



# Federated Autonomic Network Access Control

Simon N. Foley, William M. Fitzgerald and Wayne Mac Adams

Department of Computer Science, University College Cork, Ireland  
email: s.foley@cs.ucc.ie, wfitzgerald@4c.ucc.ie, w.macadams@4c.ucc.ie

## Ontologies for NAC Configuration

OWL-DL ontologies are developed for iptables, TCP-Wrapper and XMPP application-level firewalls.

### Linux iptables Firewall Ontology Excerpt

The *IPTRule* concept defines an iptables rule as being composed of a rule index, exactly one chain, one or more filter conditions and a single permission action.

```
IPTRule ≡ ∃=1hasIndex.Integer ⊓
          ∃=1hasChain.ChainType ⊓
          ∃≥1hasGenericFilter.Condition ⊓
          ∃=1hasAction.Action
```

### Example Invalid TCP Packet Filtering

The following iptables rule, drops packets identified as part of the XMAS TCP port scan where TCP flags FIN, PSH and URG are simultaneously set.

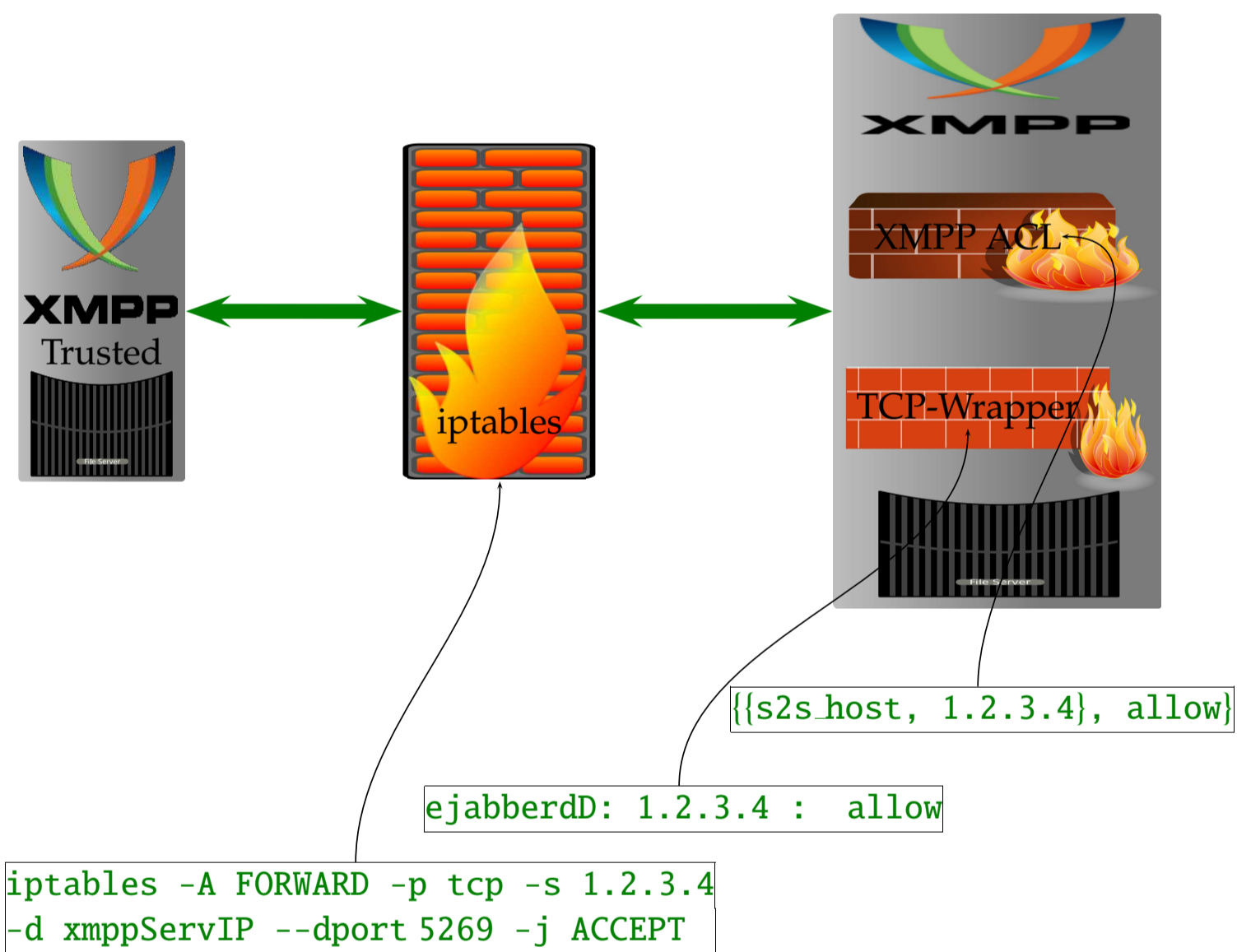
```
iptables -A FORWARD -p tcp --tcp-flags
          FIN,PSH,URG FIN,PSH,URG -j DROP
```

This firewall rule is represented as:

```
IPTRule(iptr_DropXMAS)
← hasChain(iptr_DropXMAS, forward) ⊓
  hasProtocol(iptr_DropXMAS, tcp) ⊓
  hasTCPFlagCheck(iptr_DropXMAS, fin_psh_urg) ⊓
  hasTCPFlagSet(iptr_DropXMAS, on) ⊓
  hasAction(iptr_DropXMAS, drop)
```

## Sample Configuration

Server-to-Server federation is permitted with XMPP server (IP address 1.2.3.4). The XMPP server is protected by a gateway firewall, a locally hosted firewall and XMPP's own application-level ACL firewall.



## Ontology for NAC Configuration

The security mechanism topology is represented as:

```
Service(ejabberd)
← protectedBy(ejabberd, xmppACL) ⊓
  protectedBy(ejabberd, tcpwrapper) ⊓
  protectedBy(ejabberd, iptables)
```

The iptables access-control rule is represented as:

```
IPTRule(iptr_AllowIP1.2.3.4XMPPpkt)
← hasChain(iptr_AllowIP1.2.3.4XMPPpkt, forward) ⊓
  hasProtocol(iptr_AllowIP1.2.3.4XMPPpkt, tcp) ⊓
  hasSrcIP(iptr_AllowIP1.2.3.4XMPPpkt, ip1.2.3.4) ⊓
  hasDstIP(iptr_AllowIP1.2.3.4XMPPpkt, xmppServIP) ⊓
  hasDstPort(iptr_AllowIP1.2.3.4XMPPpkt, 5269) ⊓
  hasTarget(iptr_AllowIP1.2.3.4XMPPpkt, accept)
```

The TCP-Wrapper access-control is represented as:

```
TCPWrapperRule(twr_Allow1.2.3.4XMPPpkt)
← hasDaemon(twr_Allow1.2.3.4XMPPpkt, ejabberd) ⊓
  hasClient(twr_Allow1.2.3.4XMPPpkt, ip1.2.3.4) ⊓
  hasAction(twr_Allow1.2.3.4XMPPpkt, allow)
```

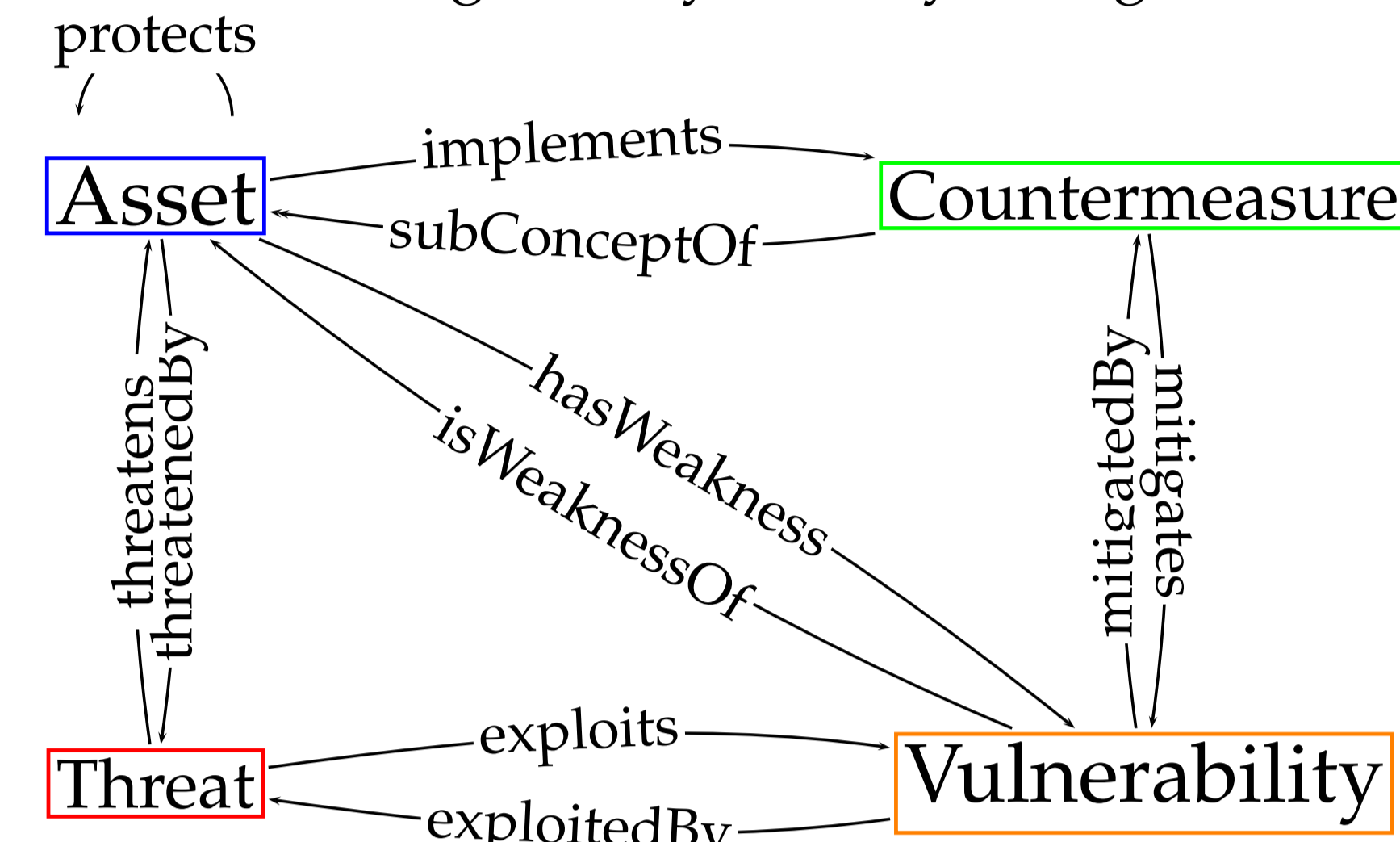
The XMPP access-control rule is represented as:

```
ACLRule(xmppr_Allow1.2.3.4XMPPpkt)
← federateWith(xmppr_Allow1.2.3.4XMPPpkt, ip1.2.3.4) ⊓
  hasPermission(xmppr_Allow1.2.3.4XMPPpkt, allow)
```

## Managing Security Configuration

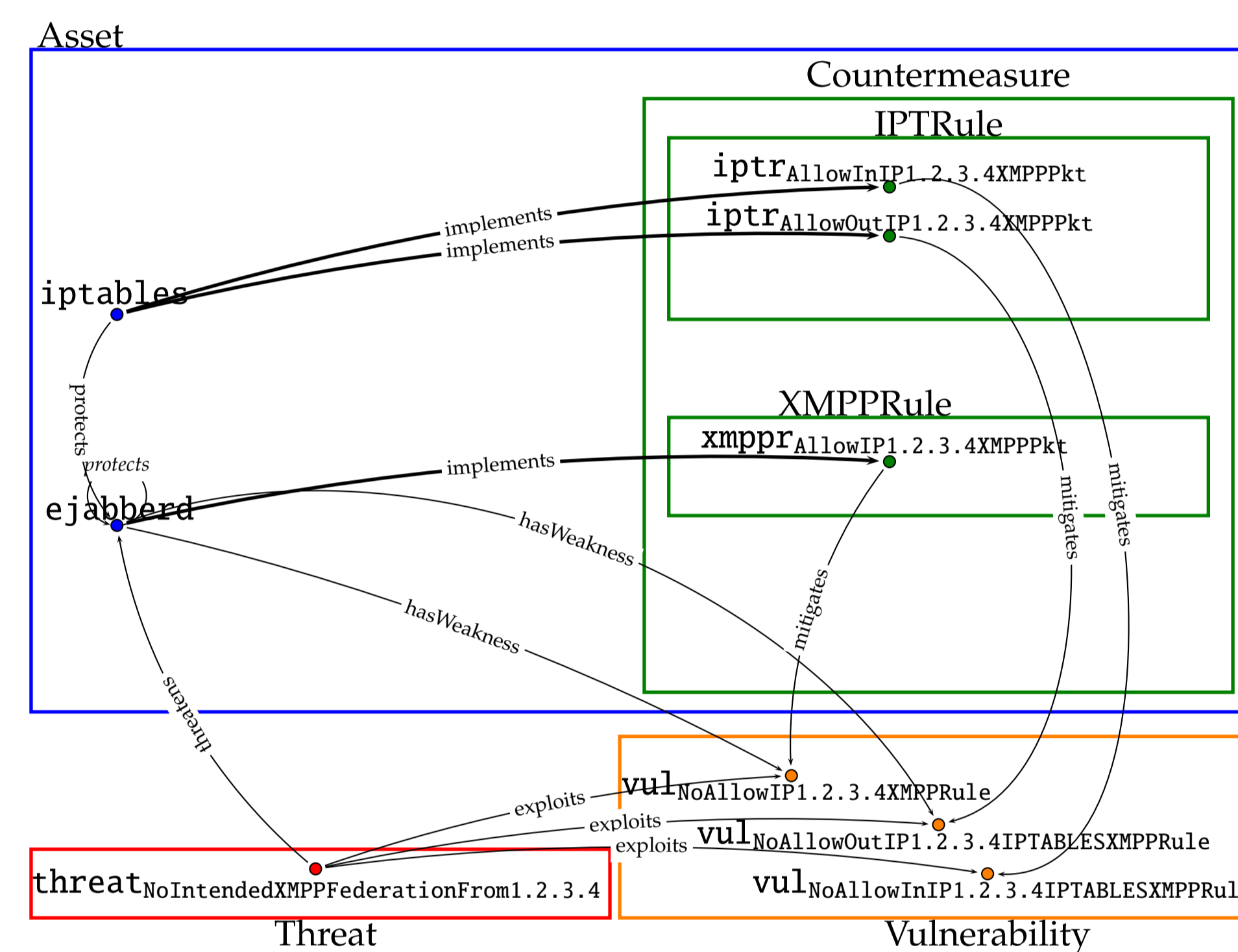
### Semantic Threat Graphs

Semantic Threat Graphs are used to model knowledge about threat mitigation by security configurations.



### Example Semantic Threat Graphs

XMPP Federation Whitelist Recommendations



## Catalogues of Best Practice

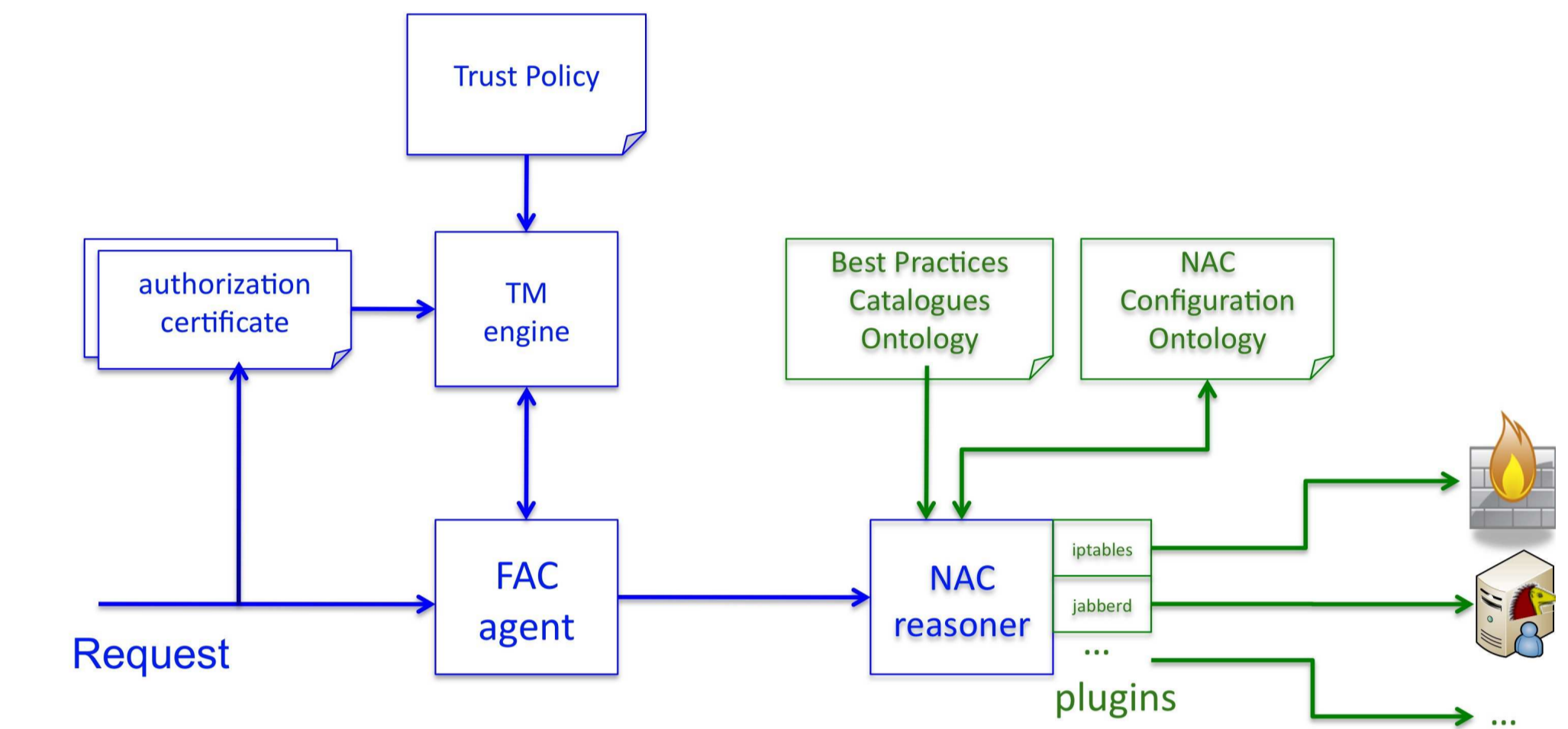
Built ontologies for NIST-800-41, NIST-800-41rev1, NIST-800-45v2, NIST-800-44v2, RFC1918, RFC3330, XEP-0205.

### Sample Ontology Excerpt for NIST-800-41

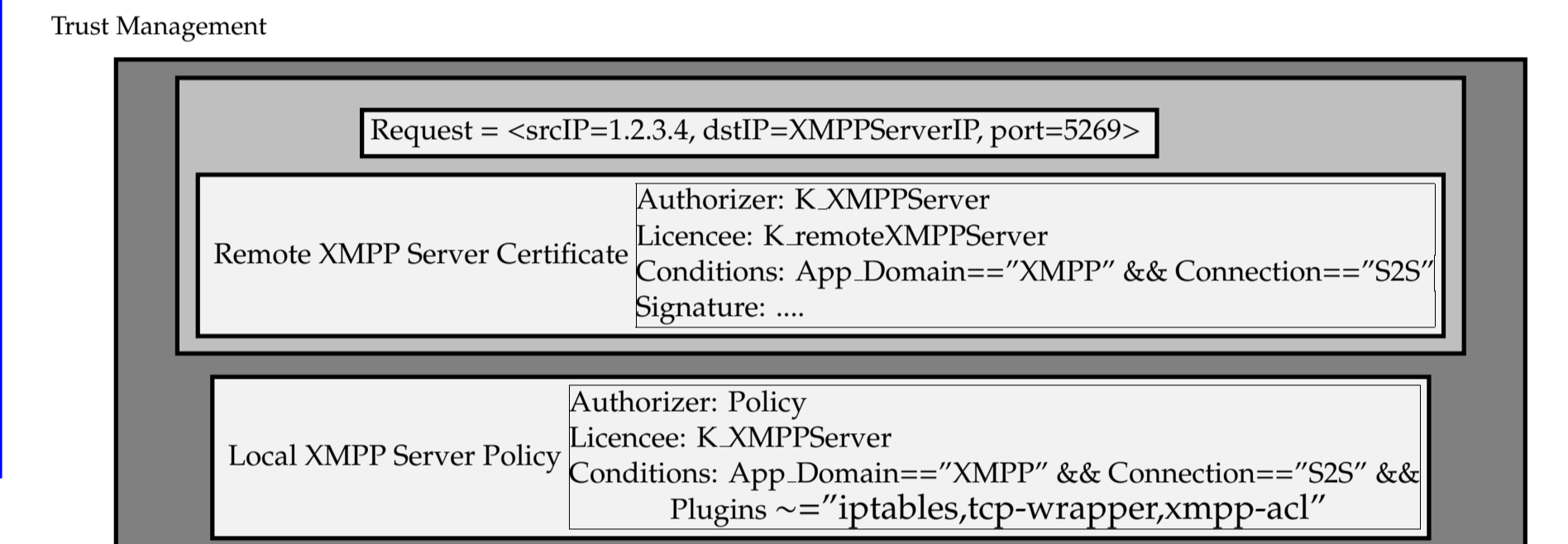
## FANAC Prototype

A FANAC Agent is developed to accept remote requests to reconfigure the Network Access Controls. The current prototype focuses on S2S XMPP federation network access control (re) configuration.

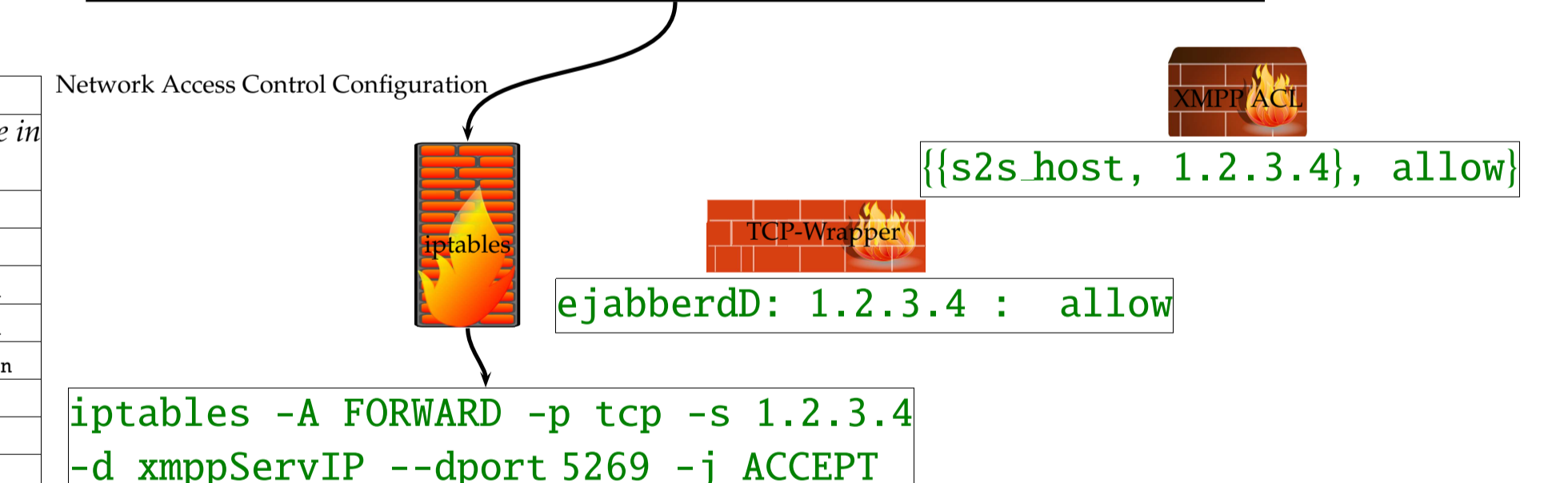
### Architecture



## Scenario



```
Asset(?a) ∧ Threat(?t) ∧ Vulnerability(?v) ∧
TemplateIPTablesRule(iptr_temp) ∧ hasWeakness(?a, ?v) ∧
exploits(?t, ?v) ∧ mitigates(iptr_temp, ?v) ∧
hasDstIP(?a, dstIP) ∧ hasSrcIP(?t, srcIP) ∧
hasDstPort(?a, dstPort)
swrlx : makeOWLIndividual(?iptr, iptr_temp, ?t, ?a, ?v)
→ IPTablesRule(?iptr) ∧ mitigates(?iptr, ?v) ∧
hasSrcIP(?iptr, srcIP) ∧ hasDstIP(?iptr, dstIP) ∧
hasDstPort(?iptr, dstPort) ∧ hasAction(?iptr, accept)
```



### iptables Firewall Example

ID Recommendation Description	Threat	Vulnerability	Countermeasure
FBP-1 Deny "Inbound or Outbound traffic from a system using a source address that falls within the address ranges set aside in RFC1918 as being reserved for private networks" [NIST-800-41].	threatInbound192.168.0.0/16SrcIPPKt	VulInAuthenInbound192.168.0.0/16PktToFW	iptrDenyIn192.168.0.0/16SrcIPPKtInputChain
	threatOutbound192.168.0.0/16SrcIPPKt	VulInAuthenOutbound192.168.0.0/16PktFromFW	iptrDenyOut192.168.0.0/16SrcIPPKtOutputChain
	threatInbound192.168.0.0/16SrcIPPKt	VulInAuthenInbound192.168.0.0/16PktToHost	iptrDenyIn192.168.0.0/16SrcIPPKtForwardChain
	threatOutbound192.168.0.0/16SrcIPPKt	VulInAuthenOutbound192.168.0.0/16PktFromHost	iptrDenyOut192.168.0.0/16SrcIPPKtForwardChain
	threatInbound10.0.0.0/8SrcIPPKt	VulInAuthenInbound10.0.0.0/8PktToFW	iptrDenyIn10.0.0.0/8SrcIPPKtInputChain
	threatOutbound10.0.0.0/8SrcIPPKt	VulInAuthenOutbound10.0.0.0/8PktFromFW	iptrDenyOut10.0.0.0/8SrcIPPKtOutputChain
	threatInbound10.0.0.0/8SrcIPPKt	VulInAuthenInbound10.0.0.0/8PktToHost	iptrDenyIn10.0.0.0/8SrcIPPKtForwardChain
	threatOutbound10.0.0.0/8SrcIPPKt	VulInAuthenOutbound10.0.0.0/8PktFromHost	iptrDenyOut10.0.0.0/8SrcIPPKtForwardChain
	threatInbound172.16.0.0/12SrcIPPKt	VulInAuthenInbound172.16.0.0/12PktToFW	iptrDenyIn172.16.0.0/12SrcIPPKtInputChain
	threatOutbound172.16.0.0/12SrcIPPKt	VulInAuthenOutbound172.16.0.0/12PktFromFW	iptrDenyOut172.16.0.0/12SrcIPPKtOutputChain
	threatInbound172.16.0.0/12SrcIPPKt	VulInAuthenInbound172.16.0.0/12PktToHost	iptrDenyIn172.16.0.0/12SrcIPPKtForwardChain
	threatOutbound172.16.0.0/12SrcIPPKt	VulInAuthenOutbound172.16.0.0/12PktFromHost	iptrDenyOut172.16.0.0/12SrcIPPKtForwardChain

### TCP-Wrapper Firewall Example

ID Recommendation Description	Threat	Vulnerability	Countermeasure
FBP-1 Deny "Inbound or Outbound traffic from a system using a source address that falls within the address ranges set aside in RFC1918 as being reserved for private networks" [NIST-800-41].	threatInbound192.168.0.0/16SrcIPPKt	VulInAuthenInbound192.168.0.0/16PktToFW	twDenyIn192.168.0.0/16Pkt
	threatInbound10.0.0.0/8SrcIPPKt	VulInAuthenInbound10.0.0.0/8PktToFW	twDenyIn10.0.0.0/8Pkt
	threatInbound172.16.0.0/12SrcIPPKt	VulInAuthenInbound172.16.0.0/12PktToFW	twDenyIn172.16.0.0/12Pkt

### Sample real-world anti-bogon iptables firewall rules

Countermeasure	iptables Rule
iptrDenyIn192.168.0.0/16SrcIPPKtInputChain	iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP
iptrDenyOut192.168.0.0/16SrcIPPKtOutputChain	iptables -A OUTPUT -o eth0 -s 192.168.0.0/16 -j DROP
iptrDenyIn192.168.0.0/16SrcIPPKtForwardChain	iptables -A FORWARD -i eth0 -s 192.168.0.0/16 -j DROP
iptrDenyOut192.168.0.0/16SrcIPPKtForwardChain	iptables -A FORWARD -o eth0 -s 192.168.0.0/16 -j DROP
iptrDenyIn10.0.0.0/8SrcIPPKtInputChain	iptables -A INPUT -i eth0 -s 10.0.0.0/8 -j DROP
iptrDenyOut10.0.0.0/8SrcIPPKtOutputChain	iptables -A OUTPUT -o eth0 -s 10.0.0.0/8 -j DROP
iptrDenyIn10.0.0.0/8SrcIPPKtForwardChain	iptables -A FORWARD -i eth0 -s 10.0.0.0/8 -j DROP
iptrDenyOut10.0.0.0/8SrcIPPKtForwardChain	iptables -A FORWARD -o eth0 -s 10.0.0.0/8 -j DROP
iptrDenyIn172.16.0.0/12SrcIPPKtInputChain	iptables -A INPUT -i eth0 -s 172.16.0.0/12 -j DROP
iptrDenyOut172.16.0.0/12SrcIPPKtOutputChain	iptables -A OUTPUT -o eth0 -s 172.16.0.0/12 -j DROP
iptrDenyIn172.16.0.0/12SrcIPPKtForwardChain	iptables -A FORWARD -i eth0 -s 172.16.0.0/12 -j DROP
iptrDenyOut172.16.0.0/12SrcIPPKtForwardChain	iptables -A FORWARD -o eth0 -s 172.16.0.0/12 -j DROP

These catalogues are searchable:

- Generate firewall and application-security configurations that mitigate identified threats.
- Analyse a firewall and application-security configuration's effectiveness at mitigating identified threats.