Netscaler Guide

Simon Lauger

Table of Contents

Best practices and example configurations for Citrix NetScaler.

Chapter 1. Intro

1.1. What's this?

In this documents i will share all my NetScaler best practices and guidelines.

This is not a "no-brainer" copy & paste guide. Double check every configuration line before you paste it in your NetScaler.

Custom configurations (like binding of ssl certificates) are not part of this document.

1.2. Build HTML

All content is written in ASCII Doc. You can convert the content to HTML via asciidoctor. A Makefile is included in this repository. Just run the following command to build it.

```
make all
```

Ruby and the asciidoctor gem need to be installed. A RPM package for Enterprise Linux (RHEL, CentOS, Fedora) is available trough EPEL.

```
yum install rubygem-asciidoctor.noarch
```

A HTML version of this documentation is also available on my website.

The asciidoc source is available on GitHub.

1.3. Contribute

Feedback and suggestion for improvement are appreciated. Just create a issue on GitHub. You can also send me an Pull requests to suggest changes.

1.4. Available Guides

- Best Practices
- SSL Hardening
- SSH Hardening
- Exchange 2016
- NetScaler Gateway (ICA only)
- NetScaler Gateway (Smartaccess)
- XenMobile
- Sharefile

- SMS Passcode

Chapter 2. Best Practices

2.1. Layer 3 mode

Layer 3 mode is enabled by default. This enables IP fowarding and is only required in a few situations (e.g. if you need to fowrard Client IPs in different procotols than HTTP with the feature "USIP").

If you don’t need this Layer 3 mode should be disabled for security reasons.

```
disable mode L3mode
```

2.2. External auth for nsroot

External auth for nsroot is enabled by default. If not needed this should be disabled for security reasons.

```
set system user nsroot -externalAuth DISABLED
```

2.3. Optmize HTTP/TCP

These are a combination of the Citrix best practices from CTX121149 and my personal experience.

```
# configure timeout (GUI, SSH) to 10 minutes
set system parameter -timeout 600 -doppler DISABLED

# tips from CTX121149
set ns tcpProfile nstcp_default_profile -WS ENABLED -SACK ENABLED -nagle ENABLED
set ns httpProfile nshttp_default_profile -dropInvalReqs ENABLED -markHttp09Inval
ENABLED -markConnReqInval ENABLED
set ns tcpParam -WS ENABLED -SACK ENABLED -nagle ENABLED

# drop invalid HTTP requests
set ns httpParam -dropInvalReqs ON -markHttp09Inval ON -markConnReqInval ON

# enable X-Forwarded-For header globaly, set Cookie version to v1
set ns param -cip ENABLED X-Forwarded-For -cookieversion 1
```

Chapter 3. Basics

3.1. Share CPU ressources on a VPX

You may want to re-enable the sharing of CPU resources (yielding) on your VPX. Yielding is disabled by default on NetScaler 12.0 and onwards.

```
set ns vpxparam -cpuyield YES
```

3.2. Timezone

If not already done you should also configure your local timezone:

```
set ns param -timezone "GMT+01:00-CET-Europe/Berlin"
```

3.3. Basic Features

This is an example for enabling the most used features. This depends by the installed license and your individual needs.

Here are some examples.

NetScaler Gateway Platform license

```
enable ns feature SSL SSLVPN REWRITE RESPONDER
```

NetScaler Standard

```
enable ns feature LB CS SSL CF SSLVPN REWRITE RESPONDER
```

NetScaler Enteprise

```
enable ns feature LB CS SSL CF SSLVPN AAA REWRITE RESPONDER RDPProxy
```

3.4. LDAP

```
add authentication ldapAction act_auth_ldap -serverName ${DC} -serverPort 636
-ldapBase "${LDAP_BASEDN}" -ldapBindDn "${LDAP_BINDDN}" -ldapBindDnPassword
"${LDAP_BINDPW}" -ldapLoginName sAMAccountName -searchFilter "${LDAP_FILTER}"
-groupAttrName memberOf -subAttributeName CN -secType SSL -ssoNameAttribute
sAMAccountName -passwdChange ENABLED -nestedGroupExtraction ON -ldapHostname
${LDAP_FQDN} -groupNameIdentifier sAMAccountName -groupSearchAttribute memberOf
-groupSearchSubAttribute CN
add authentication ldapPolicy pol_auth_ldap ns_true act_auth_ldap
```

Admin authentication

```
bind system global pol_auth_ldap
```

```
add system group "${LDAP_ADMINS}"
bind system group "${LDAP_ADMINS}" -policyName superuser 100
```

3.5. Syslog

```
add audit syslogAction act_audit_syslog ${SYSLOG} -logLevel ${SYSLOG_LOGLEVEL}
add audit syslogPolicy pol_audit_syslog ns_true act_audit_syslog
bind system global pol_audit_syslog -priority 100
```

3.6. SNMP

```
add snmp community ${SNMP_COMMUNITY} GET_NEXT
set snmp mib -name ${SNMP_NAME} -contact ${SNMP_CONTACT} -customID ${SNMP_CUSTOM_ID}
-location ${SNMP_LOCATION}
```

3.7. NTP

```
add ntp server ${NTP1}
add ntp server ${NTP2}
```

NTP synchronisation is disabled by default.

```
enable ntp sync
```

3.8. VLAN Binding

You should configure a SNIP for each network interface and bind it to a VLAN. This avoids MAC flapping between ports (NetScaler auto discovery).

```
add vlan ${VLAN_EXT} -aliasName ${VLAN_EXT_ALIAS}
bind vlan ${VLAN_EXT} -IPAddress ${SNIP_EXT} ${SNIP_EXT_MASK}
bind vlan ${VLAN_EXT} -ifnum ${VLAN_EXT_INTERFACE}
```

3.9. DNS Loadbalancing

```
add server ${DNS1_FQDN} ${DNS1_IP}
add server ${DNS2_FQDN} ${DNS2_IP}
```

```
add serviceGroup sg_dns_ad DNS -maxClient 0 -maxReq 0 -usip NO -useproxyport NO
-cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
```

```
bind serviceGroup sg_dns_ad ${DNS1_FQDN} 53
bind serviceGroup sg_dns_ad ${DNS2_FQDN} 53
```

```
add lb vserver vs_lb_dns_ad DNS 0.0.0.0 0 -persistenceType NONE -cltTimeout 180
```

```
bind lb vserver vs_lb_dns_ad sg_dns_ad
```

```
add dns nameServer vs_lb_dns_ad
```

3.10. DNS Suffix

```
add dns suffix ${DNS_SUFFIX}
```

3.11. Generic SSL Redirect

```
add responder action act_responder_ssl_redirect_generic redirect "\"https://\" +
HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.HTTP_URL_SAFE" -responseStatusCode 301
add responder policy pol_responder_ssl_redirect_generic true
act_responder_ssl_redirect_generic
```

3.12. Access Log

```
add audit messageaction access_log INFORMATIONAL q/CLIENT.IP.SRC + HTTP.REQ.HEADER("X-Forwarded-For").HTTP_HEADER_SAFE + " " + HTTP.REQ.HOSTNAME + " \"" +HTTP.REQ.METHOD + " " +HTTP.REQ.URL.PATH_AND_QUERY.HTTP_URL_SAFE + "\"" +" \"" + HTTP.REQ.HEADER("User-Agent").HTTP_HEADER_SAFE + "\"" /
```

Chapter 4. SSL Hardening

4.1. Secure Ciphers

High secure ciphers only - TLS 1.3 and TLS 1.2.

```
add ssl cipher CUSTOM_MODERN
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.3-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.3-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.3-CHACHA20-POLY1305-SHA256
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-RSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
```

High secure ciphers for TLS 1.3 and TLS 1.2, but keep some legacy ciphers enabled.

```
add ssl cipher CUSTOM_INTERMEDIATE
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.3-AES256-GCM-SHA384
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.3-AES128-GCM-SHA256
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.3-CHACHA20-POLY1305-SHA256
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-RSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-RSA-AES-256-SHA384
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-RSA-AES-128-SHA256
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-ECDSA-AES256-SHA384
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_INTERMEDIATE -cipherName TLS1.2-ECDHE-ECDSA-AES128-SHA256
```

If support for TLS 1.0 or TLS 1.1 is required. Also a good choice for your backend profile.

```
add ssl cipher CUSTOM_MODERN_LEGACY
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.3-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.3-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.3-CHACHA20-POLY1305-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-RSA-CHACHA20-POLY1305
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-DHE-RSA-AES256-GCM-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-RSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-DHE-RSA-AES128-GCM-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-ECDSA-AES256-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-RSA-AES-256-SHA384
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-ECDSA-AES128-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1.2-ECDHE-RSA-AES-128-SHA256
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-ECDHE-RSA-AES256-SHA
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-ECDHE-RSA-AES128-SHA
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-DHE-RSA-AES-256-CBC-SHA
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-DHE-RSA-AES-128-CBC-SHA
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-AES-256-CBC-SHA
bind ssl cipher CUSTOM_MODERN_LEGACY -cipherName TLS1-AES-128-CBC-SHA
```

4.2. Default Profile

```
set ssl parameter -defaultProfile ENABLED
Y

set ssl profile ns_default_ssl_profile_frontend -denySSLReneg NONSECURE
bind ssl profile ns_default_ssl_profile_frontend -cipherName CUSTOM_INTERMEDIATE
unbind ssl profile ns_default_ssl_profile_frontend -cipherName DEFAULT

set ssl profile ns_default_ssl_profile_backend -denySSLReneg NONSECURE
bind ssl profile ns_default_ssl_profile_backend -cipherName CUSTOM_INTERMEDIATE
unbind ssl profile ns_default_ssl_profile_backend -cipherName DEFAULT_BACKEND
```

4.3. Disable TLS 1.0 and TLS 1.1

- disable TLS 1.0
- disable TLS 1.1
- enable TLS 1.2
- enable TLS 1.3

```
set ssl profile ns_default_ssl_profile_frontend -tls1 DISABLED -tls11 DISABLED -tls12
ENABLED -tls13 ENABLED

set ssl profile ns_default_ssl_profile_backend -tls1 DISABLED -tls11 DISABLED -tls12
ENABLED -tls13 ENABLED
```

4.4. Enable HSTS

This adds the STS header to all HTTP responses. This is required to get an A+ rating on ssllabs.com.

```
add rewrite action act_rewrite_inject_http_sts_header insert_http_header Strict-
Transport-Security "\"max-age=31536000\""
add rewrite policy pol_rewrite_inject_http_sts_header true
act_rewrite_inject_http_sts_header
bind rewrite global pol_rewrite_inject_http_sts_header 100 NEXT -type RES_OVERRIDE
```

NetScaler 12.0+ Since version 12.0 NetScaler has a builtin HSTS feature. The rewrite policy from above is no longer needed.

```
set ssl profile ns_default_ssl_profile_frontend -HSTS ENABLED -maxage 31536000
```

4.5. Diffie-Hellman key

Create a Diffie-Hellman key and bind it to the default frontend profile (be aware, this could take a lot of time).

```
create dhParam /nsconfig/ssl/ECDHE.key -gen 5 2048
set ssl profile ns_default_ssl_profile_frontend -dh ENABLED -dhFile
"/nsconfig/ssl/ECDHE.key"
```

Chapter 5. SSH Hardening

5.1. Create ed25519 hostkey

The default configuration does not use ed25519 hostkeys.

```
ssh-keygen -t ed25519 -f /nsconfig/ssh/ssh_host_ed25519_key -N ""
```

5.2. Secure configuration

This will disable legacy ciphers, the dsa hostkey and the password authentication.

Copy this file to /nsconfig/sshd_config and reboot your NetScaler.

```
Port 22
ListenAddress 0.0.0.0
ListenAddress ::
#HostKey /nsconfig/ssh/ssh_host_dsa_key
#HostKey /nsconfig/ssh/ssh_host_ecdsa_key
HostKey /nsconfig/ssh/ssh_host_rsa_key
HostKey /nsconfig/ssh/ssh_host_ed25519_key
LoginGraceTime 120
PermitRootLogin without-password
LogLevel INFO
IgnoreRhosts no
StrictModes yes
X11Forwarding no
X11DisplayOffset 10
PrintMotd no
KeepAlive yes
SyslogFacility AUTH
PasswordAuthentication no
PermitEmptyPasswords no
UsePam no
UseDNS no
ClientAliveInterval 10
ClientAliveCountMax 5
Subsystem sftp /usr/libexec/sftp-server
AllowTcpForwarding no
MaxStartups 10:30:60
MaxAuthTries 3
KexAlgorithms curve25519-sha256@libssh.org
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-
etm@openssh.com
```

Chapter 6. Exchange 2016

This is an example configuration for Exchange 2016 on NetScaler.

- Content Switching for every single service/directory
- Formbased Preauth for Outlook Web Access
- 401 Preauth for Active Sync Traffic
- NTLM Preauth for Outlook Anywhere Traffic

To use all features (e.g. preauth) at least a Enterprise license is required (AAA).

This configuration was built for my Citrix Networking specialist exam and Exchange 2013. I added some improvements and support for Exchange 2016 (e.g. MAPI over HTTPS) afterwards.

This configuration runs in production on 10+ customer setups (not all of them use preauth).

6.1. Content Switching

Add your exchange servers

Be sure that you use the real FQDN of the machine as name for the server object. This is important if you plan to use Kerberos preauth for Outlook Anywhere.

```
add server ${EX1_FQDN} ${EX1_IP}
add server ${EX2_FQDN} ${EX2_IP}
```

Service groups

```
add serviceGroup sg_ssl_ex2016_owa SSL -maxClient 0 -maxReq 0 -cip ENABLED X-
Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360 -CKA YES
-TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_ews SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_activesync SSL -maxClient 0 -maxReq 0 -cacheable YES
-cip ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout
360 -CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_rpc SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_autodiscover SSL -maxClient 0 -maxReq 0 -cacheable YES
-cip ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout
360 -CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_ecp SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_oab SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
add serviceGroup sg_ssl_ex2016_mapi SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
```

Add Exchange servers to the servicegroups

```
bind serviceGroup sg_ssl_ex2016_owa ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_owa ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_ews ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_ews ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_activesync ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_activesync ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_rpc ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_rpc ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_autodiscover ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_autodiscover ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_ecp ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_ecp ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_oab ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_oab ${EX2_FQDN} 443

bind serviceGroup sg_ssl_ex2016_mapi ${EX1_FQDN} 443
bind serviceGroup sg_ssl_ex2016_mapi ${EX2_FQDN} 443
```

Monitoring

```
add lb monitor mon_ex2016_owa HTTP -respCode 200 -httpRequest "GET
/owa/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_ews HTTP -respCode 200 -httpRequest "GET
/ews/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_activesync HTTP -respCode 200 -httpRequest "GET /Microsoft-
Server-ActiveSync/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_rpc HTTP -respCode 200 -httpRequest "GET
/rpc/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_autodiscover HTTP -respCode 200 -httpRequest "GET
/Autodiscover/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_ecp HTTP -respCode 200 -httpRequest "GET
/ecp/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_oab HTTP -respCode 200 -httpRequest "GET
/oab/healthcheck.htm" -LRTM ENABLED -secure YES
add lb monitor mon_ex2016_mapi HTTP -respCode 200 -httpRequest "GET
/mapi/healthcheck.htm" -LRTM ENABLED -secure YES
```

```
bind serviceGroup sg_ssl_ex2016_owa -monitorName mon_ex2016_owa
bind serviceGroup sg_ssl_ex2016_ews -monitorName mon_ex2016_ews
bind serviceGroup sg_ssl_ex2016_activesync -monitorName mon_ex2016_activesync
bind serviceGroup sg_ssl_ex2016_rpc -monitorName mon_ex2016_rpc
bind serviceGroup sg_ssl_ex2016_autodiscover -monitorName mon_ex2016_autodiscover
bind serviceGroup sg_ssl_ex2016_ecp -monitorName mon_ex2016_ecp
bind serviceGroup sg_ssl_ex2016_oab -monitorName mon_ex2016_oab
bind serviceGroup sg_ssl_ex2016_mapi -monitorName mon_ex2016_mapi
```

Internal LB vServers

```
add lb vserver vs_lb_http_ex2016_owa HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_owa_redirect HTTP 0.0.0.0 0 -cltTimeout 180
add lb vserver vs_lb_http_ex2016_ews HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_activesync HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_rpc HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_autodiscover HTTP 0.0.0.0 0 -lbMethod
LEASTRESPONSETIME -cltTimeout 180
add lb vserver vs_lb_http_ex2016_ecp HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_oab HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
add lb vserver vs_lb_http_ex2016_mapi HTTP 0.0.0.0 0 -lbMethod LEASTRESPONSETIME
-cltTimeout 180
```

Connect LB vServers and the servicegroups

```
bind lb vserver vs_lb_http_ex2016_owa sg_ssl_ex2016_owa
bind lb vserver vs_lb_http_ex2016_owa_redirect sg_ssl_ex2016_owa
bind lb vserver vs_lb_http_ex2016_ews sg_ssl_ex2016_ews
bind lb vserver vs_lb_http_ex2016_activesync sg_ssl_ex2016_activesync
bind lb vserver vs_lb_http_ex2016_rpc sg_ssl_ex2016_rpc
bind lb vserver vs_lb_http_ex2016_autodiscover sg_ssl_ex2016_autodiscover
bind lb vserver vs_lb_http_ex2016_ecp sg_ssl_ex2016_ecp
bind lb vserver vs_lb_http_ex2016_oab sg_ssl_ex2016_oab
bind lb vserver vs_lb_http_ex2016_mapi sg_ssl_ex2016_mapi
```

Content Switching policies

```
add cs action act_cs_ex2016_owa -targetLBVserver vs_lb_http_ex2016_owa
add cs action act_cs_ex2016_ews -targetLBVserver vs_lb_http_ex2016_ews
add cs action act_cs_ex2016_activesync -targetLBVserver vs_lb_http_ex2016_activesync
add cs action act_cs_ex2016_rpc -targetLBVserver vs_lb_http_ex2016_rpc
add cs action act_cs_ex2016_autodiscover -targetLBVserver
vs_lb_http_ex2016_autodiscover
add cs action act_cs_ex2016_ecp -targetLBVserver vs_lb_http_ex2016_ecp
add cs action act_cs_ex2016_oab -targetLBVserver vs_lb_http_ex2016_oab
add cs action act_cs_ex2016_mapi -targetLBVserver vs_lb_http_ex2016_mapi
```

```
add cs policy pol_cs_ex2016_owa -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/owa\")\n||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/owa\")\n||HTTP.REQ.URL.SET_TEXT_M
ODE(IGNORECASE).EQ(\"/cgi/selfauth\")
||\nHTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/cgi/selfauth\")" -action
act_cs_ex2016_owa
add cs policy pol_cs_ex2016_ews -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/ews\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/ews\")" -action act_cs_ex2016_ews
add cs policy pol_cs_ex2016_rpc -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/rpc\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/rpc\")" -action act_cs_ex2016_rpc
add cs policy pol_cs_ex2016_autodiscover -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/Autodiscover\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/Autodiscover\")" -action
act_cs_ex2016_autodiscover
add cs policy pol_cs_ex2016_ecp -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/ecp\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/ecp\")" -action act_cs_ex2016_ecp
add cs policy pol_cs_ex2016_oab -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/oab\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/oab\")" -action act_cs_ex2016_oab
add cs policy pol_cs_ex2016_mapi -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/mapi\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/mapi\")" -action
act_cs_ex2016_mapi
add cs policy pol_cs_ex2016_activesync -rule
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).EQ(\"/Microsoft-Server-ActiveSync\") ||
HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).STARTSWITH(\"/Microsoft-Server-ActiveSync\")"
-action act_cs_ex2016_activesync
```

Create CS vServer

```
add cs vserver vs_cs_ssl_ex2016 SSL ${EX_VIP} 443 -cltTimeout 180
add cs vserver vs_cs_http_ex2016 HTTP ${EX_VIP} 80 -cltTimeout 180
```

Add the CS policies to the CS vServer

```
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_owa -priority 100
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_ews -priority 110
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_activesync -priority 120
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_rpc -priority 130
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_autodiscover -priority 140
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_ecp -priority 150
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_oab -priority 160
bind cs vserver vs_cs_ssl_ex2016 -policyName pol_cs_ex2016_mapi -priority 170
bind cs vserver vs_cs_ssl_ex2016 -lbvserver vs_lb_http_ex2016_owa_redirect
```

Redirect all HTTP requests to SSL

```
bind cs vserver vs_cs_http_ex2016 -policyName pol_responder_ssl_redirect_generic
-priority 100 -gotoPriorityExpression END -type REQUEST
```

Redirect invalid requests to OWA

```
add responder action act_responder_ssl_redirect_owa redirect "\"https://\" +
HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + \"/owa/\"" -responseStatusCode 302
add responder policy pol_responder_ssl_redirect_owa "true"
act_responder_ssl_redirect_owa
```

```
bind lb vserver vs_lb_http_ex2016_owa_redirect -policyName
pol_responder_ssl_redirect_owa -priority 100 -gotoPriorityExpression END -type REQUEST
```

6.2. Persistence

This creates a SOURCEIP persistence group for Exchange services. This is not required, but it can make your troubleshooting a lot easier.

```
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_ews
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_activesync
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_rpc
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_autodiscover
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_oab
bind lb group persistency_group_ex2016_web vs_lb_http_ex2016_mapi
set lb group persistency_group_ex2016_web -persistenceType SOURCEIP
```

6.3. Preauth for Activesync

AAA vServer

```
add authentication vserver vs_aaa_ex2016_auth_basic SSL 0.0.0.0 -AuthenticationDomain
${AAA_DOMAIN}
```

Bind LDAP policy

```
bind authentication vserver aaa_vs_ex2016_auth_basic -policy pol_auth_ldap -priority
100
```

Enable 401 auth for ActiveSync

```
set lb vserver vs_lb_http_ex2016_activesync -authn401 ON -authnVsName
aaa_vs_ex2016_auth_basic
```

6.3.1. Group Filtering for ActiveSync

Authorization policy

```
add authorization policy pol_authorization_activesync
"HTTP.REQ.USER.IS_MEMBER_OF(\"Netscaler-ActiveSync\").NOT" DENY
```

Bind authorization policy

```
bind lb vserver vs_lb_http_ex2016_activesync -policyName pol_authorization_activesync
-priority 100 -gotoPriorityExpression END -type REQUEST
```

6.4. Preauth for OWA

AAA vServer

```
add authentication vserver vs_aaa_ex2016_auth_form SSL ${AAA_VIP} 443
-AuthenticationDomain ${AAA_DOMAIN}
add cs vserver vs_cs_http_ex2016_auth_form HTTP ${AAA_VIP} 80 -cltTimeout 180
```

Redirect all requests to SSL

```
bind cs vserver vs_cs_http_ex2016_auth_form -policyName pol_responder_ssl_redirect_owa
-priority 100 -gotoPriorityExpression END -type REQUEST
```

Bind authentication policies

Bind your authentication policies as you like. This is an example with 2-factor (LDAP+RADIUS).

```
bind authentication vserver aaa_vs_ex2016_auth_form -policy pol_auth_ldap -priority
100
bind authentication vserver aaa_vs_ex2016_auth_form -policy pol_auth_radius -priority
100 -secondary
```

Enable formbased auth for OWA and ECP

```
set lb vserver vs_lb_http_ex2016_owa -AuthenticationHost ${AAA_FQDN} -Authentication
ON -authnVsName vs_aaa_ex2016_auth_form
set lb vserver vs_lb_http_ex2016_ecp -AuthenticationHost ${AAA_FQDN} -Authentication
ON -authnVsName vs_aaa_ex2016_auth_form
```

6.4.1. Group Filtering for OWA

Authorization policy

```
add authorization policy pol_authorization_owa
"HTTP.REQ.USER.IS_MEMBER_OF(\"Netscaler-OWA\").NOT" DENY
```

Bind authorization policy

```
bind lb vserver vs_lb_http_ex2016_owa -policyName pol_authorization_owa -priority 100
-gotoPriorityExpression END -type REQUEST
bind lb vserver vs_lb_http_ex2016_ecp -policyName pol_authorization_owa -priority 100
-gotoPriorityExpression END -type REQUEST
```

FormSSO and Logout from OWA

```
add tm formSSOAction prof_formsso_exchange_sso -actionURL "/owa/auth.owa" -userField
username -passwdField password -ssoSuccessRule
"HTTP.RES.SET_COOKIE.COOKIE(\"cadata\").VALUE(\"cadata\").LENGTH.GT(70)"
-nameValuePair
"destination=https://${EX_FQDN}/owa/#authRedirect=true&flags=4&forcedownlevel=0&passwo
rdText=&isUtf8=1" -responsesize 60000 -submitMethod POST

add tm trafficAction prof_traffic_exchange_sso_login -appTimeout 1 -SSO ON
-formSSOAction prof_formsso_exchange_sso -persistentCookie ON -InitiateLogout OFF
-kcdAccount NONE
add tm trafficAction prof_traffic_exchange_sso_logout -appTimeout 1 -SSO ON
-persistentCookie OFF -InitiateLogout ON -kcdAccount NONE

add tm trafficPolicy pol_traffic_exchange_sso_login
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS(\"/owa/auth/logon.aspx\")"
prof_traffic_exchange_sso_login
add tm trafficPolicy pol_traffic_exchange_sso_logout
"HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS(\"/owa/logoff.owa\")"
prof_traffic_exchange_sso_logout
```

6.5. Preauth for Outlook Anywhere

You should already have an Keytab file. See CTX129314 for more information.

AAA vServer

```
add authentication vserver vs_aaa_ex2016_auth_kerberos SSL 0.0.0.0
-AuthenticationDomain ${AAA_DOMAIN}
```

Enable 401 based auth with Kerberos/NTLM

```
set lb vserver vs_lb_http_ex2016_ews -authn401 ON -authnVsName
vs_aaa_ex2016_auth_kerberos
set lb vserver vs_lb_http_ex2016_rpc -authn401 ON -authnVsName
vs_aaa_ex2016_auth_kerberos
set lb vserver vs_lb_http_ex2016_autodiscover -authn401 ON -authnVsName
vs_aaa_ex2016_auth_kerberos
set lb vserver vs_lb_http_ex2016_oab -authn401 ON -authnVsName
vs_aaa_ex2016_auth_kerberos
set lb vserver vs_lb_http_ex2016_mapi -authn401 ON -authnVsName
vs_aaa_ex2016_auth_kerberos
```

Negotiate policy

For "NTML external path" a url with NTLM enabled auth on the target is required. This example may work if formbased auth is disabled for OWA.

```
add authentication negotiateAction act_negotiate_ex2016 -keytab
"/nsconfig/krb/ns_kcd.keytab" -NTLMPath "https://${INT_EX_LB}/owa/"
add authentication negotiatePolicy pol_negotiate_ex2016 ns_true act_negotiate_ex2016
```

Bind negotiate policy

```
bind authentication vserver aaa_vs_ex2016_auth_kerberos -policy pol_negotiate_ex2016
-priority 100
```

Add KCD account

The realm should be your internal domain name in uppercase.

```
add aaa kcdAccount svc_ctxnetscaler_krb -keytab "/nsconfig/krb/nskrb.keytab"
-userRealm ${REALM}
```

Session policy

```
add tm sessionPolicy pol_session_ex2016 true prof_session_kerberos
add tm sessionAction act_session_ex2016 -sessTimeout 60 -defaultAuthorizationAction
ALLOW -SSO ON -ssoCredential PRIMARY -ssoDomain ${SSO_DOMAIN} -httpOnlyCookie NO
-kcdAccount svc_ctxnetscaler_krb
```

Bind session policy

```
bind authentication vserver aaa_vs_ex2016_auth_kerberos -policy pol_session_ex2016
-priority 100 -gotoPriorityExpression NEXT
```

6.5.1. Group Filtering for Outlook Anywhere

It’s not possible to extract the groups from a user when NTLM or Kerberos authentication is used. Instead you could use a HTTP callout to a PHP script with and simple ldapsearch. You can find an example written in PHP inside of the contrib directory.

It’s possible to host the file directly on the NetScaler (NetScaler has php and php-ldap installed). I only recommend this if a Platinum license is available (so caching of callout responses is possible). Otherwise consider hosting the script on a different machine.

Just put the file ldapgroups.php inside of /nsconfig and create a symlink from /var/ns_gui/shared/ldapgroups.php to /nsconfig/ldapgroups.php. Use the nsbefore.sh to make this persistent across reboots.

Callouts to the NSIP are blocked. Make sure you use a SNIP with management access (HTTP or HTTPS) enabled.

nsbefore.sh

```
# /nsconfig/nsbefore.sh
cd /var/ns_gui/shared && ln -s /nsconfig/ldapgroups.php .
```

HTTP Callout

```
add policy httpCallout callout_http_usergroups -IPAddress ${MGMT_SNIP} -port 80
-returnType TEXT -hostExpr "\"${MGMT_SNIP}\"" -urlStemExpr
"\"/shared/ldapgroups.php\"" -parameters username(HTTP.REQ.USER.LOGIN_NAME) -scheme
http -resultExpr "HTTP.RES.BODY(99999)" -cacheForSecs 3600 -comment "get groups from
active directory"
```

Responder filtering policy

The HTTP response code is set to 503. If a user is blocked he will receive an error in his Outlook client.

```
add responder action act_responder_filter_outlook respondwithhtmlpage ex2016_deny.html
-responseStatusCode 503
add responder policy pol_responder_filter_outlook
"SYS.HTTP_CALLOUT(callout_http_usergroups).CONTAINS(\"Netscaler-Outlook\").NOT"
act_responder_filter_outlook
```

Bind responder policy

```
bind lb vserver vs_lb_http_ex2016_ews -policyName pol_responder_filter_outlook
-priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_lb_http_ex2016_rpc -policyName pol_responder_filter_outlook
-priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_lb_http_ex2016_autodiscover -policyName
pol_responder_filter_outlook -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_lb_http_ex2016_oab -policyName pol_responder_filter_outlook
-priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_lb_http_ex2016_mapi -policyName pol_responder_filter_outlook
-priority 100 -gotoPriorityExpression END -type REQUEST
```

6.6. SMTP, IMAP, POP3

Be sure that you use USIP to forward the client IPs to the Exchange servers.

LB vServers

```
add lb vserver vs_lb_tcp_ex2016_pop3 TCP ${EX_VIP} 110 -persistenceType NONE -state
DISABLED -cltTimeout 180
add lb vserver vs_lb_tcp_ex2016_imap TCP ${EX_VIP} 143 -persistenceType NONE -state
DISABLED -cltTimeout 180
add lb vserver vs_lb_ssl_tcp_ex2016_imaps SSL ${EX_VIP} 993 -persistenceType NONE
-state DISABLED -cltTimeout 180
add lb vserver vs_lb_ssl_tcp_ex2016_pop3s SSL ${EX_VIP} 995 -persistenceType NONE
-state DISABLED -cltTimeout 180
add lb vserver vs_lb_tcp_ex2016_smtp TCP ${EX_VIP} 25 -persistenceType NONE -state
DISABLED -cltTimeout 180
add lb vserver vs_lb_ssl_tcp_ex2016_smtps SSL ${EX_VIP} 465 -persistenceType NONE
-state DISABLED -cltTimeout 180
add lb vserver vs_lb_tcp_ex2016_msa TCP ${EX_VIP} 587 -persistenceType NONE -state
DISABLED -cltTimeout 180
```

@TODO: configuration of service groups

@TODO: configuration of peristence

Chapter 7. NetScaler Gateway

7.1. ICA only

```
add vpn sessionAction action_session_receiver -splitTunnel OFF
-transparentInterception OFF -defaultAuthorizationAction ALLOW -SSO ON -icaProxy ON
-wihome "https://${SF_FQDN}/Citrix/StoreWeb" -ClientChoices OFF -ntDomain CTXDEMO
-clientlessVpnMode OFF -storefronturl "https://${SF_FQDN}"
add vpn sessionAction action_session_web -transparentInterception OFF
-defaultAuthorizationAction ALLOW -SSO ON -homePage
"https://${SF_FQDN}/Citrix/StoreWeb" -icaProxy ON -wihome
"http://${SF_FQDN}/Citrix/StoreWeb" -ClientChoices OFF -ntDomain CTXDEMO
-clientlessVpnMode OFF

add vpn sessionPolicy pol_session_receiver "REQ.HTTP.HEADER User-Agent CONTAINS
CitrixReceiver" action_session_receiver
add vpn sessionPolicy pol_session_web ns_true action_session_web
```

```
add vpn vserver vs_vpn_citrix SSL ${VIP_GW} 443 -icaOnly ON -downStateFlush DISABLED
-Listenpolicy NONE
```

```
bind vpn vserver vs_vpn_citrix -staServer "http://${XDC1_FQDN}"
bind vpn vserver vs_vpn_citrix -staServer "http://${XDC2_FQDN}"
```

```
bind vpn vserver vs_vpn_citrix -policy pol_auth_ldap
```

```
bind vpn vserver vs_vpn_citrix -policy pol_session_receiver -priority 100
bind vpn vserver vs_vpn_citrix -policy pol_session_web -priority 200
```

Workaround: Inject internal Storefront FQDN into each Requests Sometimes the request to Storefront is sent with the external FQDN. This causes that the request is dropped in some situations (e.g. with Loadbalancing on a Sophos UTM).

```
add rewrite action act_rewrite_hostname replace HTTP.REQ.HOSTNAME "${SF_FQDN}"
add rewrite policy pol_rewrite_hostname true act_rewrite_hostname
bind vpn vserver vs_vpn_citrix -policy pol_rewrite_hostname -priority 100
-gotoPriorityExpression END -type REQUEST
```

7.2. SmartAccess

Rewriting of Storefront Cookies need to be disabled.

```
add policy patset patset_storefront_cookies
bind policy patset patset_storefront_cookies CsrfToken -index 1
bind policy patset patset_storefront_cookies ASP.NET_SessionId -index 2
bind policy patset patset_storefront_cookies CtxsPluginAssistantState -index 3
bind policy patset patset_storefront_cookies CtxsAuthId -index 4
add vpn clientlessAccessProfile profile_clientless_storefront
set vpn clientlessAccessProfile profile_clientless_storefront -URLRewritePolicyLabel
ns_cvpn_default_inet_url_label -ClientConsumedCookies patset_storefront_cookies
add vpn clientlessAccessPolicy pol_clientless_storefront true
profile_clientless_storefront
```

Setting for Storefront (web.config) to allow use in HTML Frames.

```
<add name="X-Frame-Options" value="SAMEORIGIN" />
<add name="Content-Security-Policy" value="frame-ancestors 'self'" />
```

```
add vpn sessionPolicy pol_session_web_portal ns_true profile_session_web_portal
add vpn sessionAction profile_session_web_portal -sessTimeout 1440
-transparentInterception ON -defaultAuthorizationAction ALLOW -clientIdleTimeout 1440
-clientCleanupPrompt OFF -forceCleanup none -clientConfiguration trace -SSO ON
-homePage none -icaProxy ON -wihome "https://${SF_FQDN}/Citrix/LabWeb"
-citrixReceiverHome "https://${SF_FQDN}/Citrix/LabWeb" -wiPortalMode NORMAL
-ClientChoices ON -ntDomain ${SSO_DOMAIN} -clientlessVpnMode OFF -emailHome
"https://${EX_FQDN}/owa" -clientlessModeUrlEncoding TRANSPARENT -storefronturl
"https://${SF_FQDN}/Citrix/Lab" -rdpClientProfileName profile_rdp_client_default
-iconWithReceiver ON
```

Chapter 8. XenMobile

8.1. MDM Loadbalancing

```
# Server
add server ${XMS_FQDN} ${XMS_IP}
```

Servicegroups

```
add serviceGroup sg_xm_mdm SSL_BRIDGE -maxClient 0 -maxReq 0 -usip NO -useproxyport
YES -cltTimeout 180 -svrTimeout 360 -CKA YES -TCPB NO -CMP NO
add serviceGroup sg_xm_mdm_ios SSL_BRIDGE -maxClient 0 -maxReq 0 -cacheable YES -usip
NO -useproxyport YES -cltTimeout 180 -svrTimeout 360 -CKA YES -TCPB NO -CMP NO
```

Server Binding

```
bind serviceGroup sg_xm_mdm ${XMS_FQDN} 443
bind serviceGroup sg_xm_mdm_ios ${XMS_FQDN} 8443
```

Monitor

```
bind serviceGroup sg_xm_mdm -monitorName tcp
bind serviceGroup sg_xm_mdm_ios -monitorName tcp
```

vServers

```
add lb vserver vs_lb_ssl_bridge_xm_mdm SSL_BRIDGE ${MDM_VIP} 443 -persistenceType
SSLSESSION -cltTimeout 180
add lb vserver vs_lb_ssl_bridge_xm_mdm_ios SSL_BRIDGE ${MDM_VIP} 8443 -persistenceType
SSLSESSION -cltTimeout 180
```

Servicegroup Binding

```
bind lb vserver vs_lb_ssl_bridge_xm_mdm sg_xm_mdm
bind lb vserver vs_lb_ssl_bridge_xm_mdm_ios sg_xm_mdm_ios
```

8.2. MAM Loadbalancing

```
# Servicegroup
add serviceGroup sg_xm_mam_http HTTP -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO

# Server Binding
bind serviceGroup sg_xm_mam_http ${XMS_FQDN} 80

# Monitor
bind serviceGroup sg_xm_mam_http -monitorName tcp

# vServer
add lb vserver vs_lb_xm_mam SSL ${MAM_LB_VIP} 8443 -persistenceType SOURCEIP
-cltTimeout 180

# Service Binding
bind lb vserver vs_lb_xm_mam sg_xm_mam_http
```

8.3. MAM Gateway

```
# Intranet Domains (Client VPN)
bind vpn global -intranetDomain customer.local

# Intranet Domains (Clientless VPN)
bind policy patset ns_cvpn_default_inet_domains customer.local -index 2
bind policy patset ns_cvpn_default_inet_domains mdm.customer.com -index 3
bind policy patset ns_cvpn_default_inet_domains mdm.customer.com:8443 -index 4

# Traffic fuer mdm.customer.com an den MAM Loadbalancer schicken
add dns addRec mdm.customer.com ${MAM_LB_VIP}

# Storefront Cookies Patternset
add policy patset storefront_cookies
bind policy patset storefront_cookies CsrfToken -index 1
bind policy patset storefront_cookies ASP.NET_SessionId -index 2
bind policy patset storefront_cookies CtxsPluginAssistantState -index 3
bind policy patset storefront_cookies CtxsAuthId -index 4

# Clientless Access Rewrite
add vpn clientlessAccessProfile prof_clientless_rewrite_sf
add vpn clientlessAccessProfile prof_clientless_norewrite
set vpn clientlessAccessProfile prof_clientless_rewrite_sf -URLRewritePolicyLabel
ns_cvpn_default_inet_url_label -ClientConsumedCookies storefront_cookies
add vpn clientlessAccessPolicy pol_clientless_rewrite_sf true
prof_clientless_rewrite_sf
add  vpn clientlessAccessPolicy pol_clientless_norewrite "HTTP.REQ.HEADER(\"User-
Agent\").CONTAINS(\"CitrixReceiver\") &&  HTTP.REQ.HEADER(\"X-Citrix-
Gateway\").EXISTS" prof_clientless_norewrite
```

```
# Session Policies
add vpn sessionAction act_session_xenmobile_os -splitDns BOTH -sessTimeout 1440
-splitTunnel OFF -transparentInterception ON -defaultAuthorizationAction ALLOW -SSO ON
-ssoCredential PRIMARY -icaProxy OFF -ClientChoices OFF -forcedTimeout 1440
-clientlessVpnMode ON -clientlessModeUrlEncoding TRANSPARENT -SecureBrowse ENABLED
-storefronturl "https://mdm.customer.com:8443"
add vpn sessionAction act_session_xenmobile_web -defaultAuthorizationAction ALLOW -SSO
ON -ssoCredential PRIMARY -homePage "https://mdm.customer.com:8443/Citrix/StoreWeb"
-icaProxy OFF -wihome "https://mdm.customer.de:8443/Citrix/StoreWeb" -ClientChoices
OFF -clientlessVpnMode ON -SecureBrowse ENABLED
add  vpn sessionAction act_session_xenmobile_ag -splitDns BOTH -splitTunnel OFF
-transparentInterception ON -defaultAuthorizationAction ALLOW -SSO ON -ssoCredential
PRIMARY -homePage "https://mdm.customer.com:8443/Citrix/StoreWeb"
 -icaProxy OFF -ClientChoices OFF -clientlessVpnMode OFF -clientlessModeUrlEncoding
TRANSPARENT -SecureBrowse ENABLED -storefronturl "https://mdm.customer.com:8443"
add  vpn sessionPolicy pol_session_xenmobile_os "REQ.HTTP.HEADER User-Agent CONTAINS
CitrixReceiver && REQ.HTTP.HEADER X-Citrix-Gateway EXISTS" act_session_xenmobile_os
add vpn sessionPolicy pol_session_xenmobile_web "REQ.HTTP.HEADER User-Agent
NOTCONTAINS CitrixReceiver && REQ.HTTP.HEADER Referer EXISTS"
act_session_xenmobile_web
add vpn sessionPolicy pol_session_xenmobile_ag "REQ.HTTP.HEADER User-Agent NOTCONTAINS
CitrixReceiver && REQ.HTTP.HEADER Referer NOTEXISTS" act_session_xenmobile_ag

# VPN vServer
add vpn vserver vs_vpn_xm_mam_gateway SSL ${MAM_VIP} 443 -Listenpolicy NONE
-cginfraHomePageRedirect DISABLED

# Binding Auth (UPN)
bind vpn vserver vs_vpn_xm_mam_gateway -policy pol_auth_ldaps_lm_xenmobile -priority
100

# Binding Session Policies
bind vpn vserver vs_vpn_xm_mam_gateway -policy pol_session_xenmobile_os -priority 100
bind vpn vserver vs_vpn_xm_mam_gateway -policy pol_session_xenmobile_web -priority 110
bind vpn vserver vs_vpn_xm_mam_gateway -policy pol_session_xenmobile_ag -priority 120

# Binding Clientless Access Policies
bind vpn vserver vs_vpn_xm_mam_gateway -policy prof_clientless_norewrite -priority 80
-gotoPriorityExpression END -type REQUEST
bind vpn vserver vs_vpn_xm_mam_gateway -policy pol_clientless_rewrite_sf -priority 100
-gotoPriorityExpression END -type REQUEST

# STA; Sharefile, AppController (obsolet)
bind vpn vserver vs_vpn_xm_mam_gateway -staServer "http://mdm.customer.com:8443"
bind vpn vserver vs_vpn_xm_mam_gateway -appController "https://mdm.customer.com:8443"
bind vpn vserver vs_vpn_xm_mam_gateway -sharefile mdm.customer.com:8443

# Default Cache Policies
bind vpn vserver vs_vpn_xm_mam_gateway -policy _cacheTCVPNStaticObjects -priority 10
-gotoPriorityExpression END -type REQUEST
```

```
bind vpn vserver vs_vpn_xm_mam_gateway -policy _cacheOCVPNStaticObjects -priority 20
-gotoPriorityExpression END -type REQUEST
bind vpn vserver vs_vpn_xm_mam_gateway -policy _cacheVPNStaticObjects -priority 30
-gotoPriorityExpression END -type REQUEST
bind vpn vserver vs_vpn_xm_mam_gateway -policy _noCacheRest -priority 40
-gotoPriorityExpression END -type REQUEST
bind vpn vserver vs_vpn_xm_mam_gateway -policy _cacheWFStaticObjects -priority 10
-gotoPriorityExpression END -type RESPONSE
```

Chapter 9. ShareFile StorageZone

```
add server %{SZC_FQDN} ${SZC_IP}
```

```
add serviceGroup sg_ssl_sharefile SSL -maxClient 0 -maxReq 0 -cacheable YES -cip
ENABLED X-Forwarded-For -usip NO -useproxyport YES -cltTimeout 180 -svrTimeout 360
-CKA YES -TCPB NO -CMP NO
```

```
bind serviceGroup sg_ssl_sharefile %{SZC_FQDN} 443
```

```
add lb vserver vs_lb_http_sharefile_storagezone HTTP 0.0.0.0 0 -persistenceType
SOURCEIP -lbMethod TOKEN -rule "http.REQ.URL.QUERY.VALUE(\"uploadid\")" -Listenpolicy
NONE -cltTimeout 180
add lb vserver vs_lb_http_sharefile_cifs_sp HTTP 0.0.0.0 0 -persistenceType
COOKIEINSERT -timeout 240 -Listenpolicy NONE -cltTimeout 180 -authn401 ON -authnVsName
vs_aaa_sharefile_auth
add lb vserver vs_lb_http_sf_zone_options HTTP 0.0.0.0 0 -persistenceType SOURCEIP
-Listenpolicy NONE -cltTimeout 180
```

```
bind lb vserver vs_lb_ssl_sharefile_storagezone sg_http_sharefile
bind lb vserver vs_lb_ssl_sharefile_cifs_sp sg_http_sharefile
bind lb vserver vs_lb_ssl_sf_zone_options sg_http_sharefile
```

```
add cs vserver vs_cs_ssl_sharefile SSL ${SZC_LB_VIP} 443 -cltTimeout 180 -Listenpolicy
NONE
add cs vserver vs_cs_http_sharefile_redirect_to_ssl HTTP ${SZC_LB_VIP} 80 -cltTimeout
180 -Listenpolicy NONE
```

```
add authentication vserver vs_aaa_sharefile_auth SSL ${AAA_VIP} 443
add cs vserver vs_cs_http_shareauth_redirect_to_ssl HTTP ${AAA_VIP} 80 -cltTimeout 180
-Listenpolicy NONE
```

```
add policy httpCallout callout_sharefile -vServer vs_lb_http_sharefile_storagezone
-returnType BOOL -hostExpr "\"ShareFile\"" -urlStemExpr
"\"/validate.ashx?RequestURI=\" +
HTTP.REQ.URL.BEFORE_STR(\"&h\").HTTP_URL_SAFE.B64ENCODE + \"&h=\"+
HTTP.REQ.URL.QUERY.VALUE(\"h\")" -scheme http -resultExpr
"HTTP.RES.STATUS.EQ(200).N192OT"
add policy httpCallout callout_sharefile_y -vServer vs_lb_http_sharefile_storagezone
-returnType BOOL -hostExpr "\"ShareFile\"" -urlStemExpr
"\"/validate.ashx?RequestURI=\" + HTTP.REQ.URL.HTTP_URL_SAFE.B64ENCODE + \"&h=\""
-scheme http -resultExpr "HTTP.RES.STATUS.EQ(200).NOT"
set policy httpCallout callout_sharefile -vServer vs_lb_http_sharefile_storagezone
-returnType BOOL -hostExpr "\"ShareFile\"" -urlStemExpr
"\"/validate.ashx?RequestURI=\" +
HTTP.REQ.URL.BEFORE_STR(\"&h\").HTTP_URL_SAFE.B64ENCODE + \"&h=\"+
HTTP.REQ.URL.QUERY.VALUE(\"h\")" -scheme http -resultExpr
"HTTP.RES.STATUS.EQ(200).NOT"
set policy httpCallout callout_sharefile_y -vServer vs_lb_http_sharefile_storagezone
-returnType BOOL -hostExpr "\"ShareFile\"" -urlStemExpr
"\"/validate.ashx?RequestURI=\" + HTTP.REQ.URL.HTTP_URL_SAFE.B64ENCODE + \"&h=\""
-scheme http -resultExpr "HTTP.RES.STATUS.EQ(200).NOT"
```

```
add responder policy pol_responder_sharefile "HTTP.REQ.URL.CONTAINS(\"&h=\") &&
HTTP.REQ.URL.CONTAINS(\"/crossdomain.xml\").NOT&&
HTTP.REQ.URL.CONTAINS(\"/validate.ashxrequri\").NOT&&
SYS.HTTP_CALLOUT(callout_sharefile) || HTTP.REQ.URL.CONTAINS(\"&h=\").NOT &&
HTTP.REQ.URL.CONTAINS(\"/crossdomain.xml\").NOT&&
HTTP.REQ.URL.CONTAINS(\"/validate.ashxrequri\").NOT&&
SYS.HTTP_CALLOUT(callout_sharefile_y)" DROP
```

Import the SAML certificate from sharefile.com.

```
add ssl certKey saml_sharefile.com -cert saml_sharefile.com.crt
```

```
add authentication samlIdPProfile profile_auth_saml_idp_sharefile -samlSPCertName
${SF_SUBDOMAIN}.sharefile.com -samlIdPCertName sharefile.customer.de
-assertionConsumerServiceURL "https://customer.sharefile.com/saml/acs" -samlIssuerName
"https://shareauth.customer.de" -rejectUnsignedRequests OFF -audience
"https://customer.sharefile.com"
add authentication samlIdPPolicy pol_auth_saml_idp_sharefile -rule
"HTTP.REQ.URL.CONTAINS(\"saml\")" -action profile_auth_saml_idp_sharefile

add cs action act_cs_sharefile_options -targetLBVserver vs_lb_http_sf_zone_options
add cs policy pol_cs_sharefile_options -rule "HTTP.REQ.METHOD.EQ(\"OPTIONS\")" -action
act_cs_sharefile_options

add cs policy pol_cs_sharefile_not_cifs_sp_proxy -rule
"HTTP.REQ.URL.CONTAINS(\"/cifs/\").NOT && HTTP.REQ.URL.CONTAINS(\"/sp/\").NOT ||
HTTP.REQ.URL.CONTAINS(\"/ProxyService/\").NOT"
add cs policy pol_cs_sharefile_cifs_sp_proxy -rule "HTTP.REQ.URL.CONTAINS(\"/cifs/\")
|| HTTP.REQ.URL.CONTAINS(\"/sp/\") || HTTP.REQ.URL.CONTAINS(\"/ProxyService/\") "

bind lb vserver vs_lb_http_sharefile_storagezone sg_http_sharefile
bind lb vserver vs_lb_http_sharefile_cifs_sp sg_http_sharefile
bind lb vserver vs_lb_http_sf_zone_options sg_http_sharefile
bind lb vserver vs_lb_http_sharefile_storagezone -policyName pol_responder_sharefile
-priority 100 -gotoPriorityExpression END -type REQUEST
bind cs vserver vs_cs_ssl_sharefile -policyName pol_cs_sharefile_options -priority 80
bind cs vserver vs_cs_ssl_sharefile -policyName pol_cs_sharefile_cifs_sp_proxy
-targetLBVserver vs_lb_http_sharefile_cifs_sp -priority 90
bind cs vserver vs_cs_ssl_sharefile -policyName pol_cs_sharefile_not_cifs_sp_proxy
-targetLBVserver vs_lb_http_sharefile_storagezone -priority 100
bind cs vserver vs_cs_http_sharefile_redirect_to_ssl -policyName
pol_responder_generic_redirect_ssl -priority 100 -gotoPriorityExpression END -type
REQUEST


add tm sessionAction act_session_sharefile -SSO ON -ssoCredential PRIMARY -ssoDomain
CUSTOMER -homePage "https://${SF_SUBDOMAIN}.sharefile.com/saml/login"
add tm sessionPolicy pol_session_sharefile ns_true act_session_sharefile

bind authentication vserver vs_aaa_sharefile_auth -policy pol_ldap_sharefile -priority
100
bind authentication vserver vs_aaa_sharefile_auth -policy pol_session_sharefile
-priority 100
bind authentication vserver vs_aaa_sharefile_auth -policy pol_auth_saml_idp_sharefile
-priority 100 -gotoPriorityExpression END
```

Chapter 10. SMS Passcode

RADIUS configuration for SMS Passcode, also known as CensorNet MFA.

```
add authentication radiusAction act_radius_smspasscode -serverName ${SMSPC_SERVER}
-serverPort 1812 -radKey ${SMSPC_PSK} -radVendorID 1 -radAttributeType 99
-radGroupsPrefix CTXUserGroups= -radGroupSeparator "," -accounting ON
-callingstationid ENABLED
add authentication radiusPolicy pol_radius_smspasscode ns_true act_radius_smspasscode
```

10.1. Hide second password field

Hide second password field via HTTP header for Citrix Receiver.

```
add rewrite action act_rewrite_auth_type_SMS insert_http_header X-Citrix-AM-
GatewayAuthType "\"SMS\""
add rewrite policy pol_rewrite_auth_type_SMS true act_rewrite_auth_type_SMS
```

```
add rewrite action act_rewrite_auth_type_CertAndRSA insert_http_header X-Citrix-AM-
GatewayAuthType "\"CertAndRSA\""
add rewrite policy pol_rewrite_auth_type_CertAndRSA true
act_rewrite_auth_type_CertAndRSA
```

Hide second password field via Cookie (Webbrowser).

```
add rewrite action act_rewrite_pwcount_cookie insert_http_header Set-Cookie
"\"pwcount=0\""
add rewrite policy pol_rewrite_pwcount_cookie true act_rewrite_pwcount_cookie
```

Chapter 11. Preauth for VMware vCenter (vCSA)

- Publishing VMware vSphere 6.5 vCSA HTML5 UI behind NetScaler
- Rewriting the internal dnsname for external access
- Preauth with NetScaler AAA (SSO: to be done)

11.1. Loadbalancing

Server

```
add server ${VCSA_FQDN} ${VCSA_IP}
```

Servicegroup

```
add serviceGroup sg_ssl_vcsa SSL -cip ENABLED X-Forwarded-For
```

Server Binding

```
bind serviceGroup sg_ssl_vcsa ${VCSA_FQDN} 443
```

vServer

```
add lb vserver vs_lb_http_vcsa HTTP 0.0.0.0 0 -persistenceType NONE -cltTimeout 180
```

Servicegroup Binding

```
bind lb vserver vs_lb_http_vcsa sg_ssl_vcsa
```

11.2. Content Switching

vServer

```
add cs vserver vs_cs_ssl_vcsa SSL ${VCSA_VIP} 443 -cltTimeout 180
add cs vserver vs_cs_http_vcsa HTTP ${VCSA_VIP} 80 -cltTimeout 180
```

CS Policy

```
add cs policy pol_cs_vcsa_ui -rule "HTTP.REQ.URL.STARTSWITH(\"/websso/\") ||
HTTP.REQ.URL.STARTSWITH(\"/ui/\")" -action act_cs_vcsa_ui
add cs action act_cs_vcsa_ui -targetLBVserver vs_lb_http_vcsa_websso
```

Enable Websockets

```
add ns httpProfile profile_http_websockets -dropInvalReqs ENABLED -markHttp09Inval
ENABLED -conMultiplex DISABLED -webSocket ENABLED
set cs vserver vs_cs_ssl_vcsa -httpProfileName profile_http_websockets
```

11.3. URL Rewriting

Transform Policy

```
add transform profile prof_transform_vcsa
add transform action act_transform_vcsa prof_transform_vcsa 100
set transform action act_transform_vcsa -priority 100 -reqUrlFrom vcsa.example.com
-reqUrlInto vcenter01.example.local -resUrlFrom vcenter01.example.local -resUrlInto
vcsa.example.com
add transform policy pol_transform_vcsa "HTTP.REQ.URL.STARTSWITH(\"/ui/login\") ||
HTTP.REQ.URL.STARTSWITH(\"/ui/logout\") || HTTP.REQ.URL.STARTSWITH(\"/ui/saml\") ||
HTTP.REQ.URL.STARTSWITH(\"/websso/\")" prof_transform_vcsa
```

Binding Policy

```
bind lb vserver vs_lb_http_vcsa -policyName pol_transform_vcsa -priority 100
-gotoPriorityExpression END -type REQUEST
```

11.4. Preauth

Traffic Policy (Logout)

```
add tm trafficAction prof_traffic_vcsa_logout -persistentCookie OFF -InitiateLogout ON
-kcdAccount NONE
add tm trafficPolicy pol_traffic_vcsa_logout "HTTP.REQ.URL.EQ(\"/ui/logout\")"
prof_traffic_vcsa_logout
```

Traffic Policy Binding (also possible on cs level)

```
bind lb vserver vs_lb_http_vcsa -policyName pol_traffic_vcsa_logout -priority 100
-gotoPriorityExpression END -type REQUEST
```

Chapter 12. External links

Common

- Netscaler HowTo Guides - Common Configuration HowTo guides
- NetScaler Deployment Guides - Some offical Deployment Guides by Citrix

CTX Articles

- CTX214033 - NetScaler Networking and VLAN Best Practices
- CTX121149 - Recommended Settings and Best Practices for Generic Implementation of a NetScaler Appliance
- CTX200278 - NetScaler VPX Loses Network Connectivity on VMware ESXi 5.1.0 2191751, VMware ESXi 5.5 2143827 and also on VMware ESXi 6.0
- CTX224576 - NetScaler VPX Loses Network Connectivity Intermittently on VMware ESXi After Upgrading to Version 12.0
- CTX201949 - One Public IP for AAA-TM Deployments on NetScaler
- CTX138055 - How to Force Secure and HttpOnly Cookie Options for Websites Using NetScaler Appliance
- CTX205578 - Back-End Connection on TLS 1.1/1.2 from NetScaler to IIS Server Breaks
- CTX225681 - Large File Uploads Fails on NetScaler with Content Length 0 POST Requests

SSL Tips

- https://testssl.sh/ - Shell script for testing TLS/SSL encryption
- https://cipherli.st/ - Strong Ciphers for Apache, nginx and Lighttpd
- https://www.ssllabs.com/ssltest/ - Qualys SSL Labs server test
- https://observatory.mozilla.org/ - Mozilla Observatory test

Monitoring

- check_netscaler - check_netscaler Nagios Plugin
- check_netscaler_gateway - check_netscaler_gateway Nagios Plugin
- check_nsupdates - check_nsupdates Nagios Plugin