ENFOCUS

SWITCH
2024 Fall

Secure Configuration

10 - 2024

Contents

1. Introduction **3**

2. System configuration **4**

2.1. Access management 4

2.2. Network configuration 4

2.2.1. Untrusted networks 4

2.2.2. Incoming connections 4

2.2.3. Outgoing connections 6

3. Switch configuration **7**

3.1. Administrator user security 7

3.2. User management 7

3.3. Securing HTTP connections 7

3.4. Recommendations for Switch flows 8

3.5. Secure connection to the Enfocus Appstore 8

3.6. Submit point URLs 9

4. Reference **11**

4.1. Local network connections 11

4.2. Creating a self-signed certificate using OpenSSL 12

4.2.1. RSA 12

4.2.2. ECDSA 12

4.3. TLS Cipher Suites 12

5. Copyrights **15**

1. Introduction

In this document we describe some baseline configuration settings for Switch Server deployments, which should help you make the server more secure. This information is intended for system administrators and for Switch administrators.

2. System configuration

This section includes settings specific to the underlying operating system.

2.1. Access management

From access management perspective, Switch expects that only system and Switch administrators have direct access to the server machine. This is related to the fact that local Switch connections grant administrative privileges to the local Switch Server.

This means that regular users should not have access to the server machine as this may compromise the Switch installation. Ideally Switch should be installed on a separate physical server or a separate virtual machine, which only administrators have access to.

You also need to make sure, that only the user that will be running Switch services, has access to the Application data root directory. This should already be the case with the default Application data root path, but you might need to check that manually in case you move it to a custom location on the machine. It is very important to configure access to the data root directory properly as failing to do so will severely compromise the security of your Switch installation.

2.2. Network configuration

2.2.1. Untrusted networks

Switch is a complex application suite, which includes multiple network daemons. With certain network configurations those network daemons may be accessed from untrusted networks like the internet. While this doesn't necessarily make the server setup insecure, it is highly advisable to limit access from untrusted networks by using tools like a VPN. Using a VPN might also be useful if some of the protocols you are using for communication are not encrypted, like if you connect to an outdated FTP server, which supports only plain FTP.

By limiting access from untrusted networks you will limit the possible attack surface of the server, making it more secure. This will also make connections easier to audit, as the number of possible network clients will be limited.

2.2.2. Incoming connections

To limit the remote attack surface of the server, it is a good idea to keep only a minimum required number of ports open to untrusted networks. All other unused ports should be blocked from such networks whenever possible.

Below you can see the non-local ports and protocols that Switch uses and why you may want to leave them open. Since they are bound to non-local interfaces, they will usually trigger firewall

warnings on first Switch start-up. In case you don't intend to use the functionality mentioned in the *Comments* column, you can safely block those ports as well.

Port	Protocol	Comments
51007	TCP	Used by Switch Designer to connect to the Switch Server. You can safely block this port if you are going to use Switch Designer only locally. This port can be changed in **Preferences** > **Internal communication** > **Port for Switch Designer**.
51080	TCP	Used by the Webhook flow element and by webhooks in legacy scripts. If you know the addresses from where the webhooks are going to be triggered, you can leave access to only those addresses. This port can be changed in **Preferences** > **Webhooks (legacy)**. Enabling HTTPS will add another user-defined port for the service.
51088	TCP	Used by Switch Web Service. It needs to be open for remote usage of either Switch Designer or Switch Web Portal. If only local connections are expected, this port can be blocked safely. This port can be changed in **Preferences** > **Web Services** > **Port for the Switch Web Service**.
51089	TCP	Used by the Switch Web Portal. It needs to be open for remote usage of the Switch Web Portal. If only local connections are expected, this port can be blocked safely. This port can be changed in **Preferences** > **Web Services** > **Port for the Switch Web Portal**.
51120	TCP	Used by Remote Processing. If you know the IP addresses from which the Remote Processing API will be used, you can leave access to only those addresses. This port can be changed in **Preferences** > **Remote processing**. Enabling HTTPS will add another user-defined port for the service.

Switch Server may listen to random TCP ports when using FTP in Active mode. For this reason Switch on Windows creates a firewall rule to allow the FTP executor to listen on random ports. Ports get open only when that functionality is used.

If you are running a setup with a remote Dashboard Service, then you need to also expose one port on the machine that runs it:

Port	Protocol	Comments
55092	TCP	Used by the Switch Data Service (i.e. Dashboard Service). This port should be accessible only from the machine running Switch Server. All other addresses can be blocked safely.

2.2.3. Outgoing connections

As with incoming connections, it is advisable to limit possible outgoing connections from the server machine.

Your firewall setup for outgoing connections will vary greatly based on the flows you have set up on the server. Flow elements like Mail receive, Mail send, FTP receive, FTP Send, HTTP request and many others will produce outgoing connections, so you might review such elements carefully and create an allow-list of possible connection targets.

For licensing, Switch initiates connections to the following hosts in the list below. They should be added to an-allow list for Switch to function properly.

Host	Port	Protocol
licensingservices.esko.com	443	TCP
external-services.enfocus.com	443	TCP

If you are running a setup with a remote Dashboard Service, then outgoing TCP connections to that services should be added to the allow-list as well.

If you're intending to use the **Switch Proxy Service**, then you should add the host IP along with the port to the allow list for Switch:

Host	Port	Protocol
IP of the machine on which Switch Proxy is installed	51069	TCP

3. Switch configuration

This section includes a number of Switch settings that may impact security.

3.1. Administrator user security

By default Switch creates an Administrator user during the installation. This user has no password, but it is highly advisable to set a secure password for this Administrator user as soon as you install Switch, especially if you intend to connect to Switch remotely.

3.2. User management

When using Switch Designer and/or the Switch Web Portal remotely, you need to set up Switch users and give them the necessary access rights to perform the tasks they need to perform. As a general rule of thumb you need to grant as few access rights to each user as possible. This should limit the possible damage if a user acts maliciously or was hacked by an attacker.

To make it harder to hack the users, it is important to use strong passwords. For example, make sure the passwords have at least 12 characters and avoid commonly used and already leaked passwords.

3.3. Securing HTTP connections

By default Switch Web Service, Switch Web Portal, Switch HTTP Listener and Switch Environment Service (Remote processing) are used with plain HTTP without encryption. Since those services can be accessed remotely and their communications contain sensitive access information like users' passwords and tokens, it is important to enable HTTPS for those daemons.

For HTTPS to work, you need to generate a TLS certificate and a private key in PEM format as separate files. We recommend using either at least a 4096-bit RSA key or a P-256 ECDSA key. If you don't have a *Certificate authority (CA)* in your company setup, which can grant you a TLS certificate, you could use a self-signed certificate instead. To learn how to create such a certificate, refer to *Creating a self-signed certificate using OpenSSL* on page 12. To learn how to make client applications trust the generated certificate, refer to the *HTTPS with self-signed certificates* section in the Switch Reference Guide.

To enable HTTPS for the daemons mentioned above, go to **Preferences** > **HTTPS services settings**. There you need to specify the path to the PEM certificate file and the PEM private key files. Files should be stored in a secure location. Private keys can be encrypted; in that case you need to enter the decryption password in the settings. In case you have a chain of certificates instead of a single one, they need to be concatenated into a single PEM file in the order from leaf to root.

After this has been set up, you need to enable HTTPS on individual services at **Preferences** > **Web services**, **Preferences** > **Webhooks (legacy)** and **Preferences** > **Remote processing**. Since HTTPS ports may differ from HTTP, don't forget to add the necessary firewall rules for incoming connections.

3.4. Recommendations for Switch flows

Because of compatibility reasons, Switch supports insecure legacy protocols in some of the flow elements, like FTP receive, Mail receive and others. In those cases the security level is limited by the target host server configuration. Ideally you should use modern secure protocols, if target servers support them. Below is some general advice regarding some of such elements:

- For *Mail send*, use encrypted SMTP connections with authentication instead of a plain text one, if possible.
- For *Mail receive*, use secure password verification and secure connections whenever possible.
- For *FTP send* and *FTP receive*, don't use plain FTP. Prefer SFTP as it is more secure and it has less compatibility issues in general. Otherwise use FTPS, which is the encrypted version of plain FTP, with *Ignore server certificate errors* set to *No*.
- For *HTTP request*, try to contact the server via HTTPS instead of HTTP, if the target host provides both. *Ignore server certificate errors* should be set to *No*.
- For *Database connect*, use encryption in the ODBC data sources, if the target server and the ODBC driver support such functionality.
- If you are using *Active Directory* for user management, make sure to use secure connections whenever possible. To set up a secure connection to the Active Directory server from Switch, you need to enable the 'Secure connection' option at **Preferences** > **Active Directory settings** > **Active Directory information**.

There are cases when it is impossible to enable more secure protocols on the target host. For example, you might have an old embedded FTP server, which cannot be updated. In that case you may want to use a VPN to encrypt such a connection by creating a private network, which includes both the Switch Server machine and the target host.

Another important note is that you need to make sure to handle data from untrusted sources with additional care. For example, you may have an Execute command element in your flow and the full command depends on data you get from a job. If that job comes from an untrusted source, then an attacker may perform a command execution attack with specially crafted strings of data and gain access to the server. So you need to be careful and sanitize/validate data from such untrusted sources.

3.5. Secure connection to the Enfocus Appstore

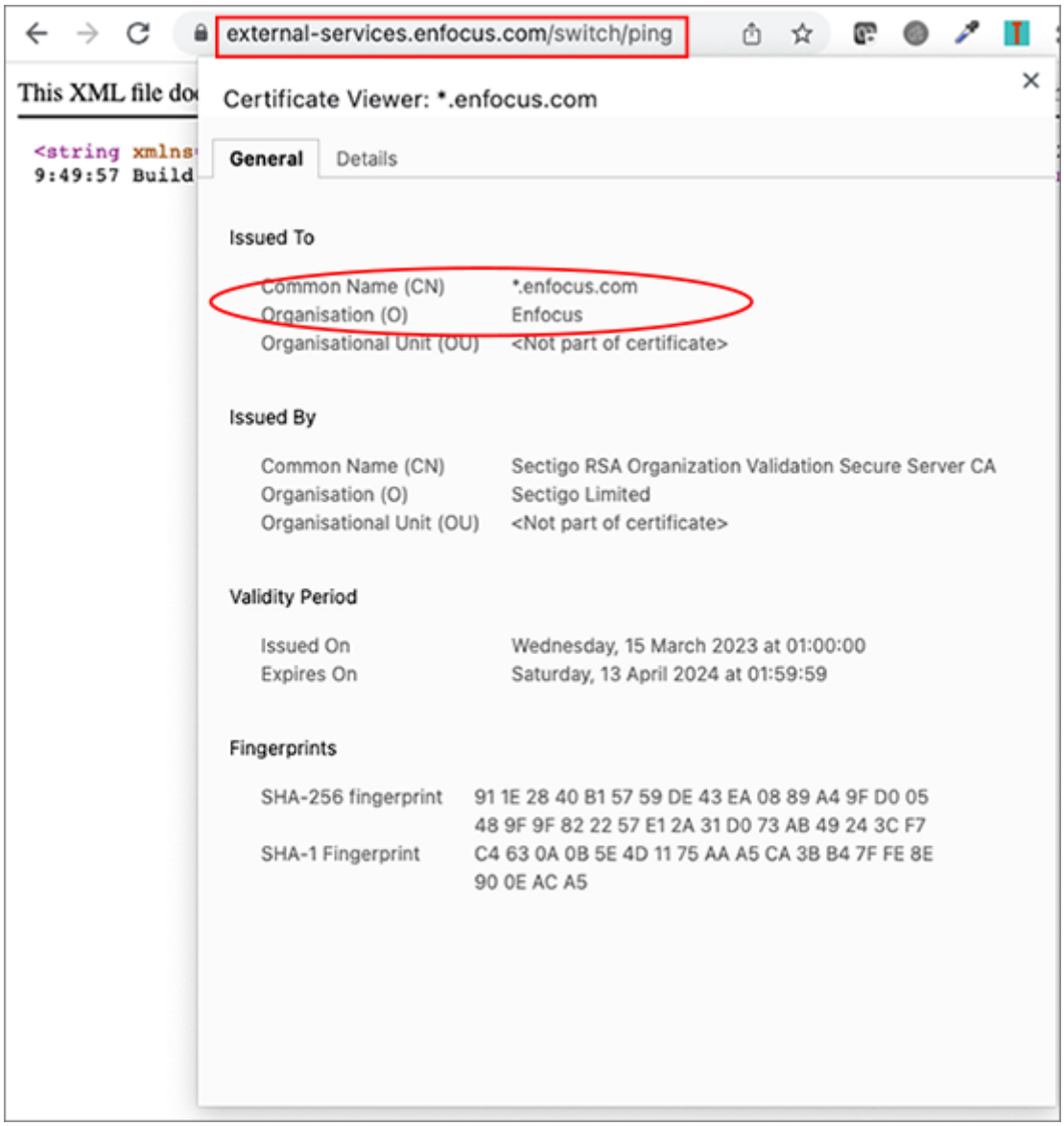
In case of a **firewall**, make sure to allow HTTPS connections to *external-services.enfocus.com* (port 443). See also *Outgoing connections* on page 6.

You can check this connection by using the following command:

```
curl -v https://external-services.enfocus.com/switch/ping
```

Note: Alternatively, you could use a browser to check the connection. However, if your system has a proxy server configured, this is not recommended: the browser will use the proxy server, but Switch will ignore it for Appstore connections.

Note that **Switch does not support a terminating proxy**. The SSL encrypted traffic between Switch and the Enfocus Appstore must be forwarded unchanged. To check if this is the case, take a look at the certificate: it should be issued by Enfocus.



3.6. Submit point URLs

When disabling authentication for Submit point URLs, we highly recommend to only share each URL with a single user only. This allows later on

- To track which user submitted a certain job, e.g. in case of abuse.

- To remove the URL for a particular user without affecting any other users.

It is also not recommended to embed Submit point URLs without authentication into a website, even when behind a login page, as anybody who can log in or access that webpage can grab the link to further spread or abuse it.

4. Reference

This section includes miscellaneous info, which may be useful during server configuration and testing.

4.1. Local network connections

Most of the network daemons in Switch bind only to *localhost* and are unavailable remotely. Since those connections are local only, most of them are unencrypted as data should be contained within the server machine. Below is a list of some local listening ports used by Switch:

Port	Protocol	Description
51057	TCP	This port is opened by Switch Watchdog and is used for service management by local connections from Switch Designer. This port can be changed in **Preferences** > **Internal communication** > **Port for Watchdog**.
51087	TCP	This port is opened by Switch Watchdog and is used by Switch Server for service management. This port can be changed in **Preferences** > **Internal communication** > **Port for Watchdog - Switch Server**.
55091	TCP	This port is opened by MongoDB and is used exclusively by the Switch Data Service (locally). Authentication on the service is enforced using SCRAM. The password is auto-generated and cannot be configured by the user.
55092	TCP	This port is opened by the local Switch Data Service (Dashboard Service). It is used by multiple other Switch services.
55093	TCP	This port is opened by Switch Logger, which handles message logging. It is used by multiple other Switch services.
55094	TCP	This port is opened by the Switch Analytics Service, which handles the Switch Insight Dashboard functionality. It is used by multiple other Switch services.
55095	TCP	This port is opened by the Switch Processor Service, which handles Node.js scripting. It is used by Switch Server. **Note:** This daemon triggers a firewall warning because of the way it checks for TCP port availability. Communications are actually local only.

Port	Protocol	Description
55096	TCP	This port is opened by Switch Designer Service, which is started alongside Switch Designer to provide additional web-based functionality. It is used only by Switch Designer.

Switch Designer may open a random TCP port, when doing OAuth authentication in flow element properties. This opened port is used for an HTTP server, which handles the OAuth local redirect.

4.2. Creating a self-signed certificate using OpenSSL

4.2.1. RSA

```
# Creates an 4096-bit RSA private key and certificate valid for 1 year
# Domain and other information will be prompted
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out crt.pem -days 365
```

4.2.2. ECDSA

```
# Generates a P-256 ECDSA private key
openssl ecparam -name prime256v1 -genkey -noout -out key.pem
# Creates the corresponding certificate valid for 1 year
# Domain and other information will be prompted
openssl req -new -x509 -key key.pem -out crt.pem -days 365
```

4.3. TLS Cipher Suites

Below you can find lists of cipher suites, which are supported by Switch TLS servers. Server preference is used for TLSv1.2 cipher suites, so when a TLSv1.2 client connects, the first matching TLSv1.2 cipher suite will be used. These lists are static and cannot be changed by the user.

Service	Protocol	Cipher suites
Switch Web Service, Switch Web Portal	TLSv1.3 TLSv1.2	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

Service	Protocol	Cipher suites
		TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_256_CCM_8 TLS_RSA_WITH_AES_256_CCM TLS_RSA_WITH_ARIA_256_GCM_SHA384 TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_128_CCM_8 TLS_RSA_WITH_AES_128_CCM TLS_RSA_WITH_ARIA_128_GCM_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA
Switch Environment Service,Switch HTTP Listener	TLSv1.3 TLSv1.2	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

Service	Protocol	Cipher suites
		TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_GCM_SHA256 TLS_RSA_WITH_AES_256_GCM_SHA384 TLS_RSA_WITH_AES_128_CBC_SHA256 TLS_RSA_WITH_AES_256_CBC_SHA256 TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA
Switch Server	TLSv1.3 TLSv1.2	TLS_AES_128_GCM_SHA256 TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

5. Copyrights

© 2024 Enfocus BV all rights reserved. Enfocus is an Esko company.

Certified PDF is a registered trademark of Enfocus BV.

Enfocus PitStop Pro, Enfocus PitStop Workgroup Manager, Enfocus PitStop Server, Enfocus BoardingPass, Enfocus Connect YOU, Enfocus Connect ALL, Enfocus Connect SEND, Enfocus StatusCheck, Enfocus CertifiedPDF.net, Enfocus PDF Workflow Suite, Enfocus Switch, Enfocus SwitchClient, Enfocus SwitchScripter, Enfocus TestDrive, Enfocus SwitchScriptTool, Enfocus Browser, PitStop Library Container, Enfocus Griffin, Enfocus Review, Enfocus FastLane and Enfocus Appstore are product names of Enfocus BV.

Adobe, Acrobat, Distiller, InDesign, Illustrator, Photoshop, FrameMaker, PDFWriter, PageMaker, Adobe PDF Library™, the Adobe logo, the Acrobat logo and PostScript are trademarks of Adobe Systems Incorporated.

Datalogics, the Datalogics logo, PDF2IMG™ and DLE™ are trademarks of Datalogics, Inc.

Apple, Mac, macOS, Macintosh, iPad and ColorSync are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. Windows and Windows Server are registered trademarks of Microsoft Corporation.

PANTONE® Colors displayed here may not match PANTONE-identified standards. Consult current PANTONE Color Publications for accurate color. PANTONE® and other Pantone, Inc. trademarks are the property of Pantone, Inc. ©Pantone, Inc., 2006.

OPI is a trademark of Aldus Corporation.

Quark, QuarkXPress, QuarkXTensions, XTensions and the XTensions logo among others, are trademarks of Quark, Inc. and all applicable affiliated companies, Reg. U.S. Pat. & Tm. Off. and in many other countries.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

This product and use of this product is under license from Markzware under U.S.Patent No. 5,963,641.

Other brand and product names may be trademarks or registered trademarks of their respective holders. All specifications, terms and descriptions of products and services are subject to change without notice or recourse.