Mestrado em Engenharia Informática
Dissertação

Laboratório de Segurança para Redes de Controlo Industrial

Jorge Diogo Gomes Proença

jdgomes@student.dei.uc.pt

Orientador:

Prof. Doutor Paulo Alexandre Ferreira Simões

Data: <30 de Agosto de 2012>

Relatório Confidencial e de Divulgação Restrita

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Nos últimos anos a segurança de sistemas SCADA tem sido alvo de grande atenção. Os recentes exemplos dos vírus *Stuxnet* e *Duqu* alarmaram a comunidade através das suas acções. Mais do que os danos que causaram, estes vírus revelaram as potencialidades de distúrbio e destruição que podem causar através da interacção directa com a componente física das infra-estruturas críticas. Estes são apenas dois exemplos de ameaças a infra-estruturas que estão na base da economia. Os vários *stakeholders* envolvidos (instituições públicas, indústria, operadores de infraestruturas críticas, comunidade científica, público em geral) têm por isso vindo a prestar cada vez mais atenção a esta área específica de segurança, por meio de legislação específica, da criação de mecanismos de alerta partilha de informação, da adopção de práticas apropriadas, e da investigação científica focada neste tópico.

Esta dissertação enquadra-se na participação da Universidade de Coimbra no projecto Europeu FP7 CockpitCI, que irá decorrer entre Janeiro de 2012 e Dezembro de 2014. Mais concretamente, este trabalho fornece um ponto de partida para a referida participação, por meio de várias contribuições: um estudo introdutório dos sistemas SCADA para controlo de sistemas industriais (de modo a reforçar os conhecimentos nesta área por parte do grupo de investigação da Universidade de Coimbra que está envolvido no projecto); uma análise do estado de arte relativamente à segurança em sistemas SCADA, a proposta de uma arquitectura de referência para uma plataforma de detecção de intrusão em ambientes de controlo industrial (para posterior adoção pelo projecto CockpitCI), e a proposta e implementação de uma bancada de testes para posterior suporte do processo de concepção e validação de mecanismos de detecção de intrusão em ambientes SCADA.

Palavras Chave

Sistemas SCADA

Segurança em sistemas SCADA

Proteção de infraestruturas críticas

Bancada de testes para segurança em sistemas SCADA

Índice

Lista de anexos

Lista de figuras

Lista de tabelas

Lista de acrónimos e abreviaturas

ADU	*Application Data Unit*
AGA	*American Gas Association*
ANSI	*American National Standards Institute*
APCI	*Application Protocol Control Information*
APDU	*Application Protocol Data Unit*
ASDU	*Application Service Data Unit*
BMS	*Backup Master Station*
CI	*Infra-estrutura Crítica*
COTS	*commercial off-the-shelf*
CRC	*Cyclic Redundancy Check*
DBMS	*Database Management System*
DMZ	*DeMilitarized Zone*
DNP	*Distributed Network Protocol*
EIA	*Electronic Industries Alliance*
EPA	*Enhanced Performance Architecture*
EtherNet/IP	*Ethernet / Industrial Protocol*
FBD	*Function Block Programming*
FN	*Field Network*
HIDS	*Host Intrusion Detection System*
HMI	*Human-Machine Interface*
Hz	*Hertz*
IDS	*Intrusion Detection System*
IEC	*International Electrotechnical Commission*
IEEE	*Institute of Electrical and Electrotechnical Engineers*
EtherNet/IP	*Ethernet Industrial Protocol*
IPS	*Intrusion Prevention Systems*
ISA	*Industry Standard Architecture*
ISO	*International Standards Organization*
kV	*Kilovolt*
LCT	*Laboratório de Comunicações e Telemática*
LD	*Ladder Diagrams*
MBAP Header	*Modbus Application Header*
MS	*Master Station*
MW	*Megawatt*
NIDS	*Network Intrusion Detection System*
PDU	*Protocol Data Unit*
PLC	*Programmable logic controller*
RTU	*Remote Terminal Unit*
SCADA	*Supervisory Control And Aquisition*
SHA	*Secure Hash Algorithm*
ST	*Structured Text*
TCP	*Transmission Control Protocol*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
TIC	*Tecnologias de Informação e Comunicação*
VoIP	*Voice over Internet Protocol*

1 Introdução

1.1 Enquadramento - segurança em redes SCADA, Projecto CockpitCI

A presente dissertação enquadra-se na área da segurança em sistemas SCADA[1] (*Supervisory Control And Aquisition*), uma designação usada para referir um conjunto diversificado de tecnologias, protocolos e plataformas usados em sistemas de controlo industrial (por exemplo para automação de linhas de produção, para controlo de centrais nucleares ou termoeléctricas e para redes de distribuição de energia.

Este projecto é também enquadrado pelo projecto europeu FP7 CockpitCI. Este projecto, com duração prevista de 36 meses e início em Janeiro de 2012, tem por objectivo o reforço da segurança em infra-estruturas críticas, por meio de mecanismos avançados de detecção de intrusões aos sistemas SCADA e posterior isolamento dos componentes afectados – mantendo o funcionamento autónomo das unidades separadas, de modo a conter os efeitos dos ataques informáticos. A Universidade de Coimbra faz parte do consórcio responsável por este projecto, sendo as suas contribuições focadas na detecção de intrusões nestes ambientes.

1.2 Objectivos da Tese

O trabalho subjacente a esta dissertação teve por principal objectivo contribuir para a participação da Universidade de Coimbra no projecto CockpitCI, por meio de uma série de atividades preparatórias.

O primeiro objectivo foi elaborar um estado de arte relativo aos sistemas SCADA em geral. Este estado de arte tem uma natureza de tutorial – focando-se nos conceitos fundamentais, e não em investigação científica – de modo a ajudar o grupo de investigação envolvido no projecto na aquisição de conhecimentos gerais sobre sistemas SCADA (uma área onde o grupo nunca tinha trabalhado anteriormente). Alguns dos temas incluem as arquitecturas SCADA tradicionais, o funcionamento dos protocolos utilizados para comunicação entre equipamentos activos, as evoluções ocorridas dos sistemas SCADA originais para os actuais, e as semelhanças e diferenças entre as redes SCADA e as redes de dados convencionais.

O segundo objectivo foi preparar um estado de arte relativo à segurança em sistemas SCADA. Este estado de arte identifica as vulnerabilidades e as técnicas utilizadas ao nível da segurança nestes ambientes. Compreende-se que estes são sistemas com um elevado tempo de vida (décadas) e que a idade de muitos dos equipamentos em funcionamento actualmente e suas limitações físicas contribuem para a dificuldade de implementação de técnicas de segurança modernas. Na época da implementação de muitos dos sistemas, a segurança não era um tema importante além de que também se encontravam isolados do exterior. Por essa altura a maioria dos ataques tinham origem interna e a sua causa devia-se maioritariamente a ex-empregados descontentes ou a sabotagens. Actualmente com a sua ligação ao mundo exterior

[1] Os sistemas SCADA são responsáveis por funções de monitorização e controlo dos aspectos físicos de processos industriais, mantendo um registo dos valores presentes em sensores (como o nível de água de um tanque de refrigeração, pressão de gás a fluir numa canalização, etc.), controlo automatizado de actuadores consoante os valores registados nos sensores, e controlo manual por meio de operadores quando necessário.

a percentagem de ataques externos tem sofrido um aumento substancial, comparado com a taxa de ataques internos. A análise de segurança engloba também os aspectos da separação com o mundo exterior, técnicas de detecção de intrusões utilizadas, e características de segurança de protocolos de comunicação SCADA.

O terceiro objectivo do trabalho foi a proposta de uma arquitectura de referência para detecção de intrusões em sistemas SCADA. Esta proposta foi depois apresentada aos restantes parceiros do consórcio, constituindo a base de trabalho para a arquitectura de referência que será adoptada pelo projecto CockpitCI (cuja versão final está presentemente a ser preparada) para posterior implementação e validação – numa fase mais avançada do projecto.

O quarto objectivo corresponde à proposta e implementação de uma bancada de testes para suporte do processo de concepção e validação das soluções de segurança que venham a ser desenvolvidas no âmbito do projecto. Para esse efeito foi proposta e implementada uma bancada de testes que recria um sistema de controlo industrial e no qual serão posteriormente pré-validadas as soluções concebidas durante o projecto (a validação final será feita no laboratório da Israel Electric Corporation, um operador de energia, com um sistema industrial de maior dimensão).

1.3 Estrutura do documento

Esta dissertação encontra-se dividida em sete capítulos.

O segundo capítulo é composto por um estado da arte em sistemas SCADA, sendo descrito o funcionamento da arquitectura, as principais evoluções que sofridas desde o seu início, e uma análise de dois dos protocolos mais utilizados na comunicação entre os equipamentos.

No terceiro capítulo é apresentado um estado da arte subordinado do tema da segurança nestes sistemas, nomeadamente no que diz respeito a problemas das soluções utilizadas em redes TI, alguns ataques e incidentes conhecidos e descrição de alguns projectos na área de protecção de infra-estruturas críticas relacionados com o tema.

O quarto capítulo apresenta a proposta de arquitectura de detecção genérica para ambientes de controlo. Esta proposta foi desenvolvida no âmbito do projecto CockpitCI, e apresenta a definição genérica de uma ferramenta para a monitorização e reacção a intrusões no sistema.

O quinto capítulo descreve dois pontos de trabalho: uma proposta para a integração das bancadas de teste existentes entre os parceiros do CockpitCI, e a bancada de testes desenvolvida internamente para auxílio do trabalho no decorrer do projecto.

No sexto capítulo é apresentado o planeamento e design do trabalho desenvolvido, onde se apresenta o planeamento inicial do trabalho, são expostos os constrangimentos ocorridos e a evolução real das tarefas.

O sétimo capítulo conclui o documento e nele são identificadas as contribuições para o projecto CockpitCI e são definidas algumas linhas de ação para trabalho futuro.

2 Estado de Arte - Sistemas SCADA

Nesta secção são abordados os sistemas SCADA. É feita uma breve introdução ao tema e as origens destes sistemas e o modo como foram evoluindo. Também são referidas quais as motivações para a instalação e utilização, e é feita uma descrição da arquitectura e dos seus equipamentos e um exemplo do modo como se relacionam. No fim é descrito o funcionamento de dois dos protocolos utilizados para a comunicação entre os equipamentos.

Conforme foi já mencionado, este estado de arte tem uma natureza essencialmente tutorial e introdutória, tendo em vista colmatar a inexperiência e os poucos conhecimentos do grupo de investigação nesta área.

2.1 Introdução

Os sistemas SCADA são sistemas de monitorização e de controlo. Estes sistemas tornaram-se populares na década de 60 do século passado. No início, eram muito rudimentares comparados com os existentes actualmente. Muitos deles eram constituídos essencialmente por painéis com mostradores e luzes indicando o estado de sensores ligados fisicamente, onde cada sensor estaria ligado a uma luz/mostrador [Bailey2003]. Com esses dados a tarefa manual do operador de ajustar o processo era facilitada. A sua simplicidade representava a grande vantagem na sua utilização, na medida que não existia processamento, sendo resumidamente um conjunto de sensores ligados directamente e individualmente às luzes/mostradores. No entanto, esta simplicidade torna-se também na sua maior desvantagem. A sua instalação torna-se viável apenas para cenários de pequena dimensão, visto que com a instalação de centenas ou milhares de sensores se tornava impossível gerir a quantidade de cabos necessários ao seu funcionamento. Não obstante, estes sistemas foram e continuam a ser utilizados em algumas indústrias [Bailey2003].

A grande simplicidade destes sistemas impossibilitava a implementação de funcionalidades como o armazenamento de dados, que poderia ser utilizado por exemplo para depuração de erros e falhas, ou para avaliações de desempenho. Isto potenciou a evolução dos sistemas para o seu estado actual, com vários servidores e PLCs (*Programmable Logic Controller*) interligados por uma rede informática. Esta arquitectura será mais explorada na secção **Erro! A origem da referência não foi encontrada.**. Os sistemas actuais possuem um grande tempo de vida, sendo frequentemente explorados por períodos superiores a 20 anos. Esta longevidade, por oposição à rapidez com que evoluem os sistemas informáticos em geral, faz com que os sistemas SCADA acabem por ficar rapidamente "obsoletos" – não ao nível das suas funções intrínsecas de controlo industrial mas sim no que se refere ás tecnologias de segurança usadas na comunicação entre os diversos componentes e na proteção desses mesmos componentes.

2.2 Aplicações

Os sistemas SCADA são utilizados em diversas áreas onde é necessário a monitorização e/ou o controlo de processos. A título de exemplo, mencionam-se algumas áreas de utilização:

- Indústrias:
  - Eletricidade (produção, transporte e distribuição);
  - Fornecimento de água;
  - Saneamento e tratamento de águas residuais;
  - Indústrias de produção de bens no geral;
- Sinalização e controlo de linhas ferroviárias;
- Semáforos de trânsito e controlo de tráfego rodoviário.

Estes sistemas serão responsáveis pela monitorização dos dados físicos dos processos captados por sensores, tais como temperaturas no interior de um forno ou o número de objectos em produção que passam numa esteira. Estes sistemas não monitorizam apenas os dados do processo, podendo também monitorizar dados que estão de algum modo relacionados com ele. Alguns exemplos são a monitorização dos valores de temperatura ambiente ou humidade de uma fábrica, visto que o processo pode necessitar de valores dentro de um parâmetro estabelecido. Outro exemplo pode passar pela existência de sensores de água/líquidos no piso de uma fábrica, desse modo na ocorrência de uma inundação podem ser lançados alertas no sistema para tomar as medidas pré-estabelecidas para o acontecimento.

2.3 Motivações

A introdução dos sistemas SCADA nas indústrias criou grandes vantagens no seu funcionamento. Foram várias as motivações para a sua introdução e evolução. Nesta secção encontramos algumas das motivações para a utilização destes sistemas.

2.3.1 Qualidade do processo

Com a sua adopção é possível aumentar a qualidade do processo e por consequência a do produto final. Tendo um controlo automatizado do processo é garantido um funcionamento óptimo dos componentes, assegurando os requisitos do processo produtivo.

2.3.2 Redução de custos

Por outro lado isto permite uma redução de custos. Esta redução de custos abrange várias áreas, como em recursos humanos e no número de operadores na monitorização, e na sua formação.

2.3.3 Rapidez do processo

Outro ganho com a implementação dos sistemas SCADA encontra-se na rapidez do processo. O controlo computorizado destes sistemas permite uma maior rapidez comparativamente ao controlo puramente humano.

2.3.4 Integração com outros sistemas

Por último, a adopção destes sistemas permite-nos integrar dados do processo com outras aplicações. Gerando informações do processo, e tornando-a disponível para outros

departamentos da organização. É possível a realização de avaliações de desempenho, ou gerir os stocks com maior controlo.

2.4 Arquitectura dos Sistemas SCADA

Nesta secção é descrita a arquitectura base de um sistema SCADA, os componentes que o compõem e o modo como comunicam entre si. De início é apresentado um exemplo simples do funcionamento, e posteriormente a descrição dos equipamentos envolvidos.

Um sistema SCADA é tipicamente composto por cinco níveis [Bailey2003]:

- Sensores / Actuadores;
- *Slaves*;
- Sistema de comunicação;
- *Master Station*(s);
- Processamento de dados.

2.4.1 Sistema SCADA simples

A Figura 2.1 representa um sistema SCADA simplificado, com alguns dos seus principais componentes. Este sistema é constituído pela *Master Station*, pelos *Slaves* e pelos sensores e actuadores do processo.

A *Master Station* (MS) é o equipamento que gere todo o sistema. Este componente é também chamado de *Master*, *Client* ou SCADA *Server*. É neste servidor que são definidos os valores que o sistema deve idealmente ter nos sensores físicos.

Os *Slaves* recebem mensagens da *Master Station*. Estas mensagens podem ser pedidos de leitura dos valores dos sensores, envio de programas, ou comandos para serem efectuados, como por exemplo o fecho de uma válvula, ou a alteração da frequência de um controlador de um motor eléctrico. Além de tratarem as mensagens provenientes da(s) *Master*(s), estes equipamentos também controlam o processo, alterando os valores dos actuadores de forma a obterem leituras dos sensores que estejam dentro dos limites impostos anteriormente pela *Master*. Estes componentes são muitas vezes mencionados por *Server* (porque processam os pedidos da *Master*), por RTU (*Remote Terminal Unit*) e também por PLC (*Programmable Logic Controller*, um equipamento popular para estas funções).

Os últimos componentes são os sensores e actuadores, que fazem a passagem de valores entre o mundo real e o mundo virtual. Os sensores fornecem informação ao sistema do processo monitorizado, e os actuadores permitem ao sistema interagir com ele. No exemplo da Figura 2.1 temos dois sensores, o sensor 1 mede o fluxo de água que passa numa canalização, e o sensor 2 mede o nível de água num depósito. Nos actuadores, temos o actuador 1 que faz o controlo da frequência de uma bomba de água, e o actuador 2 que controla a abertura e fecho de uma válvula de saída de água do depósito.

Quando o sistema se encontra em funcionamento, pode ser descrito por vários passos. Em primeiro lugar, quando o sistema inicia, a *Master Station* envia pedidos de leitura ao *Slaves* 1 e 2. Deste modo fica informado sobre quais os valores de fluxo e de nível que são encontrados nos sensores, e qual a rotação da bomba de água e o estado da válvula de saída do depósito.

De seguida, define quais os valores pretendidos para os sensores de fluxo e de nível, e comunica-os aos *Slaves* 1 e 2. A partir desse momento os dois *Slaves* podem controlar autonomamente através dos actuadores para garantir que os valores lidos dos sensores coincidem com os definidos pela *Master*. Se o *Slave* 1 detecta que o valor lido do sensor de fluxo é mais baixo que o pretendido, aumenta a frequência de funcionamento da bomba de água, e vice-versa. Neste exemplo o *Slave* 2 vai efectuar as mesmas operações para a válvula. Se detectar que o nível do depósito se encontra com um valor inferior ao pretendido, fecha a válvula, e se o valor for superior abre-a. A *Master* efectua leituras periódicas aos *Slaves* e guarda os valores na base de dados. Esta base de dados pode ter várias funcionalidades, como representação do estado do sistema, ou do seu histórico. Tipicamente estas funcionalidades são implementadas em bases de dados diferentes. Uma base de dados de tempo real para o estado do processo, e uma base de dados relacional para o histórico [Björkman2010].

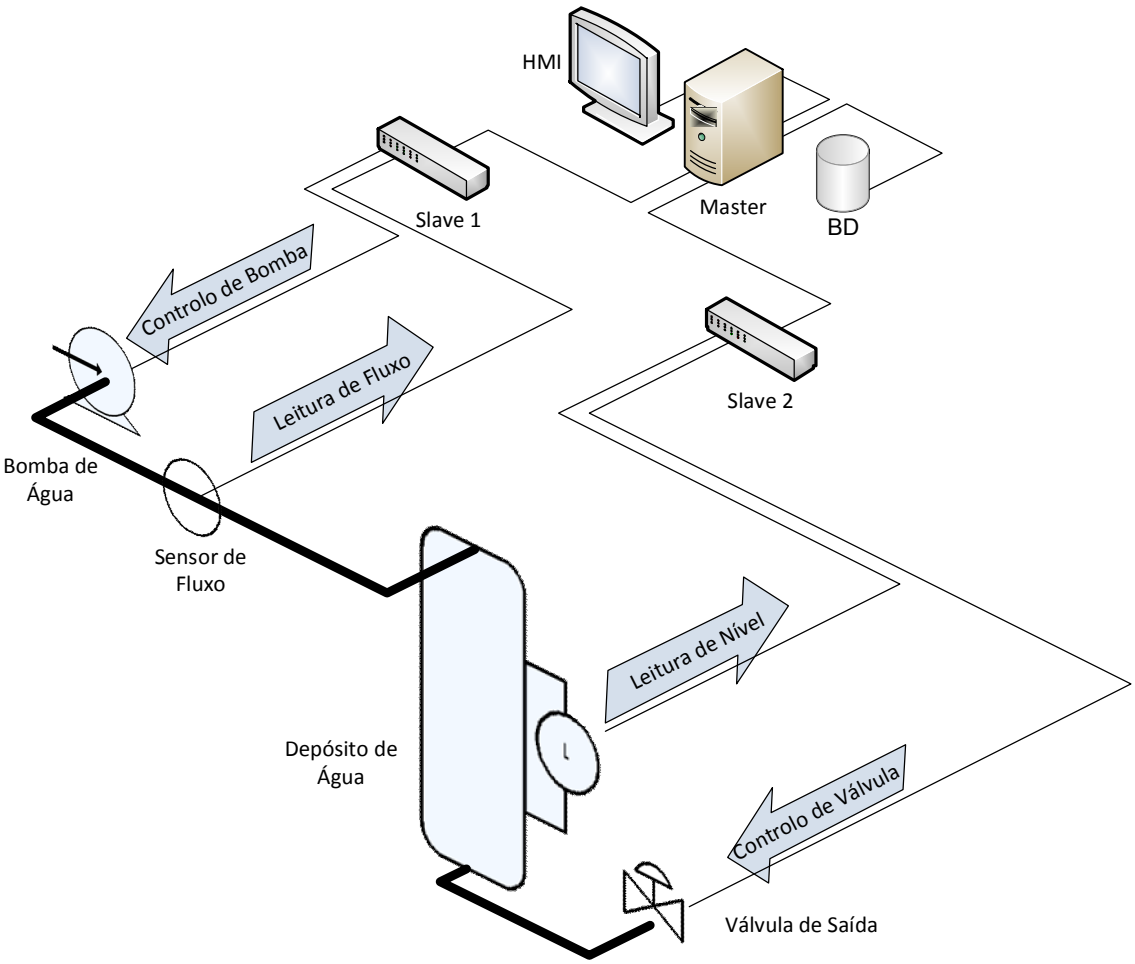

Figura 2.1- Arquitectura de um sistema SCADA simples. (Adaptado de [Wikipedia])

2.4.2 Master / Master Station

Como foi descrito anteriormente, este é o componente principal do sistema, e é a ele que são conectados todos os *Slaves*. A *Master Station* monitoriza-os e envia-lhes comandos quando necessário. É também à *Master Station* que se ligam outros componentes de um sistema

SCADA, tais como os HMI (*Human-Machine Interface*, os componentes que fazem a interface entre o sistema e os operadores humanos. Desse modo podemos ter um ou mais operadores a efectuar acções de monitorização e controlo. São esses operadores que são responsáveis por garantir que o sistema possui o comportamento esperado, estando atentos a eventuais alarmes que surjam e/ou a enviar manualmente comandos para os *Slaves*, caso seja necessário.

É normal encontrar as *Masters* ligadas a outras aplicações. Um exemplo é a ligação a bases de dados. Podem ser vários tipos de bases de dados, cada uma com um propósito diferente. Podemos ter uma base de dados relacional comercial para manter o histórico do estado de todos os dados provenientes do processo, como por exemplo dos sensores. O estado actual do sistema é também guardado em bases de dados. No entanto, não são utilizadas bases de dados relacionais, sendo utilizadas normalmente bases de dados de tempo real [Björkman2010]. Esta escolha prende-se com a necessidade de reflectir o estado do sistema o mais perto possível do real. Essa diferença entre o estado do processo e a sua representação na base de dados situa-se na gama de poucos segundos [VIKING2010d]. Esta performance não é possível com bases de dados relacionais comerciais, sendo apenas atingidas com bases de dados de tempo real, normalmente proprietárias [Björkman2010].

Nos sistemas SCADA actuais, as funções de *Master* costumam ser desempenhadas por computadores com sistemas operativos comerciais, normalmente baseados em Windows ou Unix [Krutz2006].

Num sistema SCADA podem existir uma ou várias *Master Stations*. A *Master* tem conhecimento da topologia do processo. Alguns protocolos SCADA permitem apenas uma *Master*, enquanto outros permitem várias. Este assunto será discutido na secção 2.6, onde são abordados os protocolos de comunicação SCADA.

2.4.3 Slave

Este equipamento está conectado a uma ou várias *Masters*. Os sensores e actuadores são também conectados a este equipamento. Este é o elemento que efectua a maioria das acções de monitorização e controlo. Normalmente não possui grande capacidade de processamento.

A sua principal função é a recolha de dados obtidos através dos sensores do processo e seu envio para a *Master*. Normalmente o envio dos dados é efectuado através de respostas a pedidos da *Master* (este tema será abordado em mais profundidade nos protocolos de comunicação SCADA, na secção 2.6). Para além da recolha dos dados, o *Slave* também pode efectuar acções de controlo. Este controlo pode ser efectuado de forma autónoma, com base em condições recebidas da *Master*, como no exemplo da secção 2.4.1, ou por um comando directo recebido também pela *Master*, através de um operador. Ao contrário da *Master*, o *Slave* não possui conhecimento da estrutura do processo, apenas controla pontos dele.

Outra funcionalidade do *Slave* é o de efectuar o encaminhamento *Store And Forward* dos fluxos de dados, esta funcionalidade é útil em casos onde o *Slave* que está a monitorizar o processo não consegue ter conectividade directa com a *Master*, como o exemplo da Figura 2.2 em que a comunicação é efectuada por uma rede sem fios e não existe linha de vista entre o

Master e o RTU #5. Nesse caso podemos ter um *Slave* intermédio a executar a função de repetidor *Store and Forward*.

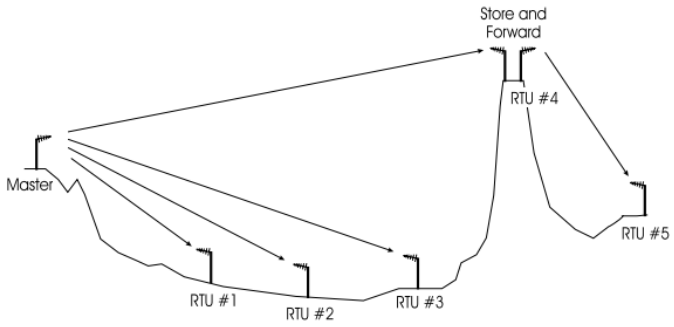

Figura 2.2 - *Slave* com função *Store And Forward*. (Retirado de [Bailey2003])

Com base nos dados recebidos da *Master*, o *Slave* monitoriza os sensores que estão conectados, podendo efectuar operações sobre actuadores para garantir que os valores dos sensores se mantêm dentro de uma gama pré-definida pela *Master*, como no exemplo descrito na secção 2.4.1.

2.4.4 Rede de comunicação

A rede de comunicação é um dos componentes base do sistema. Sem ela não seria possível a comunicação entre os equipamentos.

Os primeiros sistemas de comunicação utilizados em sistemas SCADA eram normalmente proprietários [Igure2006]. Posteriormente, os sistemas passaram a usar normas baseadas em comunicação série, como EIA-485 [IDC2009], que se manteve popular até aos anos 80 e 90. Actualmente, os sistemas estão progressivamente a adoptar as redes Ethernet [IEEE802.3] e TCP/IP, tendo vários dos protocolos desenhados para redes serial sido entretanto portados para redes *Ethernet* e redes TCP/IP em geral. Vários sistemas suportam redes de comunicação e servidores redundantes para maior disponibilidade.

2.4.5 Sensores/Actuadores

Os sensores e actuadores (também chamados de *field devices*) são os elementos ligados directamente ao processo. Os sensores podem ter diversas funções como a leitura de um valor de temperatura, de nível de um líquido num depósito, velocidade de um tapete rolante, fluxo de um líquido numa tubagem, etc. Desse modo, qualquer alteração ao processo (por exemplo, uma variação de temperatura) altera o valor lido pelo sensor e posteriormente o seu valor no resto do sistema SCADA.

Ao mesmo nível também encontramos actuadores. Estes fornecem ao sistema SCADA a capacidade para alterar o funcionamento do processo. Estes actuadores podem controlar diversos aspectos do processo, como por exemplo a frequência de um motor eléctrico ou a abertura e fecho de uma válvula.

2.5 Evolução dos sistemas SCADA

Como foi mencionado na secção 2.1, os sistemas SCADA estão presentes há várias décadas. No entanto, os primeiros sistemas eram bastante diferentes dos actuais. Esses sistemas eram isolados, sem comunicação com o mundo exterior. Esse isolamento inicial está gradualmente a ser abandonado, começando a existir uma interligação com o exterior e também, com cada vez maior frequência, a partilha das redes de comunicação com outro tipo de aplicações. Essa ligação pode ser com a rede corporativa da organização (para interacção com aplicações de gestão de stocks ou avaliações de performance, entre outras). Muitas vezes essa ligação é distante, para interligar estações separadas por várias dezenas ou milhares de quilómetros (por exemplo duas centrais eléctricas), ou para interligação de várias estações a um centro de monitorização. Essa ligação foi inicialmente concretizada utilizando linhas alugadas, ou modems *dial-up*. Actualmente é utilizada a Internet para a maioria das interligações [Ten2008] [Davis2006]. A ligação ao exterior dos sistemas nem sempre serve para a interligação com outras estações, também pode ser um meio útil do fabricante efectuar operações de manutenção do sistema e actualizações de *software*.

Os equipamentos e protocolos utilizados nos primeiros sistemas também eram proprietários. Essa característica não permitia a interoperabilidade entre equipamentos de fabricantes diferentes. Desse modo o cliente acabava por ficar preso a um fabricante ao longo do ciclo de exploração, devido aos altos custos de migração. Actualmente tanto os equipamentos como os protocolos seguem normas, o que permite uma evolução mais simples e económica. Seguindo esta tendência, começou também a ser utilizado equipamento COTS (*Commercial Off-The-Shelf)*, comum em redes convencionais [Igure2006].

Na secção seguinte são descritos dois dos principais protocolos atualmente utilizados para a comunicação entre equipamentos SCADA.

Em conjunto com os equipamentos e protocolos, também os sistemas operativos utilizados sofreram alterações. Normalmente proprietários no início, actualmente são utilizados sistemas operativos comerciais baseados em Windows ou Unix [Creery2005] e sistemas operativos *Real-Time* (por exemplo *VXWorks* ou *Real-time Linux*) [Davis2006], dependendo do equipamento e da natureza do processo gerido. Deste modo é possível uma redução dos custos de desenvolvimento das aplicações.

Não obstante os óbvios benefícios que estas evoluções trouxeram, do ponto de vista funcional e de racionalização de custos, constata-se também que eles estão na origem de algumas das questões mais prementes relacionadas com a segurança dos sistemas SCADA. Esse tema será abordado na secção 3.2.

2.6 Protocolos de comunicação

Os protocolos de comunicação SCADA são responsáveis pela boa interacção dos equipamentos do sistema. Existem entre 150 e 200 protocolos SCADA, sendo a maioria deles proprietários [Igure2006]. Nesta secção é descrito o funcionamento de dois dos protocolos mais utilizados para comunicação entre equipamentos SCADA [ESCoRTS2010c] [Igure2006]: o Modbus [Modbus] e o DNP3 [DNP].

2.6.1 Modbus

O protocolo Modbus foi desenvolvido para permitir a comunicação entre equipamentos SCADA. Foi desenvolvido em 1979 pela Modicon (actualmente Schneider Electric). Este é um dos protocolos mais utilizados na área devido à sua simplicidade e facilidade de aplicação e manutenção.

Existem várias variantes do protocolo. As mais conhecidas e utilizadas são o Modbus RTU e o Modbus TCP, que serão descritas com mais detalhe nas secções 2.6.1.1 e 2.6.1.2. Existem outras variantes, tais como o Modbus ASCII, que é idêntico à variante Modbus RTU, mas com todas as mensagens legíveis textualmente (sendo utilizado para depuração do sistema). Esta última variante não é utilizada em ambientes de produção devido a ser menos eficiente na troca de mensagens (as mensagens da variante ASCII têm cerca do dobro do tamanho comparadas com as do RTU [Pauli2003]). Outras variantes conhecidas são o Modbus UDP, que é similar ao Modbus TCP mas pretende minimizar o *overhead* do protocolo TCP eliminando as mensagens de controlo. Por último podemos mencionar a variante Modbus Plus, esta é uma variante proprietária da Schneider Electric com algumas funcionalidades extra. De seguida é descrito com mais pormenor o Modbus RTU e são identificadas as principais diferenças na implementação da variante TCP/IP.

2.6.1.1 Modbus RTU

O Modbus RTU foi desenvolvido para meios físicos baseados em interfaces série, sendo utilizada por exemplo com a norma EIA-485. Este é um protocolo de uma camada (camada de aplicação). O seu funcionamento baseia-se no mecanismo de *pooling*, onde apenas a *Master* pode iniciar a comunicação. Esta possui uma lista com todos os seus *Slaves* e envia os pedidos de dados de forma sequencial. Estas mensagens são enviadas em *broadcast* e apenas o destinatário responde. A *Master* possui conhecimento do estado dos sensores apenas quando é efectuado um *pooling*. Alguns comandos podem ser enviados para todos os *Slaves* em simultâneo. Nesse caso a *Master* envia a mensagem com o endereço de destino “0”. Os *Slaves* não respondem a mensagens enviadas para esse endereço.

Com a utilização deste protocolo é possível a existência de apenas uma *Master* no sistema. Existe um limite teórico de 247 *Slaves* ligados a esta *Master* (devido ao número de bits utilizados no endereçamento). No entanto, na prática o número máximo de *Slaves* com uma performance aceitável por parte do sistema será bastante inferior [IDC2009].

Os *Slaves* possuem quatro tabelas distintas de dados. Duas de *coils* e duas de *registers*. Para cada um dos tipos existe uma tabela de leitura e uma de leitura e escrita. As *coils* são valores de um bit, utilizados para representar sensores com dois estados, como uma válvula com o estado aberto ou fechado. Já os *registers* possuem o tamanho de dezasseis bits (uma palavra) para permitirem o armazenamento de valores mais complexos.

A descrição das mensagens do protocolo Modbus RTU encontra-se no Anexo A - .

2.6.1.2 Modbus TCP

A variante TCP/IP do Modbus foi desenvolvida com o intuito de oferecer suporte ao protocolo nas redes TCP/IP, em crescente adopção. O seu funcionamento é igual ao da versão RTU, com algumas alterações ao nível da estrutura de mensagens. As portas TCP utilizadas são a porta

TCP/502 para os equipamentos com função de servidor (*Slaves*), e uma porta com um valor superior a 1024 para os clientes (*Master*). É possível ligar equipamentos Modbus RTU a uma rede Modbus TCP utilizando *gateways* que fazem a ponte entre os dois protocolos [IDC2009].

O controlo de erros presente no protocolo Modbus RTU deixa de ser necessária porque a pilha protocolar TCP/IP já oferece esse controlo nativamente.

A descrição das alterações das mensagens do protocolo Modbus TCP/IP em relação ao Modbus RTU encontram-se no Anexo A - .

2.6.2 DNP3

O DNP3 é um protocolo utilizado para comunicação entre equipamentos SCADA. Possui mais funcionalidades, quando comparado com o Modbus, mas é também mais complexo. É mais recente que o Modbus, tendo sido desenvolvido em 1990 pela Westronic. Em 1993 passou para o DNP Users Group e em 2010 passou a integrar as normas do IEEE [IEEE1815-2010]. O desenvolvimento deste protocolo teve em mente a indústria da electricidade [IEEE1815-2010], podendo no entanto ser aplicado noutras áreas. Inicialmente foi desenhado para operar em redes *serial* (ex. EIA-485). Mais tarde foi também implementado sobre TCP/IP.

Muitas das tecnologias de rede (ao nível físico) utilizadas em redes SCADA possuem pouca capacidade, em termos de débitos de rede. O DNP3 utiliza alguns mecanismos para oferecer uma maior poupança dos recursos de rede. Um destes mecanismos é o *Event Data Reporting*: ainda que seja usado um mecanismo de *pooling* como no Modbus, no caso do DNP3 os *Slaves* guardam todas as alterações dos sensores na forma de eventos. Desta forma a *Master* passa a possuir conhecimento de todas as alterações que possam ter ocorrido entre *poolings*. No entanto, o sistema pode ser configurado de modo a que apenas seja enviada informação quando existem alterações nas leituras, permitindo uma menor utilização dos recursos de rede.

Os dados são divididos em classes (Classe 0, 1, 2 e 3). A classe 0 é a classe estática. Não são guardados eventos. Uma *pool* a esta classe retorna o valor de todos os sensores encontrados no *Slave* nesse momento. Esta *pool* é efectuada por exemplo no início do funcionamento da plataforma SCADA, para que a *Master* tenha conhecimento de todo o sistema. As restantes classes são normalmente utilizadas para definir prioridade dos dados, embora isso não seja uma definição do protocolo.

Utilizando o DNP3 a comunicação é possível sem que seja *Master* a iniciar, por meio da funcionalidade *Unsolicited Responses*. Quando activa, esta permite aos *Slaves* o envio de dados sem necessitarem de esperar por uma *pool*.

Na troca de mensagens com o DNP3 podemos requerer confirmação das mensagens enviadas. Esta funcionalidade pode ser desactivada para poupar recursos da rede, no entanto, a desactivação desta funcionalidade em alguns tipos de mensagens pode originar mau comportamento do sistema na falha destas. As *Unsolicited Responses* são um bom exemplo, enquanto uma falha de uma mensagem de pedido ou resposta pode ser detectada utilizando

um *timeout*, na *Unsolicited Response*, a *Master* não tem conhecimento da mensagem antes da sua recepção, logo não é possível controlar a falha.

Este protocolo utiliza a EPA (*Enhanced Performance Architecture*) que é constituída por três camadas:

- Camada de aplicação;
- Camada de ligação;
- Camada física.

O DNP3 adiciona ainda uma camada extra, a camada de pseudo-tranporte, entre as camadas de aplicação e de ligação, para eventual fragmentação das mensagens provenientes da camada de aplicação.

A descrição das camadas, dos campos constituintes, e das mensagens trocadas pode ser consultada no Anexo B - .

2.7 Conclusão

Como conclusão deste capítulo podemos salientar alguns dos pontos importantes dos sistemas SCADA. São sistemas com grande presença na indústria e em diversas áreas como no controlo de tráfego, controlo ferroviário, etc. A grande maioria destes sistemas possui um grande tempo de vida, na ordem de décadas. Essa característica, juntamente com o facto de os sistemas terem sido originalmente desenhados para operarem num ambiente isolado e sem preocupações com a sua segurança, origina grandes dificuldades na implementação de técnicas capazes de evitar e/ou prever e tomar acções contra as ameaças. O capítulo seguinte detalha a situação atual relativamente à sua segurança, focando alguns pontos específicos destes sistemas.

3 Estado de Arte - Segurança em ambientes SCADA

Neste capítulo é feita uma abordagem ao tema da segurança em ambientes SCADA, sendo referidos os aspectos relacionados com a sua evolução, tais como a adopção de soluções COTS em sistemas SCADA e os problemas daí decorrentes, com base nas diferenças existentes entre estas redes e as redes de dados convencionais utilizadas os contextos das TIC (*Tecnologias de Informação e Comunicação*), em ambientes empresariais ou domésticos. Serão também enunciados alguns ataques e incidentes nos sistemas, as suas causas e consequências.

3.1 Introdução

A segurança nos sistemas SCADA é um tema que durante muitos anos foi ignorado. Actualmente ainda existem algumas questões em aberto sobre o assunto. Vários protocolos mais antigos como o Modbus, ainda em utilização, não garantem a boa segurança dos sistemas associados. Estão disponíveis algumas normas e guias de boas práticas na implementação e operação de sistemas SCADA, para aumentar a sua protecção, tais como a norma ANSI/ISA-99.00.01-2007 [ISA-99.00.01] ou o AGA-12 da American Gas Association [AGA12]. No entanto, constata-se que estas normas e guias raramente são adotados de forma integral em contextos reais de exploração. Adicionalmente, as limitações intrinsecas dos sistemas em exploração fazem com que mesmo com a adoção dessas normas subsistam várias fragilidades do ponto de vista de segurança.

3.2 Evolução dos sistemas SCADA e implicações de segurança

A evolução dos sistemas SCADA foi trazendo grandes melhorias nas funcionalidades e desempenho. No entanto, várias dessas evoluções levantaram algumas questões relativamente à sua segurança. As evoluções aqui apresentadas já foram descritas na secção 2.5. Aqui são referidas implicações ao nível da segurança relacionadas com as mesmas.

Estes eram sistemas que inicialmente se encontravam isolados e limitados à rede do processo. solidificando a imagem de que estes eram sistemas seguros (fosse ou não verdade). No entanto, a crescente interligação com a rede informática da organização e mesmo com o exterior (e.g. internet, acessos remotos para manutenção e controlo) criou vários problemas de segurança e gerou uma nova fonte de ataques. A taxa de ocorrência de ataques a partir do exterior tem sofrido um crescimento maior, quando comparada com a taxa de crescimento de ataques originados internamente [Kang2011].

Os protocolos iniciais eram proprietários e o modo como funcionavam não era público, fazendo com que fabricantes e operadores confiassem na segurança por obscuridade [Clarke2004]. As falhas eram apenas conhecidas pelos fabricantes e pelos atacantes, e nenhuma das partes tinha interesse na sua publicação. Quando os protocolos começaram a ser abertos e a seguir normas abertas o seu funcionamento passou a estar disponível para todos. Esta segurança por obscuridade (que não é boa prática, de qualquer modo) deixou de existir, diminuindo um nível de segurança que por si mesmo já era fraco. As capacidades de auto configuração dos equipamentos pioram esta a situação, já que a natureza *plug-and-play* desta característica permite aos atacantes pedir aos equipamentos que se descrevessem [Clarke2004], fornecendo-lhes deste modo informação valiosa para o planeamento de posteriores ataques.

Outro ponto referido anteriormente foi a adopção pelos fabricantes de sistemas operativos comerciais. Apesar dos ganhos devidos a um menor custo de desenvolvimento, os sistemas SCADA passaram a padecer das mesmas vulnerabilidades dos sistemas operativos que lhes dão suporte. São vários os casos conhecidos de ataques e de incidentes devido aos sistemas operativos de base (e não aos sistemas SCADA em si mesmo). Na secção 3.8 são descritos alguns ataques e incidentes para os quais, entre outros motivos, as vulnerabilidades nos sistemas Windows constituíram uma das causas para a sua concretização.

Apesar da mutação ao nível da segurança destes sistemas, em alguns casos ainda persiste a ideia que eles continuam seguros e isolados [Krutz2006]. A necessidade de efectuar actualizações de segurança no *software* nem sempre é cumprida, e muitos dos sistemas raramente são atualizados, aumentando a probabilidade da sucesso de um ataque. Uma das razões que contribui para a não actualização dos sistemas é necessidade de estes estarem sempre disponíveis, podendo estar alguns equipamentos estar meses ou anos sem serem reiniciados [ESCoRTS2010c] [Zhu2011]. Além disso, disso, as actualizações de software necessitam de ser minuciosamente testadas pelos fabricantes antes de serem instaladas. Por último, o sistema SCADA a actualizar pode utilizar software legado já sem suporte por parte do fabricante.

3.3 Perfil de atacantes

Existem diferentes perfis de atacantes envolvidos em ataques a sistemas de controlo. É um conjunto amplo, variando desde pequenos *hackers* até grupos financiados por nações.

Lone Wolf: Normalmente constituído por uma pessoa sozinha ou um grupo pequeno de pessoas. Este pode possuir poucos conhecimentos, utilizando *scripts* de modo a realizar o ataque. Este é o tipo mais comum dentro do género. Normalmente são cativados pela descoberta e pelo acesso a ambientes não autorizados. Por norma é fácil conhecer a identidade do atacante devido às fracas capacidades de apagar o seu rasto.

Por outro lado, o atacante pode ter conhecimentos avançados, descobrindo vulnerabilidades obscuras e tirando partido destas, sendo mais difícil de descobrir a sua identidade.

The insider: Uma pessoa interna é uma potencial fonte de sabotagem e crime, seja um empregado descontente ou um espião. Necessitam de menos conhecimentos a nível de segurança de sistemas para realizar um ataque devido aos elevados conhecimentos internos da organização. Existem registos de ataques realizados internamente, como o caso apresentado na secção 3.8.2.1.

Outra fonte de ameaça são os grupos com maior número de elementos. Podem-se destacar os grupos criminais, e os grupos terroristas. Estes constituem uma ameaça elevada, devido ao potencial impacto que podem causar numa ou em várias organizações. Por vezes possuem grande apoio financeiro (por exemplo uma nação), aliado a uma grande janela temporal de acção. Com uma forte fonte financeira, são capazes de recrutar especialistas de diversas áreas (especialistas de segurança, controlo, entre outros). Com isto têm acesso a software proprietário, conseguindo identificar vulnerabilidades normalmente desconhecidas do público em geral. Além disso, podem persuadir os fabricantes ou seus empregados a inserir vulnerabilidades (*backdoors*) nos seus produtos.

3.4 Diferenças entre redes de dados convencionais e SCADA

A adopção de *hardware* e *software* COTS é uma tendência da área, sendo uma forma de se reduzirem os custos e tempos de desenvolvimento. Estes são utilizados inclusive nos componentes de segurança, tais como *firewalls*, sistemas de detecção de intrusão e outros componentes similares. Apesar das suas vantagens, a adopção de componentes COTS introduz alguns riscos de segurança, por exemplo porque fazem algumas assunções (legítimas em redes convencionais) que nem sempre estão presentes em ambientes SCADA.

Como exemplo destes riscos temos as *firewalls* tradicionais. Algumas delas assumem que se encontra um encaminhador nas suas interfaces que bloqueia ataques de *TCP SYN Flood*. Deste modo algumas não implementam esta funcionalidade [Byres2005]. Na Figura 3.1 encontramos um exemplo desse cenário. Numa *firewall* encontram-se três ligações: rede corporativa, rede do processo, DMZ (*DeMilitarized Zone*). Caso exista um TCP SYN Flood com origem na rede corporativa pode ocorrer perda de comunicação entre a rede do processo e a DMZ, onde se encontra o servidor de base de dados necessário ao suporte da plataforma SCADA.

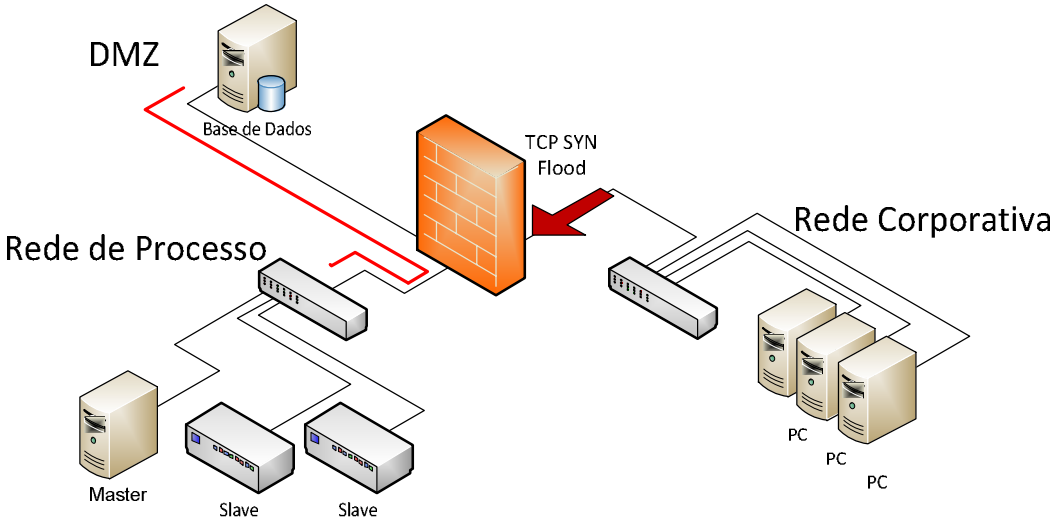

Figura 3.1 - Cenário TCP SYN Flood. (Adaptado de [Verba2008])

A execução de comandos em redes SCADA tem impacto no mundo real. São sistemas com elevados requisitos de disponibilidade e tempos de resposta normalmente baixos e, mais importante, sem tolerâncias ao nível de latência/atrasos. Existem aplicações *Soft Real-Time* em ambientes TIC genéricos, tais como VoIP (*Voice over Internet Protocol*), onde os requisitos de atrasos são já baixos (idealmente latência menor que 150 ms) mas para os quais eventuais atrasos podem ser compensados com descarte de pacotes e/ou ajustes na codificação. No entanto, as aplicações SCADA são em geral *Hard Real-Time*, e os limites de atraso têm de ser estritamente cumpridos. Um comando atrasado pode ser inútil, causar falhas no sistema ou mesmo, em casos extremos, resultar em danos ou destruição do equipamento e perda de vidas humanas [Zhu2011].

Outro factor que contribuiu para o fraco desenvolvimento de segurança em SCADA relativamente à segurança em redes convencionais encontra-se nas diferentes prioridades existentes. Como se pode observar na Figura 3.2, nas redes convencionais a confidencialidade

é a prioridade máxima, seguida pela integridade das comunicações e, por último, pela disponibilidade. Nas redes SCADA, devido à sua natureza crítica, a disponibilidade é a prioridade máxima. Em segundo lugar encontra-se a integridade das comunicações e em último lugar a confidencialidade [ISA-99.00.01]. Esta diferença de prioridades vai condicionar a implementação de alguns mecanismos de segurança, devido à possibilidade de causarem atrasos intoleráveis em alguns casos.

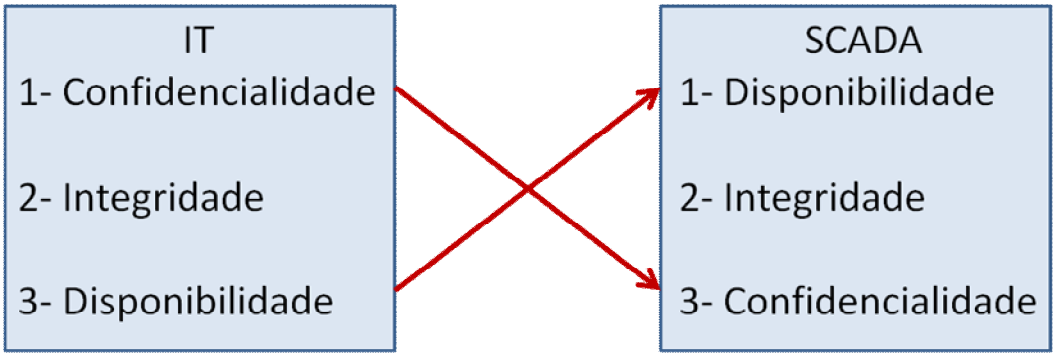

Figura 3.2 - Prioridades redes IT e SCADA. (Adaptado de [ISA-99.00.01])

Por último é necessário referir o tempo de vida destes sistemas já mencionado na secção 2.1. O elevado tempo de vida contrasta com as infraestruturas TIC que possuem ciclos de vida substancialmente inferiores e nas quais os sistemas são renovados na totalidade com alguma frequência. Essa característica condiciona a implementação de alguns mecanismos de segurança devido à fraca capacidade de processamento de alguns equipamentos [Igure2006].

3.5 IDS

Os IDS (*Intrusion Detection System*) fazem parte do grupo de soluções de segurança de ambientes TIC introduzidos nas redes SCADA. Apesar do aumento de segurança, as primeiras adopções de IDS em redes de controlo não exploravam algumas características específicas destes ambientes e relevantes do ponto de vista de segurança. Não obstante, algum trabalho foi já (e continua a ser) efectuado nesta área.

Os NIDS (*Network Intrusion Detection System*) utilizados em ambientes convencionais normalmente efectuam a análise do tráfego com base em assinaturas ou anomalias nos padrões de tráfego. O uso de assinaturas permite a detecção de intrusões através de padrões de ataque conhecidos, garantindo uma detecção eficaz de ataques conhecidos, com baixas taxas de falsos positivos. A desvantagem destes mecanismos encontra-se na sua incapacidade de detectar novos ataques que ainda não estejam documentados. O outro género de IDS avalia possíveis ataques através de observação estatística do comportamento de rede. Ele avalia o tráfego na rede e cria um modelo do tráfego "normal" do sistema. A detecção de ataques baseia-se nas diferenças encontradas entre o tráfego analisado e o modelo previamente construído. Este modelo de IDS tem como vantagem a detecção de ataques desconhecidos, mas tem a desvantagem de ser propenso a obter uma maior taxa de falsos positivos, quando comparado com o modelo baseado em assinaturas.

Estes IDS tradicionais são úteis a detectar ataques como *network scans* ou pacotes mal-formados. No entanto a falta de autenticação típica de ambientes SCADA permite que um

atacante consiga executar um ataque sem recorrer a estes mecanismos, forjando pacotes e enviando-os para os equipamentos [Verba2008]. É pois necessário fornecer aos IDS conhecimento dos protocolos para garantir uma maior protecção [Igure2006].

Como foi referido anteriormente, as redes de controlo possuem algumas características que são aproveitadas podem fornecer aos IDS uma melhor performance na protecção contra intrusões. Existem alguns parâmetros específicos destas redes, como as topologias estáticas [Verba2008], ou o tráfego que também segue parâmetros relativamente constantes após a sua implementação. Os fluxos não se alteram e podem ser mapeados [Verba2008]. Na Figura 3.3 encontramos o esquema de ligações de um sistema. Este mapeamento permite-nos conhecer quais as comunicações existentes entre os diversos equipamentos, incluindo portas, protocolo de transporte e o sentido das comunicações.

Observando a Figura 3.3 apenas podem existir comunicações entre a *Master* e os restantes equipamentos. Não teria sentido o HMI enviar um pedido directamente para um *Slave*. Neste caso se o IDS detectasse uma comunicação directa entre o HMI e um *Slave*, indicaria que o primeiro se encontrava comprometido, podendo despoletar um alarme, ou outra medida pré-definida.

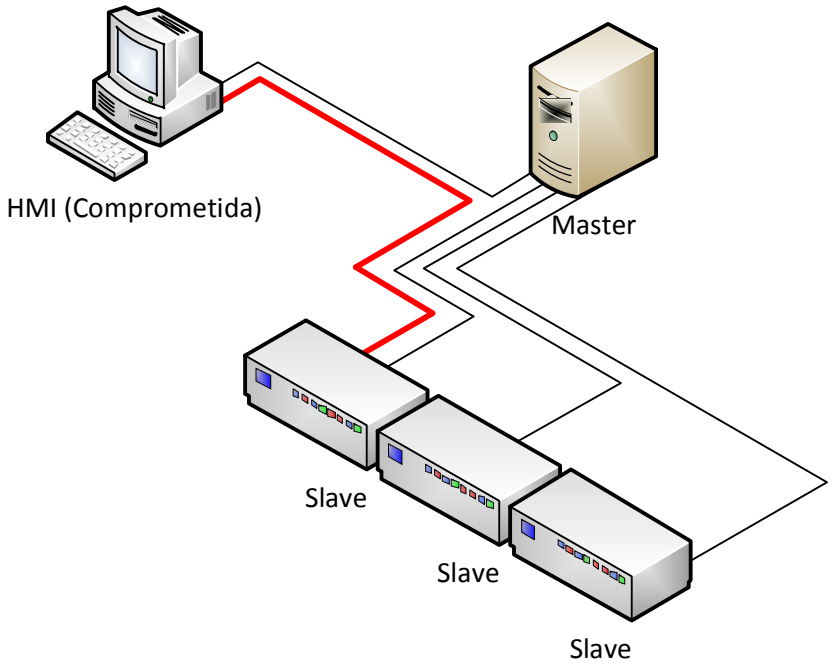

Figura 3.3 - IDS baseado em fluxos. (Adaptado de [Verba2008])

Nos protocolos SCADA os pacotes têm um tamanho máximo para o seu envio. Por exemplo se um pacote Modbus exceder o tamanho máximo de 256 bytes, um pacote com um tamanho superior pode causar um *buffer overflow* num *Slave*, devido às características físicas do equipamento [Zhu2011].

Este tipo de IDS baseado no tráfego da rede detecta comunicações entre dispositivos não autorizadas e pacotes mal-formados mas não seria capaz de detectar ataques *man-in-the-*

middle. Os pacotes enviados podem ser bem formados e entre equipamentos autorizados e no entanto as acções efectuadas não serem as correctas para o estado do sistema. Esta análise pode ser efectuada comparando os pacotes que circulam na rede com uma base de dados de estados e acções efectuadas. Como exemplo desta análise podemos observar na Figura 3.4 um cenário onde a *Master* se encontra comprometida. Um operador envia um comando de fecho de uma válvula para a *Master*, para esta reenviar para o *Slave* 3. A *Master* comprometida pode não enviar o comando ou alterar o comando recebido, mascarando a sua acção. Um IDS baseado em estados pode detectar esta inconsistência entre as mensagens 3 e 4.

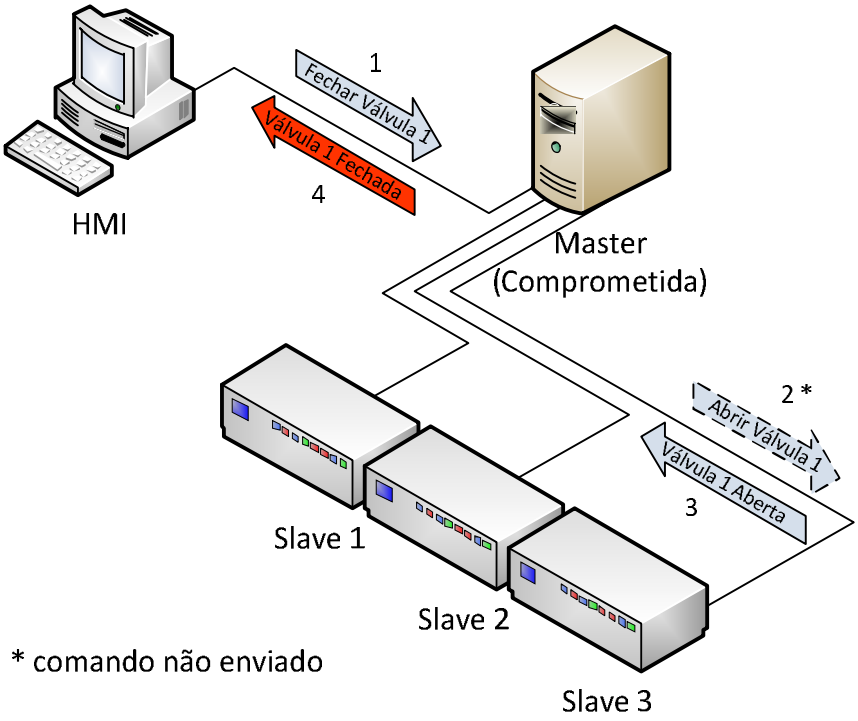

Figura 3.4 - IDS baseado em estados de sistema. (Adaptado de [Verba2008])

O *Snort* [Snort] é uma das ferramentas utilizadas para detectar intrusões em redes SCADA, existindo já (algumas) assinaturas específicas para ambientes SCADA. Um exemplo de pacotes de assinaturas pode ser encontrado em [Digital Bond]. São disponibilizados pacotes de assinaturas para vários protocolos, tais como DNP3, Modbus e EtherNet/IP (*Ethernet Industrial Protocol*). Existem também assinaturas para vulnerabilidades gerais.

3.6 Segurança em protocolos SCADA

Nesta secção são descritos alguns assuntos relativos à segurança dos protocolos de comunicação entre equipamentos SCADA. São referidas as questões relativas às vulnerabilidades de segurança dos dois protocolos apresentados na secção 2.6, o Modbus e o DNP3.

3.6.1 Modbus

O protocolo Modbus não foi desenhado com a segurança em mente, por isso não possui mecanismos para garantir:

- Confidencialidade;
- Integridade de mensagens;
- Não repúdio;
- Protecção contra *Replay Attacks*.

Não existe nenhuma norma para variantes do protocolo que implementem segurança. O único trabalho existente é um estudo de uma possível solução, proposta por Nai *et al*. [Fovino2009] [ESCoRTS2010c]. Este estudo garante integridade utilizando uma função de *Hashing SHA2*, autenticação baseada no mecanismo de assinatuas RSA, não-repúdio garantido com as assinaturas de autenticação e protecção contra *Replay Attacks* com a introdução de *time-stamps* no PDU.

3.6.2 DNP3

Apesar de o protocolo DNP3 ser mais recente que o Modbus, a segurança também não fez parte do seu desenho original. O DNP3 também não possui mecanismos para assegurar:

- Confidencialidade;
- Integridade de mensagens;
- Não repúdio;
- Protecção contra *Replay Attacks*;

No entanto, existem duas variantes deste protocolo com algum suporte de segurança: o Secure DNP3 e o DNPSec [ESCoRTS2010c]. As alterações na primeira variante foram efectuadas ao nível da camada de aplicação, enquanto no DNPSec as alterações foram efectuadas na camada de ligação. Devido à implementação numa camada mais baixa da pilha protocolar, a variante DNPSec tem algumas vantagens relativamente à Secure DNP3 porque protege contra ataques às camadas de transporte e de ligação.

3.7 Separação entre redes

A temática da separação entre redes apareceu com o início da ligação dos sistemas SCADA à rede da organização e ao exterior. A separação utilizando *firewalls* é a primeira barreira de ataques vindos do exterior. A configuração incorrecta da *firewall* pode originar vários problemas ao nível da segurança. Mais de metade (52%) dos incidentes ocorridos na área tiveram origem em más configurações de *firewall* [Byres2005] que permitiam a passagem directa de ligações *dial-up*, *wireless*, entre outras. Num artigo sobre erros de configuração de *firewalls* foi demonstrado que cerca de 80% das *firewalls* estavam configuradas para aceitar qualquer serviço na regra de *inbound*, o que pode ser considerado um erro grave [Wool2004]. Nesse estudo ainda se pode observar que a mesma percentagem de *firewalls* também não implementava mecanismos de acesso seguro.

Existem várias opções de separação da rede de processo da rede da empresa, tentando mantê-la mais isolada possível do exterior. Em [Byres2005] são apresentados resultados de um estudo efectuado junto a 25 organizações. Do total, 10 são fabricantes de *firewalls*, sistemas de controlo ou segurança em redes TI. Outros 15 são utilizadores industriais das áreas do petróleo, química, alimentação, automóvel e produção de energia. Nesse estudo foram

analisadas as técnicas de separação entre as redes de processo e da organização. Os resultados foram englobados em três grupos principais de arquitecturas de separação:

- Utilização de computadores com duas placas de rede (*Dual Homed*);
- Separação em duas zonas;
- Separação em três ou mais zonas, com DMZ.

Na Figura 3.5 podemos observar a primeira arquitectura. É a forma mais simples de separar as duas redes. São utilizadas duas placas de rede nos computadores que necessitam de ter acesso a ambas as redes. Apesar de ser a forma mais simples, esta apresenta um elevado risco de segurança nos computadores da fronteira. Apesar dos riscos associados, foi utilizada em várias organizações [Byres2005].

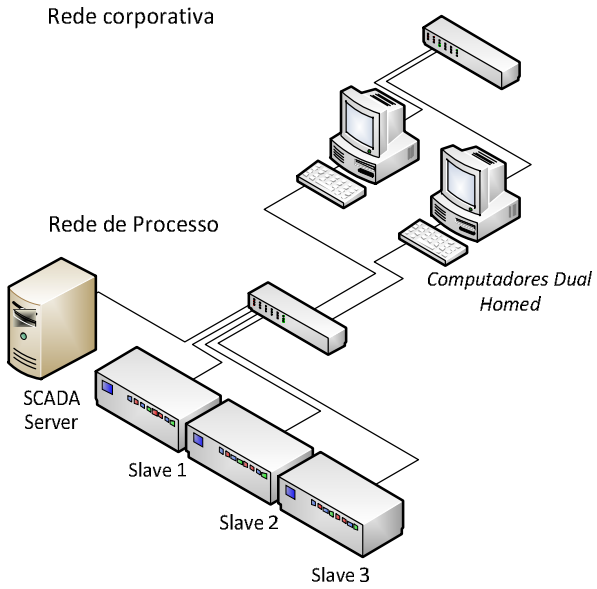

Figura 3.5 - Separação de redes utilizando computadores *Dual Homed*. (Adaptado de [Byres2005])

A segunda arquitectura apresenta algumas melhorias de segurança, mas ainda se encontram alguns problemas na sua implementação. Podemos observar na Figura 3.6, que com esta já não encontramos as vulnerabilidades de segurança associadas aos computadores de fronteira da arquitectura anterior. No entanto, desta forma é necessário ponderar onde são colocados os serviços utilizados por ambas as redes, como as bases de dados do processo. Outra questão de segurança que se encontra nesta arquitectura é a necessidade de configurar ligações directas entre as duas redes na *firewall*. Estes dois problemas são resolvidos com a terceira arquitectura. Muitas das empresas estudadas utilizavam esta configuração com duas zonas.

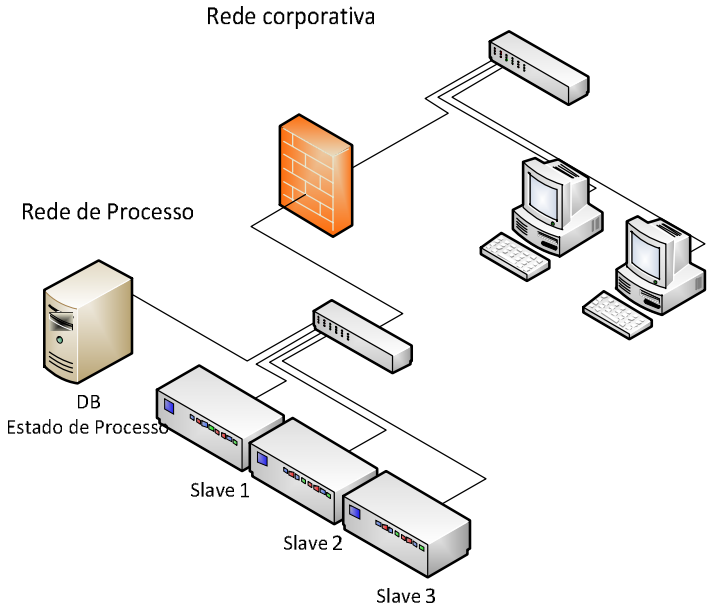

Figura 3.6 - Separação de redes em duas zonas com firewall. (Adaptado de [Byres2005])

A terceira arquitectura de separação é a que oferece maior segurança das três observadas [NISCC2005]. Nesta, a separação é feita em três zonas incluindo uma DMZ onde são alojados os serviços partilhados entre as redes. Normalmente as organizações costumam possuir DMZ para separar a rede da empresa da internet, mas neste caso é referida uma DMZ a separar a rede da organização da rede do processo. Idealmente não são possíveis comunicações directas da rede da empresa para a rede do processo e vice-versa. No entanto, normalmente é necessário existir uma ligação directa para que o fabricante possa efectuar remotamente operações de manutenção e de actualização do *software*. Apesar de ser a arquitectura com maior nível de segurança das três, foi encontrada apenas em algumas indústrias mais desenvolvidas, nomeadamente da área automóvel e do petróleo [Byres2005].

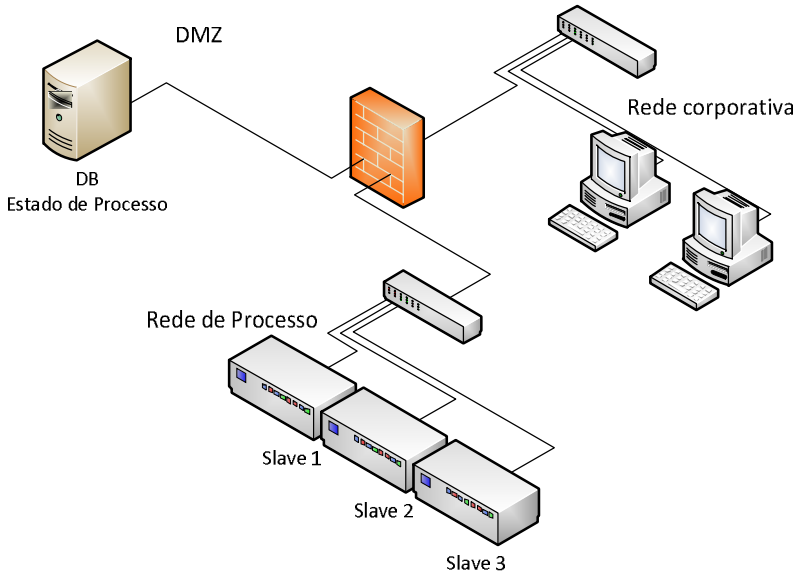

Figura 3.7 - Separação em três ou mais zonas, com DMZ. (Adaptado de [Byres2005])

3.8 Ataques a sistemas SCADA

Os sistemas SCADA não são imunes a ataques que comprometam a integridade do seu funcionamento. Os ataques podem ter várias origens, sendo os dispositivos de armazenamento como *USB Flash Drives* ou *Compact Disc* (CD) um dos possíveis pontos de entrada no sistema. Estes dispositivos são um dos possíveis meios de realizar operações de actualização dos sistemas. Com estes meios também está associado outro risco, nomeadamente a sua utilização por parte dos funcionários para efectuar tarefas pessoais não relacionadas com o sistema. Estes podem infectar negligentemente todo sistema com código malicioso.

No início dos anos 90, a ligação dos sistemas ao exterior introduziu um novo meio para a entrada de potenciais ameaças, a internet. Esta ligação trazia diversas vantagens ao nível de operação, como interligação de infra-estruturas ou a realização de operações de manutenção e actualização por parte dos fabricantes. Por outro lado, esta abertura do sistema ao exterior também possibilitou a exploração de novos vectores de ataque não presentes anteriormente nos sistemas. São diversas as possíveis formas de penetrar no sistema remotamente. Algumas destas podem ser vistas sob a forma de mensagens de correio electrónico com falsas propostas de negócio, com o objectivo de persuadir o receptor a abrir anexos maliciosos (como o exemplo do vírus duqu descrito na secção 3.8.2.5). Além disso, esta abertura dos sistemas possibilita a exploração de falhas de segurança desconhecidas no software, facilitando a instalação e replicação de vírus.

Nesta secção são descritos alguns dos ataques e incidentes conhecidos e tendo por alvo sistemas de controlo considerados relevantes no contexto desta discussão. Refira-se que estes ataques representam apenas uma pequena fração dos ataques efetivamente registados, já que muitos dos ataques que atingiram sistemas SCADA não são publicados, por receio que se crie uma má imagem na indústria atingida. Está disponível online uma base de dados com incidentes em processos industriais [RISI] mas o seu acesso não é livre.

Os ataques aqui descritos foram separados em duas secções, não direccionados a SCADA e direccionados a SCADA. Nos primeiros encontram-se alguns acidentes ocorridos em sistemas de controlo e ataques que tiveram consequências em sistemas de controlo mas que não os tinham como principal alvo, como o vírus *Blaster* ou o *Slammer*. Na segunda secção encontramos alguns ataques que foram planeados com os sistemas de controlo em mente, como sabotagens por parte de ex-funcionários ou vírus desenhados especificamente para esse objectivo.

3.8.1 Ataques não direccionados a sistemas SCADA

Nesta secção são descritos incidentes não direccionados especificamente aos sistemas de controlo.

3.8.1.1 CSX Train Signaling Systems

Este incidente ocorreu em Agosto de 2003, a sua principal consequência foi a inutilização do sistema de sinalização dos comboios em 23 estados a este do rio Mississípi nos EUA (*Estados Unidos da América)*. Houve várias viagens canceladas e vários comboios sofreram atrasos entre 15 minutos a seis horas, dependendo se eram de curta ou longa distância [Niland2003]. A causa encontrada para o acontecimento foi o vírus SoBig [Nahorney2003]. O vector de entrada do vírus na organização foi um e-mail contendo um anexo que quando aberto infectava o computador e se reenviava para outras vítimas a partir dos endereços do computador infectado. Este vírus deixa uma *backdoor* aberta onde o hacker tem acesso sem detecção. Esse acesso pode ser utilizado por *spammers* para descarregar aplicações que produzam *spam* anonimamente.

3.8.1.2 Northeast Power Blackout

Este caso representa um apagão eléctrico na zona norte dos Estados Unidos da América e em Ontário, Canadá [DOE2004]. O acontecimento ocorreu no dia 14 de Agosto de 2003 e teve uma duração de quatro dias. Aproximadamente 50 milhões de pessoas foram afectadas, equivalendo a uma carga de 61.800 MW (*megawatt*) de energia. Foram estimadas perdas entre 4.000 e 10.000 milhões de dólares. A causa do acidente deveu-se parcialmente a uma falha de alarmes do sistema SCADA, que teve por consequência que os operadores de controlo ficaram sem informação sobre mudanças na rede eléctrica. Algumas linhas com uma tensão de 354 kV (*kilovolt*) foram cortadas devido a contacto com árvores. Essas árvores deveriam ter sido identificadas anteriormente como potencial perigo e abatidas.

3.8.1.3 Hatch Nuclear Power Plant Shutdown

Este incidente aconteceu mais recentemente que os anteriores, a 7 de Março de 2008. O local do acontecimento foi numa central nuclear nos Estados Unidos da América, no estado de Geórgia [Aalto2008]. Como consequência do acidente, a componente de segurança do sistema SCADA detectou falta de dados e calculou erradamente o nível de água no sistema de refrigeração das barras de combustível como estando extremamente baixo, o que causou um alarme e a desactivação de emergência da central nuclear. A desactivação da central causou milhões de dólares em perda de receitas, além dos custos para retomar o bom funcionamento da central. A causa do acidente deveu-se a actualizações de *software* na rede corporativa da empresa que causou uma sincronização de dados entre essa rede e a rede de processo devido através de uma ligação entre elas pela *firewall*. Quando o computador da organização reiniciou, repôs os valores do sistema para os iniciais, causando o mau cálculo do nível de água no sistema de refrigeração. Os responsáveis tinham conhecimento da ligação entre as duas redes, mas não tinham conhecimento que as actualizações originavam uma sincronização de dados.

3.8.1.4 Slammer

Este é outro exemplo das consequências de vírus que apesar de não serem desenhados para danificar sistemas SCADA, por vezes acabam por os afectar. Em Janeiro de 2003 a central nuclear de Davis-Bess em Ohio nos Estados Unidos da América, teve o seu sistema de

monitorização de segurança desactivado durante cerca de cinco horas. A máquina que controlava o processo também esteve parada e demorou cerca de seis horas a estar novamente disponível [Poulsen2003]. Apesar de o vírus não possuir *payload* destrutivo, consome recursos excessivos. Quando uma máquina se encontra infectada, o vírus gera endereços IP aleatórios e replica-se. Como tem um tamanho reduzido (376 bytes), ocupa apenas um pacote UDP o que aumenta a sua velocidade de propagação. A causa da infecção na rede da central nuclear encontrava-se numa linha de dados T1 entre a rede da organização e a rede do processo. Apesar das duas redes estarem separadas por uma *firewall* e essa bloquear a porta 1434 onde o vírus operava, a linha T1 tinha passagem directa pela *firewall* sem ser filtrada. A Microsoft já tinha lançado actualizações para os sistemas operativos Windows. No entanto, os servidores da rede corporativa ainda não tinham as actualizações instaladas.

3.8.1.5 Bellingham, Washington, Gasoline Pipeline Failure

Este acidente ocorreu em Junho de 1999 em Bellingham, Washington. Uma ruptura de uma tubagem derramou cerca de 897.000 litros de combustível num ribeiro [McClary1999]. O acidente vitimou duas crianças de dez anos, um jovem de dezoito e fez ainda oito feridos. Foram estimados danos materiais num valor superior a 45 milhões de dólares. A causa do acidente deveu-se a vários factores em cadeia. As tubagens foram danificadas numa construção em 1994 e não chegaram a ser reparados. Na altura do acidente, o sistema falhou o envio de alarmes a reportar um acumular de pressão dentro dos tubos. Também houve uma falha numa válvula de segurança para aliviar a pressão das tubagens. Por último a falta de treino dos operadores para uma situação de emergência agravou as consequências.

Figura 3.8 - Incêndio em Washington [The Bellingham Herald]

Figura 3.9 - Incêndio em Washington (2) [The Bellingham Herald]

3.8.2 Ataques Direcionados a Sistemas SCADA

Nesta secção são descritos incidentes e ataques direcionados especificamente a sistemas SCADA.

3.8.2.1 Maroochy Shire Sewage Spill

Este ataque ocorreu em Janeiro do ano 2000 numa estação de tratamento de esgotos em Queensland, Austrália [Crawford2006]. As bombas da estação de tratamento não iniciavam ou paravam quando comandadas, os alarmes do sistema não eram reportados aos operadores, e existia uma perda intermitente de comunicações entre a *Master* e os *Slaves*. As consequências do acidente resultaram num derrame de cerca de 999.500 litros de esgoto que provocaram cheias em hotéis, parques e rios das proximidades da estação de tratamento. A causa dos acontecimentos resultara de um ex-empregado da empresa instaladora. Ele acedeu ao sistema de controlo a partir da sua viatura no exterior das instalações para alterar o funcionamento do sistema, utilizando um computador portátil com uma placa de rede sem fios. Os danos eram causados esperando ser contratado pela estação de tratamento para os resolver.

3.8.2.2 Kevin Finisterre

Em Setembro de 2008 o investigador de segurança Kevin Finisterre publicou código que permitia o controlo de sistemas SCADA. O investigador tinha a intenção de alertar para as vulnerabilidades existentes nestes sistemas. O código publicado explorava falhas no software CitectSCADA da Citect. A empresa lançou um patch para corrigir a vulnerabilidade e informou os clientes que apenas estariam em risco os sistemas ligados directamente à internet e sem *firewall* [McMillan2008].

3.8.2.3 Luigi Auriemma

Em Março de 2011, Luigi Auriemma [Auriemma2011] tornou público código que permitia explorar várias vulnerabilidades em sistemas SCADA [Zetter2011]. Da mesma forma que Kevin Finisterre tinha feito cerca de dois anos antes, o objectivo de Luigi Auriemma era alertar para os perigos e vulnerabilidades destes sistemas.

3.8.2.4 Stuxnet

O Stuxnet é considerado um dos vírus mais bem desenhados e planeados até aos dias de hoje [Zetter2011a]. Foi descoberto em Junho de 2010 com várias variantes. Estima-se que a infecção da primeira variante ocorreu por volta de Junho de 2009. Este vírus explora várias vulnerabilidades desconhecidas dos sistemas operativos Windows e do sistema de controlo Siemens Simatic WinCC. Os seus vectores de ataque são dispositivos de armazenamento USB ou CD-ROM. Este é o seu método de infecção inicial. Depois propaga-se para outros computadores através da rede de área local. A infecção por dispositivos de armazenamento ocorre quando o operador utiliza o explorador de ficheiros para visualizar o conteúdo da directoria raiz do dispositivo. A falha de segurança MS10-046 [Microsoft2010] explorava uma vulnerabilidade segurança nos ficheiros de atalho (extensão LNK) que permitia a execução de código. Outras vulnerabilidades do Windows que o vírus explorava eram: MS10-061 [Microsoft2010a], MS10-073 [Microsoft2010b], MS10-092 [Microsoft2010c] e MS08-67 [Microsoft2008]. No total, o Stuxnet utiliza quatro vulnerabilidades desconhecidas do Windows, um número invulgar para apenas um vírus tendo em conta o seu valor. Além das desconhecidas, ele toma partido de outras vulnerabilidades, incluindo algumas do sistema de controlo (por exemplo palavras-chave *hard-coded* no sistema da Siemens).

O vírus utilizava ainda dois certificados de segurança aparentemente legítimos para a sua execução. Esses certificados foram furtados no Hsinchu Science Park na Tailândia a duas empresas fidedignas, a J-Micron e a Realtek [Gross2011].

Apesar de o método de propagação ser massivo, o vírus só opera em determinadas configurações de sistema, tanto ao nível dos servidores como da configuração do sistema de controlo (por exemplo: número e distribuição de centrifugadoras). Assim, no início de uma infecção, o Stuxnet efectua algumas verificações como: arquitectura do sistema operativo (o Stuxnet opera apenas em sistemas Windows XP/2k ou Vista/Win7 de 32-bit), verifica os direitos administrativos e o antivírus. Na Figura 3.10 podemos observar as verificações para a instalação efectuadas pelo Stuxnet. Se o sistema não possuir as condições necessárias, o vírus torna-se inerte e não causa nenhum dano ao sistema.

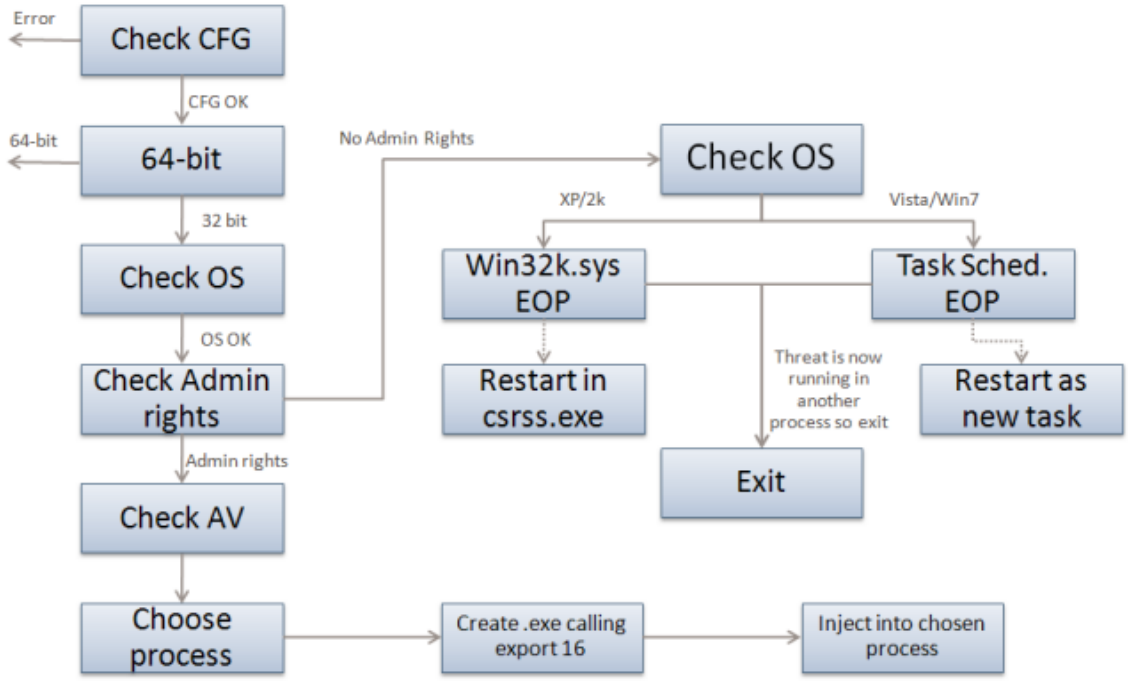

Figura 3.10 - Processo de instalação do Stuxnet (retidado de [Falliere2011])

Acredita-se que o alvo do vírus se encontrava no Irão, numa instalação de enriquecimento de urânio. O vírus tinha como objectivo a destruição de centrifugadoras utilizadas nessa central. O Stuxnet, quando instalado no sistema, enviava comandos aos *Slaves* e mascarava-os dos operadores apresentando valores normais de funcionamento no HMI.

Em funcionamento normal as centrifugadoras tinham uma frequência de rotação de 1064 Hz. O Stuxnet aumentava a frequência para 1410 Hz durante 15 minutos. Passados os 15 minutos, a centrifugadora mantinha a frequência normal durante 27 dias, altura em que o Stuxnet a alterava para 2 Hz durante 50 minutos. Depois, a frequência era novamente alterada para os 1064 Hz originais durante mais 27 dias. Após esse tempo, o Stuxnet voltava ao activo e iniciava a sequência de novo. Esta variação de frequência causava vibrações nas sensíveis centrifugadoras causando-lhes danos de modo a elevar a taxa de avarias mas sem levantar suspeitas, o suficiente para abrandar a produção de urânio. Estas alterações de frequência são ocultas pelo vírus. Para atingir esse objectivo, o Stuxnet intercepta os comandos entre os PLCs e o Windows, mascarando a sua actividade aos operadores do sistema.

Os servidores de comando e controlo localizam-se na Dinamarca e na Malásia e já foram desligados no esforço de desabilitar o vírus. Com 58,85% das infecções conhecidas, o Irão é o país com a maior percentagem, seguido da Indonésia com 18,22% e a Índia com 8,31%.

3.8.2.5 Duqu

Em Outubro do ano 2011, foi descoberto um novo vírus com os sistemas SCADA como alvo [Fisher2011]. O Duqu (nome atribuído devido ao nome de ficheiros que cria) é suspeito de ter sido desenvolvido pelos mesmos responsáveis do Stuxnet, ou pelo menos alguém com acesso ao seu código. No entanto, este possui objectivos muito distintos comparado com o Stuxnet. O Duqu recolhe informação para se utilizada em ataques futuros, através de imagens dos ecrãs e *keyloggers* [Gostev2011], não executando acções de sabotagem.

Ao contrário do Stuxnet, o Duqu não possui uma propagação massiva, sendo a infecção efectuada através de um e-mail escrito individualmente para cada empresa alvo. O conteúdo do e-mail sugere uma intenção de negócio com a empresa. O vector de ataque é um documento Word em anexo [Aleks2011], explorando uma vulnerabilidade previamente desconhecida no kernel do sistema operativo Windows [Microsoft2011], relacionado com o tipo de letra TrueType do Windows. Do mesmo modo que o Stuxnet, o Duqu também utiliza um certificado genuíno para a sua execução, furtado à C-Media.

A máquina infectada fica com ligação aos atacantes que podem efectuar o download de mais componentes para o computador infectado e espalhar o vírus por outros computadores na organização. Computadores infectados que não tenham ligação directa à internet também conseguem comunicação com os atacantes. Foram descobertos ficheiros de configuração que permitiam uma ponte com outro computador infectado com ligação ao exterior, de modo a conseguirem comunicação com os atacantes [Fisher2011a].

Espera-se mais informações sobre o Duqu no futuro, já que ainda se encontram investigações em curso sobre a ameaça.

3.9 Trabalho relacionado

Nesta secção são abordados os dois dos principais projectos de investigação na área de segurança em sistemas de controlo industrial: os projectos VIKING e ESCoRTS.

3.9.1 VIKING

O projecto VIKING tem como objectivo promover a segurança de sistemas SCADA e avaliar os riscos de ataques em todas as partes do sistema [VIKING]. O projecto teve uma duração de 36 meses, com início em Novembro de 2008. Conta com a colaboração de parceiros de diversas áreas Industriais como a ABB, E.ON, MML e Astron Informatics. Tem também a colaboração de três universidades: ETH [ETH], KTH [KTH] e a University of Maryland [UDM].

Tem também como objectivo avaliar os custos e consequências de potenciais ataques na sociedade. O projecto tem como alvo a área da produção da energia por duas razões: em primeiro lugar esta é uma das mais importantes, visto que todas as outras estão dependentes dela. E em segundo lugar os resultados do projecto também podem ser aplicados nas outras áreas.

Nas seguintes secções são abordadas três ferramentas desenvolvidas no decurso do projecto:

- ViCiSi - Cities Simulator;
- CySeMoL - Cyber Security Modeling Language;
- Viking Testbed;

3.9.1.1 ViSiCi - Cities Simulator

Esta ferramenta pretende efectuar a simulação de uma sociedade virtual [VIKING2010]. Esta é simplificada, mas no entanto funcional. É constituída pelas características vitais de uma sociedade normal. A sociedade possui estruturas estáticas e dinâmicas. Tem infra-estruturas como apartamentos, ruas e redes de electricidade. Existem empresas públicas e privadas que produzem bens e pessoas que os consomem.

A essa sociedade foi dado o nome de país VIKING. Este é implementado com base em ficheiros de template. Estes templates estão disponíveis para todos os países da Europa e possuem valores demográficos, estatísticas económicas do Eurostat, e estatísticas energéticas dos operadores de energia.

A sociedade relaciona a necessidade de energia para uma vida económica. Possui as actividades para a produção de consumo. Este consumo é baseado no produto interno bruto do país simulado.

O simulador cria cenários de falhas energéticas para avaliar os impactos monetários e não monetários na sociedade. As falhas de energia são simplificadas. Na altura da falha, toda a produção pára. Não existe produção ou consumo de bens durante esse período. O restabelecimento de energia já possui uma modelação mais complexa. Dependendo da actividade pode ou não existir um maior consumo durante o restabelecimento de energia. Por exemplo uma arca congeladora que consome alguma da energia perdida durante a falha, observando-se um maior consumo energético no período do restabelecimento de modo a restabelecer o estado em que se encontrava anteriormente à falha. Por outro lado, uma

lâmpada incandescente já não terá esse restabelecimento de energia, visto fica num estado igual ao da altura da falha no momento que a energia é restabelecida.

A avaliação dos custos e consequências possui duas dimensões:

- Monetárias: custo calculado com base na diferença do produto interno bruto com e sem falha energética;
- Não Monetária: perspectivas Macro e Micro das consequências da sociedade, avaliadas pelo tempo da falha e sua incidência;

A micro perspectiva foca-se na avaliação do ponto de vista individual. Foca-se nos aspectos como perda de bens, problemas com transportes, falta de combustíveis, entre outros. É avaliado pela duração da falha. Na macro perspectiva são calculadas as consequências dependendo do número de pessoas afectadas. São avaliados riscos na sociedade como proliferação de doenças, motins, etc.

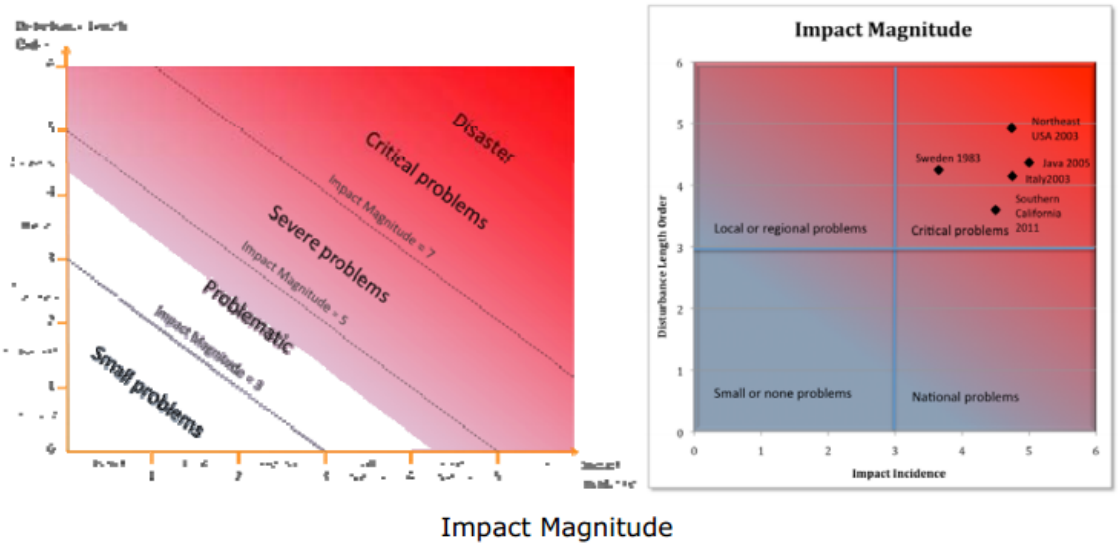

Figura 3.11 - Avaliação de impacto na sociedade nas Micro e Macro perspectivas. (Retirado de [VIKING2010a])

3.9.1.2 CySeMoL - Cyber Security Modeling Language

Esta ferramenta foi desenvolvida com o intuito de fornecer uma avaliação da segurança de um sistema [VIKING2010a]. Possibilita a descrição da arquitectura, bem como a sua avaliação. Podem ser representadas várias entidades como serviços, fluxos de dados, sistemas operativos, sistemas de detecção de intrusões, etc. Em cada entidade podem ser definidas propriedades que possibilitam uma avaliação mais precisa, como por exemplo se os serviços possuem vulnerabilidades conhecidas ou se o sistema operativo utiliza memória executável.

Dependendo da arquitectura definida, vão ser possíveis diferentes processos de ataque. Em cada processo podem ser atribuídas probabilidades condicionais como podemos observar na Tabela 3.1 onde estão ilustradas as probabilidades de um ataque de *Denial-of-Service* semântico.

Tabela 3.1 - Probabilidades condicionais de um ataque. (Adaptado de [VIKING2010a])

Existem vulnerabilidades no software	Atacante possui credenciais de acesso	Probabilidade de sucesso do ataque
Sim	Sim	0.72
Sim	Não	0.53
Não	Sim	0.60
Não	Não	0.38

Para a avaliação da segurança do sistema o CySeMoL utiliza dados publicados em estudos anteriores e informações recolhidas através de inquéritos a especialistas de segurança. Os inquéritos tiveram um número máximo de 165 participantes e os cenários assumem que o atacante possui bons conhecimentos e uma semana de preparação. Para verificar a qualidade dos inquéritos foi utilizado o método de Cooke.

A ferramenta pretende fornecer uma análise do sistema do ponto de vista da segurança sem que o utilizador seja especialista. O utilizador define a arquitectura, e os alvos de ataque e seus pontos de início. O CySeMoL fornece resultados para cada par de início e destino. Se o utilizador pretender obter uma visão geral sobre a segurança de todo o sistema terá de definir vários pares alvo - início. Variando a arquitectura, o utilizador pode perceber como aumentar a segurança do sistema.

A precisão dos resultados está fortemente relacionada com as informações adquiridas empiricamente (inquéritos). A utilização do método de Cooke permitiu uma melhor precisão nos resultados, e no fim foi utilizado o método de turing para avaliar os resultados da ferramenta. Alguns cenários foram avaliados pela ferramenta e por um grupo de especialistas. Todas as soluções foram apresentadas a um segundo grupo de especialistas para tentarem diferenciar quais as soluções resultantes da ferramenta e do primeiro grupo de especialistas. Esse segundo grupo de especialistas não as conseguiu diferenciar.

Apesar do possível grau de incerteza resultante dos inquéritos, um sistema que obtenha uma probabilidade de ataque de 10% será sempre mais seguro que um que obtenha 30%. As probabilidades condicionais necessitam de ser actualizadas de tempos em tempos porque os conhecimentos dos atacantes e dos defensores varia no tempo. A ferramenta reflecte a situação para o ano de 2010.

3.9.1.3 VIKING Testbed

A bancada de testes foi desenvolvida para demonstrar os ataques e avaliar as suas consequências nas redes eléctricas e na sociedade [VIKING2010b]. Esta possui elementos reais e simulados. O sistema SCADA está na parte real da bancada. Nos objectos simulados, temos a modelação das características eléctricas, o modelo de sociedades virtuais apresentado em 3.9.1.1 e um emulador de comunicações entre as duas primeiras partes. As diferentes partes constituintes podem ser observadas na Figura 3.12.

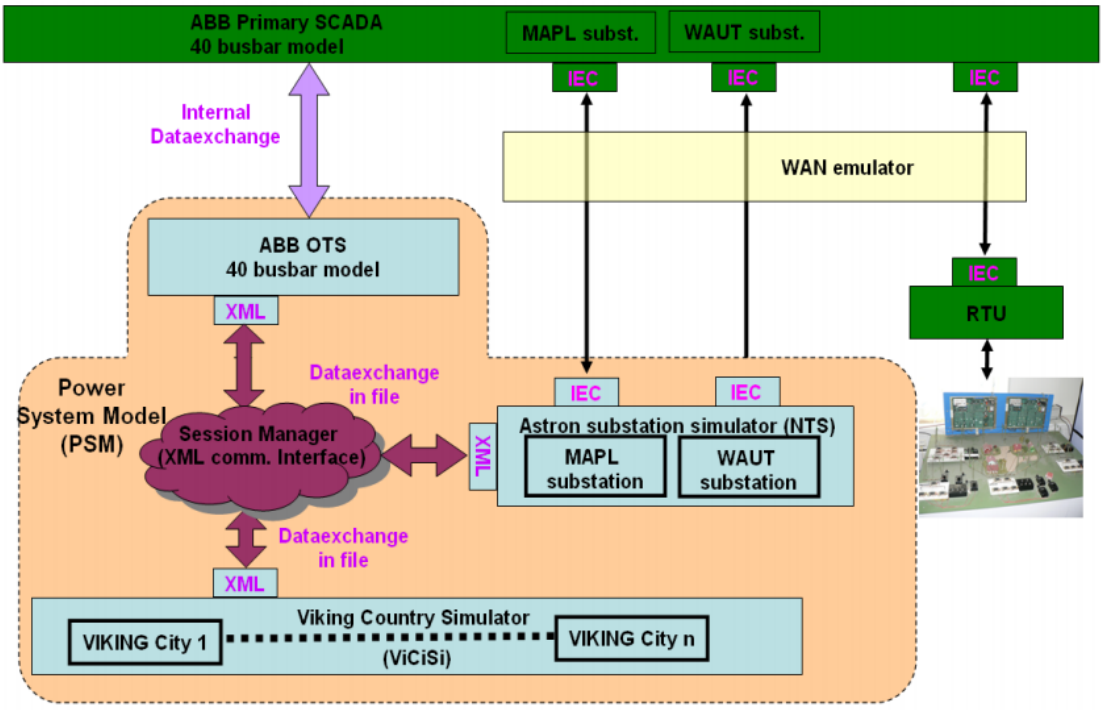

Figura 3.12 - Bancada de testes do projecto VIKING. (Retirado de [VIKING2010b])

3.9.2 ESCoRTS

O projecto ESCoRTS tem como principais objectivos o aumento da segurança em sistemas de controlo através da disseminação de boas práticas aplicadas na área e a criação e aplicação de standards [ESCoRTS]. Entre os vários parceiros encontram-se alguns fabricantes de equipamento: ABB, Areva, Siemens. Também se encontram alguns utilizadores dos sistemas: Enel Produzione (companhia eléctrica italiana), Transelectrica (distribuição eléctrica romena), Mediterranea delle Acque (gestão hídrica, italiana). A análise centra-se na aplicação de normalização ao nível dos sistemas de controlo, não na avaliação da sua eficácia. Relativamente à aplicação de normas de segurança, concluiu-se que a mais promissora e que possui mais abrangência ao nível de segurança em sistemas de controlo é a norma ISA99/IEC62443 [ESCoRTS2010].

Existem várias normas que abordam o tema da segurança em sistemas de controlo (37 no total). Desses, 14 são originários de comités dos EUA, 10 têm origem em organismos europeus e 13 internacionais. A maioria destas normas são definidas para o sector energético, 5 são genéricos, 13 abordam as áreas da automação, 4 abordam as indústrias do petróleo e gás, e 2 do sector químico [ESCoRTS2010a].

Na Europa existe menos sensibilização para as questões de segurança comparando com os EUA. Estas diferenças estão em parte relacionadas com a falta de procura por estes sistemas, e muitas vezes os custos de falha dos sistemas críticos não serem comparados com os custos de implementação de medidas de segurança rigorosas [ESCoRTS2008].

As métricas de segurança também são abordadas no projecto. Estas são geralmente muito amplas sem definições precisas para serem utilizadas em cenários reais, ou demasiado restritas para abranger várias áreas da segurança [ESCoRTS2010b]. As avaliações de segurança podem ser executadas por várias razões: imposta por lei, necessárias devido a obrigações contratuais, ou decorrentes de um processo de certificação. Na área de controlo industrial não existem fortes imposições legais. Pelo estudo podemos ver também que as métricas na área de segurança são um tema recente, ainda em desenvolvimento, com poucas aplicações práticas. São propostas algumas métricas que apesar de não serem totalmente completas, permitem responder a algumas questões [ESCoRTS2010b].

O projecto também aborda a temática de cenários onde podem ser recolhidas informações para a criação de standards e guias de boas práticas [ESCoRTS2011]. Dos dois ambientes possíveis de recolher informações (sistemas reais e bancadas de testes em laboratório) é abordado o ambiente de laboratório. Esta opção deve-se à falta de dados de ambientes reais associado à pouca frequência de incidentes de segurança e da possível revelação de dados sensíveis sobre as indústrias. São apresentadas algumas considerações no seu desenho e são definidos alguns requisitos para conseguir uma bancada de testes que recrie um sistema capaz de obter dados com a maior proximidade possível com um sistema real.

No decorrer do projecto também foram deduzidas taxonomias para ataques e vulnerabilidades em ambientes SCADA apresentadas em [ESCoRTS2010c]. Estas vulnerabilidades são agrupadas em quatro categorias principais:

- Arquitectura;

- Políticas de segurança;
- Software;
- Protocolos de comunicação;

Na primeira categoria são descritos a fraca separação entre as diferentes redes do sistema, a falta de autenticação dos equipamentos activos e a pouca atenção para o balanceamento de carga e redundância de rede.

Na segunda categoria destacam-se a falta de actualizações do software, desleixo nas políticas de acesso que normalmente são bem definidas mas a sua implementação acaba por não cumprir os requisitos. A documentação do sistema normalmente é bem definida no início mas não é actualizada com a evolução do sistema. Do mesmo modo, as auditorias de segurança não são frequentes (quando existem) o que também é um erro visto que um sistema seguro hoje pode não o ser amanhã.

As vulnerabilidades de software estão normalmente associadas à falta de rigor na aplicação das políticas de actualização.

Na última categoria de vulnerabilidades também são apresentados alguns problemas em relação a dois dos protocolos de comunicação mais utilizados (DNP3 e Modbus) como descrito na secção 3.6. Ambos são extremamente vulneráveis a ataques clássicos como *Man-In-The-Middle*, execução de comandos não autorizada, ataques por repetição de pacotes e escuta.

Além da taxonomia de vulnerabilidades, também são descritos alguns cenários de ataque a estes ambientes, onde se destacam:

- Ataques orientados aos protocolos;
- Ataques a redes de processos;
- Ataques às *Exchange Networks*[2] que estão divididos em ataques aos sistemas de diagnóstico e às bases de dados de tempo real;

Por último, ainda em [ESCoRTS2010c] encontramos contra medidas para minimizar o impacto das vulnerabilidades e ataques descritos anteriormente.

[2] Nas Exchange Networks encontra-se equipamento que faz a tradução de protocolos entre sistemas SCADA com tecnologias diferentes, ou comunicação com a rede organizacional.

4 Arquitectura proposta para CockpitCI (*Detection Layer*)

No âmbito do projecto Europeu FP7 CockpitCI, uma das tarefas onde a Universidade de Coimbra tem um papel preponderante corresponde á definição e posterior implementação de uma plataforma de detecção de ataques informáticos às aplicações SCAAD e/ou aos processos industriais que estas controlam. A Universidade de Coimbra coordena o *Workpackage* relacionado com esta vertente (WP3000).

Nesse âmbito, um dos objectivos propostos para esta dissertação foi a definição do primeiro esboço do que virá a ser a arquitectura de referência deste componente do CockpitCI. Dado o desfasamento entre o calendário do trabalho de dissertação e o calendário do projecto, o trabalho aqui apresentado não corresponde necessariamente à arquitectura final que virá a ser adotada para o CockpitCI (que ainda se encontra em discussão e refinamento com os restantes parceiros do projecto), mas sim um primeiro esboço – produzido antecipadamente pela Universidade de Coimbra para suporte a essa atividade. Não obstante, tendo em conta as reações dos restantes parceiros às propostas da Universidade de Coimbra, espera-se que a arquitectura final dos componentes de detecção de intrusão não seja substancialmente diferente da arquitectura aqui apresentada.

Na Secção 4.1 é feita uma introdução sumária ao Projecto CockpitCI. Enquanto as Secções seguintes se focam na arquitectura proposta especificamente para os componentes de detecção de intrusões (âmbito, conceitos, componentes, etc.).

4.1 Projecto CockpitCI

Como referido no início do documento, o projecto CockpitCI aborda a temática de segurança de infra-estruturas críticas, mais concretamente nos sistemas de controlo industrial SCADA. Neste contexto, a segurança sempre andou a um ritmo separado dos sistemas de informação convencionais, deixando estas infra-estruturas com défices de segurança quando comparadas com os sistemas de segurança implementados actualmente em plataformas TIC convencionais.

Este projecto com duração prevista de 36 meses tem como alvo os ataques informáticos, estando os ataques físicos de força bruta parcialmente fora do espectro do projecto (como por exemplo corte de cablagem ou danificação de equipamento).

A ferramenta resultante do projecto estará associada a uma CI (infra-estrutura crítica, *Critical Infrastructure*), e será capaz de interagir com outras CIs que tenham a ferramenta implementada.

A arquitectura geral do projecto podem ser observada na Figura 4.1 (os números incluídos na Figura correspondem às tarefas especificas onde estes componentes são concebidos e/ou desenvolvidos, de acordo com o plano de trabalhos estabelecido).

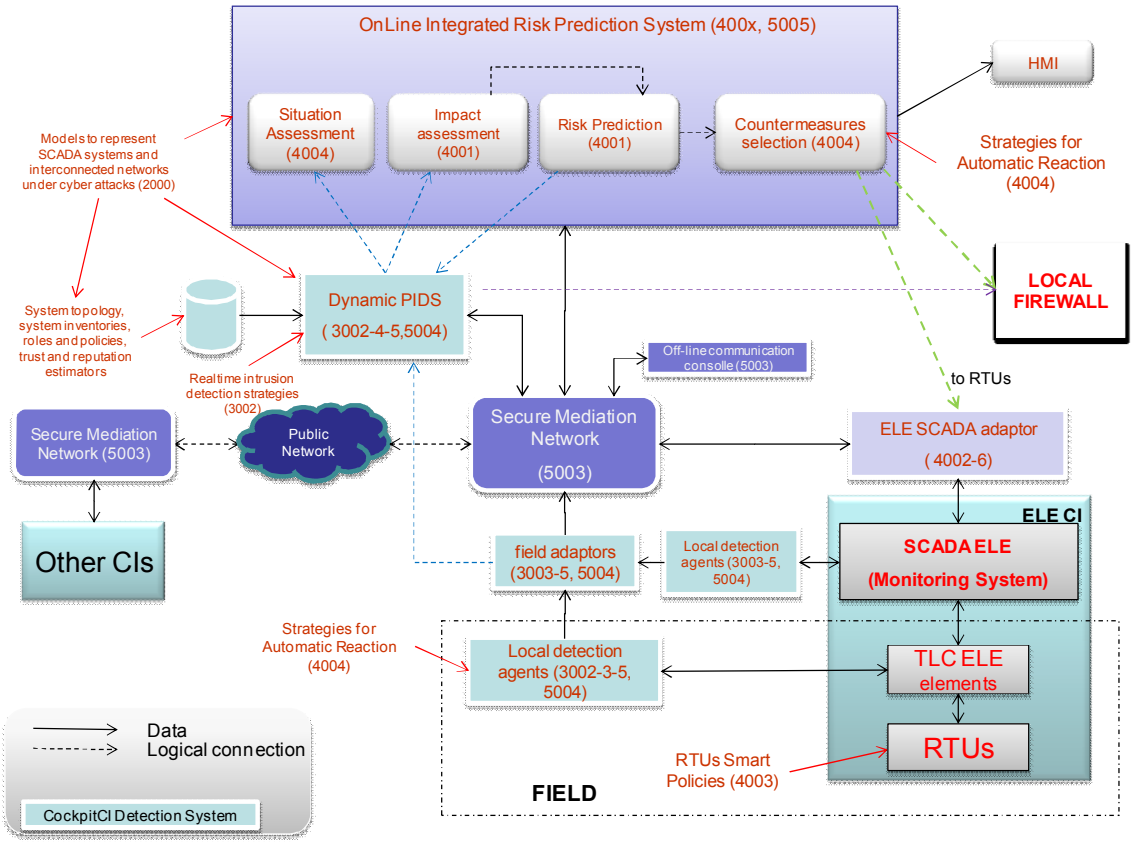

Figura 4.1 - Descrição geral do projecto CockpitCI

4.2 Âmbito de Arquitectura proposta

A arquitectura proposta nesta dissertação corresponde especificamente aos componentes de detecção de intrusão (ao nível da rede e dos diversos componentes da plataforma SCADA). De acordo com a abordagem do projecto, assume-se que estes componentes servem essencialmente para *detecção*, sendo as medidas de resposta a eventuais intrusões – sejam elas preventivas ou reativas – executadas noutros componentes do sistema CockpitCI (pelo menos a nível conceptual).

Dentro deste contexto, foi apresentada uma proposta de arquitectura de detecção de intrusões abrangente mas com diversas particularidades referentes aos ambientes de controlo e aos sistemas SCADA. As próximas secções apresentam os objectivos desta arquitectura, os seus principais requisitos e o modo de operação esperado. São também descritos os seus principais componentes.

É também importante mencionar que até este momento ainda não foi efectuada nenhuma implementação ou validação da arquitectura proposta. Neste momento encontra-se em curso a definição da arquitectura ao nível do consórcio (prevendo-se que a sua versão preliminar fique concluída até 30 de Setembro de 2012). Este esboço será depois gradualmente refinado, até à versão final da arquitectura, a concluir até Junho de 2013. O trabalho de implementação e validação irá decorrer posteriormente.

4.3 Conceitos Subjacentes

Como foi mencionado anteriormente, esta arquitectura tem como principal objectivo definir uma estrutura genérica de detecção para aplicar à ferramenta CockpitCI. Esta arquitectura possui sensores para a monitorização de tráfego de rede, eventos nos *hosts*, e componentes de análise que possibilitam a detecção de comportamentos anormais. Existe também a possibilidade da ferramenta tomar decisão e reagir a ameaças, em situações específicas (por exemplo, se o evento necessitar de uma reacção numa janela temporal pequena).

Esta arquitectura tem em vista a detecção de vários cenários de ataque, onde se podem apresentar a título de exemplo os seguintes:

- Envio de comandos não autorizados para equipamento de controlo;
- Envio de informação falsa para um operador, fazendo-o tomar acções inapropriadas;
- Alterar o funcionamento do sistema, atrasando ou bloqueando os fluxos de informação na rede de controlo;
- Efectuar alterações não autorizadas dos equipamentos de controlo, nomeadamente modificação de alarmes, ou outras configurações;
- Afectar a disponibilidade de recursos através da propagação de programas maliciosos (por exemplo, vírus ou *worms*).

A arquitectura apresenta dois conceitos relativamente inovadores para a captura e análise de pacotes: *Shadow RTU* e BMS (*Backup Master Station*). O primeiro tem como função a monitorização das interacções existentes entre os RTUs e o restante sistema. A BMS ficará responsável pela monitorização de uma determinada (FN) *Field Network*, sendo também capaz de tomar algumas medidas de reacção a determinados eventos.

Devido aos elevados requisitos de disponibilidade e não tolerância a atrasos, a arquitectura de detecção funcionará uma rede conceptualmente separada do sistema SCADA (mesmo que em alguns casos possa ser implementada sobre a mesma rede física e/ou lógica), não existindo assim o risco de a ferramenta atrasar, bloquear, ou alterar o fluxo de pacotes entre os diversos componentes do sistema. De modo a suportar ambientes de grandes dimensões, a solução deverá ser escalável.

4.4 Camada de Monitorização (*Generic Probing Layer*)

Para a detecção de intrusões, é necessário capturar informações sobre o funcionamento do sistema. A arquitectura de captura apresentada nesta secção – *CockpitCI Generic Probing Layer* – descreve o modo de alcançar esse objectivo. De seguida é descrita a divisão do sistema em zonas e os sensores para a aquisição dos dados necessários.

4.4.1 Definição de zonas

A arquitectura encontra-se dividida em três zonas distintas de captura de tráfego, como se pode observar na Figura 4.2:

1. ***IT Network*** - Representa a rede da organização. Esta rede não se encontra no sistema SCADA, mas em alguns casos encontram-se componentes do sistema, por exemplo

consolas HMI. Esta rede é também um possível ponto de entrada de ameaças no sistema, como por exemplo o *worm* Slammer, descrito na secção 3.8.1.4;

2. ***Operations Network*** - Representa a rede de operações do sistema. Aqui encontram-se os servidores responsáveis pela gestão e comando do sistema. Alguns dos componentes que esta zona contém são: *Master Stations*, servidores de *DBMS* (*Database Management System*), consolas HMI;
3. ***Field Network*** - Esta zona representa o nível mais baixo do sistema. Aqui encontram-se os RTU e os sensores e actuadores do sistema.

A separação em zonas permite uma distinção entre os diversos contextos do sistema SCADA anteriormente referidos. Deste modo a colocação de sensores para a captura de tráfego e a comunicação entre os diversos níveis é simplificada. Os sensores de rede (*Network* IDS) são colocados nas fronteiras das redes, analisando pacotes trocados entre redes adjacentes. A informação adquirida referente aos componentes principais do sistema SCADA (por exemplo numa *Master Station*) está entregue a *Host* IDS.

4.4.2 NIDS

Um NIDS monitoriza o tráfego numa determinada rede. Os sensores deste componente serão colocados na fronteira de cada rede, capturando o tráfego trocado entre redes adjacentes. A captura é efectuada através de *port mirroring*, deste modo a análise não é intrusiva para o sistema. Com a captura de pacotes é possível efectuar uma análise de eventos indesejados ao nível de rede, através de análise de padrões (*pattern-based*) ou procurando desvios no comportamento normal do sistema (*anomaly-based*).

4.4.3 HIDS

Com o HIDS torna-se possível analisar intrusões ao nível dos *hosts*. Instalados nos componentes principais dos sistemas (como por exemplo *Master Stations*), analisam alguns parâmetros relativos ao *host*. Um dos exemplos é a integridade de ficheiros de sistema, configurações, e de programas, através de assinaturas de ficheiros.

4.4.4 HoneyPots

A arquitectura prevê a utilização de *HoneyPots* em várias zonas do sistema. Estes componentes simulam o funcionamento de componentes reais do sistema, mas não desempenham nenhuma tarefa de produção. As comunicações recebidas por eles são por norma ilegítimas (devido a estes não receberem comunicações do resto do sistema), permitindo obter informações sobre ataques. Os *HoneyPots* operam no espaço de endereçamento da rede não utilizado.

Ataques a componentes de produção podem ser reencaminhados para estes componentes, permitindo a recolha de mais informações sobre o ataque.

4.4.5 Shadow RTUs

Um *Shadow RTU* representa um dispositivo capaz de monitorizar as interacções que um determinado RTU tem com o sistema. Esta tarefa pode ser desempenhada por um equipamento com capacidade de processamento e de I/O limitada (por exemplo um Arduino [Arduino]).

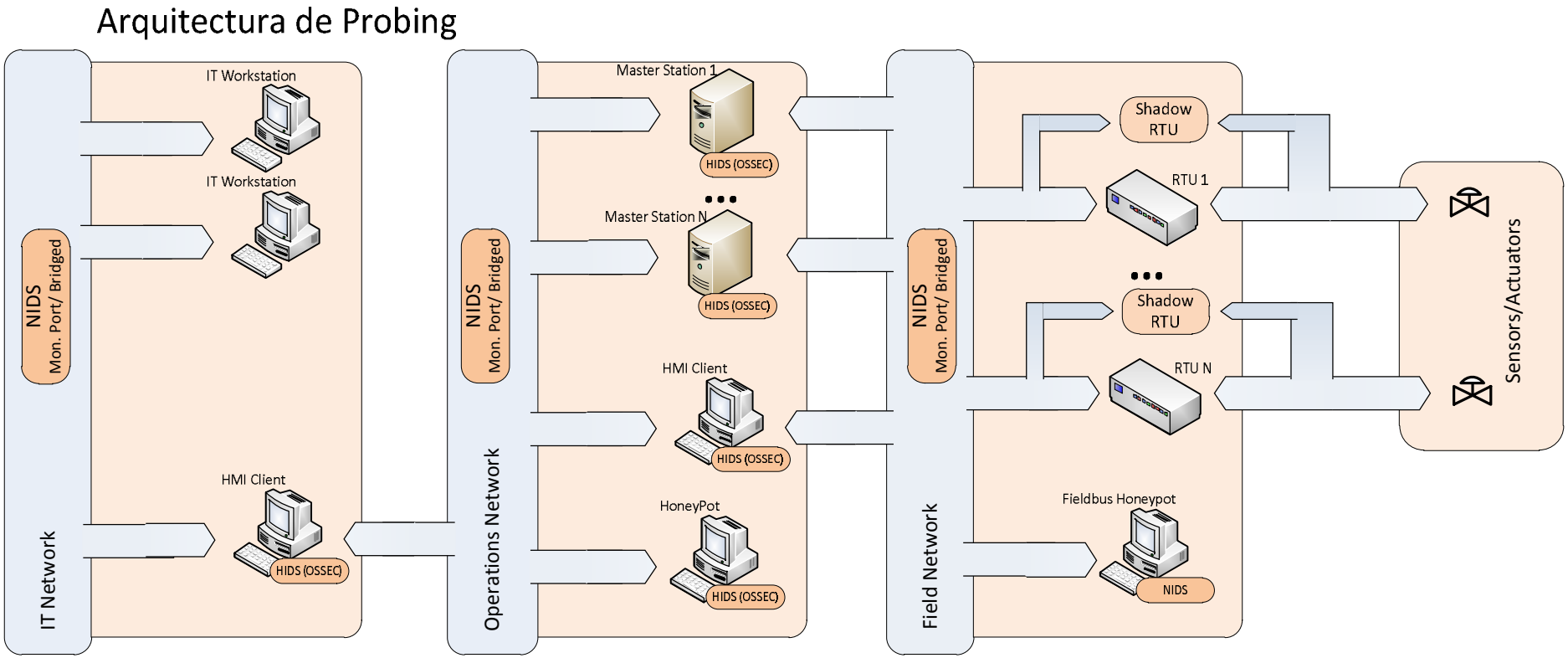

Figura 4.2 - Camada de captura.

4.5 Camada de Tratamento de Dados

É necessária uma infra-estrutura para analisar os dados adquiridos através da camada de captura. Esta secção descreve a camada responsável pela correlação/inferência de dados – com base nas informações capturadas através da camada de captura apresentada na secção anterior. A correlação encontra-se dividida em dois níveis: correlação local, e principal. Este princípio aumenta a escalabilidade da proposta e melhora também a sua funcionalidade. A informação trocada pelos diversos componentes utiliza o formato IDMEF (*Intrusion Detection Message Exchange Format*), adoptado por grande parte dos IDS, simplificando a troca de mensagens.

Correlação Local

O nível de correlação local é contido em cada segmento de rede, como se pode observar na Figura 4.3. Neste nível, a correlação é efectuada apenas com os dados da rede local. Resultados desta análise são enviados para o correlacionador principal. Os correlacionadores locais podem ter também alguma capacidade de decisão e reacção em situações específicas. Esta segmentação possui vantagens na organização, permitindo uma melhor gestão dos vários componentes. Outra vantagem encontra-se na melhor distribuição das necessidades de processamento, quando implementado em sistemas de grande dimensão. Os correlacionadores locais de cada FN encontram-se integrados na BMS (descrita na secção 4.5.1).

Correlação Principal

O correlador principal encontra-se num nível superior aos correlacionadores locais, recebendo os seus eventos. Assim obtém uma visão mais abrangente do sistema, podendo despoletar acções de reacção abrangendo toda a infraestrutura. Em ambientes de maior dimensão esta unidade de correlação local poderá ser separada em várias unidades, por exemplo de acordo com critérios geográficos ou funcionais para maior escalabilidade.

4.5.1 Backup Master Station

A *Backup Master Station* consiste numa *Master Station* local numa determinada FN, com algumas funcionalidades extra. As funcionalidades de controlo dos RTUs encontram-se inactivas durante o normal funcionamento do sistema. O controlo destes é activo apenas na eventualidade do isolamento da FN. Nessa circunstância, ela possui um conjunto de regras para operar os RTUs sem intervenção de operadores, como por exemplo o encerramento dos componentes da FN de forma ordeira.

4.5.2 Autonomous System

Um Autonomous System consiste numa FN com um grau de autonomia suficiente para desempenhar tarefas pré-determinadas na eventualidade do seu isolamento. A BMS desempenha um papel importante na autonomia de acções.

4.5.3 Plataforma de gestão

A plataforma de gestão torna possível a gestão da ferramenta de detecção por parte dos operadores.

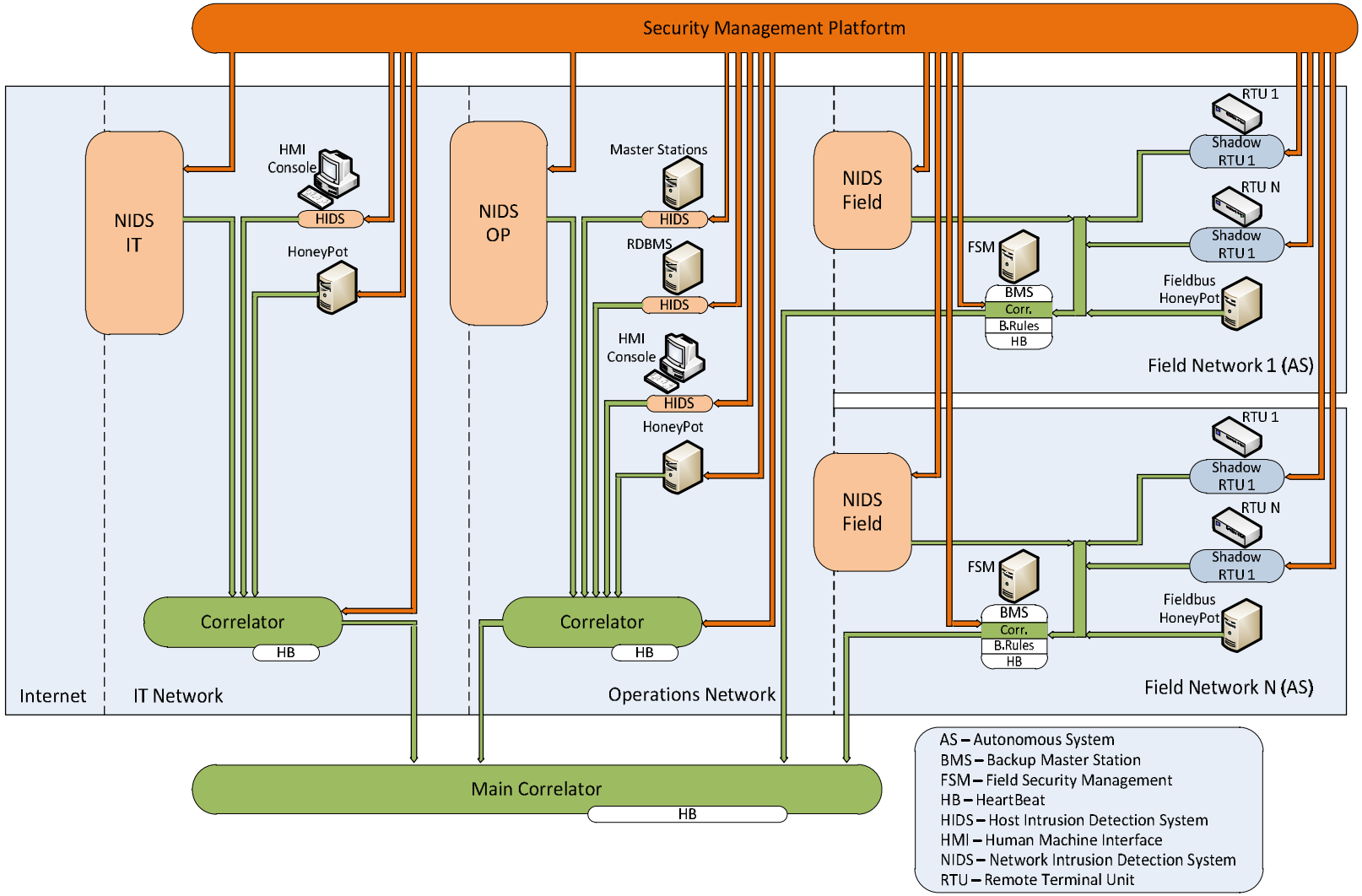

Figura 4.3- Camada de detecção.

4.5.4 HeartBeat

Com esta técnica torna-se possível de detectar falta de comunicação com o sistema. O envio periódico de um sinal para outros componentes do sistema e sua resposta permite a confirmação da sua conectividade. A falta de resposta denuncia isolamento e permite a execução das acções definidas para esse estado.

4.5.5 Plataforma de gestão

A plataforma de gestão permite definir as configurações dos diversos componentes do sistema e sua monitorização.

5 Laboratório de segurança

Este capítulo descreve os dois pontos mais práticos do trabalho, relacionados com a preparação da bancada de testes para suporte do projecto CockpitCI.

Conforme foi já mencionado, o grupo da Universidade de Coimbra envolvido no projecto não tem qualquer experiência no uso de ferramentas SCADA, e por isso um dos objectivos do projecto consistiu na proposta e implementação de uma bancada de testes que possa dar, em primeiro lugar, mais experiência *hands-on* na experimentação de soluções de segurança em ambientes SCADA e, em segundo ligar, suporte às atividades de validação do projecto CockpitCI (mesmo tendo em conta que grande parte da validação será feita num laboratório de maiores dimensões de um outro parceiro do projecto: a *Israel Electric Corporation* (IEC). Tendo em conta estes objetivos, depois de um trabalho inicial de levantamento bibliográfico, foram selecionados e adquiridos equipamentos SCADA, que foram posteriormente integrados num laboratório, em conjunto com algumas ferramentas já existentes de detecção de intrusão. Por vicissitudes às quais o candidato é alheio (nomeadamente o arranque tardio do projecto CockpitCI, em relação à data prevista, e a chegada ainda mais tardia dos equipamentos SCADA encomendados), este trabalho não foi tão longe quanto teria sido possível em outras circunstâncias. Não obstante, foi já possível montar toda a bancada de testes e fazer diversas experiências preliminares.

Um outro aspecto, dentro deste objetivo, relaciona-se com a interligação das diversas bancadas de teste dos diversos parceiros do projecto. Ainda que este aspeto não estivesse inicialmente previsto no plano de trabalhos, por iniciativa da Universidade de Coimbra irá proceder-se à interligação das diversas bancadas de teste, por meio de redes seguras (VPN – *Virtual Private Network*), de planos de endereçamento articulados e de ferramentas diversas de acesso remoto. Deste modo será possível ter acesso a uma bancada de testes de maior dimensão e com maior diversidade de cenários, equipamentos e tecnologias SCADA. Nesse contexto, coube ao candidato a preparação da proposta que fui apresentada ao consórcio e que se encontra presentemente em fase de implementação por parte de vários parceiros.

5.1 Integração de Bancadas de teste

Como foi referido anteriormente, vários parceiros do consórcio possuem bancadas de teste internas para auxílio no desenvolvimento do projecto CockpitCI. O LCT (Laboratório de Comunicações e Telemática, o grupo da UC envolvido no Projecto CockpitCI) propôs a interligação destas através de uma VPN de nível 3 com o espaço de endereçamento partilhado. Desse modo, o ambiente de desenvolvimento e teste de soluções torna-se mais diversificado e completo para os parceiros envolvidos.

Originalmente, a proposta incluía suporte para comunicações de nível 2, opção posteriormente abandonada devido ao aumento de complexidade da solução.

Para atingir os objectivos, foi eleito o protocolo IPSec principalmente devido às vantagens trazidas em termos de latência quando comparado com outras alternativas como o PPTP ou OpenVPN. Os detalhes técnicos da proposta podem ser visualizados no Anexo C - **.** O esquema de interligação proposto pode ser visualizado na Figura 5.1.

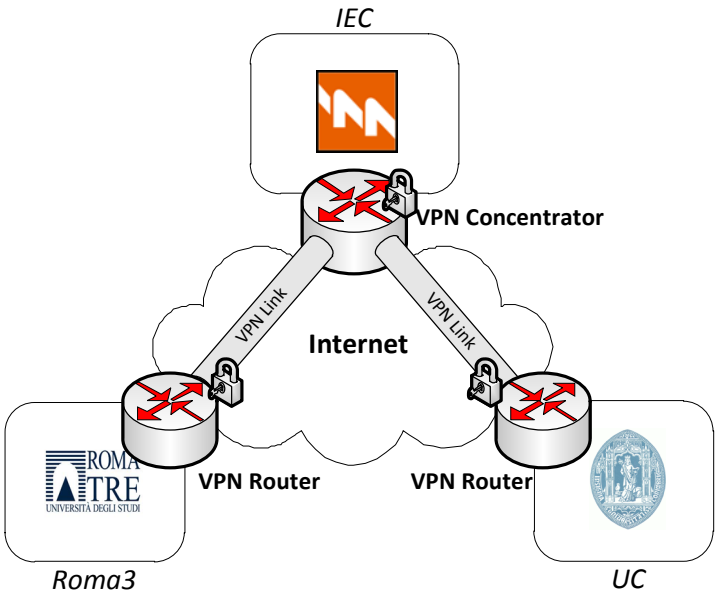

Figura 5.1 - Esquema de VPN para integração de bancadas de teste.

Até ao momento, três parceiros do consórcio possuem bancadas de teste: IEC, Universidade de Roma 3 e Universidade de Coimbra. Outros parceiros manifestaram já interesse na futura integração, tais como a ENEA (Agência Nacional de Energia Atómica – Itália). A proposta prevê a utilização de *links*, permitindo a troca de informação de forma segura. Um concentrador de VPNs que permite agregar os *links*, encontrando-se na IEC.

Com a aprovação do plano, foi desenvolvida internamente uma proposta para a integração destas bancadas, existentes entre os parceiros. A proposta encontra-se no Anexo C - desta dissertação.

Para validar a proposta, implementou-se o cenário recorrendo ao uso de máquinas virtuais. Primeiro com a inclusão da camada de rede de nível 2, e posteriormente apenas com a camada de nível 3 aquando do abandono da camada de nível 2.

5.2 Bancada de Testes Local

De seguida é descrita a bancada de testes desenvolvida e implementada localmente no LCT. Esta vai apoiar o laboratório no desenvolvimento de soluções durante o decorrer do projecto CockpitCI (por exemplo, na validação da arquitectura de detecção apresentada na secção 4). Esta bancada será também incorporada na bancada integrada apresentada na secção 5.1. As seguintes secções apresentam a disposição, os componentes, e o trabalho desenvolvido.

5.2.1 Componentes

A bancada é composta por vários componentes, onde se podem destacar o PLC Modicon M340 da Shneider os simuladores (*Schneider PLC Simulator*) e a *Master Station*. Com a bancada, é possível recriar cenários de controlo, através da interação dos diferentes componentes.

Modicon M340

O PLC Modicon M340 foi adquirido pelo laboratório para apoiar o trabalho desenvolvido no decorrer do projecto CockpitCI. Este funciona utilizando o protocolo Modbus e apresenta um

conjunto de entradas e saídas físicas. Com ele podemos recriar cenários de controlo, com um conjunto de entradas e saídas, controlando-as com lógica escrita num conjunto de linguagens, entre elas: ST (*Structured Text*), LD (*Ladder Diagrams*) e FBD (*Function Block Programming*).

Modicon PLC Simulator

O Modicon PLC Simulator, integrado no kit M340 adquirido, torna possível a simulação de um PLC real. Este pode ser programado com a mesma lógica que o M340, permitindo criar cenários de teste mais elaborados com um conjunto de PLCs reais e simulados em execução. Sendo um componente simulado, tem como desvantagem a inexistência de entradas e saídas físicas comparado com o real. No entanto, estas entradas e saídas podem ser recriadas a partir do envio de comandos através do protocolo Modbus. A principal vantagem deste simulador comparado com os analisados anteriormente é a sua capacidade de ser programado. Muitos dos simuladores existentes têm como principal objectivo o teste de ligação entre *Master Station* e PLC, e consequentemente apenas permitem operações simples como a escrita e leitura de registos.

A Figura 5.2 representa o esquema da bancada de testes implementada com os seus componentes e respectivas ligações.

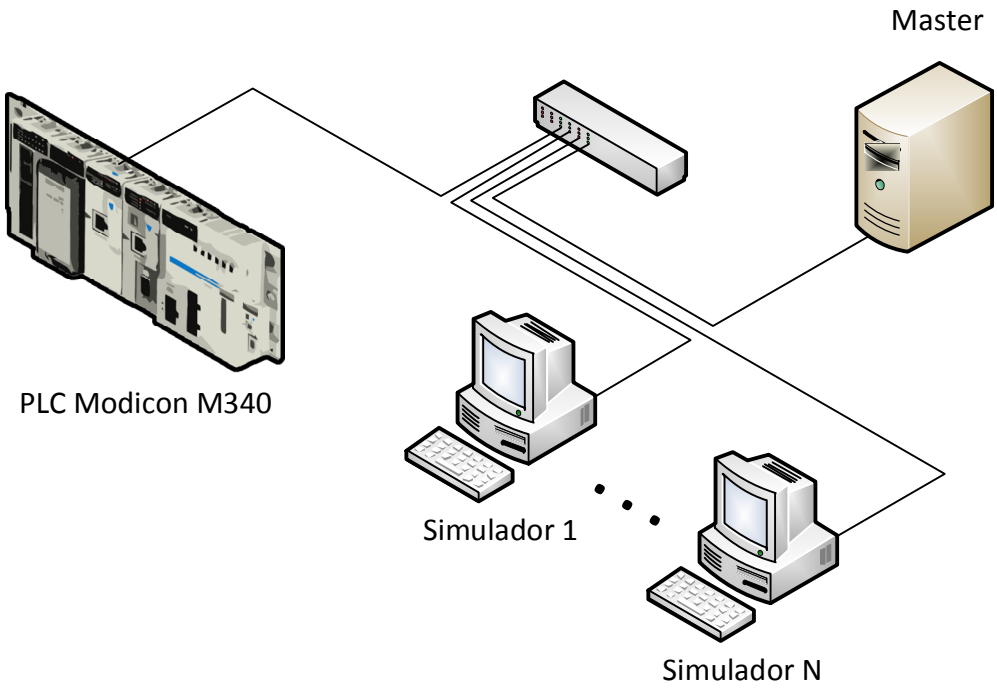

Figura 5.2 - Esquema da bancada de testes implementada

5.2.2 Trabalho desenvolvido

Na bancada de testes implementada, os controladores (reais e simulados) foram programados com lógica de forma a simular o controlo de uma garagem automóvel (exemplo fornecido com o pacote de *software* incluído no *kit* do M340). Na Figura 5.3 podemos observar a consola de operador (HMI) da garagem. Nesta, é possível visualizar o estado do sistema através de diagramas animados e efectuar algumas operações de controlo como a inicialização do sistema ou restabelecer manualmente o número de carros estacionados na garagem para zero. No PLC

real, as entradas podem ser manipuladas através de um comando integrado neste. No exemplo apresentado estas são: a leitura de um cartão de identificação e a passagem do automóvel pelos dois sensores (entrada e saída da garagem).

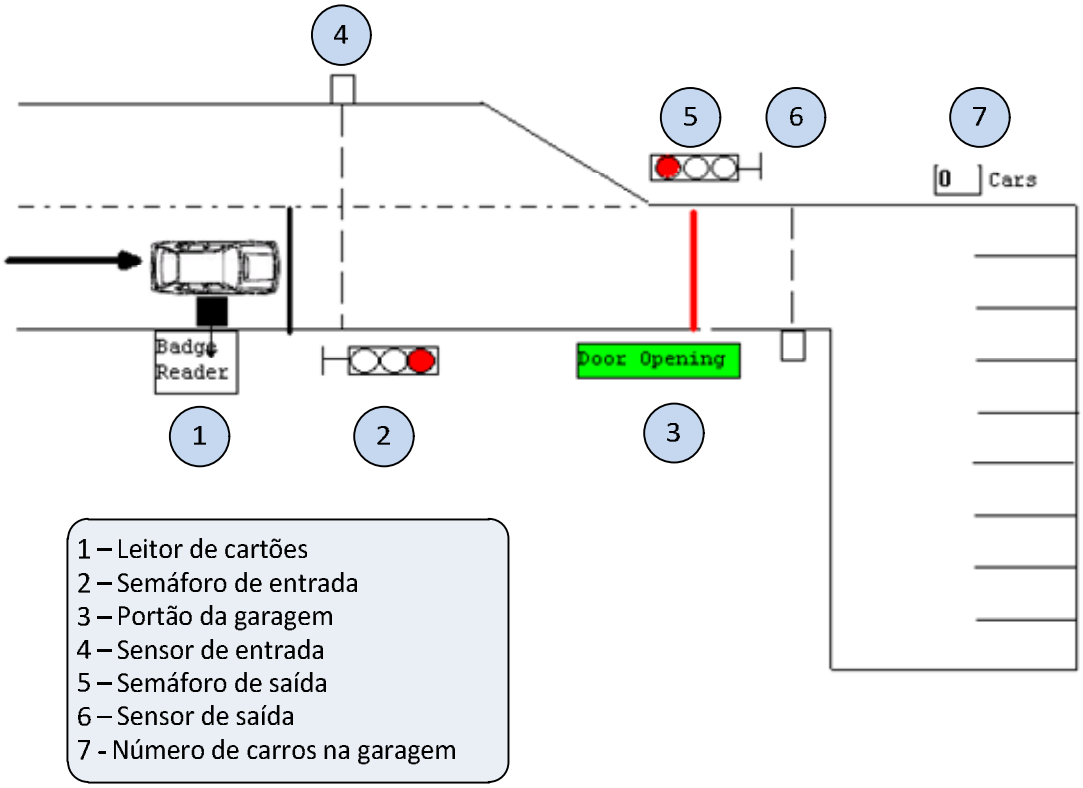

Figura 5.3 - Consola HMI do cenário de garagem automóvel

Os estados do sistema variam consoante as alterações nas entradas físicas. Apresenta-se como exemplo os principais estados (apenas os principais por motivo de simplicidade da explicação) para a entrada de um automóvel na garagem:

1. Introdução de cartão de entrada no leitor de cartões;
2. Inicialização de abertura da porta (duração de 10 segundos);
3. Com a porta aberta, o semáforo de entrada torna-se verde, permitindo a entrada do automóvel;
4. O automóvel atravessa o sensor de entrada;
5. O automóvel atravessa o sensor de saída;
6. Incremento do número de carros no interior da garagem;
7. Após a entrada do automóvel na garagem, é inicializado o fecho do portão da garagem (duração de 10 segundos).

A saída do automóvel possui algumas diferenças, já que não é necessária a apresentação do cartão de identificação. Novamente, são apresentados os passos principais do processo:

1. O automóvel atravessa o sensor de saída;
2. Inicializa-se a abertura do portão da garagem (duração de 10 segundos);
3. O semáforo torna-se verde, permitindo a saída do automóvel;
4. O automóvel atravessa o sensor de entrada;

5. O número de carros no interior da garagem decrementa;
6. É inicializado o fecho do portão da garagem (duração de 10 segundos).

Teste do cenário implementado no simulador

O cenário acima descrito foi também implementado no simulador de PLC. No entanto, devido à falta de entradas/saídas físicas foi necessário efectuar algumas alterações ao programa, mais concretamente nos diagramas de *Ladder*. Nessas alterações, as entradas físicas do PLC ficam ligadas a variáveis internas, podendo essas ser alteradas através da sua escrita via mensagens Modbus. Desse modo, a alteração de uma variável troca o respectivo sinal de entrada.

Para validar as alterações de estado no simulador, o desempenho deste foi comparado com o desempenho do PLC real, ambos a executar a versão alterada do cenário descrito no parágrafo anterior. Este objectivo foi atingido através da escrita de um programa na linguagem C#. Assim, através da livraria "Modbus TCP class" [ModbusTCPClass] para enviar comandos Modbus, foram implementados os métodos de escrita e leitura de variáveis para efectuar de forma automática as diferentes etapas da entrada de um automóvel na garagem. Cada variável exibida na aplicação representa um registo de memória do controlador (por exemplo número de carros estacionados na garagem). É possível observar o estado do sistema através da leitura das variáveis. Existem variáveis de leitura e variáveis de escrita, as primeiras representando as saídas do sistema (por exemplo o estado de um semáforo) e as de escrita representam as entradas do sistema (por exemplo a inserção de um cartão de identificação no leitor). Para alterar o estado do sistema, é analisado um conjunto de variáveis (dependendo da transição a efetuar) e é realizado as respectivas alterações nas variáveis de entrada, permitindo avançar para o estado seguinte do sistema. Um exemplo desta análise é a entrada de um automóvel após a apresentação do cartão de identificação. A passagem do automóvel pelo sensor de entrada só é possível após a abertura completa do portão e cancela de entrada.

O *layout* do programa pode ser observado na Figura 5.4.

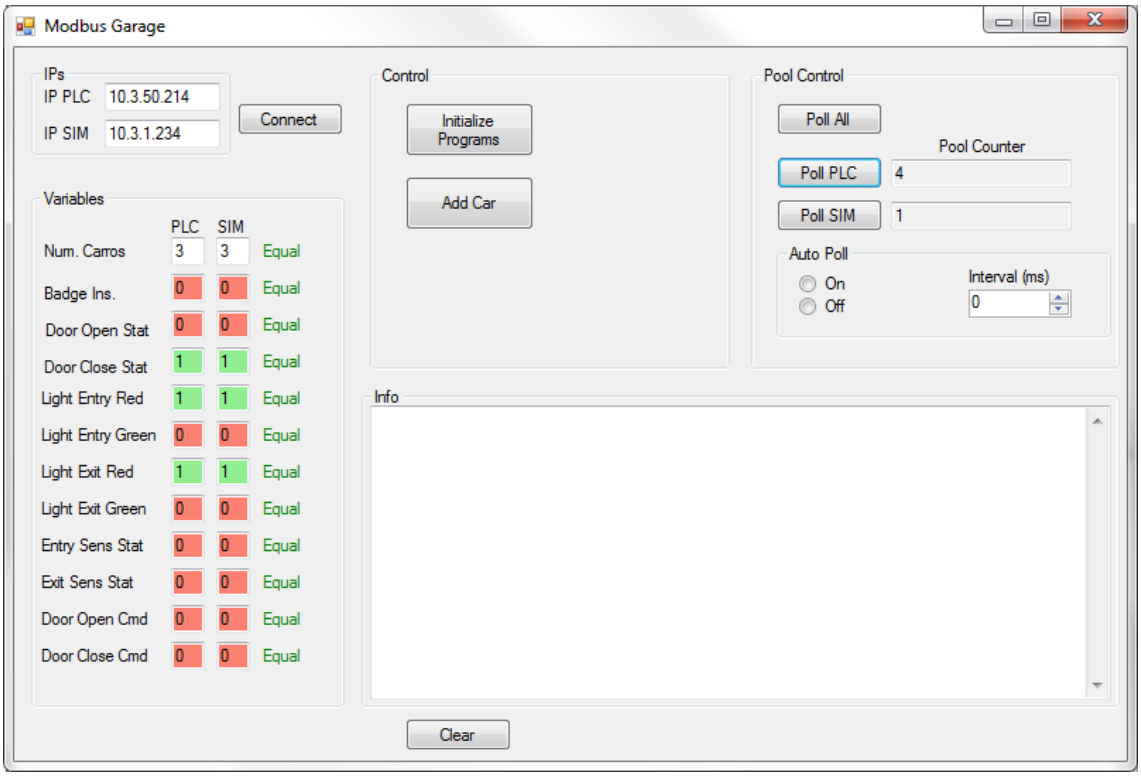

Figura 5.4 - Validação de estados de PLC real e simulado

Com o auxílio deste programa foi possível validar as transições de estado, sendo que ambos os PLCs (real e simulado) obtiveram os mesmos estados durante as diversas etapas de execução.

5.2.3 Conclusão

Em resumo, este capítulo apresentou a proposta de VPN integrada para uma futura interligação das bancadas de teste dos parceiros envolvidos. Com isto será possível recriar cenários mais extensos, e permitir também a sua utilização por parte de parceiros que não possuam bancadas próprias.

Nas últimas secções, foi apresentada a bancada de testes local e seus componentes. Eventualmente, é possível que esta seja expandida com novos componentes no decorrer do projecto CockpitCI consoante as necessidades do projecto.

6 Planeamento e Execução do Trabalho de Dissertação

Este capítulo apresenta o planeamento do trabalho de dissertação (Secção 6.1) e os principais constrangimentos que afectaram a sua execução (Secção 6.2). Na Secção 6.3 é apresentada a execução real do trabalho desenvolvido.

6.1 Plano inicial

O trabalho inicial possuía algumas linhas de orientação. Tendo o trabalho início antes do projecto CockpitCI seria normal este sofrer algumas alterações durante o seu desenvolvimento.

Originalmente existiam dois pontos importantes no plano de trabalhos:

1. Desenvolvimento de um estado de arte abrangente sobre a temática dos sistemas SCADA e sua segurança;
2. Desenho e implementação de uma bancada de testes que permitisse recriar cenários de controlo;

O primeiro ponto, como foi referido no inicio deste documento, tinha como princípio a aquisição de conhecimento relativo a estes sistemas. Isto devido a tratar-se de uma área nova no LCT, sem existir experiência prévia significativa por parte da equipa envolvida.

A bancada de testes tinha o objectivo de apoiar o laboratório no decorrer do projecto CockpitCI, auxiliando o desenvolvimento, teste e validação de soluções desenvolvidas. Neste ponto houve algumas dificuldades logísticas que afetaram o desenrolar do trabalho, relatadas na próxima secção.

6.2 Constrangimentos

No decorrer do trabalho surgiram constrangimentos que afetaram a sua progressão.

Em primeiro lugar, registou-se um desfasamento inesperado entre o arranque do projecto CockpitCI. Previsto para Outubro/Novembro de 2011 (já após sucessivos atrasos, por motivos administrativos ao nível da Comissão Europeia), o projeto acabou por arrancar apenas no final de Janeiro de 2012. Este aspecto condicionou o trabalho de dissertação, atrasando substancialmente a aquisição de equipamentos para a bancada de testes e a discussão com os restantes parceiros da arquitetura geral da plataforma CockpitCI (âmbito, prioridades ao nível funcional, cenários de referência, etc.).

Cumulativamente com o atraso no arranque do projeto, diversos obstáculos administrativos atrasaram ainda mais a aquisição do equipamento para a bancada de testes, atrasando não apenas a sua instalação mas também a execução de experiências de ambientação e avaliação de soluções e tecnologias. O equipamento SCADA começou a chegar apenas a 18 de Abril, afetando a profundidade e amplitude das tarefas associadas à bancada de testes.

Já no decurso do projeto, surgiram outros constrangimentos de menor impacto, relacionados com a lentidão do consórcio CockpitCI na discussão e validação das propostas apresentadas (arquitetura de detecção, interligação de bancadas de testes). Muito embora o *feedback* dos parceiros tenha sido positivo e ambas as propostas tenham sido aceites, a lentidão no

processo (apesar de tudo normal, neste tipo de projetos) atrasou a sua posterior execução. Só agora, por exemplo. Será possível começar a proceder á interligação das bancadas de teste.

6.3 Execução

Esta secção apresenta o desenvolvimento real das tarefas.

Nas Tabelas 6.1 e 6.2 apresenta-se a execução das tarefas durante o 1º período, enquanto as Tabelas 6.3, 6.4 e 6.5 fazem o mesmo para o 2º período.

Tabela 6.1 - Tarefas Primeiro período

ID	Nome da Tarefa	Início	Fim	Duração
1	Estudo SCADA	11-09-2011	16-10-2011	5
2	Estudo Segurança SCADA	16-10-2011	13-11-2011	4
3	Análise de ataques e incidentes na área	23-10-2011	13-11-2011	3
4	Análise de ferramentas de simulação Modbus	13-11-2011	27-11-2011	2
5	Estudo Projectos Relacionados	27-11-2011	18-12-2011	3
6	Elaboração da Dissertação	18-12-2011	22-01-2012	5

Tabela 6.2 - Duração de tarefas segundo período

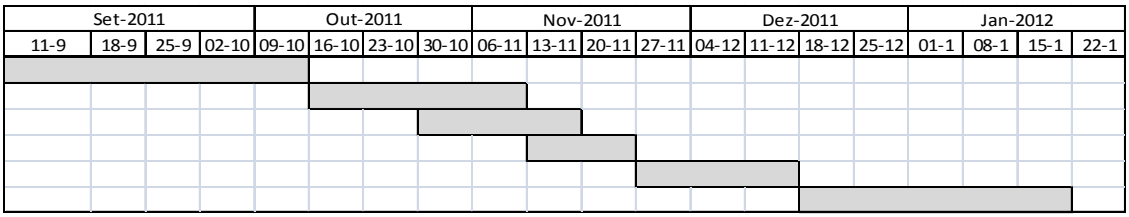

Tabela 6.3 - Tarefas segundo período

ID	Nome da Tarefa	Início	Fim	Duração
8	Proposta VPN	06-02-2012	28-05-2012	16
9	Arquitectura de Detecção	20-02-2012	16-04-2012	8
10	Desenvolvimento Testbed	13-02-2012	16-07-2012	22
11	Elaboração da Dissertação	23-07-2012	03-09-2012	6

Tabela 6.4 - Duração de tarefas segundo período (1)

ID	Fev-2012				Mar-2012				Abr-2012				
	06-2	13-2	20-2	27-2	05-3	12-3	19-3	26-3	02-4	09-4	16-4	23-4	30-4
8													
9													
10													
11													

Tabela 6.5 - Duração de tarefas segundo período (2)

ID	Mai-2012				Jun-2012				Jul-2012				Ago-2012				
	07-5	14-5	21-5	28-5	04-6	11-6	18-6	25-6	02-7	09-7	16-7	23-7	30-7	06-8	13-8	20-8	27-8
8																	
9																	
10																	
11																	

7 Conclusão

Esta dissertação abordou os sistemas de controlo industrial SCADA, mais concretamente a sua segurança. Estando integrada no projecto CockpitCI, apresenta um estado de arte nos sistemas, e descreve o trabalho realizado no âmbito do projecto. Relativamente ao último ponto, foram realizadas uma proposta de uma VPN para integração de bancadas de teste, e uma proposta para uma arquitectura de detecção de intrusões de genérica direccionada aos ambientes de controlo industrial. Por fim, foi também descrita a bancada de testes desenvolvida para apoio do trabalho do LCT a realizar no decorrer do projecto.

Nas secções seguintes são descritas as contribuições efectuadas no âmbito do projecto e é indicado o trabalho futuro a realizar.

7.1 Contribuições

No decorrer do trabalho foram realizadas várias contribuições relativamente ao projecto CockpitCI, nomeadamente: proposta de VPN para integração das bancadas de teste existentes no projecto, a proposta de uma arquitectura de detecção de intrusões genérica e uma contribuição em relatório do projecto (Deliverable 2.1 [CockpitCI D2.1]). A proposta de VPN e da arquitectura de detecção já foram descritos nas secções 5.1 e 4, respectivamente. Relativamente à última contribuição (D2.1), este é o primeiro relatório da *work package* 2000 do projecto realizado com a contribuição de vários parceiros do projecto e detalha um estado de arte nas ferramentas de segurança direccionados a ambientes SCADA. As contribuições realizadas relacionam-se com segurança nos sistemas SCADA, mais concretamente nos tópicos da utilização de firewalls, IDS e Honeypots/Honeynet para assegurar uma maior segurança aos sistemas.

As contribuições encontram-se anexadas ao presente documento nos Anexo C - , Anexo D - e Anexo E - .

7.2 Trabalho futuro

Relativamente ao futuro, o trabalho apresentado neste documento será continuado, mais concretamente na arquitectura de detecção. Esta será refinada no futuro em pontos mais objectivos sendo também implementada e realizada a sua validação. É também esperado uma evolução na bancada de testes consoante necessidades no desenrolar do projecto CockpitCI (por exemplo, de modo a integrar os componentes da arquitectura de detecção).

8 Bibliografia

[Aalto2008] Michael Aalto, Reuters. Disponível em: http://www.reuters.com/article/2008/07/30/idUS158811+30-Jul-2008+PRN20080730 (Último acesso em: Dez. 2011)

[AGA12] American Gas Association, *AGA Report no.12 - Cryptographic Protection of SCADA Communications Part 1: Background, Policies and Test Plan (AGA12, Part 1)*, 2006

[Aleks2011] Aleks, In *The Duqu Saga Continues: Enter Mr. B. Jason and TV's Dexter*. Securelist. Disponível em: http://www.securelist.com/en/blog/208193243/The_Duqu_Saga_Continues_Enter_Mr_B_Jason_and_TVs_Dexter (Último acesso em: Dez. 2011)

[Arduino] http://www.arduino.cc

[Auriemma2011] Luigi Auriemma, Luigi Auriemma Website. Disponível em: http://aluigi.altervista.org/ (Último acesso em: Nov 2011)

[Bailey2003] David Bailey and Edwin Wright, *Practical SCADA for Industry*. Great Britain: IDC Technologies, 2003.

[Björkman2010] Gunnar Björkman, "The VIKING Project - Towards more Secure SCADA Systems" , 2010.

[Byres2005] Byres, E.; Chauvin, B.; Karsch, J.; Hoffman, D.; Kube, N.; , "The special needs of SCADA/PCN firewalls: architectures and test results," Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on , vol.2, no., pp.8 pp.-884, 19-22 Sept. 2005
doi: 10.1109/ETFA.2005.1612765

[Clarke2004] Gordon Clarke, Deon Reynders, and Edwin Wrigth, *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. Great Britain: IDC Technologies, 2004

[CockpitCI D2.1] CockpitCI, Cyber-security on SCADA: risk prediction, analysis and reaction tools for Critical Infrastructures. 2012

[Crawford2006] Michael Crawford, In *Utility hack led to security overhaul*. Computer World. Disponível em: http://www.computerworld.com.au/article/151361/utility_hack_led_security_overhaul/ (Último acesso em: Dez. 2011)

[Creery2005] Creery, A.; Byres, E.J.; , "Industrial cybersecurity for power system and SCADA networks," Petroleum and Chemical Industry Conference, 2005. Industry Applications Society 52nd Annual , vol., no., pp. 303- 309, 12-14 Sept. 2005 doi: 10.1109/PCICON.2005.1524567

[Davis2006] Davis, C.M.; Tate, J.E.; Okhravi, H.; Grier, C.; Overbye, T.J.; Nicol, D.; , "SCADA Cyber Security Testbed Development," Power Symposium, 2006. NAPS 2006. 38th North American , vol., no., pp.483-488, 17-19 Sept. 2006 doi: 10.1109/NAPS.2006.359615

[Digital Bond] Digital Bond. Quickdraw SCADA IDS. Disponível em: http://www.digitalbond.com/tools/quickdraw/ (Último acesso: Nov. 2011)

[DNP] DNP Users Group, DNP - Distributed Network Protocol. Disponível em: http://www.dnp.org (Último acesso: Out. 2011)

[DOE2004] U.S. - Canada Power Systems Outage Task Force, "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recomendations," April, 2004

[Eriksson2007] G. Ericsson, “Toward a framework for managing information security for an electric power utility—CIGRÉ experiences,” IEEE Trans. Power Del., vol. 22, no. 3, pp. 1461–1469, Jul. 2007.

[ESCoRTS] ESCoRTS - Security of Control and Real Time Systems. Disponível em: http://www.escortsproject.eu/ (Último acesso em: Dez. 2011)

[ESCoRTS2008] ESCoRTS, "Survey of Stakeholders Needs - Deliverable 1.1", Dez. 2008.

[ESCoRTS2010] ESCoRTS, "R&D and standardization Road Map - Devilerable 3.2", Dez. 2010

[ESCoRTS2010a] ESCoRTS, "Survey of Existing Methods, Procedures and Guidelines - Deliverable 2.1", Jan. 2010

[ESCoRTS2010b] ESCoRTS, "Security metrics for cyber security assessment and testing - Deliverable 4.2", Ago. 2010.

[ESCoRTS2010c] ESCoRTS, "TAXONOMY of SECURITY SOLUTIONS for the SCADA Sector - Deliverable 2.2, Version 1.1" , Mar, 2010.

[ESCoRTS2011] ESCoRTS, "Requirements for future cyber security laboratories - Deliverable 4.3", Jan. 2011.

[ETH] Eidgenössische Technische Hochschule Zürich. Disponível em: http://www.ethz.ch/index_EN (Último acesso em: Dez. 2011)

[Falliere2011] L. O’Murchu N. Falliere. W32.Stuxnet dossier, Symantec White Paper, February 2011

[Fink2006] K. Raymond Fink, F. David Spencer, and A. Rita Wells, Lessons Learned From Cyber Security Assessments of SCADA and Energy Management Systems, Sept 2006.

[Fisher2011] Dennis Fisher, In *Researchers 'Convinced' Duqu Written By Same Group as Stuxnet*.ThreatPost.Disponível em: http://threatpost.com/en_us/blogs/researchers-convinced-duqu-written-same-group-stuxnet-111611 (Último acesso em: Dez. 2011)

[Fisher2011a] Dennis Fisher, In *Duqu Installer Contains Windows Kernel Zero Day*. Threadpost. Disponível em: http://threatpost.com/en_us/blogs/duqu-installer-contains-windows-kernel-zero-day-110111 (Último acesso em: Dez. 2011)

[Fovino2009] Igor Nai Fovino, A. Carcano, and Marcelo Masera, "Secure Modbus Protocol, implementation, tests and analysis," in *In Proceeding of the Third Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, Dartmouth College, Hanover, New Hampshire, USA, 22-25 March 2009.

[Gostev2011] Alex Gostev, *In The Mystery of Duqu*. ThreadPost. Disponível em: http://threatpost.com/en_us/blogs/mystery-duqu-102011 (Último acesso em: Dez. 2011)

[Gross2011] Michael Joseph Gross, In *A Declaration of Cyber-War*. Vanity Fair. [Online]. Disponível em: http://www.vanityfair.com/culture/features/2011/04/stuxnet-201104 (Último acesso em: Dez. 2011)

[IDC2009] *Practical Fieldbus, DeviceNet and Ethernet for Industry*.: IDC Technologies, 2009.

[IEEE1815-2010] IEEE, *IEEE Standard for Electric Power Systems Communications - Distributed Network Protocol (DNP3) - IEEE 1815-2010*, 2010

[IEEE802.3] IEEE 802.3 Ethernet Working Group. Disponível em: http://www.ieee802.org/3/ (Último acesso: Out. 2011)

[Igure2006] Vinay M. Igure, Sean A. Laughter, Ronald D. Williams, Security issues in SCADA networks, Computers & Security, Volume 25, Issue 7, October 2006, Pages 498-506, ISSN 0167-4048, 10.1016/j.cose.2006.03.001.

[ISA-99.00.01] American National Standard , ANSI/ISA-99.00.01-2007 - Security for Industrial Automation and Control Systems - Part 1: Terminology, Concepts, and Models, 29 Oct. 2007

[Kang2011] D.J. Kang, J.J. Lee, B.H. Kim, D. Hur, Proposal strategies of key management for data encryption in SCADA network of electric power systems, International Journal of Electrical Power & Energy Systems, Volume 33, Issue 9, November 2011, Pages 1521-1526, ISSN 0142-0615, 10.1016/j.ijepes.2009.03.004.

[Krutz2006] Ronald L. Krutz, Securing Scada Systems. USA: Wiley Publishing, Inc., 2006.

[KTH] KTH - Royal Institute of Technology. Disponível em: http://www.kth.se/?l=en_UK (Último acesso em: Dez. 2011)

[McClary1999] Daryl C. McClary, In *Olympic Pipe Line accident in Bellingham kills three youths on June 10, 1999*. HistoryLink. Disponível em: http://www.historylink.org/index.cfm?DisplayPage=output.cfm&File_Id=5468 (Último acesso em: Dez. 2011)

[McMillan2008] Robert McMillan, In *Computer Threat for Industrial Systems Now More Serious*. PCWorld. Disponível em: http://www.pcworld.com/businesscenter/article/150888/computer_threat_for_industrial_systems_now_more_serious.html (Último acesso em: Dez. 2011)

[Microsoft2008] Microsoft, Microsoft Security Bulletin MS08-067 - Critical. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/ms08-067 (Último acesso em: Dez. 2011)

[Microsoft2010] Microsoft, Microsoft Security Bulletin MS10-046 - Critical. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/MS10-046 (Último acesso em: Dez. 2011)

[Microsoft2010a] Microsoft, Microsoft Security Bulletin MS10-061 - Critical. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/MS10-061 (Último acesso em: Dez. 2011)

[Microsoft2010b] Microsoft, Microsoft Security Bulletin MS10-073 - Important. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/MS10-073 (Último acesso em: Dez. 2011)

[Microsoft2010c] Microsoft, Microsoft Security Bulletin MS10-092 - Important. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/MS10-092 (Último acesso em: Dez. 2011)

[Microsoft2011] Microsoft, Microsoft Security Bulletin MS11-087 - Critical. Disponível em: http://technet.microsoft.com/en-us/security/bulletin/ms11-087 (Último acesso em: Dez. 2011)

[Modbus] Modbus Organization, Disponível em: http://www.modbus.org/ (Último acesso: Dez. 2011)

[Modbus2006] Modbus-IDA, "Modbus Application Protocol Specification V1.1b", Dez. 2006

[ModbusTCPClass] http://www.codeproject.com/Articles/16260/Modbus-TCP-class

[ModbusTools] Modbus Tools. Disponível em: http://www.modbustools.com (Último acesso em: Dez. 2011)

[Nahorney2003] Benjamin Nahorney, Symantec. Disponível em: http://www.symantec.com/security_response/writeup.jsp?docid=2003-081909-2118-99 (Último acesso em: Dez 2011)

[Niland2003] Marty Niland, In *Computer Virus Brings Down Train Signals*.Information Week. Disponível em: http://www.informationweek.com/news/13100807 (Último Acesso em: Dez 2011)

[NISCC2005] NISCC, *NISCC good practice guide on firewall deployment for SCADA and process control networks.*: National Infrastructure Security Cordinaton Centre (NISCC), February 2005.

[Pauli2003] Pauli, S.; Krahl, B.; Leuschner, B.; , "A guide to specifying, justifying and installing substation monitoring and control systems," Petroleum and Chemical Industry Conference, 2003. Record of Conference Papers. IEEE Industry Applications Society 50th Annual , vol., no., pp. 71- 80, 15-17 Sept. 2003 doi: 10.1109/PCICON.2003.1242600

[PLCSimulator] Modbus PLC Simulator. Disponível em: http://www.plcsimulator.org (Último acesso em: Dez. 2011)

[Poulsen2003] Kevin Poulsen, In *Slammer worm crashed Ohio nuke plant network*. Security Focus. Disponível em: http://www.securityfocus.com/news/6767 (Último acesso em: Dez 2011)

[RISI] Risi - The Repository of Security Incidents. Disponível em: http://www.securityincidents.org/ (Último acesso em: Nov. 2011)

[Snort] Snort ®. Disponível em: http://www.snort.org (Último acesso: Nov. 2011)

[Ten2008] Chee-Wooi Ten; Chen-Ching Liu; Manimaran, G.; , "Vulnerability Assessment of Cybersecurity for SCADA Systems," Power Systems, IEEE Transactions on , vol.23, no.4, pp.1836-1846, Nov. 2008 doi: 10.1109/TPWRS.2008.2002298

[The Bellingham Herald] The Bellingham Herald. Disponível em: http://www.bellinghamherald.com/ (Último acesso em: Dez. 2011)

[UDM] University of Maryland. Disponível em: http://www.umd.edu/ (Último acesso em: Dez. 2011)

[Verba2008] Verba, J.; Milvich, M.; , "Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)," Technologies for Homeland Security, 2008 IEEE Conference on , vol., no., pp.469-473, 12-13 May 2008
doi: 10.1109/THS.2008.4534498

[VIKING] VIKING Project. Disponível em: http://www.vikingproject.eu/ (Último acesso em: Dez. 2011)

[VIKING2010] VIKING, "VIKING Cities Simulator (ViCiSi)". Vital Infrastructure, NetworKs, INformation and Control System ManaGement, 2010

[VIKING2010a] VIKING, "Cyber Security Modeling Language (CySeMoL)". Vital Infrastructure, NetworKs, INformation and Control Systems ManaGement, 2010

[VIKING2010b] VIKING, "VIKING Testbed Overview". 2010

[VIKING2010d] VIKING Project, "Report D2.3" , 2010.

[Wikipedia] Wikipedia - SCADA. Disponível em http://en.wikipedia.org/wiki/SCADA (Último acesso em: Dez. 2011)

[Wool2004] Wool, A.; , "A quantitative study of firewall configuration errors," Computer , vol.37, no.6, pp. 62- 67, June 2004
doi: 10.1109/MC.2004.2

[Zetter2011] Kim Zetter, In *Attack Code for SCADA Vulnerabilities Released Online*. Wired. Disponível em: http://www.wired.com/threatlevel/2011/03/scada-vulnerabilities/ (Último acesso em: Dez. 2011)

[Zetter2011a] Kim Zetter, In *How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History*. Wired. Disponível em: http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/1 (Último acesso em: Dez. 2011)

[Zhu2010] Bonnie Zhu and Shankar Sastry, SCADA-specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy In proceedings the First Workshop on Secure Control Systems (SCS'10), Stockholm, Sweden, 2010

[Zhu2011] Bonnie Zhu, Anthony Joseph, and Shankar Sastry, "A taxonomy of Cyber Attacks on SCADA Systems," , Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, 2011

Anexo A - Mensagens Modbus

Este anexo contém a estrutura de mensagens do protocolo de comunicação Modbus.

A.1 - Modbus RTU

Em relação à estrutura das mensagens, podemos observar os campos constituintes na Figura A-1. A mensagem é constituída por duas partes. O PDU (*Protocol Data Unit*) que compreende os campos *Function Code* e *Data*, e o ADU (*Application Data Unit*) que adiciona ao PDU os campos *Additional Address* e *Error Check*.

No primeiro campo (*Additional address)* encontra-se o endereço do *Slave* destinatário da mensagem. Este endereço mantém-se mesmo na resposta do *Slave* à *Master*, é este o modo dela conhecer o emissor da mensagem. Este mecanismo funciona bem porque não existe troca de mensagens entre *Slaves*.

O Segundo campo (*Function code*) mantém o registo da função enviada ao *Slave*. Este campo tem tamanho de um byte e pode receber valores entre 0 e 255. A *Master* pode enviar o campo com valores entre 1 e 127 (o valor 0 não pode ser utilizado). Os valores entre 128 e 255 são utilizados pelo *Slave* no caso de ocorrer algum erro na execução do comando. Uma resposta com o mesmo valor que o pedido, indica que o *Slave* executou o comando sem problemas. No caso de um erro, o *Slave* envia o campo com o bit mais significativo activo, informando a *Master* do erro.

O campo *Data* possui dados da mensagem. O seu tamanho é variável e na mensagem de pedido pode conter informações adicionais que o *Slave* necessite para completar o comando. A mensagem de resposta pode conter dois tipos de informação. Com a boa execução do comando, contém os dados requisitados pela *Master*, na eventualidade de um erro, contém informação adicional sobre o erro.

O último campo contém uma verificação de erros utilizando CRC-16 (*Cyclic Redundancy Check*). Esta verificação assegura que o equipamento destino não reage a mensagens que tenham sido danificadas durante a transmissão.

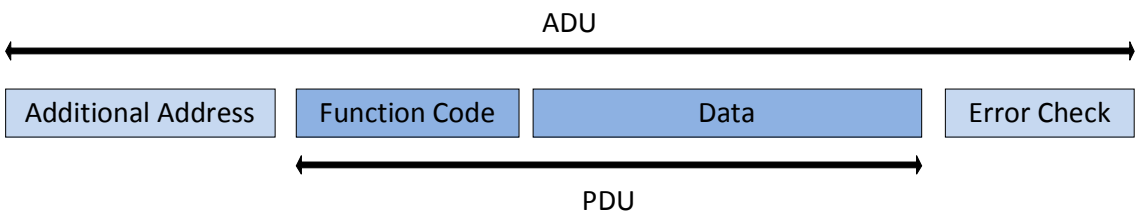

Figura A-1 - Estrutura de mensagens Modbus RTU. (Retirado de [Modbus2006])

A.2 - Modbus TCP/IP

As mensagens do protocolo Modbus TCP/IP seguem a estrutura base do Modbus RTU, mantendo o PDU idêntico. O campo *Addicional Address* e *Error Check* não são utilizados, e é adicionado um MBAP Header como é possível obsernar na figura Figura A-2.

O primeiro campo do MBAP Header (*Transaction Identifier*) tem o tamanho de dois bytes e identifica o pedido de um cliente. Como as respostas podem não vir na mesma sequência dos pedidos, a *Master* consegue associar os pedidos com as respostas.

O campo *Protocol Identifier* tem dois bytes de tamanho e é preenchido pela *Master* e tem sempre o valor 0000x para esta implementação.

O penúltimo campo do MBAP Header possui o número de bytes utilizados pelos campos seguintes. Este campo também tem dois bytes de tamanho.

O último campo é o *Unit Identifier* e é utilizado para endereçar equipamento da variante RTU do protocolo que possa existir na rede com a utilização de gateways.

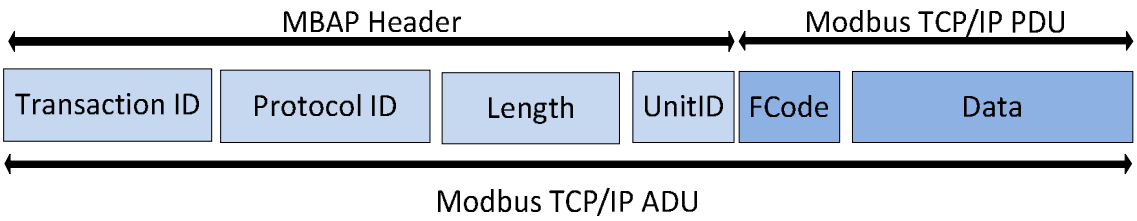

Figura A-2 - Estrutura de mensagens Modbus TCP. (Retirado de [Modbus2006])

Anexo B - Camadas do DNP3

Este anexo contém a descrição das camadas constituintes da pilha protocolar DNP3.

B.1 - Camada de Aplicação

A camada de nível mais alto é a de aplicação, estes são os maiores fragmentos da pilha protocolar do DNP3, tendo um tamanho máximo de 2048 bytes. Caso seja necessário transferir mais dados numa mensagem, esta camada é responsável por efectuar a sua divisão. Os APDU (*Application Protocol Data Unit*) são criados juntando os APCI (*Application Protocol Control Information*) e os ASDU (*Application Service Data Unit*) como está ilustrado na Figura 8.1. O primeiro bloco é o *header* de aplicação e o segundo são os dados a enviar. O *header* de aplicação é diferente para as mensagens de pedido e resposta. As primeiras são enviadas pela *Master*, e possuem campos de controlo e código de função a executar pelo *Slave*. A resposta possui um campo adicional (*Internal Indications*) que possui informação sobre alguns estados e erros num *Slave*.

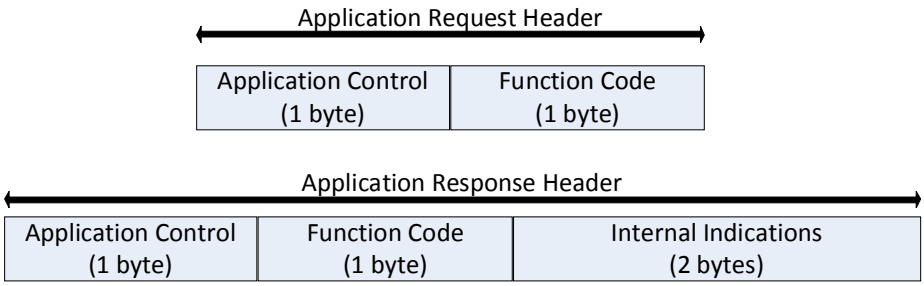

Figura 8.1 - Header de aplicação DNP3. (Retirado de [IEEE1815-2010])

Os campos de controlo podem ser observados na Figura 8.2:

- FIR - Activo quando é o primeiro fragmento de uma mensagem;
- FIN - Activo quando é o último fragmento de uma mensagem;
- CON - Quando activo, é necessário enviar uma mensagem a confirmar a recepção;
- UNS - Quando activo, indica que a mensagem é uma *Unsolicited Response*;
- SEQ - Número de sequência, utilizado para garantir a reconstrução dos vários fragmentos constituintes de uma mensagem;

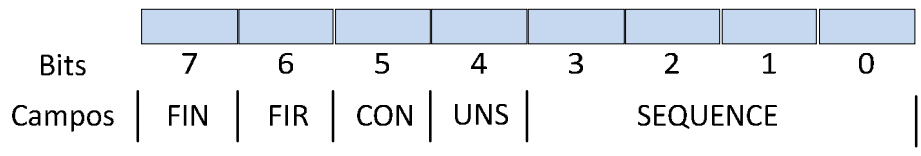

Figura 8.2 - Campos de controlo do header de aplicação do DNP3. (Retirado de [IEEE1815-2010])

O campo *Function Code* tem um byte de tamanho e indica qual a função executada. Os valores entre 0 e 128 são preenchidos pelo *Slave* e os valores entre 129 e 255 são preenchidos pela *Master*.

Em conjunto com o *header* temos os dados com tamanho variável. Não havendo um limite para o tamanho de dados numa mensagem de aplicação, o DNP3 fragmenta em vários segmentos com um máximo de 2048 bytes cada.

B.2 - Camada de Pseudo Transporte

A camada de transporte situa-se entre as camadas de aplicação e de ligação. A sua principal é fragmentar os pacotes da camada de aplicação para terem uma quantidade de dados que possa ser enviada pelos meios físicos *serial*. Nesta camada os dados têm um tamanho compreendido entre 1 e 249 bytes, ao qual é adicionado um *header* de um byte. Na Figura 8.3 podemos observar os campos que compõem o *header* da camada de pseudo-transporte:

- FIN - Quando activo indica que é o primeiro fragmento;
- FIR - Quando activo indica que é o último fragmento;
- Sequence - Número de sequência do fragmento para permitir a reconstrução pelo receptor;

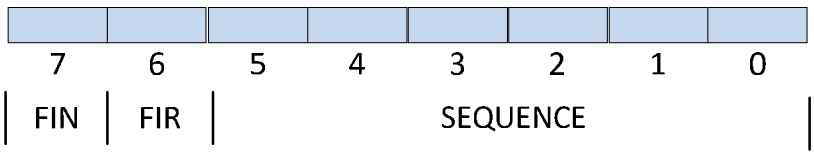

Figura 8.3 - Header da camada Pseudo-Transporte do DNP3. (Retirado de [IEEE1815-2010])

B.3 - Camada de Ligação

A camada de ligação está encarregue de gerir as variáveis relacionadas com a ligação lógica entre o emissor e o receptor. É também nesta camada que é adicionado a informação sobre o endereço da origem e destino. Esta encapsula os segmentos de transporte para a sua transmissão. Faz isto adicionando-lhe um *header* de 10 bytes, e adicionando um controlo de erros (CRC-16) por cada 16 bytes de dados. Na Figura 8.4 podemos observar os campos constituintes da camada de ligação:

- Start - É constituído por dois bytes (0564 hexadecimal) e define o início da frame;
- Length - Representa o tamanho do resto do segmento excluindo os bytes de controlo de erros;
- Control - O byte de controlo possui alguns campos internamente:
    - DIR (Direction Bit) - Bit de direcção, se o segmento tiver origem numa *Master* o bit é activo;
    - PRM (Primary Bit) - Quando o bit está activo indica que a frame é primária (inicial) ou secundária (resposta). Este bit é utilizado para interpretação do *Function Code* (FC) do bloco de controlo. Existem seis funções válidas para segmentos primários e cinco para frames secundários;
    - Frame Count Bits - Constituído por dois bits utilizados nas mensagens primárias. São utilizados para detectar segmentos perdidos ou repetidos. Quando o FCV está activo, o valor do FCB será trocado a cada mensagem confirmada;

        - FCB (Frame Count Bit);
        - FCV (Frame Count Valid Bit);
    - RES (Reserved) - Reservado, estará sempre com o valor zero;
    - DFC (Data Flow Control Bit) - Este bit é utilizado nas mensagens secundárias. Quando está activo significa que o envio de outra mensagem resultará num *buffer overflow* no *Slave*. A *Master* pára de enviar dados, mas envia pedidos de estado da ligação até receber uma mensagem onde o DFC se encontre igual a zero;
    - FC (Function Code) - Este campo representa vários códigos de funções possíveis de executar. O seu significado depende se o segmento é primário ou secundário. Com ele é possível confirmar recepções, reiniciar uma ligação, pedir o estado da ligação, entre outros[Clarke2004].
- Destination Address - Endereço de destino com um tamanho de dois bytes;
- Source Address - Endereço de origem com um tamanho máximo de dois bytes;
- CRC - Controlo de erros com um tamanho de dois bytes;
- User Data - Cada bloco tem 16 bytes de dados. O tamanho do último bloco pode variar entre 1 e 16. Se o segmento a enviar tiver o tamanho máximo, o último bloco terá um tamanho de 10 bytes;

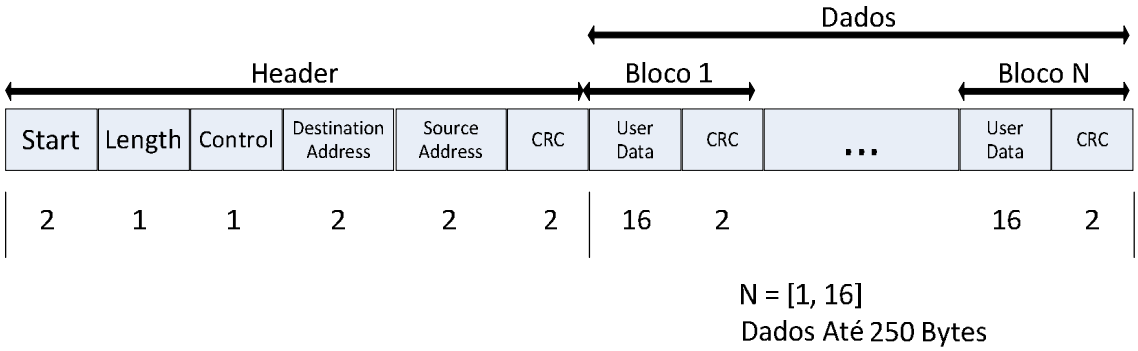

Figura 8.4 - Segmento da camada de ligação do DNP3. (Adaptado de [IEEE1815-2010])

B.4 - Camada Física

A camada física define entre outras, as características da interface física, as especificações eléctricas. Esta camada deve fornecer os seguintes serviços:

- Conexão;
- Desconexão;
- Envio;
- Recepção;
- Estado;

B.5 - DNP3 em redes TCP/IP

O funcionamento do DNP3 adaptado para redes TCP é igual ao descrito anteriormente nas camadas de aplicação, pseudo-transporte e ligação. A camada física é trocada pela pilha protocolar TCP/IP e é adicionada uma camada intermédia, *Connection Management* [IEEE1815-2010]. Esta camada faz a ligação entre as camadas do DNP3 e da pilha TCP/IP. Entre outras tarefas, estabelece as ligações TCP, efectua o envio ou recepção de pacotes UDP e de os enviar para as camadas superiores do DNP3.

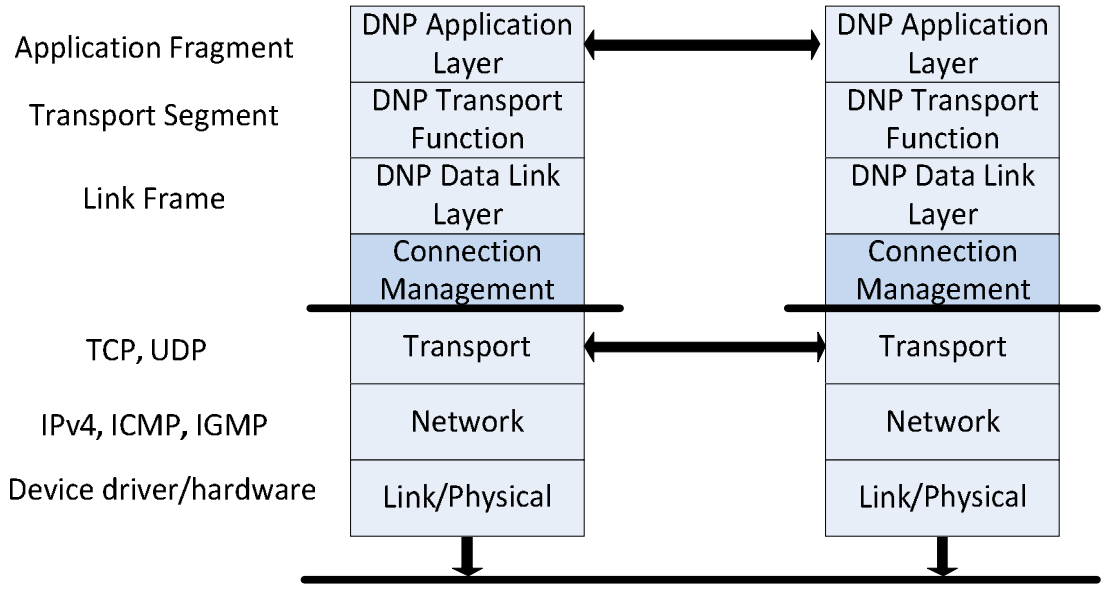

Figura 8.5 - Camadas DNP3 sobre TCP/IP. (Adaptado de [IEEE1815-2010])

Anexo C - Proposta de VPN

VPN L3 Bridging for SCADA Testbed Aggregation

CockpitCI Technical Note

Version	Authors	Affiliation	Changelog
1.2	Jorge Proença, Tiago Cruz	UC	Removed L2 support. Added technical specifications.
1.1	Jorge Proença	UC	Added L2TP. Minor corrections.
1.0	Jorge Proença	UC	Original release.

Description

This technical note documents a proposal for the implementation of a mechanism interconnecting several SCADA laboratory testbeds at the Layer-3level using Virtual Private Networking mechanisms.

Proposal

In order to fulfill the requirements of CockpitCI Work Packages, the responsible entities need to establish and validate the requirements of the stakeholders, namely IEC (Israel Electrical Corporation) which has offered to provide access to its SCADA testbed. Accordingly, it was agreed that the laboratory testbeds at Roma3, UC (University of Coimbra) and IEC would be safely interconnected in a common Layer-3 domain with the same IP addressing space, using secure VPN (Virtual Private Networking) technologies for LAN-to-LAN communication. Figure 1 illustrates the proposed topology:

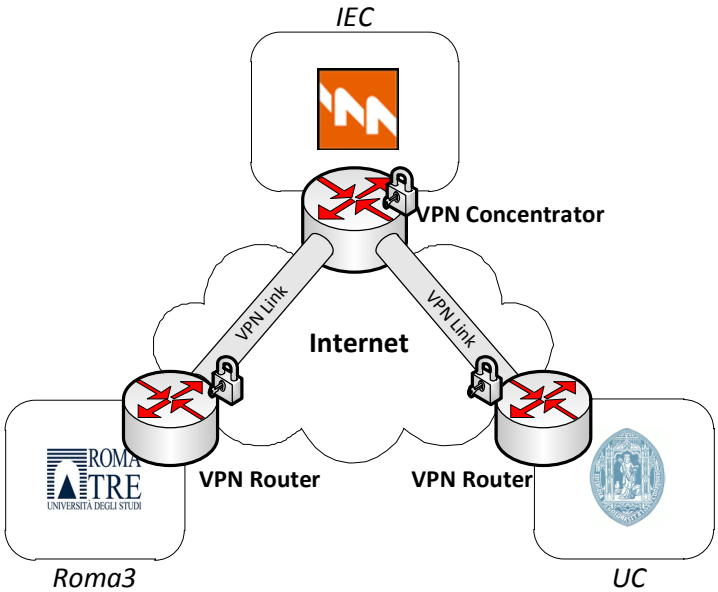

Figure 1: Logical VPN Topology for Testbed Aggregation

In this scenario, IPSEC (RFC4301) is the chosen protocol for establishing the proposed VPN topology, because of its advantage in terms of communication latency over other alternatives such as PPTP or OpenVPN, which commonly rely on userspace implementations (and therefore, more affected by factors such as context switching).Figure 2 illustrates this approach:

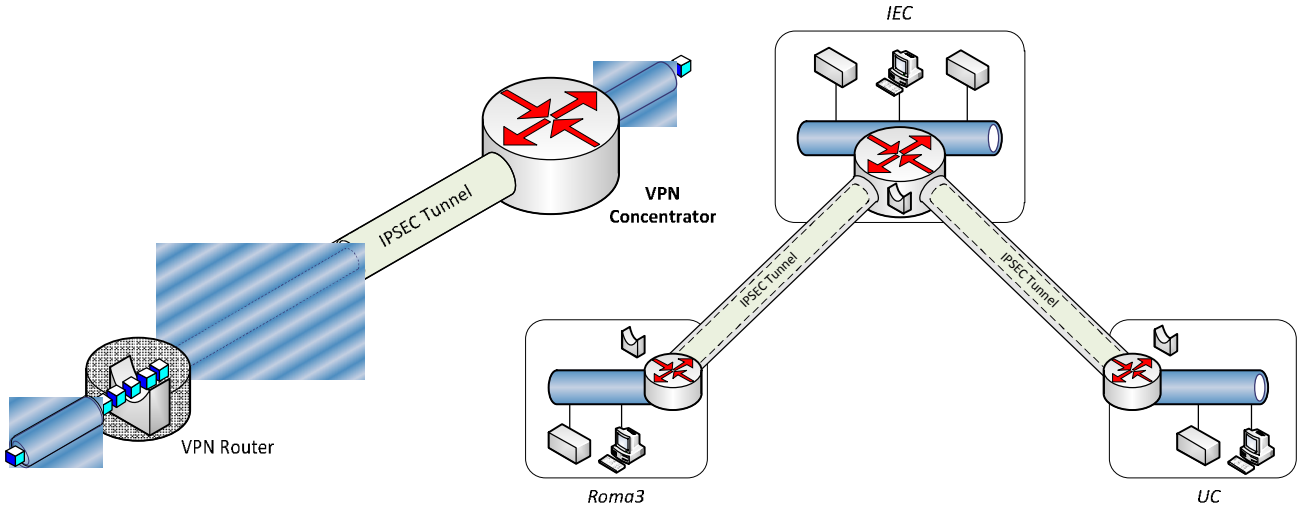

Figure 2: Layer-3 Bridging using VPN Links

A VPN concentrator, placed in the IEC testbed network will be responsible for bridging the tunnels from the VPN routers on UC andRoma3 with the physical network, therefore constituting a topology aggregating together all the networks behind each device (router or concentrator) in a single Layer-3 entity. The VPN concentrator can also be optionally configured to enable secure and authenticated permanent and *ad-hoc* accesses from isolated hosts or other networks.

As for the specific VPN configuration, Table 1 illustrates the proposed configuration. This configuration is fine-tuned to enable an adequate balance between security (use of 3DES encryption and reduced encryption key life) and latency overhead -however, it is possible that some fine-tuning will be required (i.e., timeouts) to enable adequate testbed component interoperability across VPN links. VPN router authentication is provided using IKE (Internet Key Exchange) Phase 1 with a preshared key which must be previously agreed and communicated over secure channels.

	Phase 1 - IKE
Authentication method	Preshared keys
Preshared Key	Established and delivered over secure channels
Key Exchange encryption	3DES
Data Integrity	SHA-1
Diffie-Hellman group (phase 1)	Group 2
Timer IKE phase 1	86400 seconds
Type	Main mode
	Phase 2 - IPSec
UDP encapsulation	Yes
Protocol	ESP
IPSEC - Encryption	3DES
Data Integrity - Authentication	SHA-1
Diffie-Hellman group	Group 2
Keylife	3600 seconds

Table 1: Proposed IPSEC Configuration Parameters

Both the VPN concentrator and the routers can be built using commodity hardware (a PC with two network interfaces) and software (a common Linux distribution has all the required means to implement this solution). In either case (VPN concentrator or VPN router), one physical network interface is to be connected to the physical network segment that is going to be bridged and the other one must be connected to the Internet, preferably with a public IP – as an alternative, if this interface is to be connected to a network that is behind a routing firewall, IPSEC NAT-T (NAT Traversal – RFC3715) must be enabled and configured accordingly.

System Specifications

System Specifications	
Operating System	Linux Fedora 16 (Verne) (3.1.0-7.fc16.i686.PAE)
Installed Packages	Openswan-2.6.37-1.fc16.i686 and dependencies
Firewall open ports	Protocol - ESP (n. 50) UDP port 500 (ike) UDP port 4500 in the case of NAT traversal
NICs	2 (One connects the VPN Concentrator and one to connect to the site's private network)
IPs	One Public IP address for all public interfaces. NAT-T (for NAT traversal) is also possible, but ads complexity to the setup.

IP Addressing plan

The specification assumes a private network in every site (see Figure 3). Each one will have a class C address (192.168.x.0/24), providing 253 hosts. If additional hosts are necessary, class B subnetting may be used through /23, /22, or /21 masks.

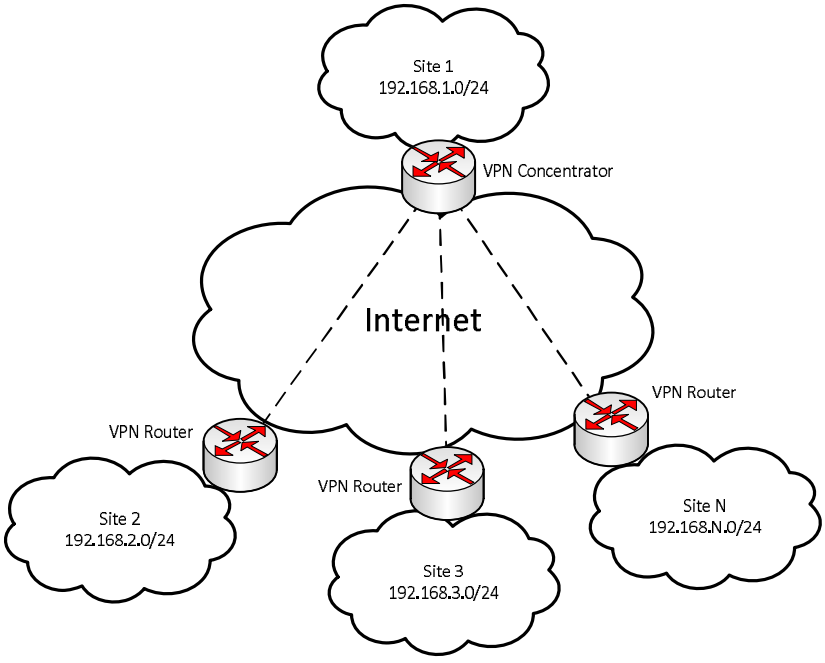

Figure 3: Addressing plan

IP Routes

Each VPN Router needs to have a route to all private network. Those routes will redirect the traffic to the gateway where the VPN Concentrator can be accessed.

IPSEC Configuration

config setup
    interfaces=%defaultroute
    klupsdebug=none
    plutodebug=none
    protostack=netkey

conn site1toConcentrator
    type=tunnel
    authby=secret
    #left -VPN concentrator
    left=’VPN concentrator public IP’
    leftsubnets={192.168.1.0/24 192.168.2.0/24 192.168.x.0/24} - every private network except the site’s private network

    #Right - the site’s VPN Router
    right-‘The site´s VPN router public IP’
    rightsubnets={192.168.1.0/24 192.168.2.0/24 192.168.x.0/24} - the site’s private network
    #cryptographic options
    phase2=esp
    phase2alg=3des-sha1;modp1024
    salifetime=3600s
    ikelifetime=24h
    forceencaps=yes
    auto=start

Anexo D - Proposta de arquitectura de detecção

Anexo E - Deliverable 2.1

Este anexo encontra-se no CD entregue com a dissertação devido ao grande número de páginas.