Kerkenbos 13-27
6546 BG Nijmegen

T : (024) 345 5000
E : info@tunix.nl
K : Arnhem 09145594
B : 813915120B01

TUNIX/Authenticatie Service

Handleiding 2FA integratie

via

TUNIX/CloudProxy REST-interface

($Revision: 1.2 $)

Classificatie: Classified Customer Document
Doelgroep: TUNIX- of partner-consultants die deze installatie uitvoeren
Eigenaar: Ronald Pikkert
Afdeling: TUN/PMG
Project: TUN/MAR212

Dit document is nog nooit geauditeerd.
De informatie in dit document is geldig tot 15-02-2021.

1. Handleiding 2FA integratie TUNIX/CloudProxy

1.1 Inleiding

1.1.1 Functie van de TUNIX/CloudProxy

De TUNIX/Authenticatie Service is een integraal onderdeel van de TUNIX/CloudProxy. Deze software zet vanuit uw serveromgeving een beveiligde SSL-verbinding op met het TUNIX/Authenticatie Service-platform. De TUNIX/CloudProxy maakt onder meer gebruik van sterke symmetrische certificaten om de veiligheid te waarborgen. De TUNIX/CloudProxy is nodig om telefoonnummers uit AD op te vragen en voorziet tevens in een lokale Radius-service. Het Radius-protocol is geschikt om sterke authenticatie toe te voegen aan verschillende producten zoals de RDP-GW, SSL-VPN appliances en toepassingen die op basis van REST integreren.

1.1.2 Doelgroep en voorkennis

Deze handleiding is bedoeld voor de ontwikkelaar die de installatie van TUNIX/Authenticatie Service voor TUNIX/CloudProxy uitvoert om vervolgens op basis van het REST-interface 2FA aan een applicatie toe te voegen.
Deze handleiding beschrijft niet primair de installatie van de TUNIX/CloudProxy omdat die is beschreven in een separate handleiding.

1.1.3 Methoden voor integratie

TUNIX/Authenticatie nodes in de cloud zorgen voor de afwikkeling van 2FA. Deze nodes zijn over het Internet bereikbaar en worden benaderd via SOAP met symmetrische SSL-certificaten. Deze methode biedt de faciliteiten die nodig zijn om een dergelijk protocol veilig over het Internet te kunnen gebruiken. Integratie op basis van SOAP is in principe mogelijk, maar heeft de volgende nadelen:

- Het afwikkelen van SOAP in combinatie met client-certificaten is niet in elke programmeer-omgeving even eenvoudig.

- TUNIX zorgt voor drie onafhankelijke authenticatie nodes in de cloud, die samen zorgen voor een hoge beschikbaarheid van de dienst. Applicaties die SOAP gebruiken moeten zelf bepalen welke server op enig moment de beste performance biedt.

TUNIX adviseert daarom om niet op basis van SOAP, maar op basis van het REST-interface van de TUNIX/CloudProxy te integreren. De TUNIX/CloudProxy verzorgt de afwikkeling van certificaten, het monitoren van de beschikbare authenticatie nodes en de afwikkeling van het SOAP verkeer en biedt lokale applicaties een REST-interface om de authenticatie af te wikkelen.

1.2 Integratie op basis van REST interface

1.2.1 Activeren REST service in TUNIX/CloudProxy

Voor het integreren van 2FA in een applicatie dient u een TUNIX/CloudProxy te installeren, of u dient in een bestaande installatie het REST-interface te activeren.

Het REST-interface faciliteert de integratie van 2FA in applicaties die daartoe een REST-call naar de TUNIX/CloudProxy uitzetten. Bij gebruik van het REST-interface vindt authenticatie uitsluitend plaats op basis van een telefoonnummer. Dat betekent dat voor deze toepassing geen gebruik wordt gemaakt van de LDAP/AD-koppeling. De TUNIX/CloudProxy kan stand-alone worden gebruikt om uitsluitend de REST-koppeling te faciliteren, maar het is ook mogelijk om de TUNIX/CloudProxy te gebruiken in combinatie met andere toepassingen zoals bijvoorbeeld RADIUS.

Installeer de TUNIX/CloudProxy zoals beschreven in de relevante handleiding. Activeer het REST-interface door in `config.xml` de volgende regels op te nemen:

```
<Rest-Server>http://localhost:81/</Rest-Server>
<Rest-Server>https://localhost:444/</Rest-Server>
```

en herstart de TUNIX/CloudProxy.

1.2.2 Authenticeren via REST

De volgende paragrafen beschrijven de manier waarop het REST-interface moet worden aangesproken en de resultaten die dit zal opleveren.

Voor een eerste test kunnen de commando's ook eenvoudig in een browser worden uitgevoerd.

1.2.2.1 Format GET voor authenticatie

Door het uitvoeren van de volgende REST aanroep wordt een 2FA-verzoek naar de TUNIX/KeyApp gestuurd van de gebruiker waarvan het telefoonnummer bij *phone to use* is ingevuld.

```
http://localhost:81/authenticatie/phone/0031000000000/user%20name[?format=status]
^^^^^^^^^^^^^^^^^^^ ^^^^^^^^^^^^^ ^^^^^ ^^^^^^^^^^^^^ ^^^^^^^^^^^ ^^^^^^^^^^^^^^
    |                   |           |         |         |           |
    |                   |           |         |         | optional format specifier
    |                   |           |         |   principal name
    |                   |           |     phone to use
    |                   |      keyword phone
    |                keyword authenticatie
URL in <Rest-Server>
```

Het veld `phone to use` bevat het telefoonnummer waar het authenticatieverzoek naar toe moet worden gestuurd.

De `principal name` is een string die ter identificatie van de gebruiker in de TUNIX/KeyApp getoond wordt. Tekens die in HTTP als special worden behandeld, moeten in het HTTP-formaat worden ingevoerd. Voor de presentatie in de TUNIX/KeyApp worden deze vertaald naar een tekst-representatie.

In `format` kan worden aangegeven welk formaat de output van deze aanroep moet krijgen.

`format=json`
: returns response as a json string

```
{"status", "no-valid-command"}
{"phone": "xxxxxxxxxxxxx", "time": "xxxx-xx-xx xx:xx:xx",
 "protocol": "REST",
 "principal": "username", "status": "ACK"}
{...., "status", "ACK"}
{...., "status", "NACK"}
{...., "status", "EXPIRED"}
{...., "status", "CHALLENGE"}
{...., "status", "UNREGISTERED"}
```

`format=status`
: returns only status as a string (e.g. `ACK`, `NACK`, `EXPIRED`, `UNREGISTERED`)

return	betekenis
ACK	Authenticatie is geslaagd en de gebruiker heeft akkoord gegeven
NACK	Authenticatie is geslaagd maar gebruiker heeft geweigerd
EXPIRED	Authenticatie is geslaagd maar gebruiker heeft niet gereageerd
CHALLENGE	Authenticatie is geslaagd maar gebruiker heeft geen App en u neemt ook SMS af De applicatie dient de gebruiker nu om een response te vragen en deze response moet via de in paragraaf 1.2.2.2 beschreven aanroep ter verificatie worden teruggegeven.
UNREGISTERED	Op het gebruikte telefoonnummer is geen TUNIX/KeyApp actief

1.2.2.2 Format GET for SMS response

Deze paragraaf is alleen van belang als u ook SMS authenticatie afneemt.
Gebruikers van TUNIX/Authenticatie Service die ook SMS authenticatie afnemen en die een authenticatieverzoek sturen naar een telefoon waarop geen TUNIX/KeyApp is geactiveerd, ontvangen uit het authenticatieverzoek (zie paragraaf 1.2.2.1) een challenge met een *state* die in onderstaand formaat samen met de door de gebruiker ingevoerde response als `state_from_challenge` moet worden teruggegeven.

```
http://localhost:81/authenticatie/response/state_from_challenge/otp/user%20name
^^^^^^^^^^^^^^^^^^^ ^^^^^^^^^^^^^ ^^^^^^^^ ^^^^^^^^^^^^^^^^^^^^ ^^^ ^^^^^^^^^^^
    |                   |           |         |                   |       |
    |                   |           |         |                   | principal name
    |                   |           |         |           user response
    |                   |           |         |
    |                   |           |     state uit 1e call
    |                   |      keyword response
    |                keyword authenticatie
URL in <Rest-Server>
```

1.2.2.3 Format GET loginformatie ophalen

Deze functie faciliteert geen authenticatie, maar faciliteert het ophalen van logregels uit de TUNIX/CloudProxy. Deze functie is primair gemaakt voor TUNIX intern gebruik.

```
http://localhost:81/getlogging/CloudProxy%20Pollaround%20index=
```

Ascii dump van logfile met regels die *CloudProxy%20Pollaround%20index*= bevatten. Dat zijn de regels in de logfile waarin de turnaround tijden naar de diverse nodes staan.


TUNIX
digital security

Table of contents