Authenticated Greeter

TheDestinationGreeterAuthenticated contract sets some permissioning constraints. It only allows itsgreeting to be updated fromSourceGreeterAuthenticated . In order to enforce this, the contract checks that the caller is the original sender from the origin domain.

Target Contract

The target contract must implement some checks to uphold its security constraints.

```

Copy // SPDX-License-Identifier: UNLICENSED pragmasolidity^0.8.15;

import{IXReceiver}from"@connext/interfaces/core/IXReceiver.sol";

/ *@titleDestinationGreeterAuthenticated* @noticeExample destination contract that stores a greeting and only allows source to update it. */ contractDestinationGreeterAuthenticatedisIXReceiver{ // The Connext contract on this domain addresspublicimmutableconnext;

// The domain ID where the source contract is deployed uint32publicimmutableoriginDomain;

// The address of the source contract addresspublicimmutablesource;

stringpublicgreeting;

/**@noticeA modifier for authenticated calls. * This is an important security consideration. If the target contract * function should be authenticated, it must check three things: * 1) The originating call comes from the expected origin domain. * 2) The originating call comes from the expected source contract. * 3) The call to this contract comes from Connext.* / modifieronlySource(address_originSender,uint32_origin) { require( *origin==originDomain&& _originSender==source&& msg.sender==connext, "Expected original caller to be source contract on origin domain and this to be called by Connext" );* ; }

constructor( uint32_originDomain, address_source, address_connext ) { originDomain=_originDomain; source=_source; connext=_connext; }

/*@noticeAuthenticated receiver function. @param_callData Calldata containing the new greeting.*/ functionxReceive( bytes32_transferId, uint256_amount, address_asset, address_originSender, uint32_origin, bytesmemory_callData )externalonlySource(_originSender,_origin)returns(bytesmemory) { // Unpack the _callData stringmemorynewGreeting=abi.decode(_callData,(string));

_updateGreeting(newGreeting); }

/*@noticeInternal function to update the greeting.@paramnewGreeting The new greeting.* / function_updateGreeting(stringmemorynewGreeting)internal{ greeting=newGreeting; } }

```

Source Contract

Nothing special has to be accounted for on the source contract.

```

Copy // SPDX-License-Identifier: UNLICENSED pragmasolidity^0.8.15;

import{IConnext}from"@connext/interfaces/core/IConnext.sol";

/ *@titleSourceGreeterAuthenticated* @noticeExample source contract that updates a greeting in DestinationGreeterAuthenticated. */ contractSourceGreeterAuthenticated{ // The connext contract on the origin domain. IConnextpublicimmutableconnext;

constructor(address_connext) { connext=IConnext(_connext); }

/*@noticeUpdates a greeting variable on the DestinationGreeterAuthenticated contract.@paramtarget Address of the DestinationGreeterAuthenticated contract.* @paramdestinationDomain The destination domain ID.*@paramnewGreeting New greeting to update to.* @paramrelayerFee The fee offered to relayers. */ functionxUpdateGreeting( addresstarget, uint32destinationDomain, stringmemorynewGreeting, uint256relayerFee )externalpayable{ // Encode the data needed for the target contract call. bytesmemorycallData=abi.encode(newGreeting);
```

connext.xcall{value:relayerFee}(destinationDomain,// _destination: Domain ID of the destination chain target,// _to: address of the target contract address(0),// _asset: use address zero for 0-value transfers msg.sender,// _delegate: address that can revert or forceLocal on destination 0,// _amount: 0 because no funds are being transferred 0,// _slippage: can be anything between 0-10000 because no funds are being transferred callData// _callData: the encoded calldata to send); } }

```

Note thatHelloSource should be deployed beforeHelloTargetAuthenticated because the latter needs the address of the former in its constructor.

Now we've enforced that the greeting inHelloTargetAuthenticated can only be updated throughHelloSource !

Previous Simple Bridge Next Ping Pong Last updated9 months ago On this page *Target Contract * Source Contract

Edit on GitHub
```