What’s new in Domino 12.0.2 Security

OpenNTF, Feb 2023

Daniel Nashed -- https://blog.nashcom.de
Agenda

• Basic Agenda
  - Domino security
  - e-mail security
  - VSS Backup Writer support on Windows
  - Demos & examples powered by DNUG LAB

• Disclaimer
  - Not a complete list of all new features!
  - Detailed slides are available for reference
Domino on Linux & Docker

• Platform support
  – Support for **RHEL 9.x** and **SLES 15.3/15.4**
  – Support for Linux **Kernel 5.x**
  – Support for **SELinux** in **enforced mode**

• Domino on Docker
  – New Container image based on the HCL Community image including Nash!Com Domino Start Script
  – Special build on RedHat Universal Base Image 8.6 (Traveler and Domino Leap)
  – The community project offers still many more options
    ▫ Including **Nomad Server**, **Verse**, **REST API** install option
Important Software Package Updates

• OpenSSL 3.0.5
  - New major OpenSSL version
  - Modular design helps with **FIPS 140-2** support
    ▫ [https://www.openssl.org/blog/blog/2022/08/24/FIPS-validation-certificate-issued/](https://www.openssl.org/blog/blog/2022/08/24/FIPS-validation-certificate-issued/)
  - Starting with Notes/Domino 12.0.2 OpenSSL is linked into core with **no separate .dll/.so files**!

• LibCurl 7.83.0
  - Important package, leveraging OpenSSL
  - Linked into core Notes/Domino since 10.x
  - Used from Lotus Script and also in the back-end for other features (CertMgr, OIDC)

• Apache Tika 2.4.1
  - Used for attachment filtering when full text indexing attachments

• **Packages are newer than in most Linux distributions!**
New Trusted Roots

- Imported from LibCurl /local/notesdata/cacert.pem
- Additional information added to new Certifier documents for Internet Trusted Roots
Improved Import/Export Dialogs

• Requires **Notes 12.0.2 Client** and fully supports ECDSA certificates
CertMgr & CertStore

Domain wide trusted root, private key & certificate management
Before Domino 12: kyr files, kyrtool & OpenSSL

- Domino used *.kyr file format for Internet Certificates
  - Old IBM format nobody else can read or write
  - The only tool available to read and write is “kyrtool”
    - Very flexible but command line driven -- Not always easy to handle
    - Replaced old certreq.nsf database which wasn’t easy to use either

- Creating keys and CSR required an external tool like “openssl” on command line
  - Very powerful, but also very cryptic tool with confusing command line for most admins

- *.kyr files have a corresponding *.sth containing the encoded password
  - Can be decoded with simple perl script

- Old kyrfile cache for internet processes always needed restart for any *.kyr change
certstore.nsf

- **Domain wide** database managed by **CertMgr** task

- **Secure**, automated deployment for TLS Credentials and trusted roots

- Private keys are **encrypted** with CertMgr server and the server specified in field “Servers with access:”
  - Special designed Vault style encryption with new API

- Easy to use with modern interface

- **CertMgr servertask** is only supported on **W64** and **Linux64**
  - **AIX** and **OS400** can still leverage **certstore.nsf** and the new TLS Cache
    - Create replica manually
  - New in Domino 12.0.2: **Full support for AIX**

- AIX and OS400 can still leverage certstore.nsf and the new TLS Cache
  - Create replica manually
  - New in Domino 12.0.2: Full support for AIX
Create Domain wide certstore.nsf

• **First** server in domain starting the “certmgr” servertask is setup as the CertMgr Server
  - Checks the Domino **directory profile** on **admin server** for an existing CertMgr server
  - If no server exists automatically creates the domain wide **certstore.nsf** database
  - Updates the directory profile on admin server to propagate the CertMgr server in the domain

• Starting the certmgr servertask on any **additional** server in the domain creates a replica
  - Each additional server acts like a “**CertMgr client**” and will just replicate the database every 2 minutes
  - Keeping the CertMgr servertask loaded is an **optional** convenience step
  - Any type of replication setup which ensures a short replication cycle can be used as well
• **TLS Credential** = private key + leaf certificate + chain (intermediates) + trusted root

• Replaces “*.kyr files”
  - Stored in **PEM** format (text with base64 encoded data)

• Can be created via
  - **ACME V2** protocol (Let's Encrypt & others)
  - Manual flows including import
  - Domino MicroCA (exportable in 12.0.2)

• Specify trusted roots used for client certificate verification
  - Used to be hidden in **kyr-file** and was difficult to manage
Manual Certificate Operations

1. **CertMgr** processes submitted requests and creates
   - Private key (RSA or ECDSA)
     - Saved locally encrypted for assigned servers
   - CSR (Certificate Signing Request) signed by private key → PEM

2. Admin copies CSR to CA

3. Admin imports certificate & chain (PEM) back
   - Paste full chain in any order and submits the form again

Duplicate certs are ignored

Missing intermediate certs and root are automatically added from “Trusted Roots” in **certstore.nsf**
certstore.nsf – Trusted Roots

- Stored in trusted, secured certstore.nsf
  - Replicated domain wide
- Used for client cert verification
- And auto complete certificate chains
  - ACME and manual flows
- Certificate chains are automatically sorted & completed
  - **Private Key → matching leaf certificate**
    → **intermediate certs** in the right order → **trusted root**
- Tip: you can import intermediate certificates as “Trusted Root” to be used to auto complete chains
Domino 12.0.2 Trusted Roots

- New certificate categories to assign trusted roots to applications like ICAP and OIDC
  - Can be used to restrict root certificates to a specific use cases

- Additional certificate details added
  - Curve name, SHA256 Fingerprint, Key usage, …

- Easier to navigate view with categories
  - Hierarchical certificate information adapted from Domino directory, was confusing
Domino 12.0.2 cerstore.nsf

• Automatically created on new servers with **One Touch Setup**
  - First server in a domain is always the `certmgr` server when setup with One Touch Setup
  - Additional servers replicate `certstore.nsf` from their setup server when setup with One Touch Setup

• Default process interval is now **2 seconds** instead of **30 seconds**
  - Important for remote request mode (like JConsole)
HCLSoftware

Domino Micro CA
Simple internal Micro CA

- If Let’s Encrypt nor an internal CA is available …
- Or you want a simple local CA for test or first server setup
- Domino 12.0.1 introduced a simple "Micro CA"
  - Managed by CertMgr
  - Available via One-touch setup for the first server in the domain
  - Or directly from certstore.nsf at any time issuing a certificate from the local CA
- Not a full CA – Only intended for testing & setup!!

![Certificate Authority](image1)

![TLS Credentials](image2)
Domino 12.0.2 Micro CA

• New created MicroCAs are **10 years** valid instead of **1 year**
  - Now supports exportable private keys!
    ▫ Can be used outside Domino if created exportable

• MicroCA is also used for **JConsole** certificates

• Remote request mode
  - Server posts request into cerstore.nsf on CertMgr server
  - CertMgr Server process request
  - Remote server polls until key & certificate is created
Domino 12.0.2 JConsole Certs

- New JConsole certificates issued by Micro CA for the server’s hostname
  - Server cert → 10 years
  - Client cert → 2 years

- Server certs are always created via command-line
  - `certmgmt create mca controller myhost.example.com`
  - `certmgmt create mca console myhost.mydomain.com`

- Client certs can be also created via UI
  - Requires exportable key with password
  - Keys are always RSA 2048 (Java 1.8 only supports RSA)

- Note: New servers automatically create JConsole TLS credentials when setup via OneTouch Setup
Domino 12.0.1+ Exportable Private Keys

- Keys create or imported into certstore.nsf are encrypted for CertMgr server and servers listed in: “Server with access”

- By design those keys cannot be exported!

- But you can create an exportable key
  - Stored in encrypted PEM format in a separate field
  - Always encrypted with a password with reasonable entropy

- Full export & import dialog for PEM, PKCS12 & KYR (import only)

- All import functionality provides
  - Certificate chain auto sorting and filtering
  - Certificate chain completion from trusted roots (even multi level)
CertMgr Certificate
URL Health Check
CertMgr Certificate URL Health Check

• Can be configured in each **TLS Credentials doc** to check certificate health on servers

• Supports all standard TLS connections (HTTPS, LDAPS, IMAPS, POP3S, …)
  - Does not support SMTP STARTTLS which starts the connection unencrypted on port 25

• Check performed once per day

• Manual check via: **tell certmgr check**

• Can send daily notification e-mail and writes statistics
Mail & Log Example

CertMgr URL Health Check - Failures: 1, Warnings: 3

**Linus.lab.dnug.eu** to Admin

Certificate failures (1)

**Icap.lab.dnug.eu:1344** - Failed to connect to icap.lab.dnug.eu port 1344 after 12 ms. Connection refused

Certificate expiration warnings (3)

**Linus.lab.dnug.eu** (37.4 days)

**Traveler.lab.dnug.eu** (37.4 days)

**www.lab.dnug.eu** (37.4 days)

```
$ tell cmgr check
> [003345:00006-00007FCE9E68EC80] CertMgr: Checking ...
[003345:00006-00007FCE8A1FB700] Checking for requests ...
: Connection refused
[003239:000017-00007F5450357007] 10/06/2022 05:18:59 Router: Message 001D342D delivered to Admin/dnug-lab
```
New Health Check URL Statistics

- Can be used to generate custom notifications via event monitoring
- Read/Yellow/Green status similar to CertStatus
OpenID Connect 1.0 (OIDC) Authentication

- Allows to use OpenID Connect 1.0 (OIDC) compliant 3rd party IdPs for authentication
  - Check [https://openid.net](https://openid.net) for details
  - “Similar” to SAML but easier to configure & more modern

- Tested providers
  - KeyCloak
  - Google
  - Yahoo
  - Microsoft Azure AD

- Untested providers (Any volunteers?)
  - Microsoft ADFS 2019+ (On-prem)
  - Okta (On-prem)
  - PingFederate (On-prem)
  - Salesforce (per customer)
Providers known to not work (and why)

• **Apple-ID**
  - Doesn't support `client_secret_basic`
  - Apple uses a custom variant of `private_key_jwt` authentication

• **AWS IAM Identity Center (successor to AWS Single Sign On)**
  - Does not support the Authorization Code Flow with PKCE

• **Facebook**
  - Does not currently support the authorization code flow and does not expose a token endpoint

• **GitHub**
  - Supports OAuth, but no well-known endpoint and will not return an `id_token`

• **Twitter**
  - Does not support OIDC
OIDC / OpenID Authentication

• Enabled in internet site document

• Requires a OIDC document in `idpcat.nsf`
  - Provider needs to support the full OIDC standard and have a valid `.well-known/openid-configuration`

• Requires end to end TLS encryption!
  - In case of TLS termination on secure proxy, use separate TLS connection between proxy and Domino
  - Tip: Domino MicroCA can be used to issue certificates
  - Only 1 year valid, but auto renewed by CerMgr
Google OpenID Configuration


- Required
  - issuer
  - authorization_endpoint
  - token_endpoint
  - jwks_uri

- Only specify the URL without `.well-known`.
  - The server always uses this standard location!
Trusted Root Configuration

• Trusted root configuration is optional
  - Without trusted root the underlying LibCurl code uses `cacert.pem` in server’s data directory

• Trusted root is selected from `certstore.nsf`

• Tip: Import trusted roots

  - `load certmgr -ImportRootFromUrl https://accounts.google.com/.well-known/openid-configuration OIDC`
    - Checks the remote site and creates a new draft trusted root document for OIDC use
    - If remote site does not send a trusted root, certificate chain is checked against Domino directory to auto complete the chain and add the trusted root into `certstore.nsf`
    - Trusted root needs to be verified in `certstore.nsf` before it can be used
Trusted Root Validation

**Main**
- **Status:** Pending Validation
- **Name:** CN=GlobalSign Root CA/OU=Root CA/O=GlobalSign nv-sa/C=BE
- **Usage categories:**
  - OIDC
  - ICAP
- **Certificate status:** Valid
- **Subject key identifier (SHA1):** 567B 561A 450D 97CA 8350 2F7D 04CD 3A48 FFFC FD4B

**OIDC**
- DigiCert Global Root CA/www.digicert.com/DigiCert Inc/U 10.11.2031 01:00:00 RSA 2048
- DigiCert High Assurance EV Root CA/www.digicert.com/E 10.11.2031 01:00:00 RSA 2048
- GlobalSign Root CA/Root CA/GlobalSign nv-sa/BE 28.01.2020 13:00:00 RSA 2048
Google OIDC Configuration

• Documentation
  - https://developers.google.com/identity/protocols/oauth2/openid-connect

• Configuration
  - https://console.cloud.google.com/apis/dashboard
Google OIDC Configuration

- Web Application
- Specify a name
- Set the URL
- Always Server URL + /names.nsf?OIDCLogin
Google OIDC Configuration

- Generates
  - Client ID
  - Client Secret *) sample secret already replaced

- OIDC Client ID + Secret need to be stored in Notes.ini
  - Notes.ini instead of form data because OpenID Support was a last minute addition based on feedback from EAP Forum

  - set config OIDC_LOGIN_CLIENT_ID=990428096234-tiq585g98arppjhujp64aj4hvru0j4d.apps.googleusercontent.com
  - set config OIDC_LOGIN_CLIENT_SECRET=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

- set config OIDC_LOGIN_ENABLE_REDIRECT=1
  - Enables login redirect for OIDC
OIDC map User Name

• Remote name is passed via **e-mail attribute**

• Add external e-mail addresses to the corresponding person document

• `notes.ini OIDC_CUSTOM_EMAIL_CLAIM_NAME` to use custom claim instead of "email" claim
Authenticated User Example

- User is mapped and first entry in **Fullname** field is used to build the **UserNamesList**
- LTPA SSO and single server sessions are supported
Sender Policy Framework (SPF)
Sender Policy Framework (SPF)

- RFC 7208 - Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1

- Defines which host are **allowed to send mails for a domain**

- DNS TXT record for a domain or sub-domain with flexible rule set

- Example:
  host -t txt lab.dnug.eu -> lab.dnug.eu descriptive text "v=spf1 mx ~all"
  - Only allows domain's defined MX record hosts to send mail

- More complex example **dnug.de**

  ```
  v=spf1 mx
  a:domino.dnug.de ip4:87.230.23.16
  include:spf.nl2go.com include:mail zendesk.com include:spf.ce.cloud-y.com
  -all
  ```
SPF Syntax

- http://www.open-spf.org/SPF_Record_Syntax

**Mechanisms**

Mechanisms can be prefixed with one of four qualifiers:

- "u" Pass
- "f" Fail
- "s" SoftFail
- "n" Neutral

If a mechanism results in a hit, its qualifier value is used. The default qualifier is "u", i.e. "Pass". For example:

- "v=spf1 -all"
- "v=spf1 a -all"
- "v=spf1 a mx -all"
- "v=spf1 +a +mx -all"

**The "ip4" mechanism**

The argument to the "ip4:" mechanism is an IPv4 network range. If no prefix-length is specified in the argument, the mechanism defaults to the longest matching network.

Examples:

"v=spf1 ip4:192.168.0.1/16 -all"

Allow any IP address between 192.168.0.1 and 192.168.255.255.

**The "include" mechanism**

The specified domain is searched for a match. If the lookup does not return a match or an error, processing reject based on a PermError.

Examples:

In the following example, the client IP is 1.2.3.4 and the current-domain is example.com.

"v=spf1 include:example.com -all"

If example.com has no SPF record, the result is PermError.

Suppose example.com's SPF record were "v=spf1 a -all".

Look up the A record for example.com. If it matches 1.2.3.4, return Pass.

If there is no match, other than the included domain's "-all", the include as a whole fails to match.
SPF Inbound Support in Domino 12.0.2

• SPF checks can be enabled in server configuration

• Can be used to deliver mail to SPAM folder
  – Not helpful in all customer scenarios
  – But also adds a SPF field to the message leveraged in other applications like Nash!Com SpamGeek

• Enable via Config Doc: **Router/SMTP / SMTP Inbound Controls**
  – Select: Log and tag message → Adds a new field **Received_SPF** to inbound SMTP messages
Enable Inbound SPF Checking

• Config Doc: Router/SMTP / SMTP Inbound Controls
  - Select: Log and tag message
  - Adds a field Received_SPF to inbound SMTP messages

• Received_SPF field
  - contains status + additional information

Field Name: Received_SPF
Data Type: RFC822 Text
"pass (notes.hashcom.de: domain of pnp-hcl.com designates 3.226.151.152 as permitted sender) client-ip=3.226.151.152; envelope-from=john.doe@pnp-hcl.com; helo-smtp1.mail.cwp.pnp-hcl.com;"
DKIM

Domain Keys Identified Mail
Domain Keys Identified Mail (DKIM)

- Allows senders to sign parts of the message to allow a receiving server to verify the signature of a published public key in DNS

- **RFC 6376** - DomainKeys Identified Mail (DKIM) Signatures

- Signing keys per domain stored in DNS TXT Records

- Example: `host -t txt ed20220604._domainkey.lab.dnug.eu`
  
  "v=DKIM1; k=ed25519; p=P+qCLYFRh7QmmqZV4ossGeZTmLyrqI8/nU0fZHd52v0="

- There can be multiple public keys with a lookup by a “selector”
  - Most environments still use RSA. Domino supports more modern **Ed25519** keys – in parallel (dual signature)
  - There can be more selectors to define keys. Also useful for **key rollover**
DKIM Signature

• Signature it calculated based on defined fields of the message

• DKIM header added to the message

• Receiving server
  - Finds the selector in the header
  - Queries the DNS TXT record for selector/domain
  - Verifies message using the public key

• Example: mail from admin@lab.dnug.eu

• DKIM-Signature: v=1; a=ed25519-sha256; c=relaxed/relaxed; d=lab.dnug.eu;
  s=ed20220604; t=1654333615;
  bh=vUKg8XaDsgHuWYIPJChtu9IF0Ycm+6Bi7pkbXtoa4qo=;
  h=To:Cc:MIME-Version:Subject:Message-ID:From:Date:Content-Type;
  b=0TenSwxCs58gqMSI0iVuDZCN4zf1IV5f6kN1qv4MoPZ8y4MyABgb5nrrAUOANOWYb
  Ef6TcaE/kYihPS5gj0FAA==
Domino 12.0.1 - Enable outbound DKIM

- Run console command to create a DKIM key
  - `keymgmt create DKIM lab.dnug.eu ed20220604 ed25519`

- Run console command to create a file containing the DNS TXT record
  - `keymgmt export DKIM DNS lab.dnug.eu ed20220604 lab_dnug_eu_ed20220604.txt`

- Create a DNS TXT record for `ed20220604._domainkey.lab.dnug.eu`

- Define DKIM key for the domain, enable DKIM outbound signing and restart router
  - `set config DKIM_KEY_lab.dnug.eu=ed20220604`
  - `set config RouterDKIMSigning=1`
  - Restart task router
Domino 12.0.2 - Enable inbound DKIM

• Enable via Config Doc: Router/SMTP / SMTP Inbound Controls

• Adds new field “DKIM_Signature” to inbound SMTP message
Domino 12.0.2 – DKIM & SPF Status

• Field **Authentication_Results** contains result from DKIM and SPF

• Field Name: **Authentication_Results**

  Authentication_Results: notes.lab 1; **spf=pass** smtp.mailfrom=nsh@notes.lab
  (sender IP 1.2.3.4); **dkim=pass** header.s=09302021 header.d=notes.lab;
  **dkim=pass** header.s=ed10122021 header.d=notes.lab

• Reference
  - [https://www.rfc-editor.org/rfc/rfc7001](https://www.rfc-editor.org/rfc/rfc7001)

• Currently only two options are available
  - Log and Tag
  - Deliver to Junk

• External tools like Nash!Com SpamGeek can leverage the new field
CScan - Antivirus

ICAP (Internet Content Adaptation Protocol)
CScan – Antivirus leveraging ICAP protocol

- Invented for **Proxy security**, but can be used for antivirus checking attachments as well

**Diagram:**
- Clean messages are released for delivery
- Virus Detected
- All actions are logged
Internet Content Adaptation Protocol (ICAP)

- RFC 2507 - Internet Content Adaptation Protocol (ICAP)

- Domino 12.0.2 natively implements the ICAP protocol and leverages it for attachment scanning

- Support for Windows 64 / Linux 64 in Domino 12.0.2

- New “mailscan” server task is integrated into mail router message flow
ICAP Providers

• Trend Micro™ Web Security

• McAfee™ Web Gateway

• For testing only
  – C-ICAP open source project using ClamAV in the back-end (https://c-icap.sourceforge.net/)
    ▫ Urlich Krause put together a detailed step by step setup documentation
      https://www.eknori.de/2022-05-31/domino-12-0-2-eap-cd-1-clamav-icap/
  – ICAP mock server available until EAP4
    ▫ Can be copied from EAP4 -- A simple internal testing tool HCL shared during early beta

• If you have other ICAP solutions in place, I would like to hear from you!
Domino 12.0.2 Mail Flow Content Scan

• Virus scanning for mail flow (mail router integration)

• Main components

  – cscancfg.nsf
    ▫ Domain wide database for server configuration and status

  – cscanlog.nsf
    ▫ Per server database to log virus events

  – cscanquarantine.nsf
    ▫ Per server database to store quarantined message data

  – mailscan server task integrated with mail router flow
Configuration Flow

- **Load mailscan** creates the *domain wide* cscancfg.nsf configuration database
  - Tries to pull a replica from admin server if started on another server already
  - Creates a replica on admin server for other servers to replicate

- Once create open csscancfg.nsf to create a ICAP configuration

- Create server configuration with assigned ICAP configuration per server

- Finally load mailscan to validate the configuration by connecting to the ICAP server
Create new Configuration

• Create new configuration document first

• Specify an unique configuration name
  − Cannot be changed once servers are assigned
  − Protected against deletion
Specify Mail Scan Settings

- Virus detection, Quarantine and Log Options are predefined
- Can be changed based on customer needs
- Log all attachments only makes sense in test environments
- Mail Tag for Notifications should be set
  - Settings are optional, but should be set
  - Used in conjunction with “Scan and Log Options”
Specify Mail Scan Settings

• Get ICAP Configuration from ICAP Admin
  - ICAP server name should be a DNS name!
  - ICAP standard port is often **1344** or **11344** for TLS
  - The port could vary depending on ICAP server
  - ICAP Service name needs to be specified
    ▫ If ICAP server does not require a service name, specify any name
    ▫ Trend Micro Web Gateway uses “Interscan”

• Specify optional “Virus name formula”
  - Formula is executed on result document and depends on headers returned by ICAP vendor
Specify Scan Configuration / TLS Certificate

- TLS/SSL is required for all ICAP connections
  - A trusted root needs to be imported and assigned to the ICAP category
  - Trusted roots are imported into certstore.nsf
  - CertMgr and certstore.nsf are required for configuring for ICAP TLS connections!
    - In case no domain wide certstore.nsf has been created, refer to Domino 12
    - CertMgr runs on one server in the domain acting as management server for all certificate operations.
  - The server running ICAP requires a certstore.nsf replica

- Tip: Load certmgr on any server will pull a certstore.nsf replica from CertMgr server
Specify Scan Configuration / TLS Certificate

- Trusted Root import into `certstore.nsf` can be performed in multiple ways, based on the configuration.

- If using a proper certificate with a SAN certificate, the `import wizard` can help to import the trusted root:
  - The wizard will try to connect to the ICAP server to retrieve the trusted root.
  - In case the trusted root is not send with the certificate chain, a lookup in Domino directory is performed to obtain the trusted root.

- If the operation completed successfully, a new draft Trusted Root document is created in `certstore.nsf`. 
Trusted Root for ICAP Connection

- Open `certstore.nsf` via action button from CScan configuration document

- If the wizard retrieved a trusted root, it will be marked for **pending validation**
  - The selected category is **ICAP**
  - If the certificate is already there, the category **ICAP** is only added to the trusted root document
  - The wizard always imports trusted roots in "Pending Validation" status and restricts the use to ICAP category

- If the trusted root should be also used for other use cases, remove "Restrict use to category"
Manually import Trusted Root for ICAP Connection

- Some ICAP appliances ship with their own **self signed CA** without **SAN** (Subject Alternate Name) certificates.

- Without a **SAN**, the wizard cannot validate the certificate.

- The trusted root can also be manually imported:
  1. Create a new Trusted Root document.
  2. Paste the PEM data.
  3. Submit the request to **CertMgr** for processing.
  4. Add the **ICAP** category to the newly created trusted root.
Resulting Trusted Root Certificate

• Check the resulting certificate

• Add the **ICAP** category to the newly created trusted root

• Some appliances use their own simple CA or self signed certificate

• In this case you might want to use “**Restrict use to category**”

• If the trusted root should be used for other use cases do not specify “**Restrict use to category**”
Verify CScan Trusted Root Configuration

• Return to `cscancfg.nsf` and refresh the document

• Verify the newly added trusted root is displayed

• By default all trusted roots in the ICAP category will be used
  - The trusted roots can be restricted to an explicit list with the selection option below the display field

• Some ICAP appliances cannot handle certificate chains with intermediate certs.
  - In this case select “Allow partial certificate chains” option and import the intermediate certificate

• Tip: The MicroCA can create an internal certificate valid for two years as well
Certificates without SAN

- **SAN** certificates are required by most applications today
  - But many ICAP appliances ship with simple self signed certificates out of the box
  - Many customers might still use those certificates
  - It is not recommended but commonly used

- CScan can **alternatively** verify the subject of the certificate in this case

- Specify the **exact subject** in the ICAP configuration

- In case the subject is wrong, the admin finds an error message including the expected subject name in the log
Create Server Configuration

• Create a new server configuration for a server

• Select the server name

• Select the configuration just created
  – If only one configuration is present, the configuration is automatically selected

• Each server can only have one configuration
  – The selection dialog hides servers with existing config
  – Note: Configurations can only be deleted if no server is assigned

• Once configured the Health Check status of the configurations is “Pending validation”
  – mailscan server task will validate the connection and set the health check status
CScan Server Status

• Server status view shows the status per server

• Includes important server errors directly written into server document by mailscan server task

• Action to open the log database directly from the view
CScan Log Database

- Log per attachment
  - Only shows viruses unless in test mode to log all attachments

- Log per message shows sender/recipient and all logged attachments
CScan Log for Attachments

- Contains details for each attachment
- Details about virus found & status returned by ICAP
- Lookup for SHA1 hash on Virus Total website
CScan Log per message

- Embedded view for attachments
- Quarantine link only shown, if quarantine document is available
- Attachment log, message log and quarantine document are linked via ReplicaID/UNID fields
Scan Status Token “$$CScanToken”

  – Private key is encrypted for server and stored in cscancfg.nsf server configuration document
  – Used to sign a JWT scan status token

• Public key is also stored in server configuration document to allow other servers to verify the token
  – Each server uses a public key cache for the validation of other server’s tokens
Decode $$CScanToken$$

- JWT token can be decoded
- Details about JWT and a decoder can be found here -> https://jwt.io
“$$CScanToken$$” decoded

• Payload contains information about
  - Virus scanner version
  - Configuration
  - Scan date
  - Reference to the key used
  - Verification hash of attachments
Secure Domino Backup

- Backup approaches & tips
- New Windows VSS Writer Support
Domino Backup

• **Backup & disaster recovery** should be part of your Domino security concept

• Ensure your backup strategy protects you against **ransomware attacks**, too!

• Backup repositories should not be writable at run-time

• If you use the basic Domino 12 Backup functionality to file storage your Domino server and the OS has access to **all backup files** – Not just the current backup!

• There is no one size fits all approach

• Depends on your backup integration and your environment
Secure Backup Approaches

• Only mount/unmount volume in pre/post backup/restore operations
  - Does only make it less likely! → Still vulnerable during backup/restore!

• Take a backup or snapshot of the target storage and/or make files read-only after backup
  - For example if the backend storage is ZFS
  - ZFS can also encrypt file-systems and send snapshots to remote locations (without the key!)

• Use a backup solution with a secure repository
  - e.g. **Borg Backup** on Linux with remote repositories secured by **SSH/SFTP** ([https://www.borgbackup.org/](https://www.borgbackup.org/))
  - Free Restic backup in combination with **VSS Snapshots** and **SSH/SFTP** or **REST server** ([https://restic.net/](https://restic.net/))

• Use a commercial solution like **Veeam Backup & Replication** to protect your backups
Domino VSS Writer?

- VSS Admin Windows tool shows all registered VSS Writers
  - VSS Write support does not require any backup integration
- The AHA idea was also high on my personal wish list
- There was not Domino VSS Writer support ... until now
Windows VSS Writer Support

• Volume Shadow Copy Service (VSS) supports application integrated **snapshot operations**

• “**VSS Writer**” allows to make an application fully “**snapshot aware**” without any direct backup application integration or scripting

• Requires Domino to become a “**VSS Writer**”

• **Flow**
  - Domino registers as a “**VSS Writer**” using a **Microsoft VSS API**
  - **Backup application** starts **VSS Snapshot**
  - **Windows VSS** sends event to all **VSS Writers** registered to “**Freeze**” their application
  - Windows takes **VSS snapshot**
  - Windows sends VSS “**Post Thaw**” event to application
  - Domino processes **delta data** accumulated during snapshot operations
VSS Writer “AutoRecover” Support

• The biggest challenge in the snapshot backup world
  - Snapshots cannot be modified
  - Delta changes need to be stored separately and need to be applied to the database on restore to make the NSF file consistent

• Solution
  - VSS Writers VSS_VOLSNAP_ATTR_AUTORECOVER Option
  - Allows a VSS Writer to update the snapshot in the OnPostSnapshot event to
    - Merge delta information occurred during backup
    - Make the database consistent for recovery without Domino restore operations
  - The Domino VSS Writer supports AutoRecovery to apply changes directly into the writable snapshot in OnPostSnapshot event
Domino Backup VSS Writer Flow

Backup application ➔ VSS backend ➔ Domino BackupVSS ➔ Domino Backup ➔ e\:notesdata NSF files ➔ Snapshot

**Backup**
- Start backup
- Run backup
  - OnIdentify()
  - Signal interest in snapshot
  - OnFreeze()
  - Start snapshot backup
  - NSFBackupStart()
  - All DBs in backup mode
  - OnFreeze() finished
  - Snapshot
- OnPostSnapshot()
  - Mount writable
  - Stop backup
    - NSFBackupStop()
    - NSF backup finished
  - OnPostSnapshot() finished
  - NSF backup finished
  - NSFBackupStartApplyChangeInfo()
- Snapshot created
- Backup snapshot
- Delete snapshot

**Snapshot**
Implementation

- Separate “backupvss” servertask registering as a **VSS backup** writer
- Invokes Domino “backup” servertask to leverage “Domino Backup Snapshot Mode”
- In **Freeze** event waits for backup task to be in snapshot before signaling the snapshot can be created

- Integrates **VSS Writer functionality** into Core Domino
  - Separate task is needed to control “backup” servertask
  - “backupvss” task is required to be permanently loaded to allow **VSS backend** to communicate with Domino

- **GitHub**: Updated, simplified Domino 12.0.2 Veeam integration for restore only
  - [https://opensource.hcltechsw.com/domino-backup/backup-providers/veeam/install_vss_writer](https://opensource.hcltechsw.com/domino-backup/backup-providers/veeam/install_vss_writer)
VSS Writer Implementation Limitations

- NSF Data is required to be on a single volume for snapshot
  - No support for external directory or NSF links pointing to a different volume
  - No support for Windows junctions and comparable mount options
  - Support for directory and NSF link on the **same physical volume**

- VSS Snapshot backup application requires to support **AutoRecovery** mode for full functionality
  - Fallback to write delta files is still possible – In the same way it is supported in 12.0.1 today

- **Restore still requires separate integration similar to Veeam integration available today**
  - Restore integration scripts are posted in GitHub repository
  - No support for VSS restore operations
  - Vendors backup to their own repository and have no direct VSS restore integration

- Only **one** Domino partition per Windows machine can be backed up via VSS
HCL Documentation & Projects

• Domino 12.0.2 New security features and enhancements
  − https://help.hcltechsw.com/domino/12.0.2/admin/wn_security.html
  − https://help.hcltechsw.com/domino/12.0.2/admin/wn_security1201.html

• HCL GitHub – CertMgr
  − https://github.com/HCL-TECH-SOFTWARE/domino-cert-manager

• HCL GitHub – Domino Backup
  − https://opensource.hcltechsw.com/domino-backup/

• HCL GitHub – Domino Container Community Project
  − https://opensource.hcltechsw.com/domino-container/
Further Reading

• GitHub – Domino Start Script Project
  – https://nashcom.github.io/domino-startscript/

• Blog Post: Domino V12 using CertMgr for certificates used outside Domino

• Blog Post: NGINX CertMgr Integrations

• Blog Post: Leveraging Domino Event Monitoring for Domino V12 CertMgr Health Checks

• Blog Post: Fail2Ban Support for Domino on Linux -- Intrusion Detection
  – https://blog.nashcom.de/nashcomblog.nsf/dx/fail2ban-support-for-domino-intrusion-detection.htm
Questions & Answers

- Thank you for your interest in “Domino 12.0.1 + 12.0.2 Security”

- Open questions in chat?
  - Presentation will be available for download from OpenNTF
  - There will be a Q&A summary on OpenNTF

- Additional information
  - [https://blog.nashcom.de](https://blog.nashcom.de)
  - nsh@nashcom.de