Ontologies for NAC Configuration

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

Managing Security Configuration

Semantic Threat Graphs

Vulnerability

Threat

Countermeasure

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 -j DROP
```

This firewall rule is represented as:

```
IPTRule(iptrfosmxml, ) <- hasChain(iptrfosmxml, forward) \ hasProtocol(iptrfosmxml, tcp) \ hasTCPFlagCheck(iptrfosmxml, fin_psh_urg) \ hasTCPFlagSet(iptrfosmxml, on) \ hasAction(iptrfosmxml, 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.

```
iptables -A FORWARD -d xmppServIP --dport 5269 -j ACCEPT
```

Catalogues of Best Practice

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

FANAC Prototype

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

Ontology for NAC Configuration

The iptables 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(iptrallowbidP, 2.1.4.xml) <- hasChain(iptrallowbidP, forward) \ hasProtocol(iptrallowbidP, tcp) \ hasSrcIP(iptrallowbidP, 1.2.3.4) \ hasDstIP(iptrallowbidP, 1.2.3.4) \ hasDstPort(iptrallowbidP, 1.2.3.4, 5269) \ hasTarget(iptrallowbidP, 2.1.4.xml, accept)
```

The TCP-Wrapper access-control is represented as:

```
TCPWrapperRule(twallowbidP, 2.1.4.xml) <- hasDaemon(twallowbidP, ejabberd) \ hasClient(twallowbidP, 1.2.3.4, ejabberd) \ hasAction(twallowbidP, allow)
```

The XMPP access-control rule is represented as:

```
ACLRule(xmppallowbidP, 2.1.4.xml) <- federateWith(xmppallowbidP, ejabberd) \ hasPermission(xmppallowbidP, allow)
```

Sample real-world anti-bogon iptables firewall rules

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.

\[\text{This research has been supported by Science Foundation Ireland grant 08/RFP/11405.}\]