constantly updated.
2. Install the latest Service Pack for the operating systems used
3. Routers and wireless should be protected with strong passwords

4. Remove unnecessary drivers
5. Do not create more than two accounts in the Administrators group
6. Disable or delete unnecessary accounts quarterly
7. Disable Non-essential services
8. Enable Audit Logs to capture successful and failed login efforts, usage of elevated privileges and all kinds of unauthorized activities
9. Secure CMOS settings.
10. File and Directory Protection – Through the use of Access Control Lists (ACLs) and file permissions.
11. File and File System Encryption – All disk partitions are formatted with a file system type with encryption features (NTFS in the case of Windows)
12. The operating system is configured to log all activity, errors, and warnings.
13. Secure separate partitions.
14. Tighten NTFS/Registry Permissions
15. Configure appropriate settings for access control on file shares, given that permissions are set through NTFS security features
16. The operating system is configured to log all activity, errors, and warnings.
17. Disable any unnecessary file sharing
18. Remove administrative shares if not needed.
19. Ensure services are running with the least-privileged accounts.
20. Strong Password management

Identify Methods for Hardening Virtual Machines:

1. Installing Antivirus Software
2. Limiting Exposure of Sensitive Data Copied to the Clipboard
3. Removing Unnecessary Hardware Devices

   (device_name.allowGuestConnectionControl = "false“)
4. Limiting Guest Operating System Writes to Host Memory

   (tools.setInfo.sizeLimit; isolation.tools.setinfo.disable)
5. Configuring Logging Levels for the Guest Operating System

   (log.rotateSize=maximum_size; log.keepOld=number_of_files_to_keep)

Virtual Machine Security Architecture:

A virtual machine(VM) is a logical process (most often an operating system) that interfaces with emulated hardware and is managed by an underlying control program.
The VMM allows multiple virtual machines to be running at the same time and transparently multiplexes resources between them.

The VMM also isolates the virtual machines from one another, preventing them from accessing each other's memory or disk space. The operating system that runs inside of a virtual machine is traditionally referred to as the guest OS, and applications running on the guest OS are referred to as guest applications.

Types of VM Environments:

Type I VMM: runs directly on the physical hardware. It does not have an operating system running below it; the Type I VMM is fully responsible for scheduling and allocating of the system's resources between virtual machines

Type II VMM: runs as an application in a normal operating system. This operating system controls the real hardware resources, and is typically referred to as the "Host OS."
The host OS has no knowledge of the Type II VMM, which is treated like any other process in the system. The operating system that runs inside of the Type II VMM is referred to as the "Guest OS."

Virtual Environment Security Mechanisms:

The security of VM-based services rests on the assumption that the underlying trusted computing base (TCB) is also secure. If the TCB is compromised, then all bets are for the VM-based

- Virtual machine monitors
- Mandatory access control
- Para-virtualization
- Policy considerations
- Virtual Layer Vulnerabilities

Benefits:

- Resource Utilization
- Security
- Robustness
- Decomposition
- Encapsulation
- Intrusion Protection

Security Risks in Virtual Environment:

- Scaling
- Transience
- Diversity
- Mobility