Mestrado em Engenharia Informática
Estágio
Relatório Final

Monitorização pró-activa de recursos em redes de média e grande dimensão

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Orientador DEI:
Professor DOUTOR Fernando Boavida

Orientador GSIIC:
Engenheiro Mário Bernardes

Data: 12 de Julho de 2012

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

Resumo

Os gestores de redes lidam a cada dia com diferentes tipos de ameaças . Estes dispõem de um conjunto de ferramentas, que lhes permitem manter a integridade e a qualidade do serviço prestado. No entanto, no contexto actual o objectivo passa por automatizar ou simplificar muitos dos sistemas construídos.

O trabalho realizado no âmbito deste estágio teve em conta essas ferramentas, em especial as ferramentas de monitorização e seus complementos, para automatização de reacções a parâmetros de rede e seus serviços.

Outro dos aspectos, foi a implementação de um sistema que permitisse responder automaticamente quem, quando e onde foi acedido determinado recurso. Por último, a implementação de um sistema que detecte automaticamente a alteração de configurações dos activos de rede, permitindo valida-las com os procedimentos.

Palavras-Chave

Monitorização, automatização, serviços de rede, recursos, controlo de acesso, contabilização, configurações

Agradecimentos

Gostaria de agradecer a todos os que estiveram comigo e me apoiaram nesta primeira etapa do estágio.

Ao meu orientador Professor Doutor Fernando Boavida pelo seu apoio e orientação.

A toda a equipa do GSIIC, em especial ao Engenheiro Mário Bernardes e ao Engenheiro Pedro Vapi pelo tempo despendido e ajuda na compreensão dos conceitos subjacentes ao tema do estágio.

À minha família, em especial à minha mãe e irmã, por todo o apoio ao logo de uma jornada que se aproxima do fim e por sempre acreditarem nas minhas capacidades.

Por fim, à minha namorada por estar sempre comigo, pelo seu carinho, compreensão e paciência.

Índice

Lista de Figuras

Lista de Acrónimos

AAA – Autenticaion, Authorization and Accounting

ACL – Access Control list

API – Application Programing Interface

BD – Bases de dados

CESNET – Czech Educational and Scientific Network

CGI – Common Gateway Interface

CIUC – Centro de Informática da Univercidade de Coimbra

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name System

GSIIC – Gestão de Sistemas e Infra-estruturas de Informação e Comunicação

HTTP – Hypertext Transfer Protocol

ICMP – Internet Control Message Protocol

IDOMOD – Icinga Data Out Module

IDO2DB- Icinga Data Out to Database

IMAP – Internet Message Access Protocol

ISP – Internet Service Provider

LDAP - Lightweight Directory Access Protocol

NAS – Network Access Server

NAT – Network Address Translation

NVT – Network Vulnerability Test

NDOMOD – Nagios Data Out Module

NDO2DB- Nagios Data Out to Database

POP – Post Office Protocol

PPP – Point-to-point Protocol

PPPoE – Point-to-point Protocol Over Ethernet

PPTP – Point-to-point Tunneling Protocol

RADIUS - Remote Authentication Dial In User Service

RCTS – Rede Ciência, Tecnologia e Sociedade

SMTP – Simple Mail Transfer Protocol

SNMP – Simple Network Management Protocol

SOAP – Simple Object Access Protocol

SSH – Secure Shell

TACAS+ - Terminal Access Controller Access-Control System Plus

TCP – Transmission Control Protocol

TIC – Tecnologias de informação e comunicação

TUI – Text User Interface

UC – Universidade de Coimbra

UDP – User Datagram Protocol

VPN – Virtual Private Network

XML – Extensible Markup Language

Capítulo 1
Introdução

O presente capítulo visa introduzir o tema de estágio desenvolvido no âmbito do Mestrado em Engenharia Informática.

1.1 Enquadramento

Numa época de grande massificação de estruturas e serviços de rede, é importante garantir aos seus utilizadores um elevado nível de segurança e disponibilidade da rede. Regidos por acordos de nível de serviço, os Administradores de Redes estão incumbidos de gerar novas formas de vigiar a rede, que permitam manter uma qualidade de serviço elevada. Isto só é possível, medindo a utilização, correcta ou incorrecta, dos recursos disponibilizados aos utilizadores.

Este Estágio pretende assim melhorar a forma como o Administrador de Redes é chamado a actuar, libertando-o para outras tarefas. Isto só é possível, automatizando algumas dessas tarefas, mais simples, através da definição de limites da rede.

Para que tal seja possível, o presente trabalho foi desenvolvido nas instalações do GSIIC, recorrendo à rede da Universidade de Coimbra. O GSIIC, antigo CIUC, é a entidade responsável pela gestão de toda a infra-estrutura e serviços de rede da UC.

1.2 Objectivos do Estágio

O objectivo do estágio, visa a implementação de procedimentos automatizados para responder a problemas na área da gestão de redes de média e grande dimensão, designadamente, monitorização de recursos, controlo de acesso de utilizadores e inventariação de equipamentos.

Assim, para cumprir com este objectivo, foi feito no primeiro semestre:

- estudo da infra-estrutura e serviços de rede existentes;
- estudo comparativo entre soluções de monitorização;
- estudo de protocolos e frameworks utilizados na área de controlo de acesso, bem como alguns sistemas;
- instalação e familiarização com ferramentas de suporte e análise de vulnerabilidades;

1.3 Contribuições

Baseado no conhecimento adquirido no primeiro semestre, e para cumprir com o objectivo proposto, construiram-se um conjunto de soluções:

- Instalação e configuração de um novo sistema de monitorização, com especial ênfase em complementos que automatizem ou facilitem alguns dos processos.

- Implementação de uma ferramenta de controlo de acesso que permita responder 'quem, quando e onde' foi acedido determinado recurso.
- Implementação de uma ferramenta de detecção de alteração de configurações de activos de rede.

1.4 Estrutura do Documento

Este documento encontra-se estruturado em 5 capítulos incluindo o presente.

No capítulo seguinte apresenta-se o estudo do estado da arte relativo aos objectivos propostos. São abordados os serviços e infra-estruturas de rede da Universidade de Coimbra (UC), sistemas de monitorização e inventário concorrentes, sistemas de registo e controlo de acesso de utilizadores, assim como os protocolos usados nos mesmos e ferramentas de suporte e análise de vulnerabilidade.

No capítulo três é dado a conhecer o planeamento ea descrição de como o trabalho foi desenvolvido.

No capítulo quatro são apresentados os testes realizados às aplicações desenvolvidas durante o estágio.

No capítulo cinco são tecidas as considerações finais em relação ao estágio seguido da bibliografia utilizada e anexos, onde se podem encontrar todos os documentos realizados durante o estágio.

Capítulo 2
Estado da Arte

Compreender toda a dinâmica por de trás da gestão de infra-estruturas e serviços de rede é crucial para o desenvolvimento do presente trabalho. Neste capítulo, é apresentado o estudo das várias soluções existentes no mercado que se enquadram com os objectivos do trabalho proposto. Desta forma, consegue-se ter uma ideia geral de como tudo funciona, o que vai permitir identificar a melhor ferramenta ou solução a usar.

Assim, neste capítulo haverá uma breve introdução às infraestruturas e serviços de rede, passando pelos sistemas de monitorização, sistemas de monitorização dos utilizadores (controlo de acesso) e terminando com as ferramentas de análise de vulnerabilidade como complemento às anteriores.

2.1 Infra-Estruturas e serviços de Rede

A rede sobre a qual será desenvolvido o trabalho é a rede da UC. Esta rede, como se pode ver pela Figura 1, é de uma dimensão considerável englobando três pólos principais. Cada pólo, é constituído por vários edifícios, departamentos, que têm equipamentos que permitem aceder à rede e seus serviços. A rede é constituída por vários equipamentos de rede como routers, switches, Hubs, firewalls, pontos de acesso ou servidores (que alojam serviços).

Na Figura 1 pode ver-se a estrutura genérica e simplificada da rede em causa.

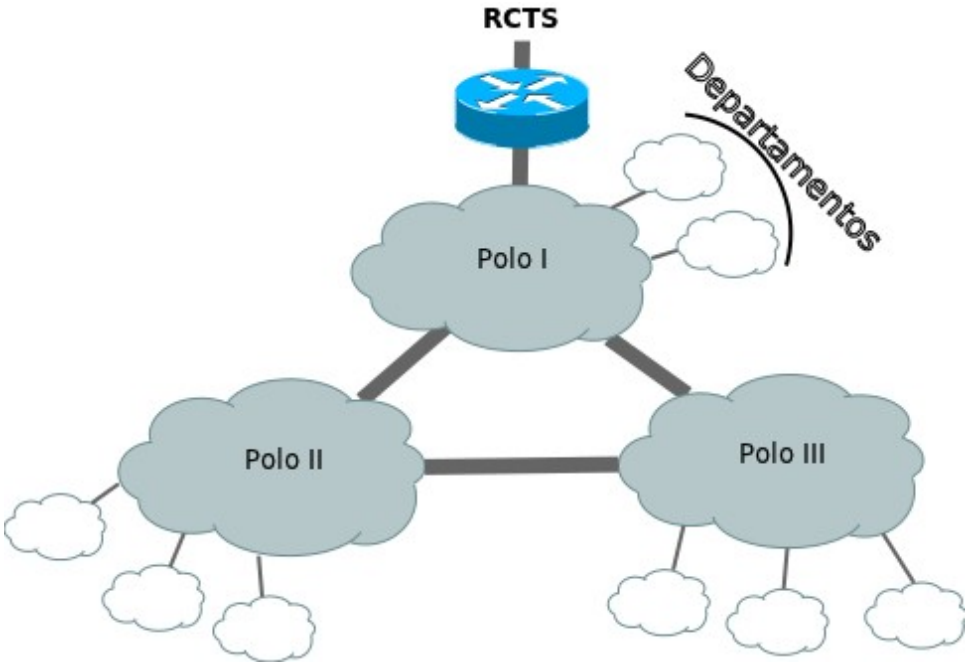

Figura 1: Infra-estrutura de rede genérica

Paralelamente à infraestrutura que permite interligar toda a UC, existe uma rede de serviços. Esta rede de serviços, exposta na Figura 2, alberga todos os serviços de rede a que os utilizadores vão ter acesso. Estes serviços são de vários tipos e englobam, por exemplo: serviço de gestão de acesso a directorias LDAP, DNS, DHCP, serviço de mail (com servidor de SMTP, POP e IMAP) ou servidores web.

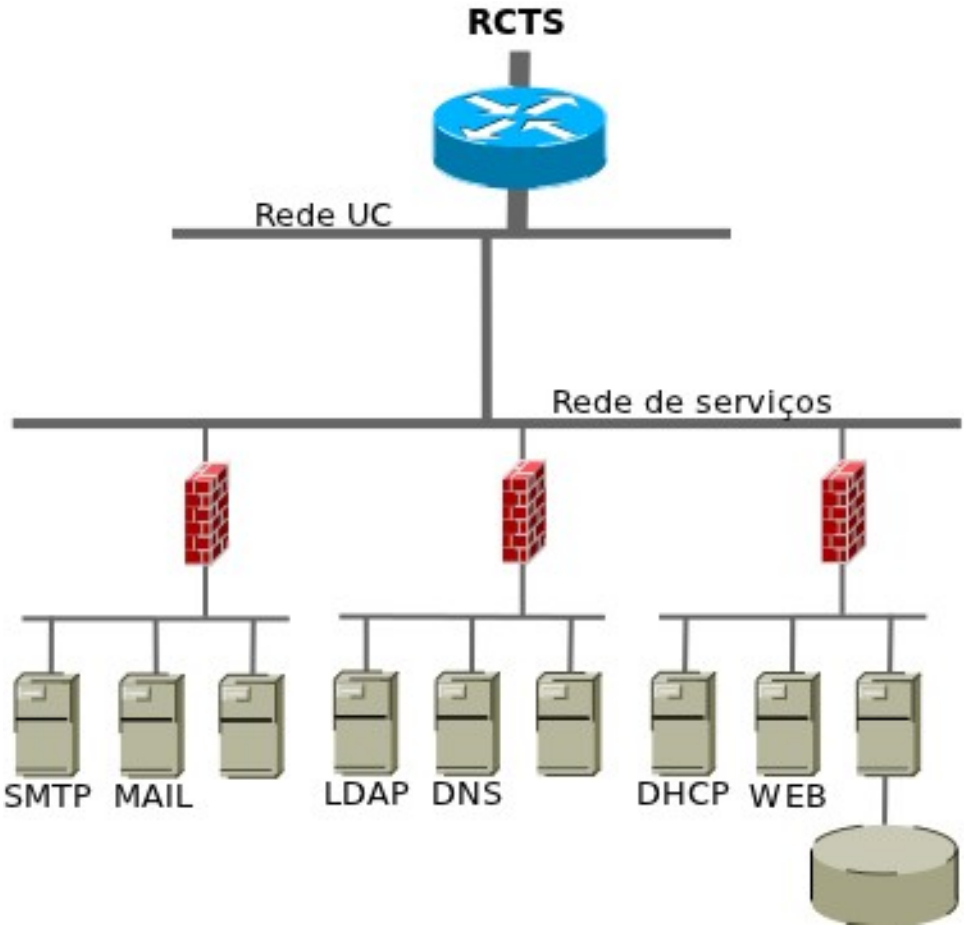

Figura 2: Infraestrutura de serviços genérica

De seguida, serão apresentadas um conjunto de ferramentas, de diversos tipos, que têm como principal propósito fornecer informações da rede e serviços disponibilizados.

2.2 Monitorização

A monitorização de uma rede pretende saber o estado e comportamento desta ao longo do tempo. Através da monitorização, o gestor consegue saber quase em estado real como se está a comportar a rede, tanto ao nível das máquinas como ao nível dos serviços, podendo assim avaliar e corrigir algum imprevisto.

2.2.1 Tipos de monitorização

São 2 os tipos de monitorização conhecidos que um gestor pode fazer uso para vigiar a rede:

- Monitorização passiva: faz uso de programas como sniffers que permitem avaliar o tráfego gerado pelas máquinas a monitorizar. Tem portanto a vantagem de lidar com tráfego real e não sobrecarregar a rede com tráfego extra. No entanto, não permite emular erros e tem problemas ao nível da privacidade e segurança dos dados.
- Monitorização activa: tem a capacidade de injectar pacotes na rede e assim obter as métricas de avaliação de desempenho. É gerado tráfego extra, mas não é significativo ao ponto de, bem configurado, afectar a performance da rede. Este permite testar o que se quer quando se quiser, emulando cenários que facilitarão a verificação da qualidade de serviço prestada ou os acordos de nível de serviço.

Para o presente trabalho, foi proposto ir mais além. Pegando nas vantagens da monitorização passiva e activa, a proposta seria ser pró-activo. Conseguir actuar sobre um problema, de forma automática, usando as métricas de avaliação da rede. Assim, consegue-se manter a rede com uma boa qualidade de serviço e retirar parte do trabalho do gestor de rede. Com o objectivo bem traçado, a fase seguinte foi o estudo de sistemas de monitorização que permitissem implementar um tipo de monitorização pró-activa.

2.2.2 Sistemas de Monitorização

Os sistemas de monitorização são uma parte importante na estrutura de monitorização de uma rede. Estes conseguem dar informação da rede em tempo real e bastante pormenorizada. Esta informação é conseguida através dos mais diversos protocolos como por exemplo o ICMP, SNMP ou TCP.

Através da flexibilidade da configuração de ferramentas com plugins, scripts ou complementos (addons), consegue-se saber a carga a que determinado recurso ou serviço de rede está sujeito, dando uma ideia, ao administrador de redes, de como deve actuar. Outro das grandes virtudes desta flexibilidade de configuração, é a possibilidade de corrigir algo que está, ou poderá vir a estar comprometido, com bastante rapidez. Esta pró-actividade, permite, como já foi dito acima, retirar carga aos administradores de redes, fazendo com que os acordos de nível de serviço sejam cumpridos mais facilmente.

De seguida vem uma breve descrição de duas soluções que servem o propósito da monitorização. Mais à frente, no capítulo 3, haverá uma comparação entre elas a fim de determinar qual a melhor para desenvolver o trabalho no segundo semestre: Nagios, já implementado no GSIIC, ou Icinga.

2.2.2.1 Nagios

O Nagios [1] é uma ferramenta, Open Source, de monitorização que "permite às organizações identificar e resolver problemas da infraestrutura da rede antes de se tornarem críticos para os processos de negócio".

A sua origem remonta a 1996. Nesse ano, *Ethan Galstad* criou uma aplicação, baseada em MS-DOS, para fazer ping aos servidores da Novell Netware. Em 1998, pegou no trabalho já feito e começou a criar um sistema que corresse em ambiente Linux. Assim, em 1999 nascia o NetSaint. Em 2002, devido a problemas com o nome NetSaint, o projecto é renomeado para Nagios ("Nagios Ain't Gonna Insist On Sainthood").

A arquitectura do Nagios é bastante simples (Figura 3). Há dois componentes: Nagios Core e Nagios Web CGIs. Estes comunicam entre si através de uma cache e um Pipe. A parte Web, Nagios Web CGIs, é uma camada de apresentação que recebe os dados de monitorização pela cache e envia comandos, via pipe, para um Ficheiro de comando. O Core, actua como um Daemon controlando todo o sistema de monitorização. Interpreta os ficheiros de configuração (ficheiros de texto simples) e executa testes de monitorização. Pode-se mesmo dizer que o Nagios é apenas um gestor de eventos, ao qual se podem adicionar funcionalidades. Mesmo a parte web, é considerada um extra ao sistema.

Os testes de monitorização, testes efectuados às máquinas e/ou serviços, são executados através de plugins. Estes plugins são escalonados pelo motor do Nagios, que os invoca de tempo a tempo, para efectuarem eles os testes às máquinas ou serviços. Os plugins só têm de obedecer a um padrão de output que o Nagios interpretará para definir o estado do sistema ou decidir as acções a realizar (enviar um mail aos gestores de redes, encurtar o tempo de verificação...). Podem ser scripts em Perl, python ou shell script, ou aplicações externas, que facilmente se integram. Assim, há a possibilidade de definir diferentes níveis de alertas ou formas de actuação, com o objectivo de mitigar ou acabar com um problema (a resolução pró-activa) na rede. O output fornece sempre um id (0 para Ok, 1 para Warning, 2 para Critical e 3 para Unknown) assim como uma descrição do estado do serviço.

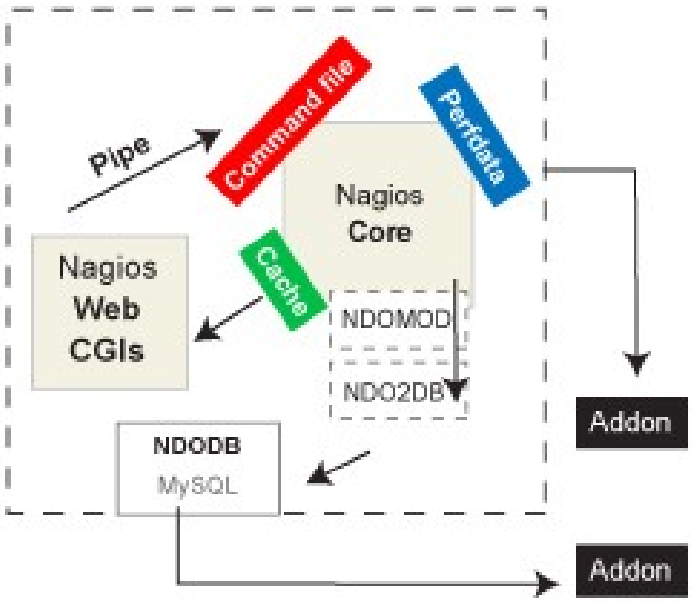

Figura 3: Arquitectura do Nagios [2]

Outro tipo de componente que estende as funcionalidades do Nagios são os complementos. Estes são pequenos programas que interpretam os dados de performance recolhidos com os plugins e os mostram de forma mais amigável (gráficos, relatórios, tabelas...). A comunicação com os complementos pode ser feita de duas formas: ou recebem os dados de performance directamente do Core ou acedem à Base de dados. A adição destes complementos, que estendem as capacidades do Nagios, torna a sua arquitectura mais

complexa. A comunicação com a base de dados, por exemplo, é feita através do complemento NDOUtils que é composto pelo módulo NDOMOD, que fica do lado do *Core*, e por um daemon, NDO2DB, que escuta num socket, do lado da base de dados *MySQL.*

O Nagios permite monitorizar diversos serviços de rede com recurso a vários protocolos de rede conhecidos (POP3, SMTP, IMAP, ICMP, SNMP, LDAP, HTTP, DHCP, SSH, TCP, UDP, etc). Assim, podem ser definidos serviços para monitorizar um servidor de mail, de directorias, de DNS, ou simplesmente os recursos usados pelo servidor, por exemplo. Permite também agregar um conjunto de serviços ou máquinas (no caso de termos vários servidores num serviço) formando assim um grupo de serviço ou grupo de máquinas. Esta monitorização pode ser feita remotamente, através de túneis SSH ou SSL.

A interface web, ainda que pareça desactualizada, torna-se bastante útil e intuitiva. Fornece diversas informações da rede (dos serviços e máquinas definidas) como por exemplo o histórico de problemas, o estado de uma máquina / serviço ou grupo, geração de relatórios ou visualização da estrutura de rede.

Existe também uma versão, Nagios XI, com suporte incluído. No entanto, esta é paga. Trata-se de uma versão mais completa com alguns addons já incluídos e uma interface redesenhada.

2.2.2.2 Icinga

O Icinga [2] é um projecto criado a partir de um fork do Nagios. Portanto, são produtos muito parecidos que, inclusivamente, podem partilhar as configurações, plugins e complementos. A sua primeira versão estável surgiu no final de 2009 e apresenta algumas diferenças relativamente ao Nagios, tanto a nível da arquitectura como ao nível da interface.

A arquitectura (Figura 4) do icinga é ligeiramente diferente da do Nagios. Primeiro, esta é constituída por três componentes: API, Core e Web. O Icinga Web, é um componente independente do Core. A comunicação entre ambos é feita através da componente API. Assim, o sistema Icinga, pode ter os vários componentes dispersos, funcionando como um sistema distribuído. Em segundo, a API,faz a ponte entre entre os complementos e o Core. Isto, permite aceder à informação da base de dados, ou directamente do Core, de forma facilitada. Uma extensão a este componente é o Icinga REST API. Este actua acima da API, estendendo as suas funcionalidades. O Icinga REST API permite a obtenção de dados via HTTP (pedidos GET ou POST), nos formatos xml ou json, admitindo a implementação de novas ferramentas independentes.

Por último, tal como o Nagios, existe ligação a um complemento que faz a gestão da comunicação com a base de dados: IDOUtils. Este complemento tem também um módulo, IDOMOD, e um daemon, IDO2DB, em tudo idênticos aos do Nagios. Para além disso, o Icinga suporta três motores de Base de dados: MySQL, Oracle e PostgreSQL. Por último, este complemento é requisito na instalação do Icinga Web, para que este consiga armazenar informação de monitorização.

O Icinga, tem praticamente as mesmas funcionalidades do Nagios. No entanto difere na forma como este interage com o gestor de redes, na arquitectura interna e em algumas novas funcionalidades.

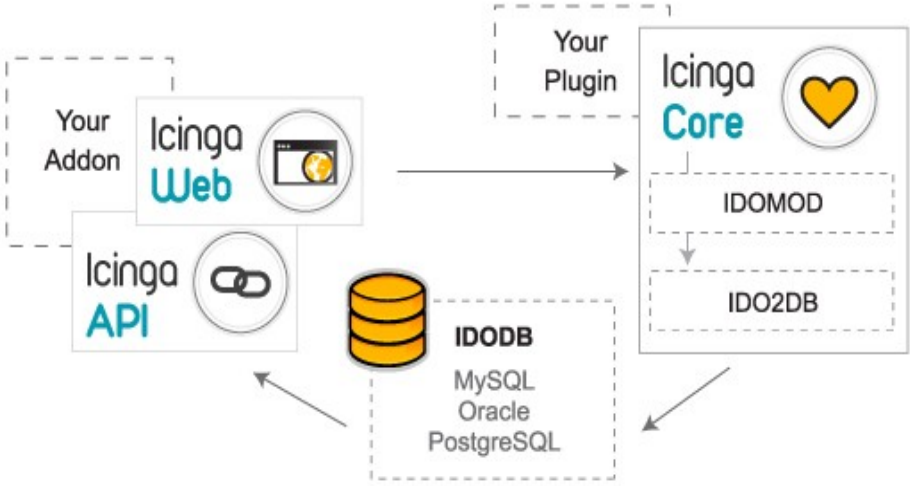

Figura 4: Arquitectura do Icinga ([2])

É possível monitorizar o sistema através de uma interface web clássica ou, através de uma nova interface web (Icinga Web). Esta nova interface, baseada em Web 2.0 e desenvolvida sobre a framework Agavi (http://www.agavi.org/), permite ter um conjunto de novas funcionalidades muito úteis. Uma delas, e talvez a mais interessante, é o Icinga Reporting (baseada no IDOUtils). Através da utilização desta biblioteca java, é possível criar relatórios do sistema e seus serviços. Esta biblioteca, trata-se de uma API SOAP que permite a comunicação do *Reporting cronk* (widget responsável pela geração de relatórios), do Icinga web, com um Jasper Server. Os templates estão predefinidos no jasper server e podem ser customizados, o que faz com que seja muito simples e personalizável a sua geração. Isto é muito importante, por exemplo, para os acordos de nível de serviço pois permite, ao gestor de redes, gerar relatórios sobre a situação da rede e seus serviços, mais detalhada e rapidamente. Outra funcionalidade, dependente do Icinga Web, é o Icinga mobile. Trata-se de uma aplicação, escrita em html 5 e que permite monitorizar o sistema através de Iphone ou de um smartphone Android.

Pela análise da Figura 5, pode-se constatar a simplicidade e usabilidade do componente Icinga web. Este é constituído por três partes distintas:

- em cima está a informação geral da rede isto é, o número total de máquinas e serviços e o estado dos mesmos;
- mais abaixo, do lado esquerdo, há um conjunto de botões, Cronks personalizáveis através de ficheiros xml, que dão acesso a informação mais detalhada das máquinas e serviços (esta informação é o resultado da análise dos dados recolhidos com os plugins), ou a uma perspectiva diferente da rede.
- À direita desta última, existe uma área onde se pode visualizar a informação dos Cronks e organizar diversas vistas por abas.

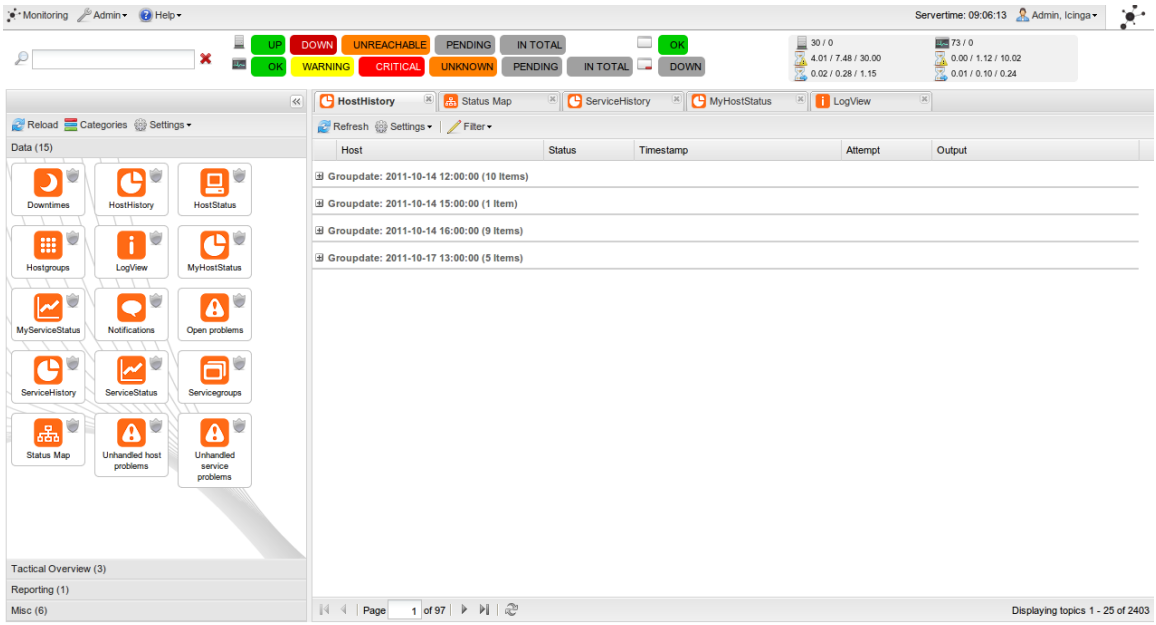

Figura 5: Painel principal do Icinga com o total de serviços e hosts em cima, a barra de Cronks à esquerda e o dashboard principal à direita

2.2.3 Monitorização pró-activa com Nagios e Icinga

A monitorização com qualquer uma destas ferramentas pode ser feita de três formas: activa, passiva ou pró-activa. A forma activa, é a mais comum. É iniciada pelo daemon e executada em intervalos de tempo regulares que estão definidos nos ficheiros de configuração. A forma passiva, é executada por aplicações exteriores que se encarregam de fazer os testes e enviar os resultados para o daemon. Ou seja, o sistema de monitorização não sabe quando receberá a informação. Só sabe que tem de verificar, periodicamente, uma queue que terá, ou não, nova informação. A forma pró-activa faz uso dos dados recolhidos com as duas anteriores para actuar automaticamente sobre um problema. Há dois tipos de monitorização pró-activa: curto prazo (reacção automática a um estado do sistema) e longo prazo (capacidade da organização para adaptar a sua estrutura de TIC).

Para implementar a pró-actividade de curto-prazo, estas duas ferramentas de monitorização têm uma funcionalidade denominada de Event handler. Este, pode ser activado nos ficheiros de configuração e permite avaliar e actuar sobre um problema com recurso aos dados de output dos plugins como o identificador numérico do estado, descrição do estado, número de vezes que a máquina / serviço foi verificado ou o tipo de estado em que se encontra. Este último pode tomar 2 valores:

- SOFT: quando uma máquina ou serviço não responde aos pedidos do sistema de monitorização (poderá estar em baixo, sem conectividade, etc), mas ainda não foi verificado o número de vezes definido nas configurações.
- HARD: quando uma máquina ou serviço não responde, aos pedidos do sistema de monitorização, e já foi verificado o número de vezes definido nas configurações; quando o estado, de uma máquina ou serviço, passa de um estado de erro para outro (Crítico para crítico); Se a máquina e o serviço, não responderem os dois.

No entanto, esta funcionalidade envolve grande conhecimento da rede e serviços alvo, nomeadamente as formas de actuação que poderão ser tomadas. É necessário ainda a instalação do módulo NRPE (Nagios remote plugin executor) que permite ao sistema de monitorização operar sobre uma máquina, actuando sobre os seus recursos locais (Figura 6). Com a instalação do NRPE, tem-se acesso a um plugin (check_nrpe) que permite comunicar com o daemon NRPE, que precisa de ser instalado na máquina remota. Funcionando como uma ponte, o sistema de monitorização vai conseguir verificar ou actuar sobre os seus recursos, serviços ou a própria máquina remota a fim de corrigir ou detectar um problema.

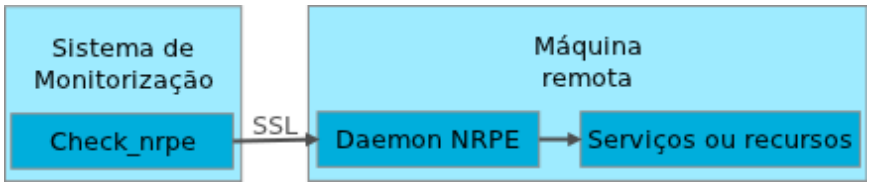

Figura 6: Arquitectura do NRPE

2.2.4 Posicionamento do sistema de monitorização

O posicionamento [3], na rede, do servidor de monitorização também é muito importante no desenvolvimento de um sistema de monitorização. O servidor, deve ser capaz de ver todas as máquinas de rede disponíveis, a fim de conseguir retirar a informação que necessita. Por exemplo, todas as máquinas (sejam firewalls ou outros dispositivos que filtrem os pacotes de rede) na rota entre o servidor de monitorização e o host monitorizado devem permitir o tráfego dos pacotes de monitorização (ICMP, SNMP por exemplo).

Se, por razões de segurança ou outras, não for possível obter essa visibilidade da rede, terá de ser implementado um sistema distribuído de monitorização, com mais servidores, em que só é autorizada a passagem de pacotes entre servidores de monitorização. Nesta configuração, haverá um único servidor central (primário) que recebe dados dos servidores remotos (secundários).No entanto, pode não ser possível, sequer, passar o tráfego entre servidores de monitorização. Nesse caso, a única solução será ter em cada zona da rede, um servidor de monitorização primário. Ambas as soluções podem ser vistas na Figura 7.

Outros aspectos a ter em conta são:

- Performance: relacionada com a distância entre host e servidor e a largura de banda utilizada. O sistema pode falhar ou ficar comprometido se o delay de recepção de resultados for muito grande.
- Disponibilidade: se houver uma configuração que suporte failover, mais do que um servidor, esta pode não ser um problema. No entanto, se houver apenas um servidor e este ou as ligações aos hosts estejam muito sobrecarregadas, toda a rede pode ficar comprometida ou apresentar resultados falsos. Isto, pode também retirar a atenção do gestor de redes, de problemas realmente importantes.
- Segurança: este coloca-se se houver servidores remotos. Principalmente, se estes precisarem de comunicar com o central via Internet. Neste caso, os dados estarão a atravessar uma ligação pública à qual os pacotes podem estar vulneráveis.

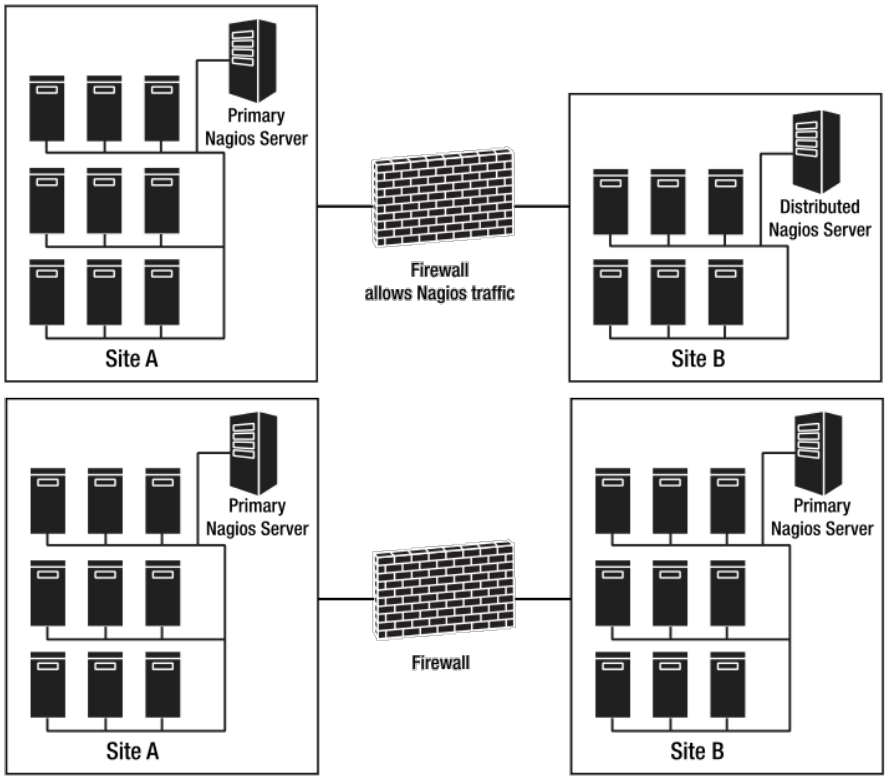

Figura 7: Monitorização distribuída e não distribuída, em cima e em baixo, respectivamente. ([3])

2.3 Sistemas de Registo e Controlo de Acesso a Utilizadores

Apesar de a rede ser constituída por máquinas, por de trás dessas máquinas há pessoas que controlam e/ou usam a rede. Para manter uma boa qualidade de serviço é necessário monitorizar estes últimos, os utilizadores, para perceber se estão a fazer bom uso dos recursos disponibilizados. Um estudo aprofundado deste tipo de sistemas , permite construir ferramentas que, apesar de não funcionarem de forma totalmente automática, vão ajudar a aliviar a carga de trabalho do gestor de redes.

O Controlo de Acesso é composto pelos processos Autenticação, Autorização e Contabilização, que definem o procedimento AAA (acrónimo em inglês). Este pode ser definido como a capacidade de permitir, negar ou limitar o acesso ao sistema e seus recursos a um determinado utilizador. Assim, a autenticação é responsável por identificar o individuo que acede ao sistema, a autorização responsável por determinar aquilo que o utilizador pode ou não fazer e a Contabilização responsável por contabilizar todos os recursos usados pelo utilizador autenticado (para efeitos de taxação, registo ou auditoria). É por isso um complemento à segurança da informação de uma instituição. Existem também diferentes formas e protocolos para implementar o controlo de acesso que serão detalhados de seguida.

2.3.1 Porquê Controlo de Acesso

A rede Universitária, em concreto a rede da Universidade de Coimbra, é composta por um conjunto de serviços. Uns mais críticos que outros mas que precisam de estar disponíveis 24/7. Para tal, é necessário cumprir um certo patamar de qualidade de serviço definido nos acordos de nível de serviço. Ora esta qualidade, diz respeito à disponibilidade de qualquer serviço da rede. Por exemplo: o acesso à conta de utilizador, ao seu email ou mesmo o acesso à rede Eduroam através de um Ponto de acesso Wireless ou rede cablada.

Estes problemas podem estar relacionados com equipamento que centralize a informação (HUBs, switches ou servidores de rede). Podem provocar lentidão na rede e seus serviços ou mesmo afectar a sua disponibilidade. Por outro lado, a má utilização, ou utilização abusiva dos serviços de rede, por parte dos seus utilizadores, podem também levar a este tipo de problemas, afectando assim quem realmente precisa da rede, para os fins para a qual foi idealizada.

No entanto, actualmente, já não pode haver preocupação só com os utilizadores habituais da rede ou mesmo com o acesso a ela a partir do seu interior. A quantidade de utilizadores e o seu tipo sofrem de um forte dinamismo e são bastante heterogéneos. A rede, tanto pode estar sobrecarregada no momento como, na hora seguinte, totalmente livre. Como também podemos ter acesso aos serviços da rede via wireless, rede cablada, a partir de casa do utilizador ou via VPN simulando o interior da rede.

Para além disso, estando integrados numa rede ainda maior, a rede Eduroam, caracterizada pela forte mobilidade dos seus utilizadores, o Administrador de redes terá de ter soluções que implementem AAA para poder controlar a autenticação, a autorização e a contabilização de recursos de cada utilizador.

Outros problemas relacionados com o controlo de acesso de utilizadores são as questões de ordem legal. Muitas vezes, as autoridades policiais solicitam informação sobre determinado IP da rede. No entanto, só conseguem fornecer um endereço IP e o horário de acesso a um recurso. Este endereço, é quase sempre público o que dificulta muito a acção do gestor de redes, caso o prevaricador esteja por de trás de um router ou ponto de acesso que camufle as suas acções. Por outro lado, as técnicas de NAT aplicadas nas grandes redes, ajudam de alguma forma a camuflar os utilizadores, o que torna difícil saber quem efectuou determinada acção com aquele endereço. Assim, torna-se crítico agrupar um conjunto de informações de acesso. O mais importante é obter os pares MAC Address – IP Address. No entanto, informação como tempo de acesso, início e fim de comunicação ou o endereço público da rede de acesso podem simplificar a descoberta do prevaricador e assim ter acesso a um conjunto de informação sobre o mesmo. Estes problemas, podem ser ultrapassados recorrendo a um conjunto de soluções e protocolos que serão detalhados de seguida.

2.3.2 Framework AAA

Este acrónimo diz respeito, como já foi dito em cima, a autenticação,autorização e contabilização. Dele fazem parte diversos protocolos que visam garantir a total integridade e segurança dos sistemas alvo. No presente trabalho, iremos debruçar-nos sobre a parte de contabilização de recursos.

Existem diversas formas de implementar sistemas que garantam a contabilização de recursos numa rede [4], [5]. A contabilização permite fazer a contagem do tráfego (numa

operadora ou instituição), saber o tempo de acesso de um cliente, saber quem acedeu a um recurso. Hoje em dia, com a massificação e facilidade do acesso à rede, bem como os recursos que lá podem ser disponibilizados, torna-se importante fazer uma correcta contabilização dos recursos utilizados pelo utilizador. Os aspectos legais são cada vez mais, e mais críticos, e são necessárias ferramentas que façam o seguimento do fluxo da informação.

2.3.2.1 Elementos arquitecturais do Sistema

Com o aparecimento de novos sistemas capazes de fazer autenticação, autorização e contabilização de recursos, surge a necessidade criar um modelo que interligue estes componentes independentemente da tecnologia usada. Os elementos que constituem o sistema AAA (Figura 8) estão descritos no RFC 2904 e são os seguintes [6–8]:

- Cliente: dispositivo (computador pessoal, smartphone, tablet, etc) que solicita o acesso à rede;
- *Policy Enforcement Point (PEP):* dispositivo de rede que permite efectuar o acesso à rede. Tipicamente é um sistema NAS (router, access point, etc) que recebe o pedido do cliente e o encaminha para o sistema de autenticação (ex.: servidor RADIUS).
- *Policy Decision Point (PDP):* dispositivo de rede responsável por autenticar e autorizar o acesso do cliente à rede. Tipicamente é um servidor RADIUS.
- Policy Information Point (PIP): repositório que contem a informação de autenticação (LDAP, base de dados, /etc/passwd) e que é consultado pelo PDP para que este tome as decisões.
- Sistema de Contabilização (Contabilização): repositório onde é guardada a informação de utilização de rede, por parte dos clientes.

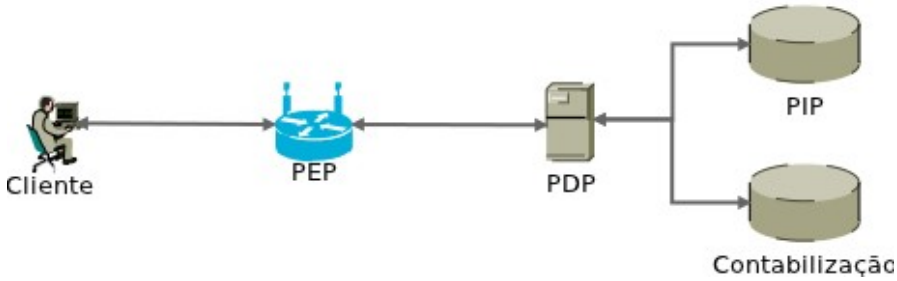

Figura 8: Arquitectura de sistema AAA e seus elementos estruturantes

2.3.2.2 Autenticação

Primeiramente, há a fase de identificação do utilizador. O utilizador fornece as suas credenciais (ID ou login, no nosso caso o email), permitindo ao sistema procurar pelos seus privilégios sobre os serviços de rede. De seguida, segue-se o processo de autenticação propriamente dito onde é validada a identificação do utilizador juntamente com uma password.

2.3.2.3 Autorização

Depois da autenticação, vem a autorização de acesso aos serviços determinados na fase de identificação. Portanto, a autorização está dependente da autenticação para manter actualizadas as Listas de controlo de acesso (ACL).

Nesta fase define-se a que é que o utilizador tem acesso, quando e como. Um utilizador autenticado, utilizando uma determinada identidade, pode pedir acesso a um recurso ou aplicação sob uma outra identidade (authorization identity). Também uma aplicação ou serviço pode actuar em nome do utilizador (impersonation). Útil num sistema de cliente servidor onde temos um servidor a correr com uma conta de servidor e a aceder a recursos sob o nome do utilizador.

2.3.2.4 Contabilização (Accounting)

Por fim, e para que se tenha total controlo sobre os serviços (garantir a sua qualidade), à que medir os recursos que um utilizador consome durante durante a sessão activa. Isto inclui o tempo de acesso ao sistema ou o volume de tráfego gerado (enviado e recebido). Quando um utilizador tenta aceder a um recurso, quer esteja autorizado ou não, é iniciada uma auditoria que fornecerá o histórico de dados e quando e como acedeu aos recursos ou como violou o sistema de autorização ou as políticas de autenticação.

2.3.2.5 Quem, quando e onde?

Esta é a pergunta a que a framework AAA, em particular a contabilização, permite dar resposta. Quem acedeu a determinado recurso da rede (que utilizador? Qual era o IP?)? Quando acedeu a esse mesmo recurso (hora, dia...)? Em que local da rede estava esse utilizador? Isto pressupõe, por parte do Gestor de redes, um bom conhecimento de toda a infraestrutura de rede a monitorizar, a fim de montar um bom sistema de monitorização.

Com a massificação da rede, como dito anteriormente, a resposta a esta pergunta, toma uma dimensão ainda maior em termos legais. A facilidade com que se partilha um recurso ou quebram alguns sistemas de segurança gera diversos problemas de ordem legal. Por exemplo, a partilha de ficheiros (audio, vídeo, imagem...), protegidos com direitos de autor, nas denominadas redes P2P ou mesmo através de uma simples página web, acarreta problemas de ordem legal (violação dos direitos de autor). O seguimento dos passos de um utilizador permitem ao gestor de redes, responder com exactidão nestas situações.

2.3.3 RADIUS

RADIUS [6], [7], [9],Remote Authentication Dial In User Service, é um protocolo de rede que implementa AAA centralizado, sobre UDP como protocolo de transporte, nos portos 1812 (autenticação e autorização) e 1813 (contabilização). Os procedimentos gerais estão definidos nos RFCs 2865 e 2866 (Accounting [6], [7], [10]). É usado para autenticar utilizadores ou máquinas na rede, autorizar esses utilizadores ou máquinas a aceder a serviços da rede e contabilizar (Accounting) o uso desses serviços.

Este protocolo, actua sobre uma arquitectura cliente-servidor, em que o cliente é responsável por fazer pedidos ao servidor a fim de autenticar, autorizar e contabilizar um

utilizador.

Como se pode ver pela Figura 9, o cliente (um NAS) é tipicamente um access point, router ou modem que aceita autenticação através de PPP, PPPoE, PPTP ou telnet. Depois de receber a informação do utilizador, envia um pacote *Access-Request* para o servidor RADIUS que trata da autenticação e autorização do acesso a serviços da rede. Se tudo correr bem ou seja, se o servidor RADIUS reconhecer a chave partilhada com o cliente NAS e os dados do utilizador, este responde com um pacote *Access-Response*. Caso não haja resposta por parte do servidor, ao fim de um intervalo de tempo predefinido, o NAS pode repetir o envio do *Access-Request* ou mesmo, enviar o pedido a um servidor RADIUS secundário (esta regra aplica-se a todos os pacotes de Request).

De seguida, o NAS envia um pacote *Acct-Request* (com o registo *acct_status_type=start* que faz com que o servidor RADIUS preencha alguns dos campos do registo de Accounting).

Por último, quando o utilizador fechar a sessão, é enviado um pacote para o NAS a pedir a sua desconexão. Por sua vez, o NAS, envia um *Acct-Request*, desta vez com *acct_status_type=stop*, para *o* servidor RADIUS que, tem agora toda a informação de que necessita para concluir o registo de Accounting. Neste pacote de Request, para além dos registos já enviados com o *acct_status_type=start*, vão:

- Acct-Session_Time: indica o tempo de ligação do utilizador em segundos;
- Acct-Input-Octets e Acct-Output-Octets: guarda o número de bytes, enviados e recebidos, durante a sessão.
- Acct-Terminate-Cause: indica o fim da sessão e o porquê.

Figura 9: Autenticação com RADIUS

2.3.4 Implementações

De seguida, apresentam-se algumas soluções que pretendem dar resposta aos problemas apresentados atrás e à pergunta de *quem, quando e onde*. Livres ou proprietárias, simples ou complexas, com ou sem aplicações, a correr, do lado do cliente. Todas elas tentam dar resposta à pergunta anterior.

2.3.4.1 Eduroam Accounting

O Eduroam Accounting [11] é uma solução de Contabilização, implementada por Daniel Studený que funcionou, durante quase dois anos, na CESNET. O seu funcionamento é simples pois a informação de acesso à rede vem em bruto, através do protocolo RADIUS, para um servidor RADIUS configurado e é armazenada para mais tarde ser acedida. No entanto, apresenta algumas desvantagens, pois não permite responder com exactidão ao problema do controlo de acesso.

Os pontos de acesso não conhecem o IP da máquina que se ligou (o protocolo 802.1x faz a autenticação antes de atribuir o IP, usando a camada de dados do modelo OSI). Assim, só tendo conhecimento dos endereços do hardware (MAC Address), o IP terá de vir de outras fontes (ficheiros de leases DHCP). Outro problema derivado do anterior, é a necessidade de pessoal especializado de agregar o IP aos dados de contabilização. Portanto é necessária uma aplicação onde os operadores possam encontrar os proprietários da máquina comprometida usando apenas o tempo de acesso do incidente e o IP da máquina que acedeu, para entregar o caso aos administradores mais capazes. Com os requisitos do sistema bastante precisos, o sistema terá de processar a informação a fim de gerar os registos de contabilização, start e stop, agrupá-los e acrescentar o endereço IP que virá do servidor DHCP.

Na Figura 10 está desenhada a arquitectura do sistema. São usados dois servidores RADIUS: um para autenticação (*Radius Server* da imagem) e outro para contabilização (*Accounting Radius*). O primeiro servidor de RADIUS, foi configurado para suportar a passagem de registos de contabilização. Logo, o ficheiro de configuração do RADIUS, irá ser alterado para suportar a passagem dos pacotes *Acconting-request* e, passar esses registos para o segundo servidor. Este segundo servidor, armazena a informação numa base de dados com os seguintes campos, vindos do RADIUS: User-Name, Acct-Status-type, Timestamp, Acct-Termnation-Cause, Calling-Station-id (onde vai o MAC Address do computador remoto), NAS-IP-Address e Acct-Session-Id.

Depois desta informação armazenada, há que adicionar o IP. Isto é feito com recurso ao servidor de DHCP e ao *notifyd* Daemon. O servidor de DHCP, assim que atribui um novo endereço, informa o servidor de DNS que, por sua vez, avisa o notifyd. Este Daemon implementa uma pequena parte do servidor DNS, que escuta no porto 53 por mensagens *DNS NOTIFY.* Após isto, é enviada uma resposta ao DNS que através de DNS Lookups procura registos do RADIUS sem IP e o atribui. Havia outras formas de implementação (criar um patch para o servidor de DHCP, por exemplo), no entanto, o autor, por razões de facilidade de upgrade do sistema, escolheu a alternativa acima referida.

Para visualização dos dados, existe uma interface, bastante simples, com uma tabela que integra toda esta informação.

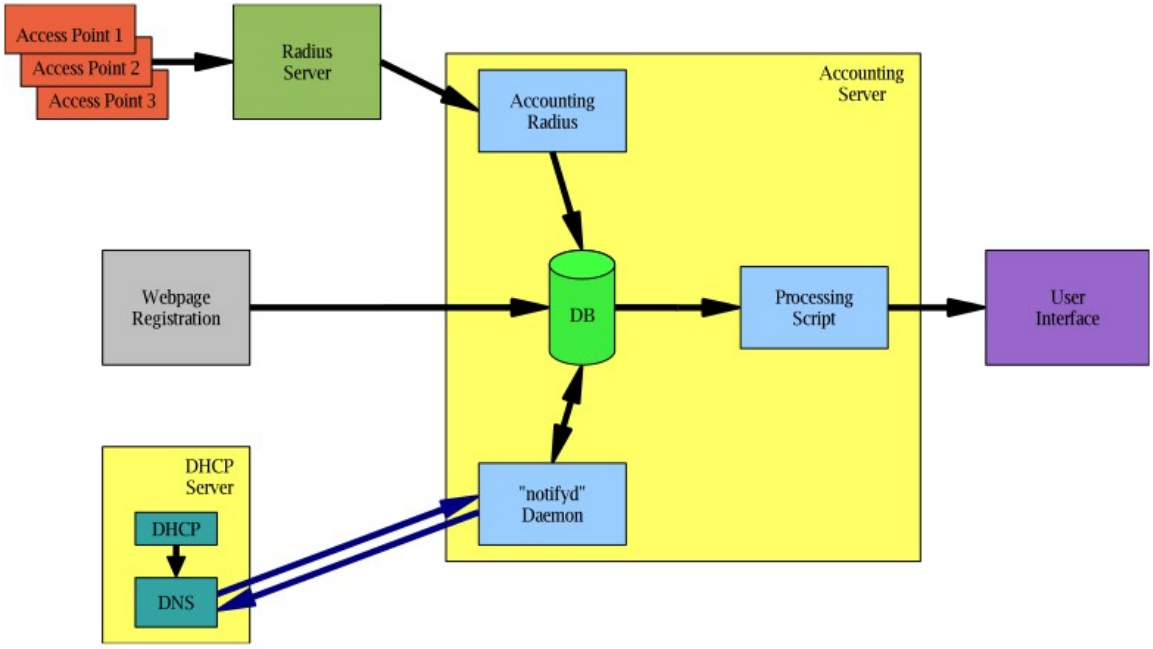

Figura 10: Fluxo de dados do Eduroam Accounting ([15])

2.3.4.2 Solução da UC

A solução implementada na UC (Figura 11) é em tudo idêntica. No entanto, há dois pontos distintos: 2 servidores de RADIUS, à escuta em portos diferentes, que não comunicam entre si e a agregação do endereço IP aos registos do RADIUS.

Quanto ao primeiro ponto, o Ponto de acesso, NAS, envia um *Access-request* ao RADIUS Auth. Depois de confirmado o acesso aos serviços, o ponto de acesso envia um *Accounting-request* para o RADIUS Acct. O segundo ponto, diz respeito à forma como o servidor de RADIUS Acct é notificado pelo servidor de DHCP, a fim de completar os seus registos com o endereço de IP. Assim que o DHCP atribui um novo endereço, um script é responsável por agregar esse endereço aos registos de accounting já criados. Por fim, o RADIUS Acct armazena a informação em ficheiros de texto (logs). Mais tarde, e porque os logs gerados pelo RADIUS são muito complexos, são gerados novos logs de interpretação mais fácil.

A Interface web, rata-se de uma página web que pegando nos dados já tratados, os agrupa a fim de disponibilizar a informação de acesso de utilizadores num curto espaço de tempo. A solução a implementar no segundo semestre, vai funcionar em paralelo com esta última e pretende responder à pergunta de quem, quando e onde foi acedido determinado recurso.

Figura 11: Arquitectura de contabilização implementada na UC.

2.3.4.3 Proprietárias – Enterasys e Cisco

Como se pode ver pelo título, estas são duas soluções pagas que permitem fazer controlo de acesso da rede. As duas funcionam de forma diferente, no entanto, o objectivo é o mesmo.

A solução da Enterasys (Enterasys NAC [12]), à direita na Figura 12, requer a instalação de uma aplicação do lado do cliente, que enviará os dados ao servidor. Este agente corre em background e serve sobretudo para detectar programas P2P. Quando algum é detectado, o agente pode terminá-lo e enviar um relatório para o servidor. De forma simples, mas muito evasiva, consegue-se controlar os utilizadores da rede. No entanto, apesar de ser bastante pró-activa, esta solução é pouco funcional pois parte do princípio que o tráfego P2P será o principal afectante da performance da rede.

A solução da Cisco [13], à esquerda na Figura 12,é em tudo diferente da anterior. Esta, suporta dois protocolos - RADIUS e TACACS+ (protocolo que permite AAA tal como o RADIUS). Os NAS da Cisco, tem como sistema operativo o Cisco IOS que têm suporte nativo para contabilização via RADIUS e TACACS+, permitindo aceder a grupos de servidores pré-configurados. Isto é, permitem distribuir a carga do sistema de contabilização pelos diversos servidores de um grupo. Quando é usado o RADIUS, usando o campo Framed-IP-Address especificado no protocolo, o NAS envia ao servidor, no pacote access-request, o IP do computador remoto.

A Cisco faz ainda distinção entre 6 tipos de contabilização. Estes tipos, armazenam informação diferente, que representam diferentes tipos de acesso:

- Network accounting: informação de sessões PPP (Point-to-Point Protocol), SLIP (Serial Line Internet Protocol) ou ARAP (AppleTalk Remote Access Protocol);
- Connection accounting: informação sobre conexões de saída feitas, a partir do servidor de acesso à rede, por telnet, LAT (Local-area transport), PAD (packet

assembler/disassembler) e rlogin.

- EXEC accounting: informação sobre sessões EXEC (acesso via linha de comando ao router);
- System accounting: informação sobre todos os eventos de sistema (reinicializações, accounting on ou off, etc);
- Command accounting: informação de sessões EXEC para níveis de privilégios específicos;
- Resource accounting: geração de registos de start e stop para utilizadores que passem na autenticação e geração do registo stop, apenas, para utilizadores que falham a autenticação.

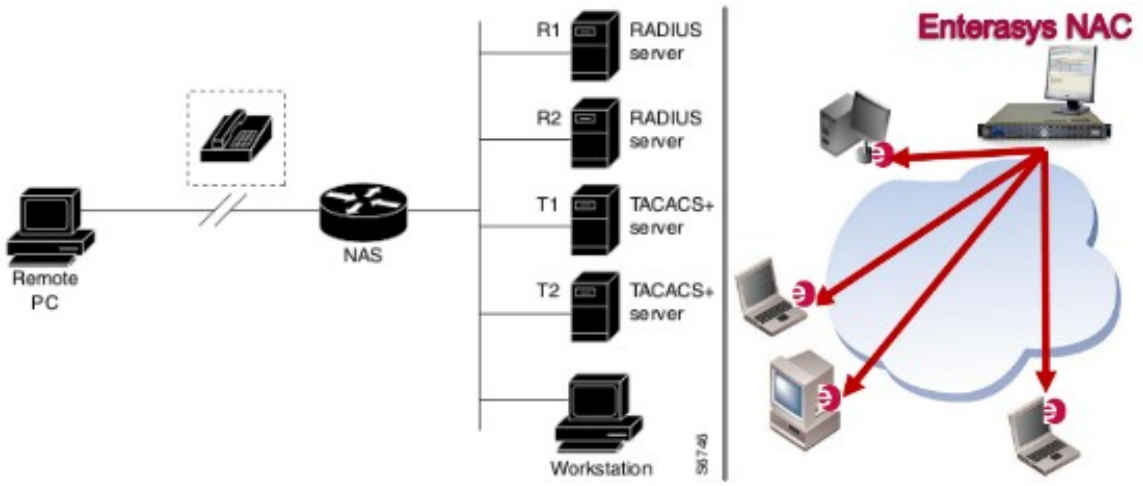

Figura 12: À esquerda a configuração de rede AAA da Cisco ([17]); à direita a da Enterasys NAC ([16])

2.4 Ferramentas de análise de vulnerabilidade

Neste sub-capítulo, apresentam-se três ferramentas simples que permitem saber rapidamente o estado da rede (serviços ou equipamentos) [14]. Estas permitem ao gestor de redes, planear a rede ou simplesmente audita-la, procurando por falhas de segurança. Fornecem também uma boa ajuda na hora de planificar o sistema de monitorização pois permitem definir, sem erros, o mapa da rede.

Apesar de o fim e a forma de actuação (através de *port scans*) de ambas, ser a mesma, estas apresentam resultados diferentes, fazendo com que estes se complementem. Todas elas actuam com recurso a um ou vários port scanners. Isto é, testam um host ou servidor à procura de portos abertos, tcp ou udp, conforme as definições da pesquisa, com o intuito de dar a conhecer aos gestores de redes o que pode estar a comprometer a rede. Com essa informação, os gestores de redes conseguem definir novas políticas de segurança que visem blindar a rede a acessos mal intencionados (ou por atacantes ou por código automatizado).

2.4.1 Nmap

O Nmap [15] surge em 1997, pelas mãos de *Gordon Lyon* e apresenta-se, como uma ferramenta de auditoria de rede que permite descobrir os equipamentos e serviços de rede disponibilizados. Actuando como um Port scanner, apresenta o estado dos portos e serviços de rede de forma rápida e precisa.

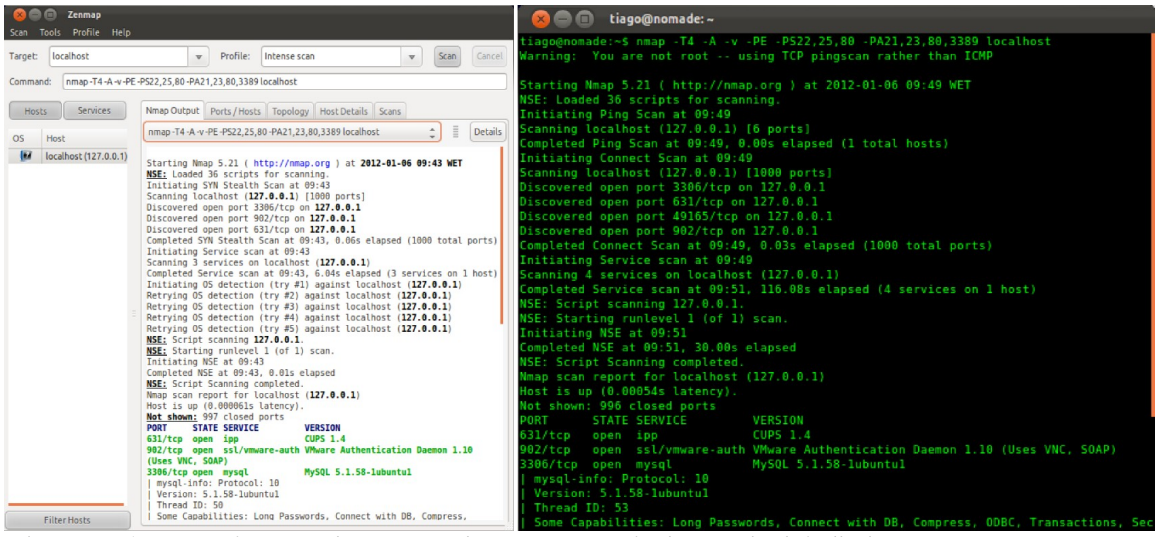

Figura 13: Zenmap à esquerda e comando Nmap através do terminal à direita

Para além disso, pode construir um pequeno mapa da rede auditada e fornecer um conjunto importante de informações acerca dos diferentes hosts: se está ligado, total de portos abertos e fechados, endereço, hostnames, sistema operativo usado, versões de protocolos e serviços usados entre outros.

A principal vantagem do Nmap é a capacidade de criação de scripts complexos, na linguagem de programação Lua, que visam melhorar e intensificar a procura de vulnerabilidades.

É possível usá-lo através da linha de comando ou de um interface gráfica (zenmap - http://nmap.org/zenmap/) para configurar as pesquisas à rede alvo e visualizar os resultados. No entanto, tratando-se, o nmap, de um programada do tipo TUI, o zenmap só disponibiliza algumas pesquisas predefinidas. As pesquisas mais complexas terão de ser sempre feitas pelo utilizador. A grande vantagem do zenmap é a visualização mais organizada e detalhada das pesquisas, como se pode ver na Figura 13.

2.4.2 Nessus

O Nessus [16], produto da Tenable Network Security, surge um ano mais tarde, em 1998, pelas mão de *Renaud Deraison* e apresenta-se, como uma ferramenta proprietária (para instituições; gratuito para uso pessoal) de auditoria de rede, que permite descobrir equipamentos de rede, analisar as suas vulnerabilidades ao nível dos portos, hierarquiza-las (baixo, médio ou alto) e dar a conhecer possíveis causas / soluções.

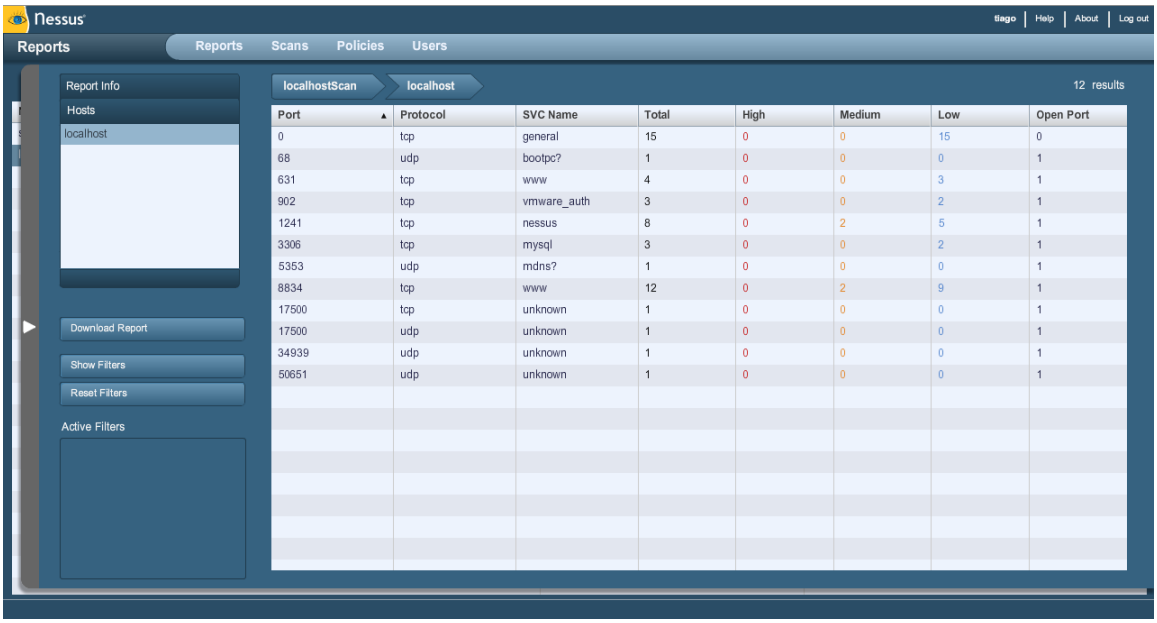

Figura 14: Relatório do Nessus

As pesquisas são feitas com recurso a vários plugins que são responsáveis por testar as vulnerabilidades, nos hosts, ao nível do sistema operativo ou serviços prestados (diversos serviços de rede, web servers e bases de dados por exemplo). Em Janeiro de 2012 estavam disponíveis mais de 46000 plugins.

Funciona sobre uma arquitectura cliente-servidor, em que o cliente, apresenta uma interface para configuração de pesquisas, apresentação de resultados e geração de relatórios. O servidor, é responsável por fazer a pesquisa com as regras definidas e enviar os resultados ao cliente.

2.4.3 OpenVAS

O OpenVAS [17], produto da Greenbone Networks, inicialmente conhecido como GnessUs, surge como um fork do Nessus, em 2008, para que este pudesse continuar a ser disponibilizado, de forma gratuita, às instituições. É, portanto, muito parecido ao anterior, nas funcionalidades disponibilizadas.

No entanto, por utilizar uma plataforma já obsoleta do Nessus (iniciou o desenvolvimento com a versão 2.2 do Nessus) tem algumas limitações. Ao nível dos NVTs, os plugins, este conta com pouco mais de 24000 (Janeiro de 2012). Isto pode-se traduzir em pesquisas mais simples e menos completas.

Trata-se também de uma plataforma menos optimizada, sendo necessária mais capacidade de processamento e memória para a executar.

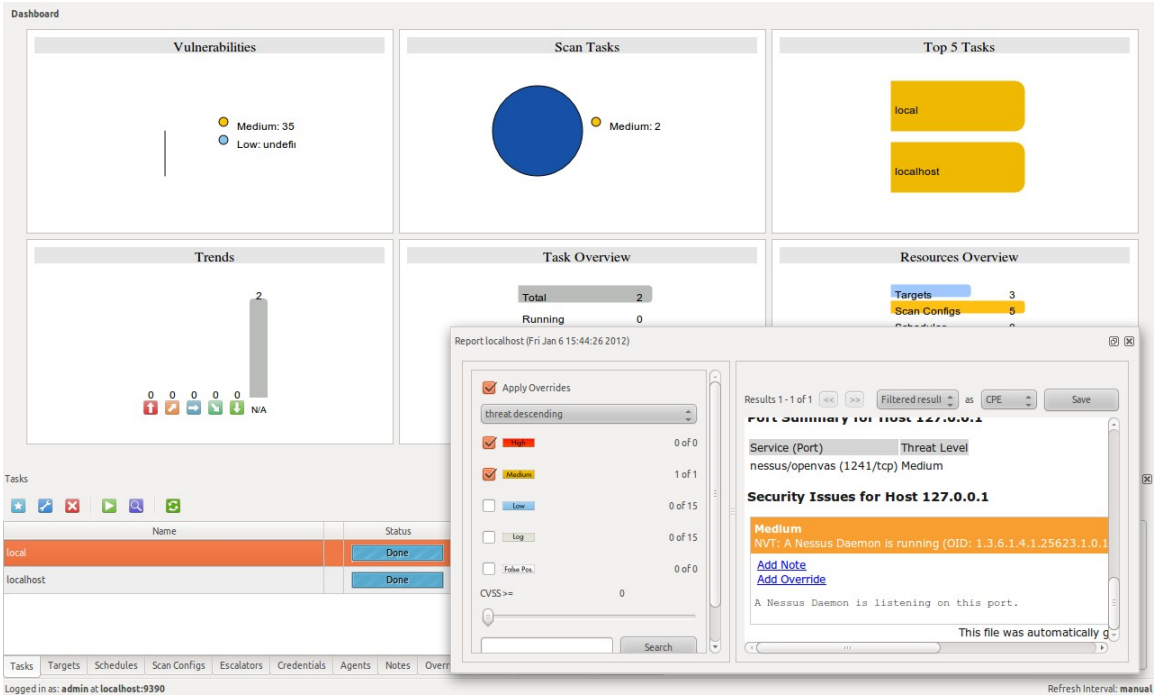

Figura 15: Interface gráfico OpenVAS com um exemplo de relatório do lado esquerdo.

Capítulo 3
Trabalho Realizado

Neste capítulo, é apresentado o plano de trabalho pormenorizado e o trabalho realizado ao longo do presente estágio.

3.1 Plano de Trabalho

O plano de trabalho do primeiro semestre envolveu um estudo aprofundado das tecnologias e técnicas usadas nesta área. Ao longo de cada tarefa, foram identificadas as limitações ou problemas e potencial do que ia sendo analisado.

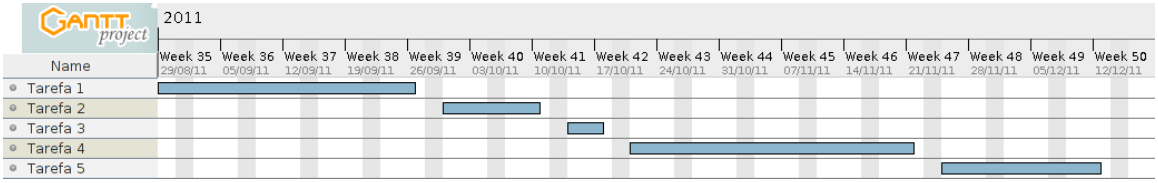

Figura 16: Diagrama de Gantt relativo ao primeiro semestre

Legenda:

Tarefa 1: Estudo de sistema de monitorização e inventário

Tarefa 2: Levantamento da infra-estrutura e serviços da rede alvo

Tarefa 3: Leituras sobre DHCP, DNS, RADIUS

Tarefa 4: Estudo de sistemas de registo e controlo de acesso de utilizadores

Tarefa 5: Instalação e familiarização com ferramentas de suporte e análise de vulnerabilidades.

Para o segundo semestre, irá ser desenvolvido um trabalho com base nos conhecimentos adquiridos no primeiro semestre. Assim, o planeamento será o seguinte:

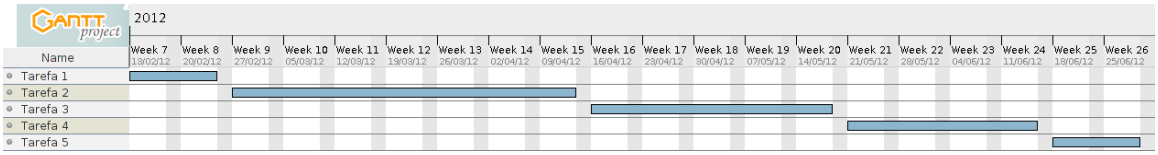

Figura 17: Diagrama de Gantt relativo ao primeiro semestre

Legenda:

Tarefa 1: Instalação do Sistema de monitorização Icinga e complementos necessários

Tarefa 2: Desenvolvimento da estrutura de monitorização e scripts que implementem a pró-actividade.

Tarefa 3: Desenvolvimento do complemento ao sistema de controlo e acessos

Tarefa 4: Desenvolvimento do complemento ao sistema de detecção de alteração de configurações

Tarefa 5: Teste e avaliação.

3.2 Comparação das Ferramentas Icinga e Nagios

Como foi possível ver no capítulo anterior, estas duas plataformas, apesar de partilharem muitos pormenores, já se começam a distinguir uma da outra. A arquitectura do sistema é muito semelhante (Figura 18). Há um Daemon central que coordena toda a monitorização. Acede aos ficheiros de configuração, a fim de lançar os plugins que irão fazer a verificação e por fim, guarda a informação de monitorização para ser acedida mais tarde.

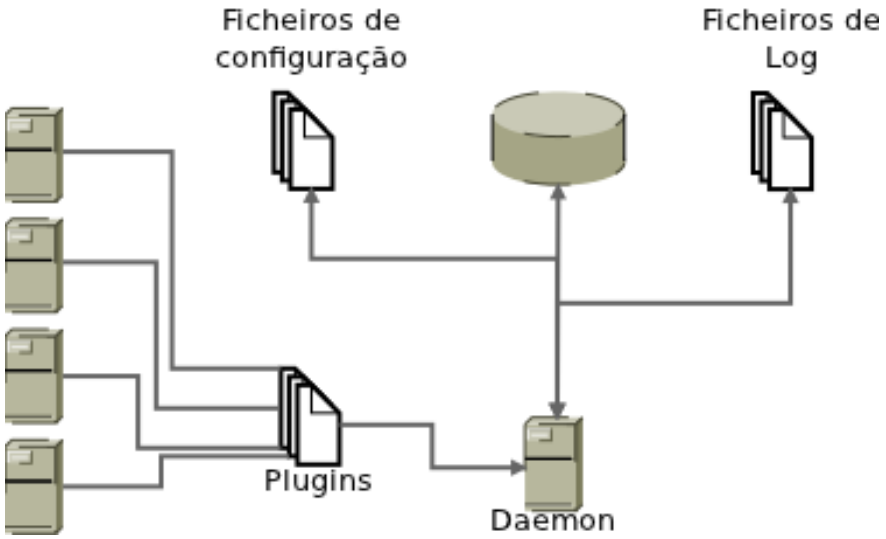

Figura 18: arquitectura genérica do sistema de monitorização

Outro ponto em que os dois sistemas se tocam é ao nível da segurança. Como se tratam de operações críticas para uma organização, há a necessidade de autorizar um determinado número de pessoas a aceder e actuar sobre o sistema. No entanto, são muitos os pontos que distanciam as duas soluções. Os requisitos que ajudaram a comparar e distinguir as duas soluções podem ser vistos de seguida:

- Plataforma extensível
- Livre (baixo custo)
- Actualizada (e em constante actualização)
- Evoluída tecnologicamente (permitirá a construção e integração com ferramentas mais poderosas)
- Alta usabilidade

Sendo assim, ao nível de custos, se se optar pelo Nagios Core, não há custos (assim como no Icinga, inclusive o Icinga Web). Se se quiser uma plataforma mais actualizada

(Nagios XI) e com suporte associado, já haverá custos anuais. Ao nível da extensibilidade, o Icinga, com a sua arquitectura distribuída, oferece melhor suporte ao crescimento ou alterações frequentes na rede. Quanto a actualização tecnológica, o Icinga também é superior:

- Suporta mais motores de base de dados (MySQL, Oracle e PostgreSQL), dando a liberdade ao gestor de redes de escolher o que mais lhe convém;
- API - Comunicação entre Core/Base de dados e Web/Addons é simplificada, assim como o desenvolvimento de addons);
- API REST permite simplificar o desenvolvimento de extensões e exportação de dados;
- Icinga Web - interface Web mais fluída, dinâmica, configurável e funcional que a clássica (presente no Nagios);
- Icinga Reporting – geração e agendamento de relatórios da rede e serviços, muito úteis para configurar melhorias da rede e serviços;
- Icinga Mobile – permite aceder ao Icinga através de um smartphone Iphone ou Android;
- Suporte para IPv6 (poderá ser necessário num futuro muito próximo).

Por último, em termos de actualização, o Icinga sendo um projecto totalmente livre e de código aberto, tem uma grande comunidade de programadores que o permite estar sempre actualizado e apresentar novas soluções. Sendo assim, o trabalho no segundo semestre será desenvolvido com o Icinga.

3.3 Análise de Vulnerabilidade

Foi feita uma análise de vulnerabilidade, a dois hosts, para testar as capacidades das ferramentas e avaliar a exposição dos hosts ao exterior. Os hosts avaliados foram o www.uc.pt (193.137.200.147) e o ftp.uc.pt (193.136.200.34). Estas ferramentas servem como complemento aos sistemas de monitorização, pois permitem ter outra ideia de problemas que haja num determinado host e ajudam a planear melhor a estrutura de monitorização e a proteger os hosts e serviços. A instalação das três ferramentas está no Anexo C – Instalação do Nmap, Nessus e OpenVAS .

3.3.1 Nmap

Foi usado o zenmap, e realizado o scan com um comando já predefinido (intense scan): nmap -T4 -A -v -PE -PS22,25,80 -PA21,23,80,3389 www.uc.pt ftp.uc.pt.

O Scan não revelou problemas de maior ao nível dos portos mas, através deste, fica-se a saber o tipo de sistema operativo (Linux em ambos) e a versão do kernel, bem como o estado dos hosts (up ou down). Os portos encontrados abertos, são os normais para o tipo de serviço que ambos disponibilizam: porto 21 no caso do servidor ftp e postos 80 e 443 no caso do servidor www. Fica-se também com uma ideia da topologia de rede e da rota da

máquina local até cada uma destes servidores, o que ajuda na definição do mapa de rede nas ferramentas de monitorização.

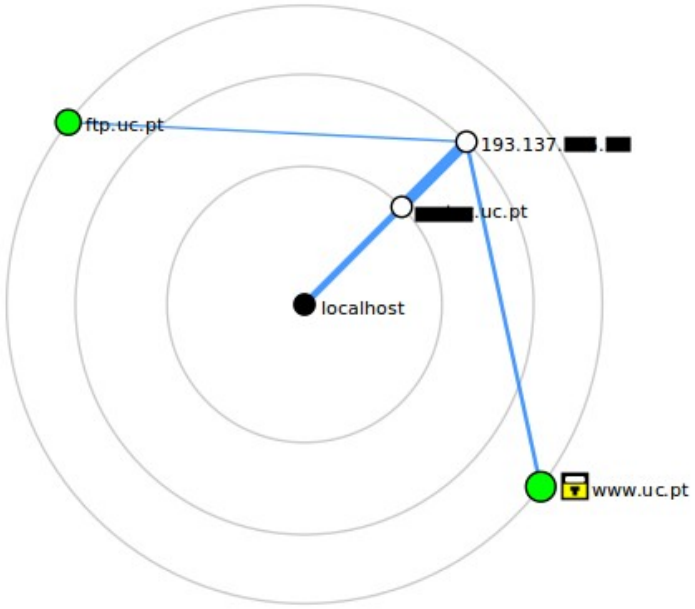

Figura 19: Topologia de rede detectada pelo nmap

3.3.2 Nessus

O Nessus, fez um relatório mais pormenorizado (contem 89 páginas). No entanto também não encontrou problemas de maior. Trata-se de um relatório extremamente extenso onde se podem observar as vulnerabilidades encontradas, organizadas por tipos e portos abertos (os mesmos que o nmap já tinha mostrado). Por exemplo, uma das de nível médio tem a ver com a assinatura do certificado SSL, por este não ser assinado por uma autoridade conhecida. Com esta informação podemos, por exemplo, procurar ou programar um plugin para a ferramenta de monitorização que informe quando um certificado está prestes a expirar. Este permite ver também algumas informações da rede (através de um traceroute) ou mesmo sobre o sistema operativo.

Exemplo do traceroute:

```
Plugin Output
For your information, here is the traceroute from 193.137.215.111 to 193.137.200.147 :
193.137.215.111
193.137.215.1
193.137.215.yyy
193.137.200.147
```

Exemplo da identificação do sistema operativo:

```
Plugin Output
Remote operating system : Linux Kernel 2.6
Confidence Level : 65
Method : SinFP
```

```
The remote host is running Linux Kernel 2.6
```

Consegue-se também saber informação (e-mails e links) através do parsing do HREF dos ficheiros html do servidor www, ou as directorias acessíveis através de um browser (vulnerabilidades baixas). No entanto, o que distingue o Nessus do Nmap, é mesmo a quantidade de informação que o relatório alberga, que pode ajudar o gestor de redes a encontrar solução para determinados problemas. Exemplo da solução para o certificado SSL:

```
Solution
Purchase or generate a proper certificate for this service.
```

De resto, foram detectadas, em ambos os servidores, um total de 52 falhas (médias e baixas).

3.3.3 OpenVAS

Por útimo, a análise com o OpenVAS. Esta foi a que mais surpreendeu pois detectou duas falhas graves, no porto 80 do servidor www. A primeira falha foi encontrada pelo NVT *http TRACE XSS attack*. A mesma, tinha sido também reportada pelo Nessus (pugin HTTP TRACE / TRACK Methods Allowed). No entanto, este identificou-a como de nível médio. No entato, a solução em ambos os casos, é a seguinte:

```
Solution :
Add the following lines for each virtual host in your configuration file :
   RewriteEngine on
   RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
   RewriteRule .* - [F]
```

A segunda, pode afectar a disponibilidade do serviço. Foi encontrada pelo NVT *Apache httpd Web Server Range Header Denial of Service Vulnerability*.

De resto, a informação detectada é muito idêntica mas, porque não tem tantos plugins como o Nessus, a análise não foi tão completa como este, detectando apenas 20 falhas (altas e baixas).

3.4 Instalação e configuração da Ferramenta Icinga

Durante a primeira parte do estágio, procedeu-se à instalação do Icinga e vários complementos, a fim de comparar esta ferramenta com a ferramenta instalada e configurada no GSIIC (Nagios). Para além de servir de comparação, serviu também para aprender como funcionam os sistemas de monitorização, como se poderá implementar a monitorização pró-activa e ver e perceber a organização de rede Universitária. Foram instalados diversos complementos e testados alguns plugins (Anexo B - Comandos Testados na monitorização).

No segundo semestre, todas essas instalações e procedimentos foram validados, a fim de se poder montar toda a estrutura de monitorização simples e pró-activa. Começou-se pelo sistema base, o Daemon e a interface clássica. Aquando desta instalação, foi necessário tomar a primeira decisão. Era necessária a instalação com o módulo IDOUTILS, essencial

para a comunicação com as bases de dados, pois é requisito de alguns dos complementos que foram instalados de seguida. Instalou-se também o Nagios Plugins, o mesmo complemento usado no Nagios, para gestão dos plugins.

Os complementos instalados, têm como objectivo facilitar o acesso e visionamento dos dados de monitorização, em forma de relatório, gráficos ou tabelas. Assim, os complementos instalados foram os seguintes:

- Icinga Web

  Como já se disse anteriormente, este componente trata-se de uma interface web mais dinâmica e intuitiva que permitirá acrescentar um conjunto de complementos específicos que irão ampliar as funcionalidades.

- Icinga Reports

  Permite gerar relatórios sobre estado das máquinas, serviços ou grupos de máquinas ou serviços, através da definição de um intervalo de tempo. Permite poupar tempo na análise de dados e torna-a mais intuitiva.

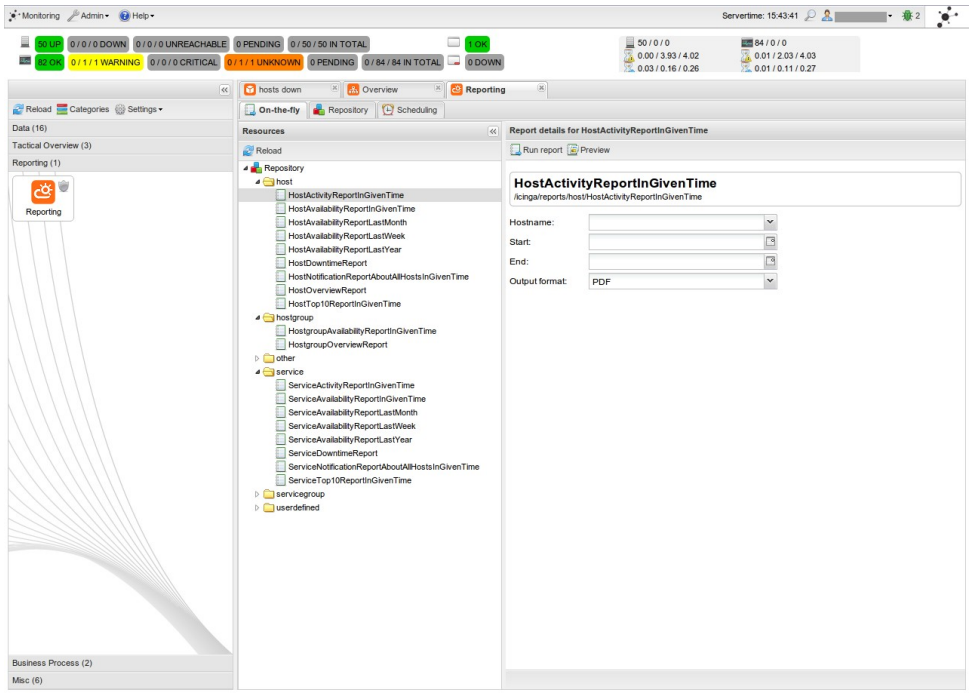

Figura 20: Painel do componente Icinga Reports

- PNP4Nagios

  Gera um conjunto de gráficos relativos às máquinas e serviços. A interpretação dos resultados de verificações é mais simples que com base nos relatórios, mas ambos se complementam.

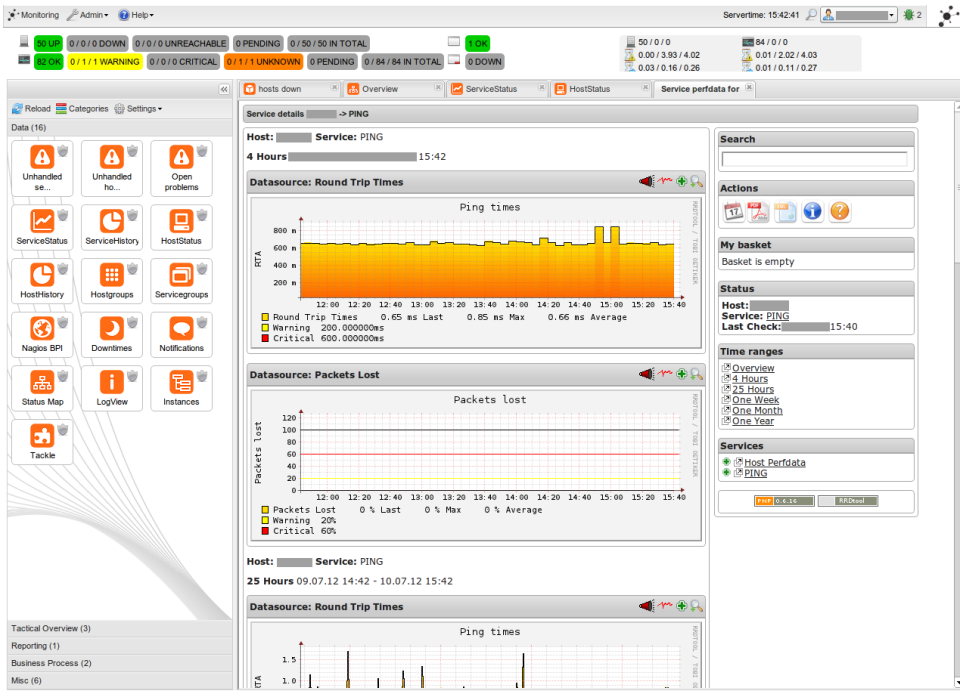

Figura 21: Painel do componente PNP4Nagios

- Business Process

Permite construir um diagrama de rede baseado nos processos de negócio da instituição. Pegando nos acordos de nível de serviço, isto é bastante útil pois permite definir um processo (serviço de mail por exemplo) com o número de máquinas, ou a máquina, indispensável ao seu bom funcionamento através da ligação entre elas. Tem também um plugin que permite monitorizar estes processos o que é muito útil para posterior geração de relatórios com base nos processos de negócio.

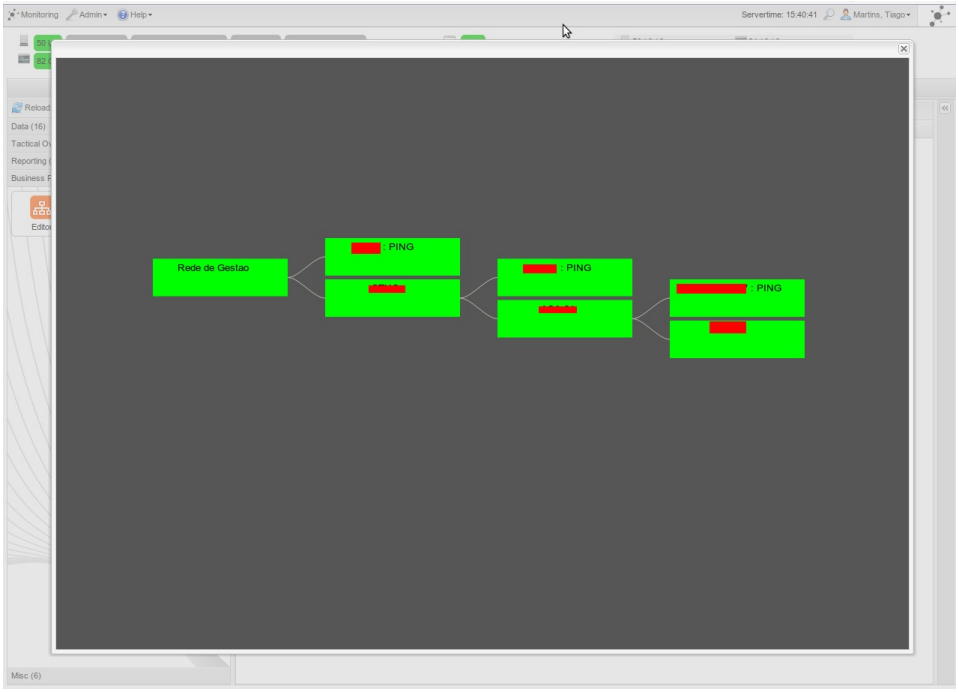

Figura 22: Painel do componente Business Process

- NRPE

  Complemento que para além de permitir implementar a monitorização pró-activa, permite verificar serviços locais em máquinas remotas como os níveis de disco, memória ou um daemon sem acesso ao exterior, através do plugin check_nrpe.

3.4.1 Opções de monitorização

Para que se torne simples e intuitivo a adição de novas máquinas e serviços, ou a interpretação e execução de procedimentos seja mais fácil e rápida, foram tomadas um conjunto de opções sobre a monitorização. As opções têm a ver com o tipo de ficheiros de configuração criados, quais as variáveis que valerá a pena predefinir ou mesmo como serão usados alguns dos complementos já apresentados. Todas estas opções foram tomadas em conjunto, em reuniões. Assim como todas as opções específicas de cada máquina ou serviço. A especificação destas opções pode ser vista no Anexo A – Documentação do Sistema de Monitorização Icinga: Opções de Monitorização.

Também foram definidos um conjunto de procedimentos para instalação de componentes, configuração do NRPE para fazer uso dos event handlers ou adição de novas máquinas e serviços. Estes seguem em Anexo A – Documentação do Sistema de Monitorização Icinga: Procedimento Criados.

3.4.2 Validação de procedimentos

Para que não haja um gestor inteiramente dedicado ao sistema de monitorização, é importante que haja um conjunto de procedimentos que permitam a um conjunto de gestores que, sempre que seja instalada uma nova máquina ou serviço, estes possam adiciona-los ao Icinga. A validação dos procedimentos é assim um meio para aliviar a carga de trabalho dos gestores de redes. À medida que o sistema de monitorização foi sendo montado, os procedimentos iam sendo feitos. Estes têm um conjunto de campos que facilitam a sua leitura e tentou-se que fossem tão simples como copiar e colar comandos num terminal unix.

Os procedimentos iniciam com uma linha que indica do que se trata o procedimento (por exemplo: Instalação do Icinga). De seguida, uma linha com o tempo estimado para a operação, seguido dos requisitos para a instalação e documentação adicional. Por fim, o procedimento propriamente dito, terminando com um pequeno teste (se necessário):

```
### INSTALAÇÃO DO ICINGA

# TEMPO ESTIMADO: 45 MIN
# REQUISITOS
SISTEMA OPERATIVO FEDORA
#
```

```
# DOCUMENTACAO
http://docs.icinga.org/1.6/en/quickstart-idoutils.html
http://docs.icinga.org/1.6/en/icinga-web-scratch.html
http://docs.icinga.org/1.6/en/reporting_1.6.html
http://jasperforge.org/espdocs/download.php?
filename=/opt/jasper/www/espdocs/Documents/112/v4.5%20Documentation/JasperReports-Server-CP-Install-Guide.pdf&ctype=application/pdf
http://docs.pnp4nagios.org/pnp-0.6/install

# INICIO
ssh xxx.xxx.xxx.xxx

# TESTE
ping yyy.yyy.yyy.yyy
```

Após o procedimento estar concluído, é necessária a sua validação. Isto é, saber se seguindo essas instruções, qualquer pessoa o consegue executar. Esta é a parte mais importante pois, depois de validado sabe-se que para realizar aquela tarefa ninguém estará dependente da pessoa que a projectou. A validação implica que alguém pegue no procedimento e o execute até o terminar. O ideal é que este processo seja feito por mais de uma pessoa e que no fim, o parecer seja positivo. Caso não consigam executar o procedimento na totalidade, este é refeito e o processo volta ao início (Figura 23).

Figura 23: Fluxo de validação de procedimento

3.4.3 Monitorização Simples

A monitorização simples, implica a construção de toda a estrutura. Desde a instalação até à configuração da ferramenta. Esta englobou 3 tipos de sistemas a monitorizar (Figura 24):

- monitorização a activos de rede através de agentes SNMP;
- monitorização de aplicações de rede;
- monitorização de Bases de dados.

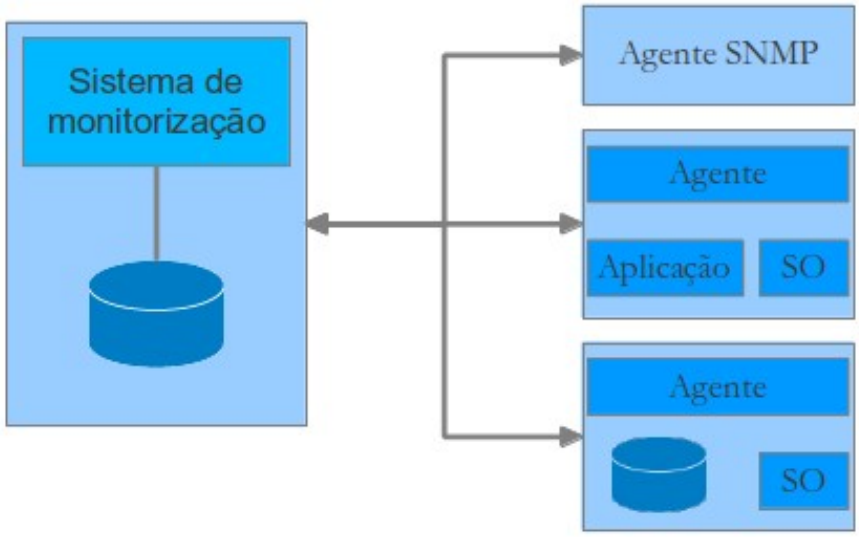

Figura 24: Sistemas a monitorizar

3.4.3.1 Plugins usados

Aquando da instalação do Nagios Plugins, são instalados os mais comuns, usados pela maioria dos gestores. No entanto, há sempre a possibilidade de instalar, ou programar, novos plugins ou mesmo desenvolver um à medida. Para se saber que plugins serão necessários, é preciso saber o que fazem os que vêm por defeito. Para além disso, um bom conhecimento dos serviços prestados aos utilizadores finais também ajuda a perceber que tipo de plugin é necessário.

Ao nível da monitorização com agentes SNMP, há routers, switches ou pontos de acesso. A monitorização através de SNMP permite saber o estado de um activo de rede e seus serviços locais, usando o protocolo SNMP. Os plugins usados formam:

- check_snmp_int.pl (http://nagios.manubulon.com/snmp_int.html): verificação de do tráfego gerado numa das interfaces de ligação.
- check_snmp_load.pl (http://nagios.manubulon.com/snmp_load.html): verificação da carga de cpu.

Os plugins usados na monitorização de aplicações de rede foram diversos e permitem verificar todo os tipo de serviços endereçados a um sistema de redes. Os plugins a seguir expostos permitem verificar as máquinas e serviços de mail, directorias ou web:

- check_ldap: verificar a resposta do servidor de ldap numa máquina
- check_ldap_syncrepl_status.pl (https://ltb-project.org/svn/nagios-plugins/trunk/check_ldap_syncrepl_status.pl): verifica se as réplicas do servidor ldap estão sincronizadas com o servidor ldap principal.
- check_http: usado na verificação de servidores web ou páginas web específicas.
- check_ping: usado para verificar a conectividade a uma máquina.
- check_ssh: verifica se é possível ligar à maquina remota por ssh. Indispensável no caso haver algum tipo de problema que só se resolva estando "dentro" da máquina.
- check_mailq: usado para verificar o tamanho da fila de email. Como se trata de um serviço local, este tem de ser usado com recurso ao NRPE.
- check_spamd.pl (http://exchange.nagios.org/directory/Plugins/Email-and-Groupware/SpamAssasin/check_spamd/details): usado na verificação do daemon do spamassassin (aplicação para filtragem de spam num servidor de mail).
- check_tcp: usado na verificação sockets de serviços (por exemplo o socket de um antivirus usado no servidor de mail).
- check_pop: usado para verificar o serviço pop com ou sem ssl.
- check_imap: usado para verificar o serviço imap com ou sem ssl.
- check_smtp: usado para verificar o serviço smtp com ou sem ssl

Na verificação de bases de dados, foram monitorizados 2 tipos de motores: mysql e postgres. No entanto, não basta verificar só os parâmetros relativos à base de dados em si. É necessário monitorizar a percentagem de disco usada, ou o nível de swap. Assim, os principais plugins usados foram os seguintes:

- check_mysql_health (http://exchange.nagios.org/directory/MySQL/check_mysql_health/details): usado na verificação de bases de dados mysql. Tem diversos modos que permitem verificar tempo de conexão à base de dados, fragmentação de tabelas entre outros que permitem mitigar problemas na inserção de dados.
- check_postgres.pl (http://bucardo.org/wiki/Check_postgres): usado na verificação de bases de dados postgres. Este, assim como o anterior, também tem diversos modos que permitem verificar a conexão à base de dados, tamanhode tabelas entre outros.
- check_swap: usado na verificação da utilização de memória swap. A utilização abusiva de swap, pode indicar problema na base de dados.
- check_disk: usado na verificação da percentagem de disco ocupada.

3.4.4 Monitorização pró-activa de curto prazo

Pegando no sistema de monitorização já configurado, partiu-se para uma nova fase. A monitorização pró-activa de curto prazo que se trata da reacção automática a um estado do sistema. Os plugins instalados fornecem um conjunto de informações bastante úteis e precisas que podem ser usadas para reagir automaticamente a um estado do sistema. Esta fase revelou-se mais complexa e morosa que a anterior pois é necessário grande conhecimento de um serviço e uma fase de testes alargada. Ainda assim, os exemplos criados são de qualidade.

Antes de tentar activar este tipo de automatismos, é necessário avaliar muito bem o problema. Perceber porque é que o serviço falha é uma ajuda na resolução do problema. Por exemplo, uma das máquinas a que foi aplicado este tipo de monitorização foi um servidor web. Este podia falhar porque o servidor de tomcat não consegue aceder à base de dados, porque o servidor tomcat se encontra em baixo, a máquina está desligada ou simplesmente está a decorrer um processo de backup. Todos estes parâmetros foram tidos em consideração na construção do event handler que tratará os erros. No caso, sempre que o servidor tomcat não estava disponível e não estava em Backup, reiniciava. Assim, os principais requisitos deste event handler foram os seguintes:

- Reiniciar serviços ao fim de 3 tentativas de verificação quando em estado UNKNOWN ou CRITICAL;
- Reiniciar imediatamente se o tipo de estado for HARD;
- Não executar qualquer tarefa se o tomcat devolver 503 (significa que está em processo de backup).

Outro exemplo foi a criação de um plugin que verificava se um script, responsável pelo armazenamento de resgistos de uma aplicação numa base de dados, estava a correr ou não. Neste não foi usado o eventhandler. O reinicio era feito automaticamente assim que se detectava que o script estava em baixo. Isto porque este script era essencial para que a aplicação não deixasse de dar resposta aos pedidos Assim, os requisitos para a construção deste plugin são:

- capacidade de prever a falha de um script específico;
- reiniciar o script caso necessário;
- enviar o estado para o Icinga notificar o administrador se não for possível reiniciá-lo após 2 tentativas.

Em ambos os casos, tentou-se que os scripts/plugins fossem o mais genéricos possível. Como a rede tem muitos serviços idênticos a diferenciação entre serviços, passa pelos parâmetros fornecidos. Assim, para além de permitir melhor organização dos ficheiros de configuração, a utilização torna-se mais simples. Ambos os exemplos estão detalhados no Anexo A – Documentação do Sistema de Monitorização Icinga: Especificação de requisitos e testes aos Event handlers.

3.4.4.1 Processo de configuração

O processo de configuração da monitorização pró-activa envolveu sempre diversas fases. A primeira era a adição das máquinas e serviços a monitorizar ao sistema de monitorização Icinga. A segunda, e depois de avaliar os dados gerados seria a construção do event handler e o ajuste dos limites dos serviços, para que não fossem gerados falsos negativos. Para esta segunda fase, era necessário testar o event handler. Como tal, este só era activado numa máquina de testes. Numa terceira fase, e depois da análise dos resultados obtidos (através dos relatórios ou gráficos do PNP4Nagios) é que se avançava para a configuração do event handler no sistema de produção.

Ao longo destes processos, tentou-se validar com os gestores os procedimentos necessários à configuração dos event handlers. Nomeadamente o procedimento de instalação do componente NRPE na máquina remota. Para que isto fosse possível era realizada uma reunião onde, para além de se discutir detalhes das máquinas a monitorizar como os serviços e os seus limites, eram validados os procedimentos. Assim, consegue-se distribuir as tarefas pelos gestores, simplificando adição de novas máquinas.

3.4.4.2 Fluxo da monitorização pró-activa

Quando é feita através de event handlers, os plugins vão estar coordenados com estes últimos. Internamente o Icinga só invoca o Event handler quando há mudança de estado ou quando o serviço permanece num estado diferente de OK, até se atingir o número máximo de verificações. A partir daí, começa a contagem para saber o número da verificação. Quando se chega ao número máximo de verificações e se satisfazem as outras condições, é enviada uma acção através do check_nrpe.

No caso da verificação e resolução do problema se fazer com o plugin, é usado apenas o comando check_nrpe, sendo que o event handler nem sequer é invocado.

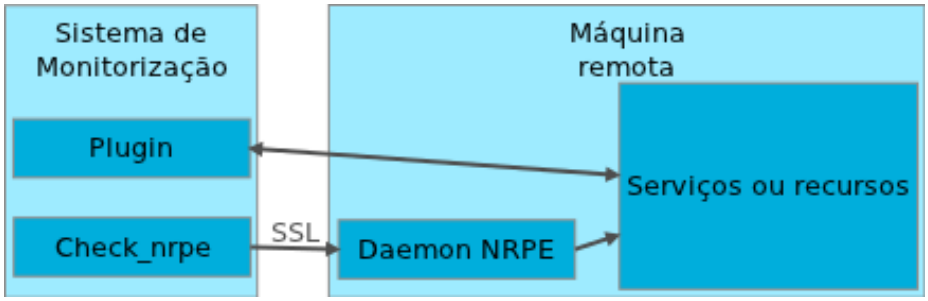

Figura 25: Fluxo da moitorização pró-activa

3.4.5 Monitorização pró-activa de longo prazo

A monitorização pró-activa de longo prazo (capacity planning) é capacidade da organização para evolução da infraestrutura de TIC em função do consumo de recursos (CPU, memória, etc) num dado intervalo de tempo. Esta permite que o gestor avalie a rede e os dados gerados na monitorização a fim de a fazer evoluir antes da mesma se ressentir.

Não foi feito nada de concreto com este tipo de pró-actividade. Este é endereçado

ao gestor de rede e à forma como vai reagir a um conjunto de problemas comuns. No entanto, a análise dos relatórios e o comportamento dos event handlers dão uma ideia de como se deve orientar a rede. A análise de todo tipo de documentação gerada pelo sistema de Monitorização permite ao gestor de redes detectar padrões, inconsistências, nos problemas encontrados. Através disto, torna-se fácil encontrar uma solução que faço evoluir a rede. Este tipo de monitorização fará parte do trabalho futuro.

3.5 Sistema de controlo de acessos

O sistema de controlo de acesso surge no estágio desenvolvido na perspectiva de aliviar a carga do gestor de redes na verificação de queixas de ordem legal. O sistema de controlo de acessos já se encontra construído, com os dados parcialmente tratados e armazenados em ficheiros, o que facilita um pouco o nosso trabalho.

A tarefa que a aplicação desenvolvida é chamada a executar já era feita. No entanto, era necessário pessoal especializado para a fazer. Esta envolvia a análise dos ficheiros em pessoa, o que a tornava muito complexa e demorada. A aplicação desenvolvida, pretende analisar os ficheiros e responder automaticamente à questão de quem, quando e onde foi acedido determinado recurso da rede. A documentação relativa ao sistema está disponível no Anexo D – Documentação do Sistema de Controlo de Acesso.

3.5.1 Requisitos

Esta aplicação tem como principal objectivo encontrar automaticamente um registo de acesso através da inserção de dois parâmetros: o endereço IP e a data de acesso ao recurso. O sua usabilidade pretende ser tão simples como inserir um conjunto de dados, accionar a procura e esperar um resultado. Para que tal fosse possível, a análise de requisitos permitiu desenhar uma aplicação que consentisse o uso por qualquer tipo de pessoa ou seja, por pessoas sem conhecimento na área de redes. Os requisitos eram:

- Definição da data de acesso (incluindo as horas)
- Definição do Timezone
- Definição do IP usado no acesso
- Apresentação da informação de acesso

3.5.2 Tecnologias usadas

Para o desenvolvimento desta aplicação, foram usadas as tecnologias representadas na Tabela 1.

Nome	Descrição
Python	Linguagem de programação usada no desenvolvimento do middleware
PHP	Linguagem de programação usada no desenvolvimento da interface web
HTML	Linguagem de programação usada no desenvolvimento da interface web

Tabela 1: Tecnologias usadas na aplicação

A escolha destas tecnologias derivou do conhecimento existente no GSSIC, onde a maioria das aplicações desenvolvidas têm por base estas três tecnologias. Assim, garante-se facilidade de suporte à aplicação e possível integração com a interface web do sistema antigo (também ele desenvolvido em PHP).

3.5.3 Arquitectura

A arquitectura é uma fase muito importante no desenvolvimento da aplicação uma vez que mostra todos os módulos envolvidos. A opção pela divisão por módulos prende-se com o facto de ser mais fácil integrar novos módulos, adicionar funcionalidades ou dar manutenção no sistema. Como se pode ver na Figura 26, a aplicação é constituída por três módulos, que estão especificados na documentação em anexo:

- Interface web
- Middleware
- Repositório de informação

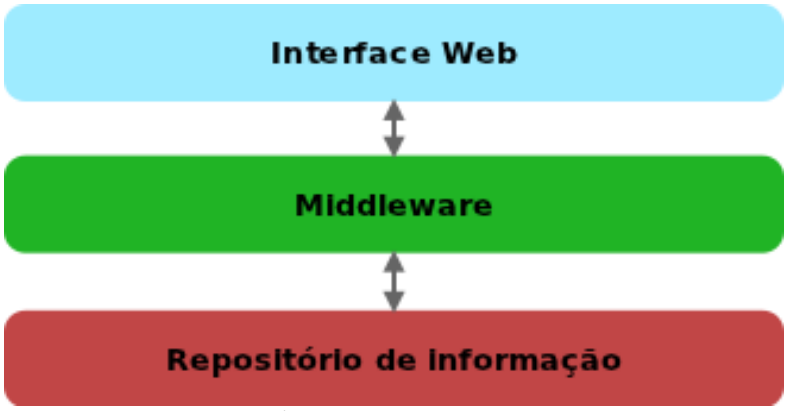

Figura 26: Modelo de três camadas usado no sistema.

Na Figura 27 pode-se ver a interface web desenhada. Através desta, pode-se definir um conjunto de parâmetros - data de acesso, timezone e endereço IP - que serão usados na pesquisa a fim de determinar os dados do utilizador ligado naquele momento.

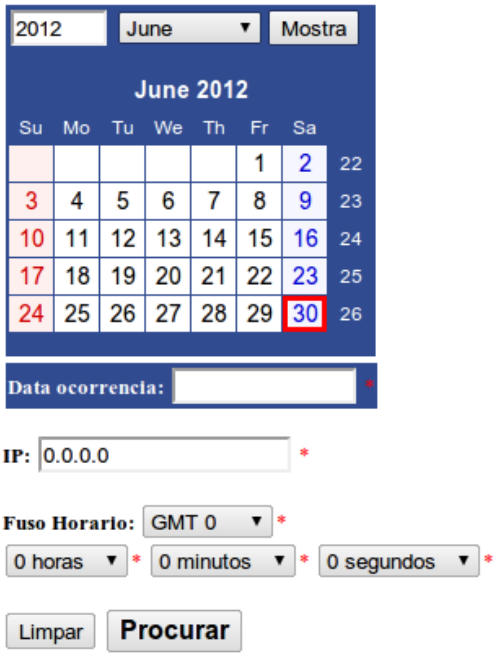

Figura 27: Aspecto final da interface web

O Middleware é o responsável por determinar que utilizador da rede se encontra dentro dos parâmetros definidos de data de acesso e endereço IP. Através da análise dos ficheiros que constituem o repositório de informação, o middleware consegue determinar um utilizador único. Devolvendo data de início e fim de sessão, endereço IP usado, a que ponto de acesso esteve ligado e o endereço MAC do equipamento, como se pode ver na Figura 28.

Data hora actual (LOCAL)	Sat Jun 30 15:41:25 2012
Hora da ocorrencia (GMT+1)	Sun Aug 12 11:42:07 2012
Hora da ocorrencia (LOCAL)	11:42:07
IP da ocorrencia (a procurar)	111.222.333.3
Data - hora de inicio	20120812-11:41:19
Utilizador (e-mail)	wwweee@student.uc.pt
Ligado ao AP	ap5648.uc.pt|10.20.50.50
Ligado com o IP	111.222.333.3
Ligado com o MAC	12sd3rty57j8
Ligacao Terminada em	20120812-11:45:19

Figura 28: Output gerado e já traduzido para html, em forma de tabela

3.6 Sistema de detecção de alteração de configurações

O sistema de detecção de alteração de configurações surge no estágio desenvolvido

na perspectiva de, mais uma vez, aliviar a carga do gestor de redes. A aplicação desenvolvida será apenas mais um componente adicionado ao sistema já existente que apenas diferenciava duas configurações do mesmo activo de rede para datas distintas.

A tarefa que a aplicação desempenha era feita esporadicamente por um gestor, caso surgisse a desconfiança de que um ponto de acesso não estava configurado como devia. Isto retirava tempo ao gestor de rede e podia manter a rede permeável a ataques caso não fosse detectado a tempo. A documentação relativa ao sistema está disponível no Anexo E – Documentação do Sistema de Detecção de Alteração de Configurações.

3.6.1 Requisitos

A aplicação desenvolvida tem a capacidade de detectar inconsistências relativamente ao procedimento de qualquer ponto de acesso da rede (validação de configuração). Assim, ao invés de o gestor ter de analisar minuciosamente todas as configurações, a sua tarefa passa apenas a ser a de verificar o output gerado pela aplicação. Este dirá em que estado se encontra o activo e ajudará na decisão a tomar.

Para isso, foram construidos três componentes para a mesma aplicação:

- Duas interfaces web – uma que integra a nova aplicação com a antiga e outra que apenas valida uma configuração.
- Validação de configurações - Aplicação que irá detectar as inconsistências com os procedimentos de todos os pontos de acesso (aproximadamente 400), a fim de as validar.

Os requisitos estão então divididos pelos três componentes e, no caso das interfaces web são:

- Escolha do dia e mês a verificar
- Facilidade de navegação entre dias
- Definição de endereço IP
- Salientar visualmente as inconsistências com os procedimentos
- No caso da interface que integra o sistema antigo, esta deve salientar também as diferenças entre as configurações de dias diferentes

No caso do validador de configurações, os requisitos são mais simples:

- verificar as inconsistências com os procedimentos de todos os backups
- Enviar email ao gestor de redes com um relatório da análise

3.6.2 Tecnologias usadas

Para o desenvolvimento desta aplicação, foram usadas as tecnologias representadas na Tabela 2.

Nome	Descrição
Python	Linguagem de programação usada no desenvolvimento do middleware e do analisador de backups
PHP	Linguagem de programação usada no desenvolvimento da interface web
HTML	Linguagem de programação usada no desenvolvimento da interface web

Tabela 2: Tecnologias usadas na aplicação

A escolha destas tecnologias derivou do conhecimento existente no GSSIC, onde a maioria das aplicações desenvolvidas têm por base estas três tecnologias. Assim, garante-se facilidade de suporte à aplicação e possível integração com a interface web do sistema antigo (também ele desenvolvido em PHP).

3.6.3 Arquitectura

Como se pode ver na Figura 29, a aplicação é constituída por três módulos:

- Módulo de apresentação (constituído por 2 tipos de componentes)
- Middleware
- Repositório de informação

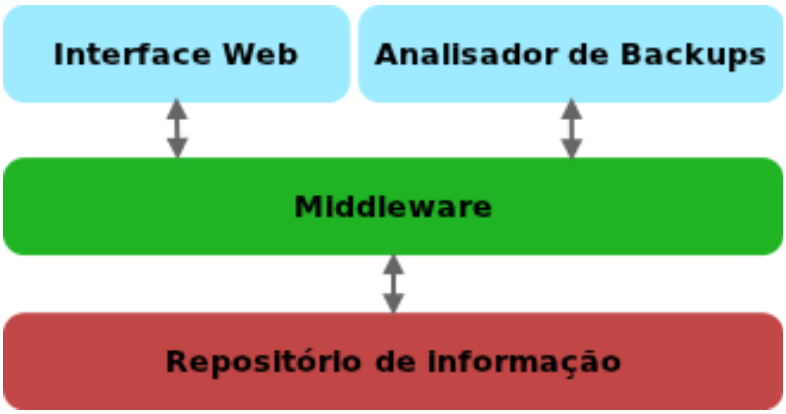

Figura 29: Modelo de três camadas usado no sistema.

Na Figura 30 pode-se ver uma das interfaces web que é responsável pela análise de uma configuração a fim de a validar. Pode-se escolher o mês e dia, tal como o endereço IP do ponto de acesso.

Figura 30: Interface web do sistema da análise da configuração

A outra interface, visível na Figura 31, procura agregar o sistema antigo, que diferenciava duas configurações do mesmo ponto de acesso, ao novo sistema, que fará a validação da configuração mais recente.

Figura 31: Interface web do sistema que diferencia duas configurações e analisa a mais recente

Por último, o analisador de backups que será executado todos os dias, depois de feitos os respectivos backups. Este será accionado pelo sistema operativo e analisará todos os backups a fim de detectar quais as configurações de pontos de acesso não estão de acordo com os procedimentos. No entanto, este ainda não se encontra automatizado. O objectivo é que o Sistema de monitorização controle esta parte, como será explicado no capítulo 5.2 - Trabalho a Realizar.

Capítulo 4
Testes realizados

Os testes seguintes tiveram como objectivo validar os sistemas desenvolvidos relativamente ao desempenho e funcionalidade dos mesmos. Os mesmos podem ser vistos em detalhe na secção de testes dos Anexos D e E. Apenas serão tratados os testes do Sistema de controlo de acesso e sistema de detecção de alteração de configurações.

4.1 Testes Funcionais

Os testes funcionais, serviram para verificar se os sistemas tinham o comportamento inicialmente previsto.

4.1.1 Sistema de controlo de acessos

Os testes funcionais ao sistema de controlo de acessos foram realizados na linha de comando. No entanto, para validar também o comportamento da interface web, estes podem la ser executados bastando para tal, preencher os campos com os parâmetros usados. Para que fosse mais fácil realizar os testes e nenhuma informação confidencial fosse tornada pública neste relatório, foram construídos 2 ficheiros que simulavam os dados recolhidos do servidor RADIUS – um com os dados de conexão e desconexão e outro com os endereços MAC usados nas ligações. No entanto, também foram realizados testes com os dados reais, os quais não são aqui divulgados.

O comando usado pode ser visto de seguida com os parâmetros dia, mês e ano, hora, minuto e segundo, timezone e endereço IP:

```
python procura_ip.py '11 08 2012 23:41:15' GMT+1 111.222.333.4
```

4.1.2 Sistema de detecção de alteração de configurações

Os testes funcionais aos componentes do sistema de detecção de alteração de configurações foram executados na nas interfaces web ou linha de comando, dependendo do componente. Os testes, foram realizados com ficheiros reais e a especificação dos mesmos encontra-se no anexo da documentação do sistema de detecção e alteração de configurações.

Estes serviram para testar os requisitos do sistema. Isto é, se eram detectadas ou não alterações entre ficheiros ou relativamente ao procedimento e se esses dados de saída eram usados para realçar as linhas modificadas nos ficheiros.

4.2 Testes de desempenho

Os testes de desempenho serviram para verificar a performance dos sistemas desenvolvidos recorrendo a 2 métricas: memória usada e tempo de processamento. A memória usada foi medida recorrendo ao ficheiro de estado do processo no sistema

operativo. O tempo de processamento através do módulo Time do Python.

4.2.1 Sistema de controlo de acessos

Os testes de desempenho ao sistema de controlo de acessos também foram realizados na linha de comando. No total foram executados oito comandos que pretendiam medir o desempenho da aplicação à medida que a pesquisa avançava no mês. Isto é, pegou-se no ficheiro de Abril (tamanho de aproximadamente 380 Megabytes) e correu-se um comando para os dias 1, 2, 10, 11, 20, 21, 29, 30. Foram realizadas três medições por cada dia. O gráfico seguinte (Figura 32) apresenta a média destas medições e pode-se ver que, exceptuando, o dia 20, à medida que se avança no mês, o tempo de processamento aumenta. Este comportamento é explicado pelo facto de não se saber quando é iniciada a ligação logo, é necessário varrer todo o ficheiro.

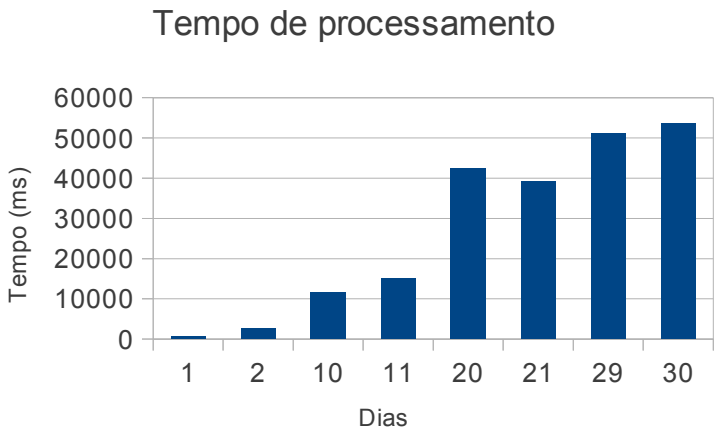

Figura 32: gráfico do tempo de processamento

O comportamento relativamente ao consumo de memória (Figura 33) também é o esperado. Ao longo da nossa pesquisa são guardadas as linhas de desconexão para precaver o facto de estas poderem aparecer antes das conexões. A memória usada, corresponde a estas.

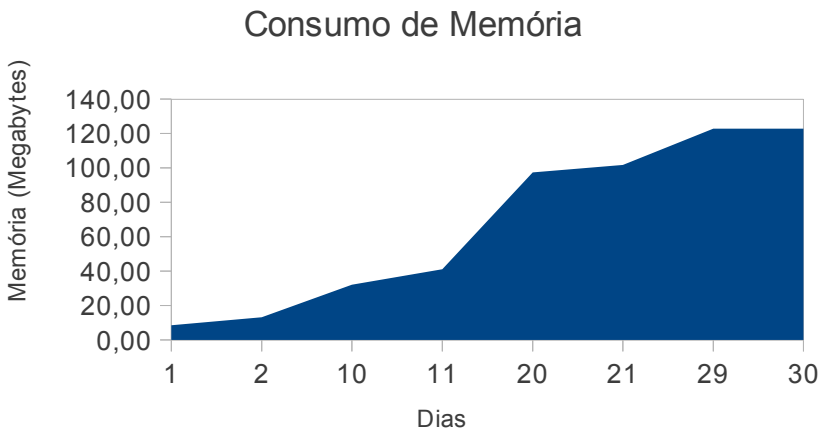

Figura 33: gráfico do consumo de memória

4.2.2 Sistema de detecção de alteração de configurações

Os testes de desempenho no sistema de detecção de alteração de configurações, foram feitos aos três componentes e, directamente na interface web, quando aplicável. Para estes testes, foram usados os ficheiros reais.

O gráfico da Figura 34 apresenta o consumo de memória para:

1. componente que verifica todas as configurações ao final do dia;
2. componente que valida uma configuração;
3. componente que agrega o sistema antigo e o validador de configuração.

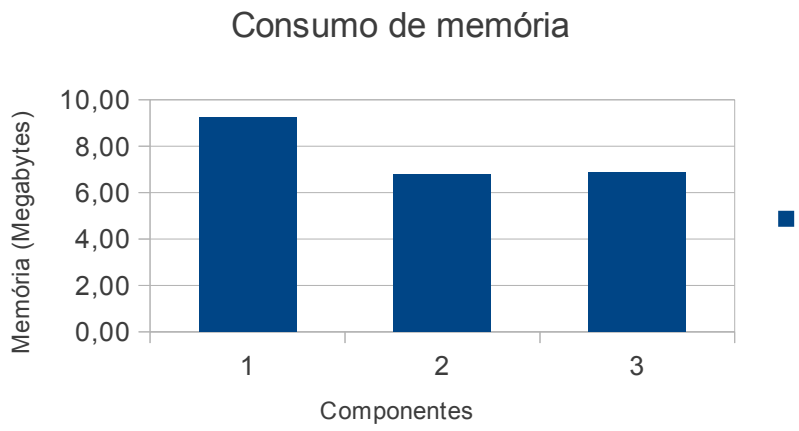

Figura 34: Gráfico do consumo de memória

Quanto ao tempo de execução, não é significativo. No caso dos componentes com interface web, o tempo de processamento anda à volta dos 100-140 milisegundos. No caso da aplicação que verifica todas configurações, o tempo é de 14112 milisegundos (aproximadamente 14 segundos).

Capítulo 5
Conclusão

5.1 Balanço do Trabalho Realizado

A área da monitorização é crítica para o sucesso das redes de comunicação. A medição do estado da rede, permite que esta esteja sempre operacional e que se cumpram determinadas quotas de nível de serviço. Assim, é essencial ter um bom sistema, de monitorização e controlo de acesso, robusto e que permita rápida e concisamente retirar todos os parâmetros que possam influenciar a performance da rede e seus serviços. Como também é importante implementar algumas soluções que ajam automaticamente ou simplifiquem as tarefas dos administradores de rede.

Através das análises às várias ferramentas disponíveis, percebe-se que cada organização constrói o seu sistema à medida das suas necessidades. Todas estas ferramentas fornecem esta flexibilidade. As ferramentas de monitorização analisadas (Nagios e Icinga), têm um nível de flexibilidade bastante elevado, permitindo construir soluções extremamente poderosas e ajustáveis ao tipo/tamanho da rede. No entanto, e para que o sistema de monitorização consiga acompanhar o desenvolvimento tecnológico (e com baixos custos), a ferramenta Icinga é sem dúvida a que leva maior vantagem. Esta traz já um conjunto de soluções (API) que facilitam, por exemplo, a programação de novos complementos. As ferramentas de análise de vulnerabilidade permitem ao gestor de rede entrar rapidamente dentro da rede a fim de a auditar e perceber melhor os problemas existentes. Outra das vantagens, é a possibilidade de se ter uma visão de fora ou dentro da rede. A informação disponibilizada por estas ferramentas pode inclusivamente aliviar a carga dos gestores de redes, na medida em que gera relatórios, pormenorizados, do estado das máquinas.

A implementação da pró-actividade através das ferramentas de monitorização é um processo complicado e moroso. No entanto quando bem feito, permite poupar muito tempo e dores de cabeça ao gestores da rede. Actuando quase na sombra do Sistema de Monitorização, o gestor praticamente só se apercebe do seu funcionamento pela análise de relatórios.

Ao nível do controlo de acesso, foram analisadas soluções que permitem fazer AAA e o que cada uma das componentes significava. Para responder aos problemas encontrados, foi criado um sistema que com a sua simplicidade permite ser usado por qualquer pessoa, pois analisa automaticamente um log confuso que só é entendido por quem tem conhecimentos na área.

A análise de configurações de activos de rede também se revelou produtiva. O sistema criado permite validar as configurações dos pontos de acesso da rede, dando uma visão simples e rápida do estado dos mesmos. Assim, consegue-se libertar o gestor para outras tarefas e manter a rede segura.

O trabalho realizado durante o primeiro semestre, permitiu o contacto com as mais diversas tecnologias desta área. Mas mais que isso, permitiu estar dentro de uma organização que lida diariamente com vários problemas e que tenta sempre encontrar novas e melhores soluções para os mesmos.

5.2 Trabalho a Realizar

Apesar de ter sido desenvolvido um conjunto interessante de aplicações, não houve tempo para desenvolver mais profundamente algumas delas. Assim, como trabalho futuro há que continuar a desenvolver a plataforma de monitorização. Com os exemplos e procedimentos feitos, esta tarefa fica mais simples para qualquer gestor do GSSIC.

A monitorização pró-activa de longo prazo (capacity planning) toma agora mais importância pois, o sistema de monitorização montado, permite ao gestor de redes perceber o que se passa na rede e com recurso aos diversos complementos, saber em que direcção esta irá prosseguir. Assim, tem a possibilidade de actuar sobre esta fazendo-a evoluir.

Aproveitando uma das funcionalidades do sistema de monitorização - sistema distribuído - irá ser construido um sistema de monitorização distribuído em que uma das instâncias (Figura 35) estará dedicada aos pontos de acesso. Foi por causa deste aspecto que o sistema de detecção de alteração de configurações não ficou completo. Juntando estes dois componentes, é possível construir um sistema autónomo que verifique os backups e notifique um conjunto de gestores.

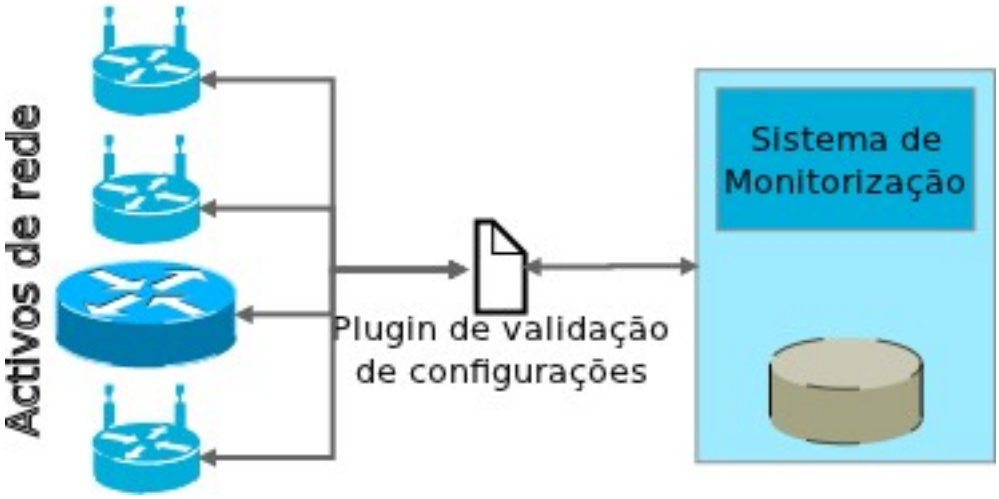

Figura 35: Arquitectura do sistema de verificação dos backups de configuração

Para além de se conseguir programar o sistema para verificar os backups num determinado período de tempo, consegue-se também eliminar alguma entropia na hora de ver o mapa da rede e, programar as notificações aos gestores de rede responsáveis por cada ponto de acesso.

No fundo, o trabalho a realizar girará sempre à volta da construção de sistemas que automatizem alguns processos endereçados ao gestor de redes.

Bibliografia

[1] 'Nagios - The Industry Standard in IT Infrastructure Monitoring', *Nagios - The Industry Standard in IT Infrastructure Monitoring*. [Online]. Available: http://www.nagios.org/. [Accessed: 06-Sep-2011].

[2] 'Icinga: Open Source Monitoring', *Icinga: Open Source Monitoring*. [Online]. Available: https://www.icinga.org/. [Accessed: 06-Sep-2011].

[3] J. Turnbull, *Pro Nagios 2.0*. Apress, 2006.

[4] G. López, O. Cánovas, A. F. Gómez, J. D. Jiménez, and R. Marín, 'A network access control approach based on the AAA architecture and authorization attributes', *Journal of Network and Computer Applications*, Jul. 2005.

[5] M. Shi, H. Rutagemwa, X. Shen, J. W. Mark, Y. Jiang, and C. Lin, 'AAA ARCHITECTURE AND AUTHENTICATION FOR WIRELESS LAN ROAMING', in *Wireless Network Security*, Springer, 2007.

[6] F. Boavida, M. Bernardes, and P. Vapi, *Administração de Redes Informáticas*, 1ª ed. FCA, 2009.

[7] E. Monteiro and F. Boavida, *Engenharia de Redes Informáticas*, 10ª ed. FCA.

[8] C. Metz, 'AAA PROTOCOLS: Authentication, Authorization and Accounting for the Internet'. 1999.

[9] C. Rigney, et al., 'RFC 2865 - RADIUS', Jul-2000. [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt. [Accessed: 02-Nov-2011].

[10] C. Rigney, et al., 'RFC 2866 - RADIUS Accounting', Jun-2000. [Online]. Available: http://www.ietf.org/rfc/rfc2866.txt. [Accessed: 02-Nov-2011].

[11] D. Studený, 'Eduroam accounting'. 27-Dec-2007.

[12] 'Controlling Peer to Peer (P2P) Traffic'. Enterasys secure Networks, 2011.

[13] 'Cisco IOS Security Configuration', Cisco Systems, Inc., Release 12.4, 2008.

[14] A. Zúquete, *Segurança em Redes Informáticas*, 3ª ed. FCA, 2010.

[15] 'Nmap - Free Security Scanner For Network Exploration & Security Audits.' [Online]. Available: http://nmap.org/. [Accessed: 24-Nov-2011].

[16] 'Tenable Nessus | Tenable Network Security'. [Online]. Available: http://www.tenable.com/products/nessus. [Accessed: 24-Nov-2011].

[17] 'OpenVAS - OpenVAS - Open Vulnerability Assessment System'. [Online]. Available: http://www.openvas.org/. [Accessed: 24-Nov-2011].

Anexos

Anexo A – Documentação do Sistema de Monitorização

- Opções de Monitorização
- Procedimentos Criados
- Especificação de Requisitos e Testes aos Event Handlers

Anexo B – Comandos Testados na monitorização

- Exemplos de comandos usados na Monitorização

Anexo C – Instalação do Nmap, Nessus e OpenVAS

- Descrição da instalação do Nmap, Nessus e OpenVAS

Anexo D – Documentação do Sistema de Controlo de Acessos

- Especificação de Requisitos e Arquitectura
- Manual de Instalação
- Manual de Utilizador
- Especificação de Testes

Anexo E – Documentação do Sistema de detecção de alteração de configurações

- Especificação de Requisitos e Arquitectura
- Manual de Instalação
- Manual de Utilizador
- Especificação de Testes

Anexo A – Documentação do Sistema de Monitorização Icinga

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Opções de Monitorização

Sistema de Monitorização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1. Opções em ficheiros de configuração

Para que se torne simples e intuitivo a adição de novas máquinas e serviços, foram tomadas um conjunto de opções sobre a monitorização. Uma delas, tem a ver com a organização dos ficheiro de configuração. O Icinga permite que se criem vários ficheiros, e se defina esses mesmos ficheiros como partes de uma configuração. O Daemon interpretará os mesmos de acordo com o tipo de objecto lá especificado. Na Figura 1 estão todos os ficheiros usados na configuração do sistema:

- Commands.cfg: definição de comandos usados na monitorização. Os plugins são invocados através destes comandos para simplificar.
- Contacts.cfg: neste ficheiro estão todos os contactos e grupos de contacto definidos para notificação. Estes foram coleccionados no inicio da configuração do sistema. Os grupos definidos têm por base o tipo de aplicação ou cargo a que cada gestor está atribuído.
- Timeperiods.cfg: ficheiro onde são definidos períodos de tempo fixos para notificação dos gestores ou verificação de máquinas e serviços.
- Templates.cfg: configurações base para definição de novo contacto, verificação de uma máquina ou serviço.
- Switches.cfg: definição dos activos de rede como routers , switches e pontos de acesso.
- Hosts.cfg: definição de servidores dos mais diversos serviços (mail, ldap, web, base de dados...).
- Host-groups.cfg: definição de grupos de máquinas que pertençam ao mesmo sistema ou serviço de rede.
- Services.cfg: definição de serviços associados a uma máquina.
- Service-groups.cfg: definição de grupos de serviço que pertençam ao mesmo tipo de serviço.

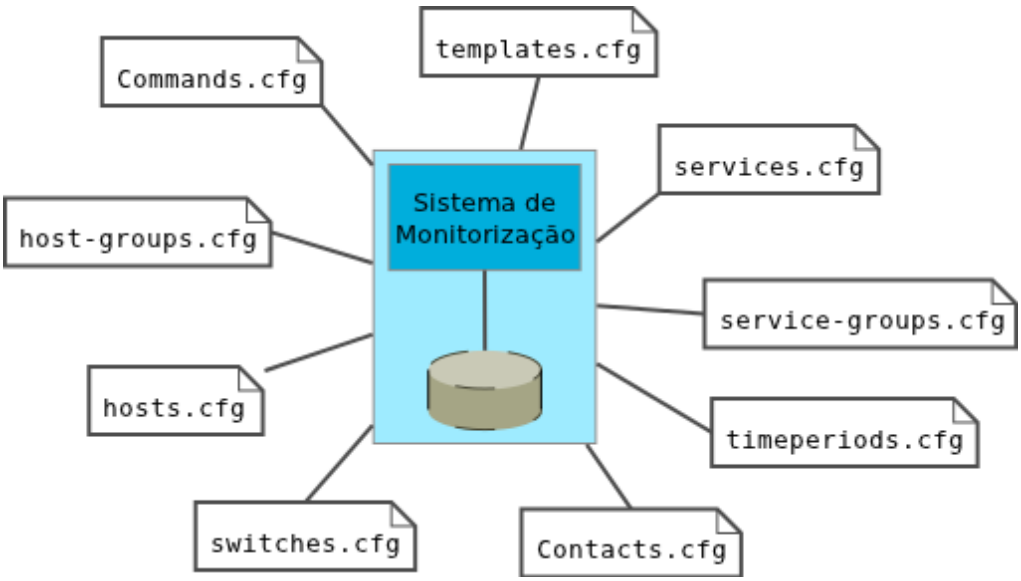

Figura 1: Ficheiros usados na configuração da monitorização

No entanto, dentro destes ficheiros há um conjunto de opções que têm de ser tomadas para uniformizar as configurações. As seguintes opções foram fixadas no ficheiro de templates:

- As notificações estão activas por defeito (notifications_enabled 1), tanto para máquinas como para serviços. Estas só funcionam se for definido um contacto na definição da máquina ou de um serviço.
- Os event handlers também estão activos por defeito (event_handler_enabled 1), uma vez que também só funcionam se for definido o comando para executar o mesmo.
- O número máximo de verificações para uma máquina é de 10 vezes. Para um serviço é de 4. Este valor é muito importante para os event handlers porque é com ele que se decidirá quando actuar sobre o serviço. Também é por este valor que as notificações se guiam. Só depois de

verificar o número máximo de vezes é que o gestor será notificado. Assim, pode-se programar a acção sobre o serviço à terceira tentativa de verificação falhada, funcionando a quarta e última tentativa como verificação. Caso não tenha sido sucedido, o gestor é realmente notificado.

- Também se tentou uniformizar o intervalo de verificações e reverificações. No caso da verificação das máquinas, com um simples ping, era efectuado de 5 em 5 minutos. No caso do serviço, de 10 em 10 minutos. Se no caso das máquinas não há problemas ou diferenças, no caso dos serviços estes têm de se ajustar de acordo com a complexidade e criticidade do serviço de rede. No entanto tem de haver sempre o cuidado de não sobrecarregar a rede ou o serviço com pedidos desnecessários.

2. Opções pelo complemento Business Process

Com vista a facilitar a percepção visual do estado da rede, pensou-se por representar visualmente apenas os activos de rede. Ou seja, no Statusmap (mapa da rede) do Icinga iriam aparecer só os routers, firewalls e switches. As restantes máquinas, que alojam os diferentes serviços, só iriam ser monitorizadas. Pensou-se esta opção primeiramente porque o que interessa é saber se o percurso até determinado serviço está ou não obstruído. Isto é, saber se a estrutura foi afectada. Em segundo para facilitar a primeira avaliação do problema, de forma a ataca-lo mais rapidamente. No entanto, tal não é possível, pois não existe nenhuma opção que desabilite a visualização de determindo host ou hostgroup, nos ficheiros de configuração. Existe, para o Icinga na sua interface clássica, uma opção que permite mostrar determinados grupos, mas para o Icinga Web não é ainda possível.

Sendo assim, para avaliar o estado de um serviço de rede, recorre-se a outra funcionalidade do Icinga: Business Process. Através desta, consegue-se montar a estrutura de um serviço como por exemplo adicionar os hosts e serviços que o compõem. Ou seja, com este componente montar-se-á a estrutura dos serviços de rede para mais fácil avaliação do seu estado pois este permite definir relações entre processos e serviços da rede. Assim, será mais fácil identificar que máquina estará a comprometer o serviço.

Esta foi uma solução adoptado no âmbito de um problema detectado ao longo da construção da estrutura de monitorização. No caso, era necessário monitorizar os serviços de rede por duas vias: rede de acesso e rede de gestão. Numa rede com esta dimensão, é necessário separar os acessos ao serviço, das actividades de gestão do serviço (backup, configurações...) a fim de não comprometer a qualidade de nenhum deles. Aqui, a monitorização assume um papel fulcral. Com a definição de Business Processes, consegue-se separar estas duas realidade e apresentar o estado delas independentemente.

3. Representação de duas interfaces na mesma máquina

O Icinga permite definir novas variáveis que poderão ser usadas nos comandos. Assim, para diferenciar a rede de acesso a serviços por parte dos utilizadores, da rede de gestão desses mesmos serviços, criou-se uma variável que irá conter o endereço IP. O nome da variável é '_gaddress' e pode ser definido um comando como se vê a seguir:

```
# 'check_ping_gestao' command definition
define command{
        command_name    check_ping_gestao
        command_line    $USER1$/check_ping -H $_HOSTGADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
        }
```

Procedimentos Criados

Sistema de Monitorização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

Aqui serão apresentados alguns dos procedimentos criados no âmbito do estágio

1. **Instalação e configuração do Icinga e componentes**

INSTALAÇÃO DO ICINGA

TEMPO ESTIMADO: 45 MIN
REQUISITOS
LINUX MINIMO
#

DOCUMENTACAO
http://docs.icinga.org/1.6/en/quickstart-idoutils.html
http://docs.icinga.org/1.6/en/icinga-web-scratch.html
http://docs.icinga.org/1.6/en/reporting_1.6.html
http://jasperforge.org/espdocs/download.php?filename=/opt/jasper/www/espdocs/Documents/112/v4.5%20Documentation/JasperReports-Server-CP-Install-Guide.pdf&ctype=application/pdf
http://docs.pnp4nagios.org/pnp-0.6/install

PREPARACAO
ssh xxx.xxx.xxx.xxx

yum install httpd gcc glibc glibc-common gd gd-devel
yum install libjpeg libjpeg-devel libpng libpng-devel
yum install net-snmp net-snmp-devel net-snmp-utils
yum install mysql mysql-server libdbi libdbi-devel libdbi-drivers libdbi-dbd-mysql
yum install php php-cli php-pear php-xmlrpc php-xsl php-pdo php-soap php-gd php-ldap php-mysql hp-soap php-pdo php-pgsql
yum install rrdtool-perl rrdtool

useradd -m icinga
groupadd icinga
groupadd icinga-cmd
usermod -a -G icinga-cmd icinga
usermod -a -G icinga-cmd apache

cd /usr/local/src/
ICINGA
wget http://downloads.sourceforge.net/project/icinga/icinga/1.6.1/icinga-1.6.1.tar.gz?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Ficinga%2F&ts=1331569276&use_mirror=garr
NAGIOS PLUGINS
wget http://downloads.sourceforge.net/project/nagiosplug/nagiosplug/1.4.15/nagios-plugins-1.4.15.tar.gz?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fnagiosplug%2Ffiles%2Fnagiosplug%2F1.4.15%2F&ts=1331294039&use_mirror=kent
ICINGA WEB
wget http://downloads.sourceforge.net/project/icinga/icinga-web/1.6.1/icinga-web-1.6.1.tar.gz?r=&ts=1331569383&use_mirror=ignum
ICINGA REPORTING
wget http://downloads.sourceforge.net/project/icinga/icinga-reporting/1.6.0/icinga-reports-1.6.0.tar.gz?r=&ts=1331569388&use_mirror=garr
JASPERSERVER
wget http://downloads.sourceforge.net/project/jasperserver/JasperServer/JasperServer

%204.5.0/jasperreports-server-cp-4.5.0-linux-x64-installer.run?r=http%3A%2F%2Fjasperforge.org%2Fuploads%2Fpublish%2Fjasperserverwebsite%2FJSJA%2520Website%2Fjs_download.html%3Fgroup_id%3D112&ts=1331569474&use_mirror=switch
PNP4Nagios
wget http://downloads.sourceforge.net/project/pnp4nagios/PNP-0.6/pnp4nagios-0.6.16.tar.gz?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fpnp4nagios%2Ffiles%2FPNP-0.6%2F&ts=1331569564&use_mirror=switch

DESCOMPACTAR
tar xvzf icinga-1.6.1.tar.gz
tar xvzf nagios-plugins-1.4.15.tar.gz
tar xvzf icinga-web-1.6.1.tar.gz
tar xvzf icinga-reports-1.6.0.tar.gz
tar xvzf pnp4nagios-0.6.16.tar.gz

INSTALACAO ICINGA
cd icinga-1.6.1
./configure --with-command-group=icinga-cmd --enable-idoutils --enable-ssl --with-icinga-user=icinga --with-icinga-group=icinga --with-web-user=apache --with-web-group=apache
make all
make install
make install-init
make install-config
make install-commandmode
make install-idoutils

CONFIGURACAO ICINGA
cd /usr/local/icinga/etc/
mv idomod.cfg-sample idomod.cfg
mv ido2db.cfg-sample ido2db.cfg

ALTERAR IDOMOD.CFG
nano idomod.cfg
#
...
use_ssl=1
output_type=tcpsocket
output=127.0.0.1
...
#

ALTERAR IDO2DB.CFG
nano ido2db.cfg
#
...
use_ssl=1
socket_type=tcp
...
#

cd module/
mv idoutils.cfg-sample idoutils.cfg

```
# PREPARAR MYSQL
service mysqld start
mysql -u root -p
#DENTRO DA BD
CREATE DATABASE icinga;
 GRANT USAGE ON *.* TO 'icinga'@'localhost'
   IDENTIFIED BY 'icinga'
   WITH MAX_QUERIES_PER_HOUR 0
   MAX_CONNECTIONS_PER_HOUR 0
   MAX_UPDATES_PER_HOUR 0;
 GRANT SELECT , INSERT , UPDATE , DELETE
   ON icinga.* TO 'icinga'@'localhost';
 FLUSH PRIVILEGES ;
 quit

# POPULAR BD
cd /usr/local/src/icinga-1.6.1/module/idoutils/db/mysql
mysql -u root -p icinga < mysql.sql

nano /usr/local/icinga/etc/ido2db.cfg
#
...
 db_servertype=mysql
 db_port=3306
 db_user=icinga
 db_pass=icinga
...
#

# INTERFACE WEB CLASSICA
cd /usr/local/src/icinga1.6.1/
make cgis
make install-cgis
make install-html
make install-webconf

# CRIACAO DO UTILIZADOR
htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin
service httpd restart

# PLUGINS
cd /usr/local/src/nagios-plugins-1.4.15
./configure --prefix=/usr/local/icinga --with-cgiurl=/icinga/cgi-bin --with-htmurl=/icinga --with-
nagios-user=icinga --with-nagios-group=icinga
make
make install

# SELINUX
setenforce 0
nano /etc/selinux/config
#
```

```
...
SELINUX= permissive
...
#

# INICIAR O ICINGA
service ido2db start
/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
service icinga start
chkconfig --add icinga chkconfig icinga on
chkconfig --add ido2db chkconfig ido2db on

# SE HOUVER PROBLEMAS A ACEDER A INTERFACE WEB!!
iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables-save

# INSTALACAO ICINGA WEB
cd /usr/local/src/icinga-web-1.6.1
./configure --prefix=/usr/local/icinga-web --with-web-user=apache --with-web-group=apache
--with-web-path=/icinga-web --with-web-apache-path=/etc/httpd/conf.d
make install
make install-apache-config
make install-done
# TESTAR DEPENDENCIAS PHP (TODOS OS REQUIRED)
make testdeps
nano /etc/php.ini
#
...
  magic_quotes_gpc = off
  safe_mode = off
  date.timezone=Europe/Lisbon
...
#

# BD ICINGA WEB
mysql -u root -p
CREATE DATABASE icinga_web;
      GRANT USAGE ON *.* TO 'icinga_web'@'localhost' IDENTIFIED BY 'icinga_web' WITH
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0;
      GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX ON
icinga_web.* TO 'icinga_web'@'localhost';
      quit
# POPULAR BD
make db-initialize

# LIMPAR A CACHE E REINICIAR HTTPD
/usr/local/icinga-web/bin/clearcache.sh
service httpd restart

# ICINGA REPORTS
cd /usr/local/src/
```

```
chmod +x jasperreports-server-cp-4.5.0-linux-x64-installer.run
# INSTALACAO COM APACHE E POSTGRESQL BUNDLED. TEMPLATES E IREPORT
./jasperreports-server-cp-4.5.0-linux-x64-installer.run
cd /opt/jasperreports-server-cp-4.5.0/
./ctlscript.sh start
cd /usr/local/src/icinga-reports-1.6.0
./configure --with-jasper-server=/opt/jasperreports-server-cp-4.5.0 --with-icinga-user=icinga --with-
icinga-group=icinga
make install-mysql-connector
/opt/jasperreports-server-cp-4.5.0/ctlscript.sh restart
make install
/opt/jasperreports-server-cp-4.5.0/ctlscript.sh restart

# INSTALACAO PNP4NAGIOS
cd /usr/local/src/pnp4nagios-0.6.16
./configure --with-nagios-user=icinga --with-nagios-group=icinga
make all
make install install-webconf install-config install-init

# CONFIGURACAO PNP4NAGIOS
cd /usr/local/pnp4nagios/etc/
cp rra.cfg-sample rra.cfg
nano npcd.cfg
# ALTERAR AS SEGUINTES LINHAS
...
log_type = file
log_level = -1
load_threshold = 10.0
user = icinga
group = icinga
log_file = /usr/local/pnp4nagios/var/npcd.log
perfdata_spool_dir = /usr/local/pnp4nagios/var/spool/
perfdata_file = /usr/local/pnp4nagios/var/perfdata.dump
...
#

su icinga
cd ../var/
touch npcd-log
touch perfdata.dump
touch perfdata.log
exit

nano process_perfdata.cfg
# ALTERAR A SEGUINTE LINHA
...
LOG_FILE = /usr/local/pnp4nagios/var/perfdata.log
...
#

nano config.php
# ALTERAR A SEGUINTE LINHA
```

```
...
$conf['nagios_base'] = "/icinga/cgi-bin";
...
#

nano /usr/local/icinga/etc/icinga.cfg
# ALTERAR AS SEGUINTES LINHAS
...
process_performance_data=1

host_perfdata_file=/usr/local/pnp4nagios/var/host-perfdata
service_perfdata_file=/usr/local/pnp4nagios/var/service-perfdata

service_perfdata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::
$TIMET$\tHOSTNAME::$HOSTNAME$\tSERVICEDESC::
$SERVICEDESC$\tSERVICEPERFDATA::
$SERVICEPERFDATA$\tSERVICECHECKCOMMAND::
$SERVICECHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::
$HOSTSTATETYPE$\tSERVICESTATE::$SERVICESTATE$\tSERVICESTATETYPE::
$SERVICESTATETYPE$

host_perfdata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNAME::
$HOSTNAME$\tHOSTPERFDATA::$HOSTPERFDATA$\tHOSTCHECKCOMMAND::
$HOSTCHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::
$HOSTSTATETYPE$

service_perfdata_file_mode=a
host_perfdata_file_mode=a

service_perfdata_file_processing_interval=30
host_perfdata_file_processing_interval=30

service_perfdata_file_processing_command=process-service-perfdata-file
host_perfdata_file_processing_command=process-host-perfdata-file
...
#

nano /usr/local/icinga/etc/objects/commands.cfg
# ADICIONAR AS SEGUINTES LINHAS
...
# pnp command definition
define command{
        command_name    process-service-perfdata-file
        command_line    /bin/mv /usr/local/pnp4nagios/var/service-perfdata
/usr/local/pnp4nagios/var/spool/service-perfdata.$TIMET$
}

define command{
        command_name    process-host-perfdata-file
        command_line    /bin/mv /usr/local/pnp4nagios/var/host-perfdata
/usr/local/pnp4nagios/var/spool/host-perfdata.$TIMET$
}
```

...
#

COPIAR OS TEMPLATES
cp /usr/local/src/icinga-web-1.6.1/contrib/PNP_Integration/templateExtensions/*.xml
/usr/local/icinga-web/app/modules/Cronks/data/xml/extensions
/usr/local/icinga-web/bin/clearcache.sh

RETIRAR AUTENTICAÇÃO PARA ACEDER AO PNP
nano /etc/httpd/conf.d/pnp4nagios.conf
COMENTAM-SE AS SEGUINTES LINHAS
...
#AuthName
#AuthType
#AuthUserFile
#Require valid-user
...
#

service npcd start
chkconfig --add npcd chkconfig npcd on
service httpd restart

PASSADOS ALGUNS SEGUNDOS JA HAVERA INFORMACAO
ls -ltr /usr/local/pnp4nagios/var/perfdata/localhost

ACEDE-SE A PAGINA WEB E CLICA-SE NUM DOS ICONES. DEVERA APARECER UM
RESUMO DA INSTALACAO
rm /usr/local/pnp4nagios/share/install.php

INSTALACAO E CONFIGURACAO DO NAGIOS BP

TEMPO ESTIMADO: 5 MIN
REQUISITOS
LINUX MINIMO
ICINGA
#

INICIO
ssh xxx.xxx.xxx.xxx
yum install perl-JSON-XS perl-CGI-Simple perl-CGI
cd /usr/local/src/
wget http://bp-addon.monitoringexchange.org/download/nagios-business-process-addon-0.9.6.tar.gz
tar xvfz nagios-business-process-addon-0.9.6.tar.gz
cd nagios-business-process-addon-0.9.6
./configure --with-nagiosbp-user=icinga --with-nagiosbp-group=icinga --with-
nagetc=/usr/local/icinga/etc --with-naghtmurl=/icinga --with-nagcgiurl=/icinga/cgi-bin --with-
apache-authname="Icinga Access"
make install

```
cd /usr/local/nagiosbp/
chown apache etc/

cd etc/
cp ndo.cfg-sample ndo.cfg
cp nagios-bp.conf-sample nagios-bp.conf
vi ndo.cfg
#...
ndo=db
ndodb_database=icinga
ndodb_username=<username>
ndodb_password=<password>
ndodb_prefix=icinga_
...

service httpd restart

cd /usr/src/icinga-web-1.6.1/contrib/businessprocess-icinga-cronk/
./install.sh
cd /usr/local/icinga-web/app/modules/BPAddon/config
vi bp.xml
#
...
<ae:parameter name="configTarget">/usr/local/nagiosbp/etc</ae:parameter>
<ae:parameter name="bin">/usr/local/nagiosbp/bin</ae:parameter>
...
#

# DEFINICAO DE UM HOST FICTICIO E COMANDO
cd /usr/local/icinga/etc/objects
vi hosts.cfg
#
...
define host{
        use             linux-server
        host_name       business_processes
        alias                Business Processes
        address         1.0.0.0
        contact_groups  admins
        check_command   check_dummy!0
        register        1
        }
...
#

vi commands.cfg
#
...
define command{
                        command_name    check_bp_status
                        command_line    /usr/local/nagiosbp/libexec/check_bp_status.pl -b $ARG1$ -f
$ARG2$
```

```
                }
...
#

# VERIFICACAO E REINICIO
/usr/local/icinga-web/bin/clearcache.sh
/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
service icinga restart
```

2. Instalação e Configuração do comanado NRPE

INSTALAÇÃO DO NRPE NO ICINGA

TEMPO ESTIMADO: 2 MIN
REQUISITOS
LINUX MINIMO
ICINGA
#

INICIO
ssh xxx.xxx.xxx.xxx
yum install gcc
cd /usr/local/src
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.13.tar.gz

tar xvfz nrpe-2.13.tar.gz

INSTALAÇÃO DO NRPE
cd nrpe-2.13
./configure --libexecdir=/usr/local/icinga/libexec/ --with-nrpe-user=icinga --with-nrpe-group=icinga
--with-nagios-user=icinga --with-nagios-group=icinga
make all
make install-plugin

TESTAR
/usr/local/icinga/libexec/check_nrpe -H <ip_de_um_cliente>
RESULTADO ESPERADO
NRPE v2.13
#

/usr/local/icinga/libexec/check_nrpe -H <ip_de_um_cliente> -c check_users
RESULTADO ESPERADO
USERS OK - ...
#

INSTALAÇÃO DO NRPE NUM CLIENTE

TEMPO ESTIMADO: 5 MIN
REQUISITOS
LINUX MINIMO
#

INICIO
ssh <ip_do_cliente>
yum install gcc xinetd

cd /usr/local/src
wget http://downloads.sourceforge.net/project/nagiosplug/nagiosplug/1.4.15/nagios-plugins-
1.4.15.tar.gz?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fnagiosplug%2Ffiles
%2Fnagiosplug%2F1.4.15%2F&ts=1331294039&use_mirror=kent

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.13.tar.gz

tar xvfz nagios-plugins-1.4.15.tar.gz
tar xvfz nrpe-2.13.tar.gz
useradd nagios

chown 777 /etc/sudoers
vi /etc/sudoers
# ADICIONAR O UTILIZADOR NAGIOS PARA CORRER COMANDOS EM MODO SUPER
UTILIZADOR
# SEM PASSWORD, NA DIRECTORIA EVENTHANDLERS. UTIL PARA EVENT HANDLERS
User_Alias NAGIOS = nagios,nagcmd
Cmnd_Alias NAGIOSCOMMANDS = /usr/local/nagios/libexec/eventhandlers/,
/usr/local/nagios/libexec/
Defaults:NAGIOS !requiretty
NAGIOS    ALL=(ALL)    NOPASSWD: NAGIOSCOMMANDS
#
chown 440 /etc/sudoers

# INSTALAÇÃO DO NAGIOS PLUGINS
cd nagios-plugins-1.4.15
./configure
make
make install

# INSTALAÇÃO DO NRPE
cd ../nrpe-2.13
./configure
make all
make install-plugin
make install-daemon
make install-daemon-config
make install-xinetd

# CONFIGURAÇÃO NRPE
vi /etc/xinetd.d/nrpe
# ADICIONAR O IP DA MAQUINA ICINGA A LINHA
only_from = 127.0.0.1 xxx.xxx.xxx.xxx
#

echo "nrpe 5666/tcp # nrpe" >> /etc/services
getenforce
# SE DER DIFERENTE DE DISABLED
setenforce 0
vi /etc/selinux/config
# ALTERAR A LINHA PARA QUE FIQUE PERMANENTE
SELINUX=disabled
#

service xinetd restart
netstat -at | grep nrpe
# RESULTADO ESPERADO
```

```
tcp 0    0 *:nrpe          *:*       LISTEN
#

# TESTAR
/usr/local/nagios/libexec/check_nrpe -H 127.0.0.1
# RESULTADO ESPERADO
NRPE v2.13
#


### CONFIGURACAO DE COMANDO NRPE - CLIENTE

# TEMPO ESTIMADO: 10 MIN
# REQUISITOS
LINUX MINIMO
"Instalacao do NRPE - CLIENTE"
#

ssh <ip_cliente> -l <login>
cd /usr/local/nagios/libexec/
# INSTALACAO DE NOVO PLUGIN
# !FACULTATIVO!
wget <url_plugin>

# DAR PERMISSÕES
chmod 755 <plugin>
chown nagios <plugin>

# CONFIGURACAO DO PLUGIN: CONSULTAR HELP PARA VER OS PARAMETROS
NECESSARIOS
./<plugin> --help

# ADICIONAR COMANDO AO NRPE
# COPIAR UM JA EXISTENTE E ALTERAR
# PODESER PRECISO PERMISSAO DE SUPER UTILIZADOR: ADICIONAR 'SUDO'
vi ../etc/nrpe.cfg
command[<check_plugin>]=/usr/local/nagios/libexec/eventhandlers/<plugin>
#
service xinetd restart

# TESTAR
./check_nrpe -H 127.0.0.1 -t 200 -c <check_plugin>
```

3. Configuração de Event handler

```
### CONFIGURACAO DO EVENT HANDLER COM SCRIPT - ICINGA

# TEMPO ESTIMADO: 10 MIN
# REQUISITOS
LINUX MINIMO
ICINGA
"Instalacao do NRPE - icinga"
"Instalacao do NRPE - cliente"
#

# NA MAQUINA ICINGA
ssh xxx.xxx.xxx.xxx
cd /usr/local/icinga/libexec/eventhandlers/
# EXEMPLO: SUBSTITUIR 'EVENT_HANDLER' POR OUTRO NOME
# A CODIFICACAO DO EVENT HANDLER REQUER CONHECIMENTO AVANCADO
touch event_handler.sh
vi event_handler.sh
chmod 755 event_handler.sh
chown icinga event_handler.sh
# O EVENT HANDLER SERA EXECUTADO USADO O NRPE
# EXEMPLO
/usr/local/icinga/libexec/check_nrpe -H $4 -t 200 -c restart_$5
#

# TESTAR
../check_nrpe -H <ip_do_cliente> -t 200 -c restart_<script_definido_no_cliente>
# OU CORRER O EVENT HANDLER DIRECTAMENTE PASSANDO-LHE AS VARIÁVEIS
NECESSARIAS
./event_handler.sh OK SOFT 1 <ip_cliente> <sufixo_script_definido_no_cliente>

# ADICIONAR SERVICO
cd ..
vi ../etc/objects/commands.cfg
# ADICIONAR O COMANDO PARA O EVENT HANDLER
define command {
      command_name     restart-script
      command_line     /usr/local/icinga/libexec/eventhandlers/event_handler.sh $SERVICESTATE$
$SERVICESTATETYPE$ $SERVICEATTEMPT$ $HOSTADDRESS$ $ARG1$
'$SERVICEOUTPUT$'
}
#

vi ../etc/objects/services.cfg
# ADICIONAR O EVENT HANDLER AO SERVICO E O NUMERO DE REPETICOES
define service{
       ...
       max_check_attempts     4
       event_handler          restart-script!<sufixo_script_definido_no_cliente>
       }
#
```

/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
service icinga start

CONFIGURACAO DO EVENT HANDLER COM SCRIPT

TEMPO ESTIMADO: 10 MIN
REQUISITOS
LINUX MINIMO
"Instalacao do NRPE - CLIENTE"
#

ssh <ip_cliente>
CRIAR O SCRIPT
su nagios
vi /usr/local/nagios/libexec/eventhandlers/restart_<script>.sh
EXEMPLO: SUBSTITUIR POR AQUILO QUE REALMENTE E QUER !!!
service httpd restart
#
chmod 777 <script>.sh

vi /usr/local/nagios/etc/nrpe.cfg
# ADICIONAR COMANDO AO NRPE
# POR UMA QUESTAO DE UNIFORMIZACAO, TODOS OS COMANDOS COMECAM POR "RESTART_"
command[restart_script]=sudo /usr/local/nagios/libexec/eventhandlers/restart_<script>.sh
#
service xinetd restart

EM MAQUINAS DE PRODUCAO NAO FAZER O SEGUINTE!
TESTAR COM HTTPD PARA O EXEMPLO
service httpd stop
VERIFICAR QUE ESTA PARADO
service httpd status

./check_nrpe -H 127.0.0.1 -t 200 -c restart_script

VERIFICAR QUE REINICIOU
service httpd status

4. Configuração de novas máquinas e serviços

```
### CONFIGURACAO DE NOVAS MAQUINAS E SERVICOS - ICINGA

# TEMPO ESTIMADO: 15 MIN
# REQUISITOS
LINUX MINIMO
INCTALACAO DO ICINGA

# INICIO

ssh xxx.xxx.xxx.xxx -l <login>
cd /usr/local/icinga/libexec/

# INSTALACAO DE NOVO PLUGIN E CONFIGURACAO DE COMANDO
# !FACULTATIVO!
wget <url_plugin>

# DAR PERMISSÕES
chmod 755 <plugin>
chown icinga <plugin>

# CONFIGURACAO DO PLUGIN: CONSULTAR HELP PARA VER OS PARAMETROS
NECESSARIOS
./<plugin> --help

# COMANDO
vi ../etc/objects/commands.cfg
# COPIAR UM COMANDO EXISTENTE E ALTERAR. EX:

define command{
command_name        check_ftp
command_line        $USER1$/check_ftp -H $HOSTADDRESS$ $ARG1$ -w 20 -c 40
}
define command{
command_name        check_novo_plugin
command_line        $USER1$/<plugin> -H $HOSTADDRESS$ $ARG1$ -w 20 -c 40
}

# CONFIGURACAO DE HOST/SWITCH E GRUPOS
# COPIAR UMA CONFIGURACAO JA EXISTENTE E ALTERAR
vi ../etc/libexec/hosts.cfg
#
define host{
use             linux-server # Para usar do template
host_name       <nomes_das_máquinas separados por virgulas>
address             <ip>
contact_groups      <grupos de contacto separados por virgula>
hostgroups      <grupo de maquinas>
}
#

# SE JA EXISTIR O GRUPO, BASTA ADICIONAR A MAQUINA AO ATRIBUTO 'MEMBERS'
```

```
vi ../etc/libexec/groups.cfg
#
define hostgroup{
hostgroup_name        <grupo de maquinas>
alias             <Grupo de maquinas do xpto>
members               <nomes_das_máquinas separados por virgulas>
}
#

# CONFIGURACAO DE SERVICOS E GRUPOS
# COPIAR UMA CONFIGURACAO JA EXISTENTE E ALTERAR
# ATRIBUTO HOST_NAME PODE SER HOSTGROUP
vi ../etc/libexec/services.cfg
#
define SERVICE{
use                   generic-service
host_name             <nomes_das_máquinas separados por virgulas>
service_description   <ip>
check_command             check_novo_plugin
contact_groups            <grupos de contacto separados por virgula>
process_perf_data     1
}
#

# SE JA EXISTIR O GRUPO, BASTA ADICIONAR A MAQUINA E SERVICO AO ATRIBUTO
'MEMBERS'
vi ../etc/libexec/service_groups.cfg
#
define servicegroup{
hostgroup_name        <grupo de maquinas>
alias             <Grupo de máquinas do xpto>
members               <<nome da maquina, nome do servico> separados por virgulas>
}
#

# VERIFICAR SE TEM ERROS E REINICIAR SERVICO
/usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
service icinga restart

# ACEDER AO ICINGA E VERIFICAR (SERVICOS PODEM ESTAR PENDENTES)
```

Especificação de Requisitos e Testes aos Event handlers

Sistema de Monitorização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1 Avaliação e afinação dos parâmetros de verificação:

O tratamento de serviços através de event handlers, obedeceu sempre a 2 fases: definição de serviços, avaliação e ajuste dos parâmetros e uma segunda que era a programação do event handler em si consoante as informações recolhidas na primeira fase.

Esta primeira fase permite-nos então chegar aos requisitos do event handler – a forma como este irá reagir perante um estado do serviço. Para tal, foram definidos uma série de procedimentos a executar tanto nas máquinas a monitorizar como na máquina do Icinga.

Primeiro, tem de haver um conjunto de reuniões onde são debatidas as máquinas e os serviços a monitorizar, contactos dos gestores para notificações, validação, numa máquina (cliente), de uma série de procedimentos necessários às verificações e aos event handlers – Instalação do NRPE, configuração do acesso a serviços (bases de dados), eventual configuração de plugins do lado do cliente e configuração dos scripts necessários aos event handlers – e apresentação do plano de monitorização (plugins a usar, tempos de verificação e acessos à máquina).

Depois das reuniões haverá sempre uma fase de testes onde são afinados os tempos de verificação dos serviços. Esta parte é importantíssima para os event handlers pois permite acompanhar o comportamento de determinado serviço e as mensagens de erro geradas. Um tempo de verificação mal definido e poderemos cair no erro de, por exemplo, estar a sobrecarregar o serviço com sucessivos reinícios. Assim como a não avaliação de uma mensagem de erro e poderemos estar a interromper o serviço a meio de uma operação crítica (um backup por exemplo).

Os exemplos seguintes, foram todos desenvolvidos em Python.

2 Resolução de problemas num servidor web

Este é um dos exemplos onde foram testados os event handlers. Trata-se de um servidor web com 3 instâncias de apache e uma base de dados postgres.

2.1 Requisitos:

Este script será usado para recuperar serviços que não estejam dentro dos padrões de QoS definidos. Isto é, quando um serviço se encontra em baixo ou demora algum tempo a responder, o estado desse mesmo serviço, segundo o Icinga, passa de OK para Warning, Unknown ou Critical. Como queremos fazer algo genérico, este Eventhandler receberá 6 parâmetros de entrada:

- estado do serviço (OK, WARNING, UNKNOWN e CRITICAL);
- tipo de estado (SOFT ou HARD);
- contador incremental de tentativas de verificação;
- o endereço IP da máquina que aloja o serviço;
- o nome do serviço (será o sufixo do comando nrpe a executar na máquina);
- o output do serviço.

Estes permitirão, em alguns casos, definir tratamentos específicos de alguns serviços.Nos estados OK e WARNING o eventhandler não actuará deixando o serviço correr normalmente.

O estado UNKNOWN é usado para tratar os serviços postgresql. O plugin que verifica o serviço postgres identifica um estado crítico como UNKNOWN (daí o uso desta flag). O estado CRITICAL é usado para tratar os restantes serviços. Ambos só actuam se há terceira tentativa de verificação, o serviço ainda permanecer neste estado, fazendo uma verificação a seguir para comprovar o reinício do serviço. Para além disto há outro problema, com serviços de tomcat. Os serviços de tomcat, fazem backup da informação a uma determinada hora. Para que o consigam fazer sem interrupções, os gestores de redes definem um nível de serviço que não permita o acesso aos utilizadores, gerando um código de erro html 503. Ou seja, no caso de ser um serviço tomcat neste estado crítico, é também verificado o código de erro devolvido pelo plugin de verificação. Caso seja diferente de 503 (500, Connection timeout...) este é reiniciado.

Requisito	Prioridade
Não fazer nada nos estados OK e WARNING	Must
Reiniciar serviços ao fim de 3 tentativas de verificação (SOFT) em estados UNKNOWN e CRITICAL	Must
Reiniciar serviços se estiver no tipo de estado HARD em UNKNOWN e CRITICAL	Must
Tratar estados UNKNOWN e CRITICAL de igual forma	Must
Não fazer nada se o serviço for Tomcat e der erro 503 (Backup)	Must

2.2 Testes:

Os testes foram realizados, variando os parâmetros de entrada do scritpt.

```
./restart-service-handler.sh <estado> <tipo> <contador> <ip> <serviço> <output>
```

1.
Comando: ./restart-service-handler.sh OK SOFT 3 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

2.
Comando: ./restart-service-handler.sh WARNING SOFT 3 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

3.
Comando: ./restart-service-handler.sh OK HARD 4 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

4.
Comando: ./restart-service-handler.sh WARNING HARD 4 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

5.
Comando: ./restart-service-handler.sh UNKNOWN SOFT 1 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

6.
Comando: ./restart-service-handler.sh UNKNOWN SOFT 2 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

7.
Comando: ./restart-service-handler.sh UNKNOWN SOFT 3 127.0.0.1 http 'http OK'
Critério: Reinicia o serviço
Saída: passou

8.
Comando: ./restart-service-handler.sh CRITICAL SOFT 1 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

9.
Comando: ./restart-service-handler.sh CRITICAL SOFT 2 127.0.0.1 http 'http OK'
Critério: Não faz nada
Saída: passou

10.
Comando: ./restart-service-handler.sh CRITICAL SOFT 3 127.0.0.1 http 'http OK'
Critério: Reinicia o serviço
Saída: passou

11.
Comando: ./restart-service-handler.sh CRITICAL SOFT 3 127.0.0.1 tomcat 'CRITICAL: error 503 found'
Critério: Tomcat está em backup! Não faz nada.
Saída: passou

12.
Comando: ./restart-service-handler.sh CRITICAL SOFT 3 127.0.0.1 tomcat 'CRITICAL: error 500 found'
Critério: Reinicia o serviço
Saída: passou

13.
Comando: ./restart-service-handler.sh CRITICAL HARD 4 127.0.0.1 tomcat 'CRITICAL: error 503 found'
Critério: Tomcat está em backup! Não faz nada.
Saída: passou

14.
Comando: ./restart-service-handler.sh CRITICAL HARD 4 127.0.0.1 tomcat 'CRITICAL: error 500 found'
Critério: Reinicia o serviço
Saída: passou

3 Resolução de problemas num servidor de bases de dados

Este exemplo é um pouco diferente do anterior, pois actua directamente a partir do plugin, sobre um script responsável pelo armazenamento dos registos de uma aplicação numa base de dados.

3.1 Requisitos:

O plugin desenvolvido, teve por base um script que estava a ser usado. No entanto, para que devolvesse o estado para o icinga, foi necessário alterá-lo. Assim, consoante o estado do serviço recebido pelo icinga, este podia notificar o gestor.

Este plugin funcionará de forma directa. Fazendo uma chamada ao sistema operativo, verifica se o script a monitorizar está a correr. Caso contrário reinicia-o logo. Do lado do Icinga as notificações foram programadas caso a primeira reverificação devolvesse um estado diferente de OK.

O plugin recebrá 2 parâmetros:

- caminho para o compilador usado na execução do script;
- caminho para o script.

Assim, conseguiu-se que o plugin fosse genérico – aplicável a mais scripts a correr nessa máquina – para além de se ter a certeza que o script verificado é realmente aquele e que a sua reinicialização é feita com o compilador adequado.

Requisito	Prioridade
Reiniciar o script quando não for encontrado no sistema	Must
Só notificar o gestor quando não conseguir reiniciar, devolvendo estado CRITICAL	Must
Passar o caminho para o compilador pois este pode ser específico	Must
Script genérico	Must

3.2 Testes:

scutu

Os testes serão realizados, com o script real e com um script de testes.

```
Python check_script.py -c <compilador> -s <script a verificar>
```

1.
Comando: python check_script.py -c python -s test_script.py
Critério: script estava em baixo
Saída: reiniciado

2.
Comando: python check_script.py -c python -s test_script.py
Critério: script estava em baixo na segunda tentativa (linha para o reiniciar comentada no plugin)
Saída: gestor notificado

2.
Comando: python check_script.py -c python -s test_script.py
Critério: script estava a correr
Saída: devolução de estado OK para o Icinga

Anexo B – Comandos Testados na monitorização

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Exemplos de comados usados na Monitorização

Sistema de Monitorização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

Aqui vão ser descritos alguns comandos testados para monitorizar um host. Inicialmente haverá a alteração do ficheiro commands.cfg, para a criação de novos comandos. Depois, surge a criação de hosts e serviços, respectivamente hosts.cfg e services.cfg. Assim como se podem criar grupos de serviços ou grupos de hosts.

Por fim, há a adição destes ficheiros de configuração ao ficheiro de configuração principal do Icinga (Icinga.cfg).

Na directoria ~/icinga/libexec/, são instalados os plugins base. É para esta directoria que irão os plugins que forem criados ou instalados à parte.

No sentido de criar mecanismos automáticos que informem o Daemon do estado dos hosts e serviços, são criados no ficheiro commands.cfg alguns comandos base. Estes comandos têm por base o uso dos plugins e a definição de quotas de alerta. Se se aceder à pasta ~/icinga/libexec, pode-se saber o conjunto de flags que cada comando suporta para configuração. Por exemplo: ./check_ldap -h. Inclusivamente, se podem testar os comandos com as respectivas flags para se poder avaliar o resultado. Todos os comandos têm a flag *-w* e *-c*. Estas definem os níveis *warning* e *critical* do serviço.

O primeiro comando de exemplo irá chamar-se check_ldap, tal como o plugin. Este irá verificar o servidor ldap e são definidos as flags de warning e critical. Quando a resposta for igual ou superior a 3 segundos será gerado um warning, quando for igual ou superior a 5 será gerado um aviso crítical. A flag -H define o Host a ser monitorizado. Aquele ao qual stá atribuído o comando:

```
define command {
	command_name	check_ldap
	command_line	$USER1$/check_ldap -b dc=ci,dc=uc,dc=pt -w 3 -c 5 -H $HOSTADDRESS$
}
```

Um segundo comando de exemplo, será para a notificação por email. Este, por exemplo já vem predefinido no ficheiro. Serve para definir o corpo da mensagem de notificação, que será enviado aos admistradores agregados ao host:

```
define command{
	command_name	notify-host-by-email
	command_line /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" |
/bin/mail -s "** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **"
$CONTACTEMAIL$
	}
```

Em terceiro, a definição do comando check_icmp. Este é muito parecido ao check_ping (o ping é feito sobre o protocolo icmp), mas mais poderoso pois permite definir o tamanho dos pacotes a enviar. Neste caso, é definido como nível warning 2 segundos ou 80% de perda de pacotes e, como nível critical 3 segundos ou 100% de perdas de pacotes. Serão enviados 5 pacotes de 4096 bytes.

```
define command{
	command_name	check-host-alive
	command_line	$USER1$/check_icmp -H $HOSTADDRESS$ -w 2000.0,80% -c 3000.0,100%
-n 5 -b 4096
	}
```

O quarto exemplo, serve para verificar o número de processos activos no host. Neste exemplo, os parâmetros de warning e critical, serão definidos como argumentos que virão da definição dos serviços e serão activados caso algum processo consuma um valor superior de CPU:

```
define command{
```

```
    command_name    check_local_procs
    command_line    $USER1$/check_procs -w $ARG1$ -c $ARG2$ --metric=CPU
    }
```

O quinto exemplo, serve para verificar serviços locais em máquinas remotas ou realizar acções sobre os mesmos a partir do Icinga. Para isso, é necessário ter o daemon NRPE a correr e os comandos que serão usados, configurados no ficheiro de configuração do NRPE na máquina remota. Dio lado do Icinga é executado apenas o plugin do NRPE com o comando:

```
define command{
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
    }
```

O comando para verificar as bases de dados mysql também foi configurado. Como argumento,receberá o tipo de acção a realizar sobre a base de dados:

```
define command{
    command_name    check_mysql_health
    command_line    $USER1$/check_mysql_health --hostname $HOSTADDRESS$ $ARG1$
    }
```

Para verificação da base de dados postgres foi configurado um comando com o respectivo plugin. Este receberá por argumento o tipo de acção a realizar sobre a base de dados:

```
define command{
    command_name    check_pgsql
    command_line    $USER1$/check_postgres.pl -H $HOSTADDRESS$ $ARG1$
    }
```

A configuração do event handler também evolve a criação do comando responsável pela execução do script. Este é um dos exemplos testados em que o event handler receberá o estado do serviço, o tipo de estado, o número de tentativas, o endereço IP da máquina, o nome do serviço e o output do mesmo:

```
define command {
    command_name    service-handler
    command_line    /usr/local/icinga/libexec/eventhandlers/service-handler.sh $SERVICESTATE$
$SERVICESTATETYPE$ $SERVICEATTEMPT$ $HOSTADDRESS$ $AR
G1$ '$SERVICEOUTPUT$'
}
```

O ficheiro seguinte, será o ficheiro de hosts. Poderão ser vários, onde poderemos agrupar os hosts de diversos tipos, para melhor organização. Poderá fazer uso de configurações predefinidas que constem do ficheiro *templates.cfg*:

```
define host{
        use                     linux-server
        host_name               localhost
        address                 127.0.0.1
        contact_groups          admins
        hostgroups              ldap-servers
        }
```

De seguida, o ficheiro de configuração de serviços. Estes também podem usar configurações predefinidas no ficheiro de templates. O comando a ser executado, pode ser atribuído a um host ou, por questões de simplicidade, a um hostgroup. Um exemplo com o comando *check_ldap*:

```
define service{
        use                     generic-service
        hostgroup_name          ldap-servers
        service_description     LDAP-Response
        check_command           check_ldap
        }
```

E outro com o *check_icmp*:

```
define service{
        use                             generic-service ;
        host_name                       localhost
        service_description             PING
        check_command                   check_icmp
        normal_check_interval           5
        retry_check_interval            1
        process_perf_data               0
        retain_nonstatus_information    0
        contact_groups                  admins
        }
```

Por fim, a definição de hostgroups e servicegroups ajuda a organizar e a atribuir serviços. Permite agrupar os hosts ou serviços que tenham as mesmas funções ou que em conjunto implementem um serviço. Isto simplifica o visionamento da informação na interface web. Um hostgroup:

```
define hostgroup{
      hostgroup_name        ldap-servers
      alias                 Servidores LDAP
      members               localhost
      }
```

Um service group:

```
define servicegroup{
      servicegroup_name     ldap
      alias                 Servico ldap
      members               localhost
}
```

Anexo C – Instalação do Nmap, Nessus e OpenVAS

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Descrição da instalaçao do Nmap, Nessus e OpenVAS

Ferramentas de análise de vulnerabilidade

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

Instalação do Nmap em Ubuntu:

Na linha de comando executar os dois comandos:

```
> sudo apt-get install nmap
> sudo apt-get install zenmap
```

Para executar pelo zenmap, basta escrever *sudo zenmap* na linha de comando. Em super utilizador, pois assim, retira mais informação dos scans.

Instalação do Nessus em Ubuntu:

Ir até http://www.nessus.org/products/nessus/select-your-operating-system. Seleccionar o pacote debian pretendido (x32 ou x64) e fazer download.

Na linha de comando fazer: sudo dpkg -i nessus.deb.

Será também necessário adicionar um utilizador para aceder ao serviço do Nessus e autoriza-lo como admin:

```
> /opt/nessus/sbin/nessus-adduser
```

Falta criar uma conta no site para poder ter acesso aos plugins. Depois, com a chave basta executar o comando:

```
> /opt/nessus/bin/nessus-fetch --register XXXX-XXXX-XXXX-XXXX-XXXX
```

Por último, executa-se o serviço:

```
> sudo service nessusd start
```

Para aceder, basta abrir um browser e ir a: https://ip_maquina:8834

Instalação do OpenVAS em Ubuntu:

Configuração do repositório para instalação via apt-get:

```
> sudo apt-get -y install python-software-properties
> sudo add-apt-repository "deb
http://download.opensuse.org/repositories/security:/OpenVAS:/STABLE:/v4/xUbuntu_10.04/ ./"
> sudo apt-key adv --keyserver hkp://keys.gnupg.net --recv-keys BED1E87979EAFD54
> sudo apt-get update
```

Instalação do OpenVAS e respectivas dependencias:

```
> sudo apt-get -y install greenbone-security-assistant gsd openvas-cli openvas-manager openvas-
scanner openvas-administrator sqlite3 xsltproc
```

Por fim, inicia-se o servidor:

```
> test -e /var/lib/openvas/CA/cacert.pem  || sudo openvas-mkcert -q
> sudo openvas-nvt-sync
> test -e /var/lib/openvas/users/om || sudo openvas-mkcert-client -n om -i
```

```
> sudo /etc/init.d/openvas-manager stop
> sudo /etc/init.d/openvas-scanner stop
> sudo openvassd
> sudo openvasmd --migrate
> sudo openvasmd --rebuild
> sudo killall openvassd
> sleep 15
> sudo /etc/init.d/openvas-scanner start
> sudo /etc/init.d/openvas-manager start
> sudo /etc/init.d/openvas-administrator restart
> sudo /etc/init.d/greenbone-security-assistant restart
> test -e /var/lib/openvas/users/admin || sudo openvasad -c add_user -n admin -r Admin
```

A primeira vez que executar a última linha, será pedido que se insira uma password. Para aceder ao serviço do openvas, basta digitar *gsd* numa linha de comando.

Anexo D – Documentação do Sistema de Controlo de Acessos

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Especificação de Requisitos e Arquitectura

Sistema de Contabilização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1 Descrição do sistema actual

O sistema actual já se encontra bastante completo. É composto por duas partes distintas: um core que trata da agregação e tratamento da informação recolhida do servidor RADIUS e DHCP e uma interface web que disponibiliza essa informação de forma mais legível.

O core recebe e trata os pacotes contabilização, agregando o endereço IP a cada um deles, juntamente com outro tipo de informação: tempo/consumo total da ligação, tempo/consumo total do presente mês, IP, email do utilizador, AP onde se ligou e a data de ligação.

A interface web mostra uma parte desta informação de forma organizada. Podemos ver todos os utilizadores que já estiveram ligados, a que tipo de rede (eduroam, guest), utilizadores que estão online, os utilizadores que estão em mobilidade ligados nos APs da UC e estatísticas sobre a utilização da rede por Pólos, Faculdades ou APs.

2 Novo sistema a implementar

O novo sistema, que será apenas um complemento ao já existente, deverá permitir pesquisar a quem pertenceu determinado IP em determinado período, com base em parâmetros bem definidos. Este terá duas partes distintas que trabalharão em conjunto para alcançar o objectivo.

A primeira parte, é uma interface web onde um utilizador poderá colocar um conjunto de dados que serão necessários à pesquisa. Data e hora de acesso, timezone e IP serão os campos necessários e obrigatórios para que a pesquisa se efectue. Todos estes dados serão obtidos de queixas efectuadas por entidades superiores e endereçadas ao GSIIC.

A segunda parte será um middleware que fará o tratamento dos dados de Accounting em bruto, pesquisando com os dados fornecidos através da interface web.

Como o sistema é para ser usado por pessoas com conhecimento informático médio e fraco conhecimento ao nível de redes de comunicação, a interface web terá de ser o mais intuitiva possível - tanto na inserção dos dados como na apresentação dos resultados.

3 Requisitos do novo sistema

ID	Descrição	Prioridade
req1	Definição da data de acesso (dia, mês e ano assim como hora, minuto e segundo)	Must
req2	Definição do IP	Must
req3	Definiçao da Timezone	Must
req4	Todos os campos anteriores são obrigatórios	Must
req6	Apresentação da informação da sessão (utilizador, Mac Address e inicio e fim de sessão)	Must

Table 1: Requisitos do sistema a implementar

4 Análise dos logs

Para que seja possível desenvolver este sistema, foi feita uma análise dos logs do sistema RADIUS que seriam usados:

- linhas que iniciam por '+' são conexões e as que iniciam com '-' são desconexões;
- Podem existir conexões sem IP logo, estes serão descartados;
- Podem existir conexões sem desconexões associadas;
- Podem existir desconexões sem conexões associadas;
- A conexão pode vir depois da respectiva desconexão;

- No campo U[...] está o nome do utilizador;
- No campo @[...] está o endereço ip e o nome do ponto de acesso;
- No campo SSID[...] está o nome da rede a que o utilizador se ligou;
- No campo IP[...] está o IP atribuido ao utilizador;
- No campo C[...] está o consumo, em Bytes, da sessão;
- No campo T[...] está o tempo de ligação em segundos;
- No campo CT[...] está o consumo total, em Bytes, desde o início do mês;
- No campo TT[...] está o Tempo total de ligação, em segundos, desde o iníciodo mês.

```
+.1x: 120811-22:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] SSID[eduroam] IP[111.222.333.4]
-.1x: 120812-00:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] C[1458889]  T[7200] TT[3588] CT[7785466]
```

O quinto caso (a conexão pode vir depois da respectiva desconexão) irá conferir um grau de complexidade mais elevado ao nosso algoritmo. À medida que o ficheiro é lido, irão ser guardadas as desconexões para, no fim, caso a desconexão seja desconhecida, podermos verificar qual dos 3 tipos de fecho de ligação nos é apresentado:

- conhecido
- desconhecido
- não existe

Com base nos pressupostos anteriores, chegou-se a um conjunto de casos típicos que se poderão encontrar:

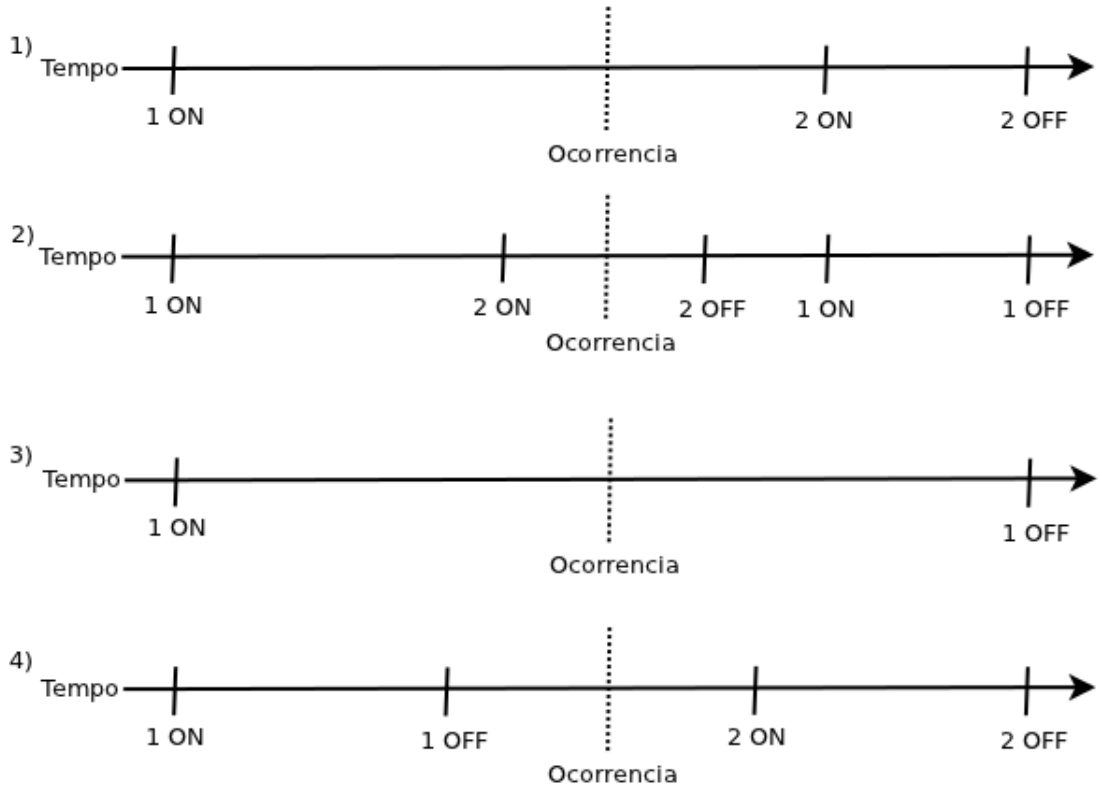

Figura 1: Casos especiais

1) A ligação é atribuída a 1 com conexão conhecida e desconexão desconhecida
2) A ligação é atribuída a 2 com conexão e desconexão conhecidos
3) A ligação é atribuída a 1 com conexão e desconexão conhecidos
4) Não existe ligação neste período

5 Casos de Uso

O único actor do sistema é identificado como "Utilizador web". É ele o responsável por interagir com a aplicação a fim de esta lhe devolver um resultado. O caso de uso da Figura 2 descreve o tipo de interacções que o actor pode exercer sobre o sistema.

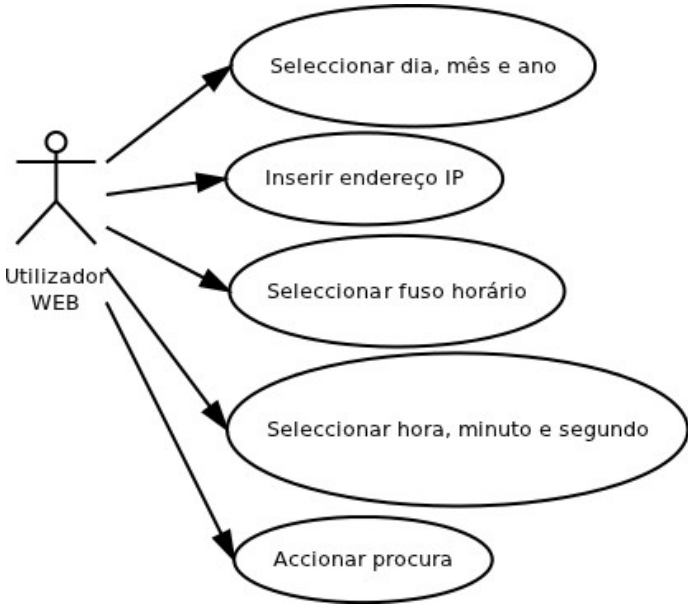

Figura 2: caso de uso do sistema, com o único actor do mesmo e as possíveis interacções.

O fluxo do sistema (Figura 3) é bastante simples. Depois de o Utilizador web inserir todos os dados, acciona a procura. A aplicação web trata os dados de entrada e constrói o comando responsável por invocar a aplicação de procura. Depois fica à espera do resultado para, por fim, o apresentar.

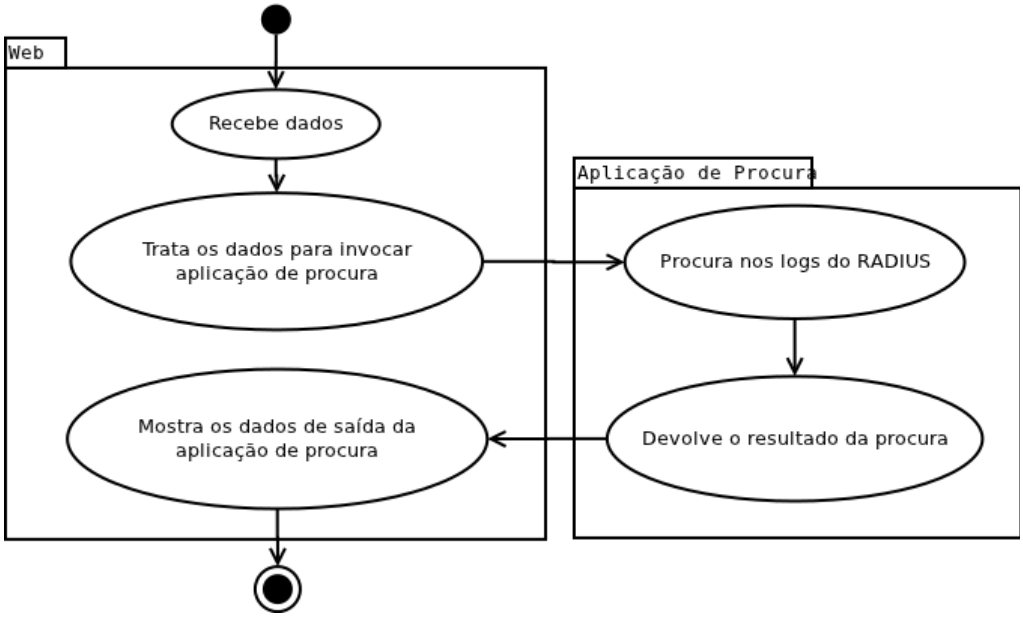

Figura 3: Fluxo do sistema desde a aquisição dos dados até à sua apresentação.

6 Arquitectura do sistema

6.1 Objectivos

O objectivo deste sistema, é a simplicidade de processos. É necessária uma aplicação que funcione localmente, usando o mínimo de recursos possível mesmo sabendo que os ficheiros de informação disponíveis são extremamente grandes e complexos.

6.2 Arquitectura Geral

A aplicação é composta por 2 componentes (Figura 4):

- interface web – desenvolvida em html e php que comunicará com o middleware através de chamadas ao sistema operativo.
- Aplicação de procura – desenvolvida em python que receberá 3 parâmetros essenciais à procura (Data e hora, fuso horário e endereço IP) e será a responsável pelo tratamento dos dados contidos nos ficheiros de informação de acesso.

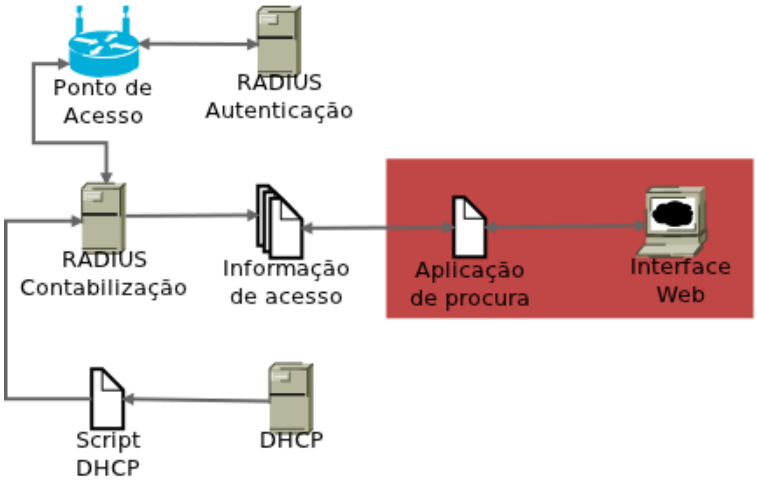

Figura 4: Arquitectura geral do sistema. A vermelho, os componentes a implementar.

6.3 Visão Geral

A solução encontrada para a implementação do sistema é composta por três camadas (Figura 5):

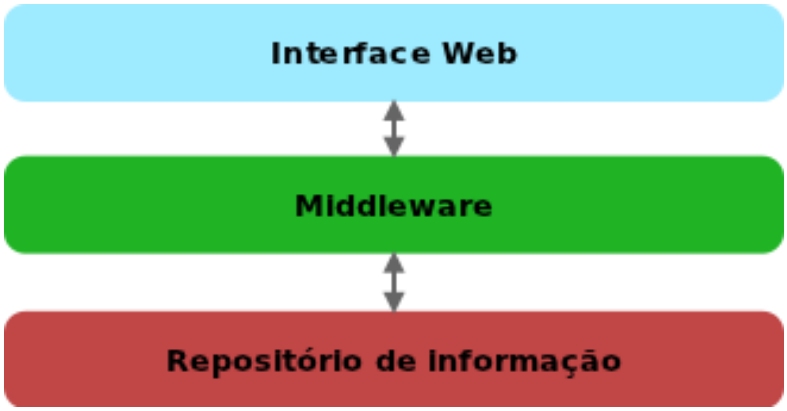

Figura 5: Modelo de três camadas usado no sistema.

A camada de Interface web é a responsável pela interacção com o utilizador. A camada Middleware, trata a informação do repositório de informação consoante os dados disponibilizados pela interface web. Por último, o repositório de informação, são os ficheiros de texto que agregam toda a informação de contabilização.

Manual de Instalação

Sistema de Contabilização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

O presente documento pretende ser uma descrição pormenorizada da instalação e configuração do sistema.

Requisitos mínimos do sistema:

- Sistema operativo fedora
- php 5.1.6
- python 2.4
- servidor RADIUS a correr

Tempo: 15 minutos

Instalação:

A instalação do presente sistema é bastante simples, bastando copiar 2 ficheiros para pastas distintas. O procedimento segue a seguir:

```
ssh <ip da maquina do serviço Radius>
# Ir até à pasta dos logs do Radius
cd /(pasta ou caminho)
# Copiar o ficheiro procura_ip.py
wget http://endereço.do.alojamento.do.programa

# Alterar o caminho para o ficheiro, no ficheiro
vi procura_ip.py
# procurar por FILE
/FILE
# alterar a variável para o caminho pretendido e guardar
# testar: deverá devolver o help
python procura_ip.py

# Ir até à pasta do servidor web
cd /var/www/html
# criar uma pasta e entrar nela
mkdir <nome_da_pasta>
cd <nome_da_pasta>
# copiar os ficheiros necessários
wget http://endereço.do.alojamento.do.index.php
wget http://endereço.do.alojamento.do.calendarphp

# Alterar o caminho para o ficheiro python, no ficheiro
vi index.php
# procurar por process
/process
# alterar a variável para o caminho do ficheiro python e guardar
# para testar, aceder ao endereço e seguir o manual de utilizador
<ip da maquina do serviço Radius>/<nome_da_pasta>
```

Manual de Utilizador

Sistema de Contabilização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1. Passos para utilização do sistema de contabilização

Os passos para a correcta utilização da aplicação são os que se seguem. Todos os campos são de preenchimento obrigatório e devem ser preenchidos na ordem a seguir exposta (Figura 1):

- Caso o mês e ano visível no calendário não sejam os pretendidos, seleccionar o ano e mês e clicar em "Mostra" (1). Desta forma é gerado um novo calendário.
- De seguida, selecciona-se um dia (2) e a data completa aparecerá no campo 3.
- No campo 4 insere-se o IP (campo bloqueado a 15 caracteres)
- No 5, o fuso horário para que a aplicação calcule a hora da ocorrência no fuso horário Português.
- No campo 6, selecciona-se a hora, minuto e segundos da ocorrência
- Por último, selecciona-se a opção "Procurar"(7)

O resultado da pesquisa aparece passados 5 a 60 segundos, em forma de tabela como se pode ver na figura.

O botão limpar (7) reinicia todas as variáveis, para que seja configurada uma nova pesquisa.

Figura 1: Página inicial da aplicação (os números a vermelho fazem parte da legenda para explicação da utilização)

Especificação de Testes

Sistema de Contabilização

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1 Introdução

Este documento tem como objectivo definir um conjunto de testes de modo a verificar se a aplicação apresenta o comportamento inicialmente previsto.

2 Testes

2.1 Funcionais

Os testes funcionais foram realizados directamente na linha de comando, executando a aplicação de procura directamente com os 3 parâmetros de entrada (data, timezone e IP):

```
python procura_ip.py 'dd mm aaa HH:MM:SS' TZ xxx.xxx.xxx.xxx
```

Os mesmos, podem ser efectuados na interface web, bastando preencher os campos específicos com os parâmetros dos comandos usados. Para que não fossem divulgados os dados reais de contabilização, foram construidos dois ficheiros de teste, idênticos aos ficheiros reais. Esses ficheiro tem o seguinte conteúdo:

- Ficheiro com os dados de acesso

```
+.1x: 120811-22:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] SSID[eduroam] IP[111.222.333.4]
-.1x: 120812-00:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] C[1458889]  T[7200] TT[3588] CT[7785466]
+.1x: 120812-08:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam]
+.1x: 120812-08:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] IP[111.222.333.444]
+.1x: 120812-08:40:19 U[aaaccc@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] IP[111.222.333.555]
-.1x: 120812-09:02:18 U[aaaccc@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] C[8583]  T[1319] TT[4589] CT[456879]
-.1x: 120812-09:10:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] C[9584]  T[1800] TT[55448] CT[4578987]
+.1x: 120812-11:40:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] IP[111.222.333.2]
+.1x: 120812-11:41:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam]
-.1x: 120812-11:45:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] C[1800]  T[240] TT[54785] CT[455878]
+.1x: 120812-11:41:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] IP[111.222.333.3]
+.1x: 120812-15:40:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] IP[111.222.333.2]
-.1x: 120812-15:45:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] C[4523]  T[300] TT[54658] CT[458795]
+.1x: 120812-16:39:19 U[jjjkkk@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] IP[111.222.333.4]
-.1x: 120812-16:40:19 U[jjjkkk@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] C[145]  T[60] TT[5487] CT[889974]
+.1x: 120812-15:40:19 U[fffggg@student.uc.pt] @[ap5252.uc.pt|10.20.10.50] SSID[eduroam] IP[111.222.333.55]
-.1x: 120812-15:45:19 U[fffggg@student.uc.pt] @[ap5252.uc.pt|10.20.10.50] C[4523]  T[300] TT[54658] CT[458795]
+.1x: 120812-16:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] IP[111.222.333.444]
-.1x: 120812-16:40:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] C[123]  T[60] TT[55508] CT[4579010]
+.1x: 120812-20:39:19 U[aaabbb@student.uc.pt] @[ap7894.uc.pt|70.20.30.40] SSID[eduroam] IP[111.222.333.444]
-.1x: 120813-01:39:19 U[aaabbb@student.uc.pt] @[ap7894.uc.pt|70.20.30.40] C[5987441] T[18000] TT[548742] CT[546589744]
+.1x: 120813-11:40:19 U[xptoxp@student.uc.pt] @[ap7849.uc.pt|10.60.50.50] SSID[eduroam] IP[111.222.123.56]
```

- Ficheiro com o MAC Address

```
+.1x: 120811-22:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] SSID[eduroam] M[ad45fg78hj96]
-.1x: 120812-00:40:19 U[jjjkkk@student.uc.pt] @[ap5478.uc.pt|10.50.30.40] M[ad45fg78hj96]
+.1x: 120812-08:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] M[1040d5f8g9j8]
+.1x: 120812-08:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] M[1040d5f8g9j8]
+.1x: 120812-08:40:19 U[aaaccc@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] M[17df58r9tdae]
-.1x: 120812-09:02:18 U[aaaccc@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] M[17df58r9tdae]
-.1x: 120812-09:10:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] M[1040d5f8g9j8]
+.1x: 120812-11:40:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] M[2090er45e4t5]
+.1x: 120812-11:41:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] M[12sd3rty57j8]
-.1x: 120812-11:45:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] M[12sd3rty57j8]
+.1x: 120812-11:41:19 U[wwweee@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] M[12sd3rty57j8]
+.1x: 120812-15:40:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] SSID[eduroam] M[2090er45e4t5]
-.1x: 120812-15:45:19 U[fffggg@student.uc.pt] @[ap5648.uc.pt|10.20.50.50] M[2090er45e4t5]
+.1x: 120812-16:39:19 U[jjjkkk@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] M[ad45fg78hj96]
-.1x: 120812-16:40:19 U[jjjkkk@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] M[ad45fg78hj96]
+.1x: 120812-15:40:19 U[fffggg@student.uc.pt] @[ap5252.uc.pt|10.20.10.50] SSID[eduroam] M[2090er45e4t5]
-.1x: 120812-15:45:19 U[fffggg@student.uc.pt] @[ap5252.uc.pt|10.20.10.50] M[2090er45e4t5]
+.1x: 120812-16:39:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] SSID[eduroam] M[1040d5f8g9j8]
```

```
-.1x: 120812-16:40:19 U[aaabbb@student.uc.pt] @[ap1234.uc.pt|10.20.30.40] M[1040d5f8g9j8]
+.1x: 120812-20:39:19 U[aaabbb@student.uc.pt] @[ap7894.uc.pt|70.20.30.40] SSID[eduroam] M[1040d5f8g9j8]
-.1x: 120813-01:39:19 U[aaabbb@student.uc.pt] @[ap7894.uc.pt|70.20.30.40] M[1040d5f8g9j8]
+.1x: 120813-11:40:19 U[xptoxp@student.uc.pt] @[ap7849.uc.pt|10.60.50.50] SSID[eduroam] M[df3of4f6g478]
```

No entanto, foram também feitos testes com os ficheiros reais a fim de validar o funcionamento de todo o sistema.

Código Teste	teste01
Condição a testar	Inicio e fim de ligação em dias diferentes
Comando	python procura_ip.py '11 08 2012 23:41:15' GMT+1 111.222.333.4
Saída	Hora da ocorrencia (GMT+1) : Sat Aug 11 23:41:15 2012 Hora da ocorrencia (LOCAL) : 23:41:15 IP da ocorrencia (a procurar) : 111.222.333.4 Data - hora de inicio: 20120811-22:40:19 Utilizador (e-mail) : jjjkkk@student.uc.pt Ligado ao AP : ap5478.uc.pt|10.50.30.40 Ligado com o IP : 111.222.333.4 Ligado com o MAC: ad45fg78hj96 Ligacao Terminada em: 20120812-00:40:19

Código Teste	teste02
Condição a testar	Sem registo encontrado
Comando	python procura_ip.py '12 08 2012 02:41:15' GMT+1 111.222.333.4
Saída	Hora da ocorrencia (GMT+1) : Sun Aug 12 02:41:15 2012 Hora da ocorrencia (LOCAL) : 02:41:15 IP da ocorrencia (a procurar) : 111.222.333.4 Nao foi encontrado qualquer registo!

Código Teste	teste03
Condição a testar	Com conexão-desconexão de outro utilizador pelo meio
Comando	python procura_ip.py '12 08 2012 08:40:15' GMT+1 111.222.333.444
Saída	Hora da ocorrencia (GMT+1) : Sun Aug 12 08:40:15 2012 Hora da ocorrencia (LOCAL) : 08:40:15 IP da ocorrencia (a procurar) : 111.222.333.444 Data - hora de inicio: 20120812-08:39:19 Utilizador (e-mail) : aaabbb@student.uc.pt Ligado ao AP : ap1234.uc.pt|10.20.30.40 Ligado com o IP : 111.222.333.444 Ligado com o MAC: 1040d5f8g9j8 Ligacao Terminada em: 20120812-09:10:19

Código Teste	teste04
Condição a testar	Encontra um utilizador em condição normal
Comando	python procura_ip.py '12 08 2012 08:50:15' GMT+1 111.222.333.555
Saída	Hora da ocorrencia (GMT+1) : Sun Aug 12 08:50:15 2012

	Hora da ocorrencia (LOCAL) : 08:50:15 IP da ocorrencia (a procurar) : 111.222.333.555 Data - hora de inicio: 20120812-08:40:19 Utilizador (e-mail) : aaaccc@student.uc.pt Ligado ao AP : ap1234.uc.pt|10.20.30.40 Ligado com o IP : 111.222.333.555 Ligado com o MAC: 17df58r9tdae Ligacao Terminada em: 20120812-09:02:18

Código Teste	teste05
Condição a testar	Sem desconexão
Comando	python procura_ip.py '12 08 2012 11:50:15' GMT+1 111.222.333.2
Saída	Hora da ocorrencia (GMT+1) : Sun Aug 12 11:50:15 2012 Hora da ocorrencia (LOCAL) : 11:50:15 IP da ocorrencia (a procurar) : 111.222.333.2 Data - hora de inicio: 20120812-11:40:19 Utilizador (e-mail) : fffggg@student.uc.pt Ligado ao AP : ap5648.uc.pt|10.20.50.50 Ligado com o IP : 111.222.333.2 Ligado com o MAC: 2090er45e4t5 Ligacao Terminada em: desconhecido

Código Teste	teste06
Condição a testar	Desconexão antes da conexão respectiva
Comando	python procura_ip.py '12 08 2012 11:44:15' GMT+1 111.222.333.3
Saída	Hora da ocorrencia (GMT+1) : Sun Aug 12 11:44:15 2012 Hora da ocorrencia (LOCAL) : 11:44:15 IP da ocorrencia (a procurar) : 111.222.333.3 Data - hora de inicio: 20120812-11:41:19 Utilizador (e-mail) : wwweee@student.uc.pt Ligado ao AP : ap5648.uc.pt|10.20.50.50 Ligado com o IP : 111.222.333.3 Ligado com o MAC: 12sd3rty57j8 Ligacao Terminada em: 20120812-11:45:19

Código Teste	teste07
Condição a testar	Inicio e fim de ligação em dias diferentes
Comando	python procura_ip.py '13 08 2012 00:44:15' GMT+1 111.222.333.444
Saída	Hora da ocorrencia (GMT+1) : Mon Aug 13 00:44:15 2012 Hora da ocorrencia (LOCAL) : 00:44:15 IP da ocorrencia (a procurar) : 111.222.333.444 Data - hora de inicio: 20120812-20:39:19 Utilizador (e-mail) : aaabbb@student.uc.pt Ligado ao AP : ap7894.uc.pt|70.20.30.40 Ligado com o IP : 111.222.333.444 Ligado com o MAC: 1040d5f8g9j8 Ligacao Terminada em: 20120813-01:39:19

Código Teste	teste08
Condição a testar	Sem desconexão (ou registo); fim do ficheiro
Comando	python procura_ip.py '13 08 2012 12:44:15' GMT+1 111.222.123.56
Saída	Hora da ocorrencia (GMT+1) : Mon Aug 13 12:44:15 2012 Hora da ocorrencia (LOCAL) : 12:44:15 IP da ocorrencia (a procurar) : 111.222.123.56 Data - hora de inicio: 20120813-11:40:19 Utilizador (e-mail) : xptoxp@student.uc.pt Ligado ao AP : ap7849.uc.pt|10.60.50.50 Ligado com o IP : 111.222.123.56 Ligado com o MAC: df3of4f6g478 Ligacao Terminada em: desconhecido

Código Teste	teste09
Condição a testar	Conversão para o timezone da ocorrencia indica o dia anterior
Comando	python procura_ip.py '12 08 2012 01:30:15' GMT+2 111.222.333.4
Saída	Hora da ocorrencia (GMT+2) : Sun Aug 12 01:30:15 2012 Hora da ocorrencia (LOCAL) : 00:30:15 IP da ocorrencia (a procurar) : 111.222.333.4 Data - hora de inicio: 20120811-22:40:19 Utilizador (e-mail) : jjjkkk@student.uc.pt Ligado ao AP : ap5478.uc.pt|10.50.30.40 Ligado com o IP : 111.222.333.4 Ligado com o MAC: ad45fg78hj96 Ligacao Terminada em: 20120812-00:40:19

Código Teste	teste10
Condição a testar	Conversão para o timezone da ocorrencia indica o dia posterior
Comando	python procura_ip.py '12 08 2012 20:30:15' GMT-4 111.222.333.444
Saída	Hora da ocorrencia (GMT-4) :Sun Aug 12 19:30:15 2012 Hora da ocorrencia (LOCAL) : 00:30:15 IP da ocorrencia (a procurar) : 111.222.333.444 Data - hora de inicio: 20120812-20:39:19 Utilizador (e-mail) : aaabbb@student.uc.pt Ligado ao AP : ap7894.uc.pt|70.20.30.40 Ligado com o IP : 111.222.333.444 Ligado com o MAC: 1040d5f8g9j8 Ligacao Terminada em: 20120813-01:39:19

2.2 Desempenho

Nos testes de desempenho, foram avaliados os tempos de execução e o consumo de memória. Em ambos os casos, foi variado o dia a pesquisar. Neste tipo de testes, foi usado um ficheiro real, pois para além de não ser necessário divulgar os dados de acesso, os ficheiros reais são maiores e mais complexos.

Estes foram executados na linha de comando e os comandos foram os seguintes:

```
python procura_ip.py '1 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '2 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '10 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '11 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '20 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '21 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '29 04 2012 19:30:15' GMT+1 193.136.206.44
python procura_ip.py '30 04 2012 19:30:15' GMT+1 193.136.206.44
```

Os resultados obtidos podem ser vistos de seguida em forma de tabela:

Dia	Memória (Bytes)	Memória (Megabytes)	Tempo (ms)				
			1	2	3	média	desvio padrão
1	8949760	8,54	757	749	747	751	5
2	13799424	13,16	2507	2520	2536	2521	15
10	33710080	32,15	11766	11388	11325	11493	239
11	43077632	41,08	15069	15093	15065	15076	15
20	102158336	97,43	43581	41802	42033	42472	967
21	106659840	101,72	39067	39330	39408	39268	179
29	128860160	122,89	51468	51150	50855	51158	307
30	128856064	122,89	53547	53749	53482	53593	139

Anexo E – Documentação do Sistema de detecção de alteração de configurações

Tiago José Santos Martins
tjmart@student.dei.uc.pt

Especificação de Requisitos e Arquitectura

Sistema de detecção de alteração de configurações

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1 Descrição do sistema actual

Para que seja possível manter a rede segura e verificar que todos os procedimentos de configuração de activos de rede estão a ser cumpridos, é necessário reconhecer as configurações diárias como válidas.

Os procedimentos existentes, definem um conjunto de directrizes às quais os gestores de rede não podem fugir. Isto é, é-lhes dada a liberdade de configurar o equipamento como quiserem – com mais de um SSID no caso dos pontos de acesso por exemplo – seguindo um padrão.

O sistema actual analisa os backups diários das configurações de todos os activos de rede. O seu funcionamento é muito simples, pois faz apenas o 'diff',com algumas modificações, entre a versão de "hoje" e a versão de "ontem", enviando um email ao gestor de rede com as respectivas diferenças.

2 Novo sistema a implementar

O novo sistema a implementar, que será um complemento ao já existente, deverá permitir verificar se as configurações estão de acordo com os procedimentos definidos. Isto é, recebendo uma configuração, o sistema fará uma análise a fim de determinar se o último backup da configuração está de acordo com as directrizes presentes no procedimento. Este sistema, terá duas partes distintas que trabalharão em conjunto para alcançar o objectivo.

A primeira parte, será uma camada de apresentação que terá dois tipos de interface web e um sistema automático de analise de backups de configurações. Através da interface web, será possível definir as datas e o endereço IP dos activos de rede a verificar. Numa verifica-se as diferenças entre dois activos e as inconsistências com o procedimento no mais recente (validação da configuração). Na outra, pode-se só verificar as inconsistências com o procedimento (validação da configuração). O sistema automático, validará todas as configurações, durante a noite, depois de ser feito um backup das mesmas.

A segunda parte será um middleware que fará a análise a uma configuração, interpretando-a à procura de inconsistências. No entanto, esta segunda parte vai permitir configurar um conjunto de aplicações mais diversificadas. A análise de todos os backups de configurações durante a noite (com envio de email ao administrador com um link directo para a página web), a integração com o sistema antigo (página web com diferenças entre configurações de dois dias e validação de procedimento para a configuração mais recente) ou mesmo com o sistema de monitorização são dois dos exemplos. Ou seja, este middleware, devolverá um conjunto de dados facilmente interpretáveis que permitirão integrar as mais variadas aplicações.

3 Requisitos do novo sistema

De seguida, são apresentados os requisitos dos três componentes a implementar. Os requisitos 4 a 6 da Tabela 1 e os requisitos 5 ao 7 da Tabela 2 são relativos às diferenças para o procedimento de configuração de activos de rede.

ID	Requisito	Prioridade
req1	Escolha do dia e mês a verificar	Must
req2	Facilidade de alteração de dias, para verificação de outras datas	Must
req3	Definição do IP	Must
req4	Salientar visualmente linhas em falta	Must
req5	Salientar visualmente linhas não alteradas que deviam ser alteradas	Must
req6	Salientar visualmente linhas alteradas	Must

Tabela 1: Requisitos do sistema de validação de uma configuração

ID	Requisito	Prioridade
req1	Escolha dos dias e meses a comparar	Must
req2	Facilidade de alteração de dias, para verificação de outras datas	Must
req3	Definição do IP	Must
req4	Salientar visualmente as diferenças entre ficheiros	Must
req5	Salientar visualmente linhas em falta	Must
req6	Salientar visualmente linhas não alteradas que deviam ser alteradas	Must
req7	Salientar visualmente linhas alteradas	Must

Tabela 2: Requisitos do sistema de validação de uma configuração e diferença entre configurações

ID	Requisito	Prioridade
req1	Verificar inconsistências com os procedimentos de todos os backups	Must
req2	Enviar email ao gestor de rede	Must

Tabela 1: Requisitos do sistema de validação de todas as configurações

4 Análise dos logs

O log gerado pelo sistema antigo, é de interpretação bastante fácil. Caso não haja diferenças entre as configurações dos dias adjacentes, o comando 'diff' devolve uma linha vazia. Caso sejam detectadas diferenças, são apresentadas em conjunto de linhas (se forem linhas seguidas) e/ou a linha diferente (se for apenas uma linha).

O caracter '<', a seguir ao IP, corresponde à configuração de 'hoje' e o '>' à antiga configuração – 'ontem'.

```
diff $hoje $ontem

>ip< linha_1_hoje
>ip< linha_2_hoje
>ip< linha_3_hoje
>ip> linha_1_ontem
>ip> linha_2_ontem
>ip> linha_3_ontem
```

Como foi aproveitada uma ferramenta em python que transforma os dados de saída do comando diff em html, tentou-se que o novo sistema gerasse logs idênticos. Assim, para que depois de construída a aplicação de validação de configurações fosse fácil de a integrar com o sistema antigo e ao mesmo tempo ficasse legível como aplicação autónoma, o log gerado terá o formato de uma lista (tratamento que a aplicação em Python dá ao log do diff antes de o converter em html). Esta será formada por tuplos em que a chave é o número da linha e o objecto será a linha em falta no procedimento ou vazio, caso a linha exista mas tenha pequenas alterações.

```
{('id1','linha em falta'),('id2',' '), ('id3',' ')}
```

5 Casos de uso

Os actores do sistema são identificado como "Utilizador web" e Sistema Operativo. São eles os responsáveis por interagir com a aplicação a fim de esta lhe devolver um resultado. O caso de uso da Figura 1 descreve o tipo de interacções que o actor pode exercer sobre o sistema. Neste caso, sobre o sistema que salienta as diferenças entre configurações e as inconsistências com o procedimento.

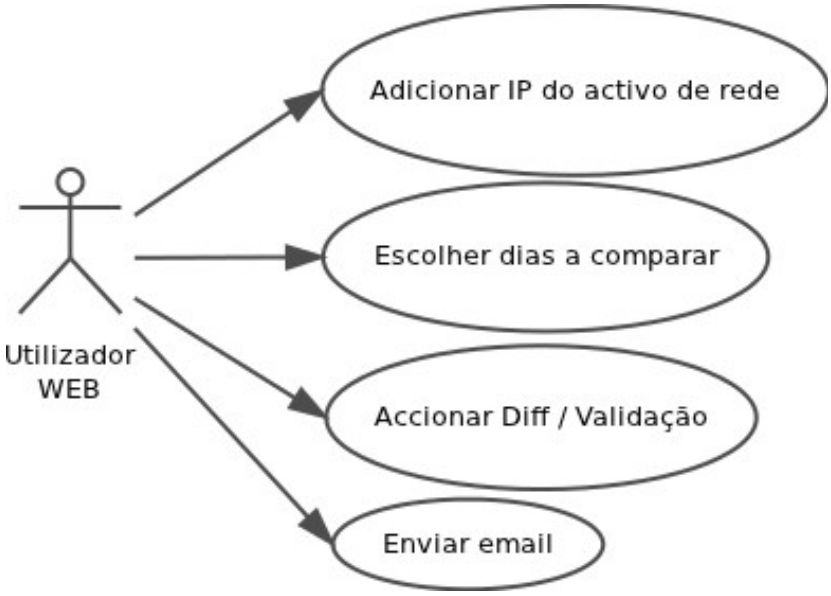

Figura 1: Caso de uso do sistema de validação uma configuração e diferença entre configurações

O fluxo (Figura 2) deste mesmo sistema, é bastante simples. Depois de o utilizador inserir todos os dados e accionar a validação, o sistema faz o diff entre as duas configurações e identifica a configuração mais recente, para poder validá-la com o procedimento.

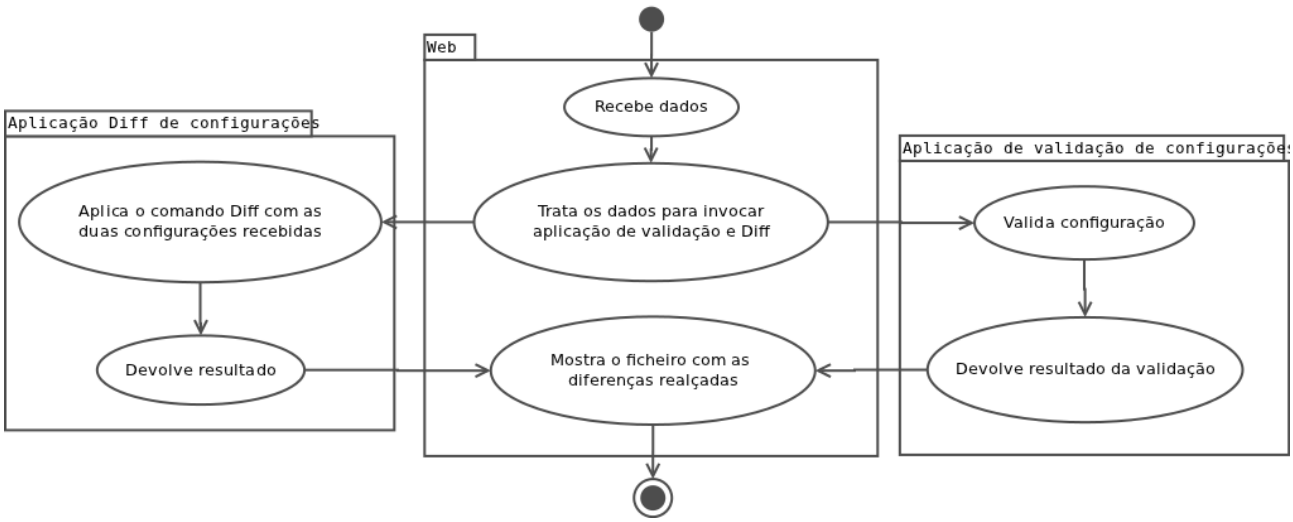

Figura 2: fluxo do sistema de validação uma configuração e diferença entre configurações

O caso de uso do sistema de verificação dos procedimentos pode ser visto na Figura 3. Como se percebe, neste caso só é dado um IP e um dia do ano. É também possível accionar a validação ou enviar um email.

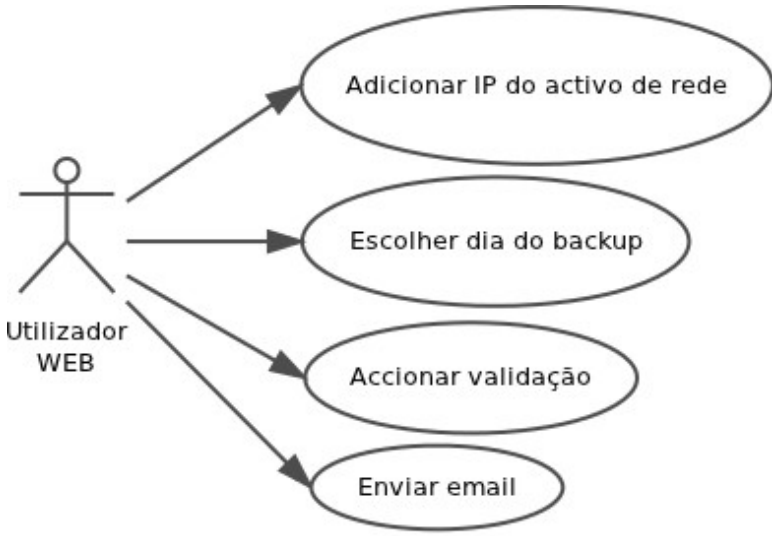

Figura 3: Caso de uso do sistema de validação uma configuração

O fluxo, que pode ser visto na Figura 4, é idêntico ao anterior. O utilizador preenche o dados e activa a validação. A aplicação valida a configuração e devolve o resultado para que a página web o possa salientar.

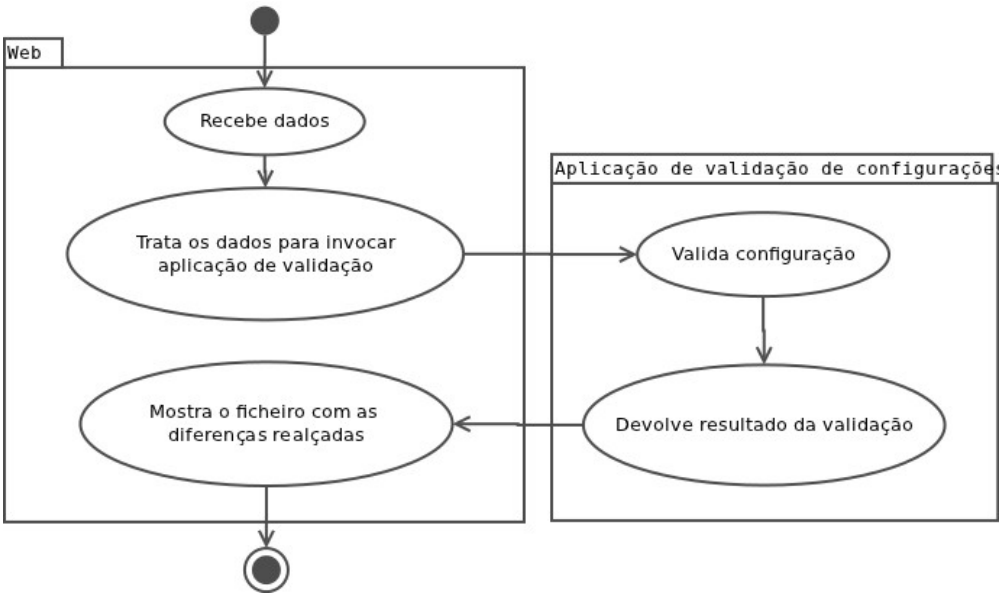

Figura 4: Fluxo do sistema de validação uma configuração

Por último, o analisador de backups que será executado todos os dias, depois de feitos os respectivos backups. O actor será o sistema operativo e o caso de uso é o da Figura 5.

Figura 5: Caso de uso do sistema de validação dos backups de configurações

Na Figura 6 pode-se ver o fluxo do sistema. A aplicação começa por determinar qual a data dos backups a verificar e depois verifica todos os backups. Por último, agrega os resultados e envia um mail com o relatório ao gestor.

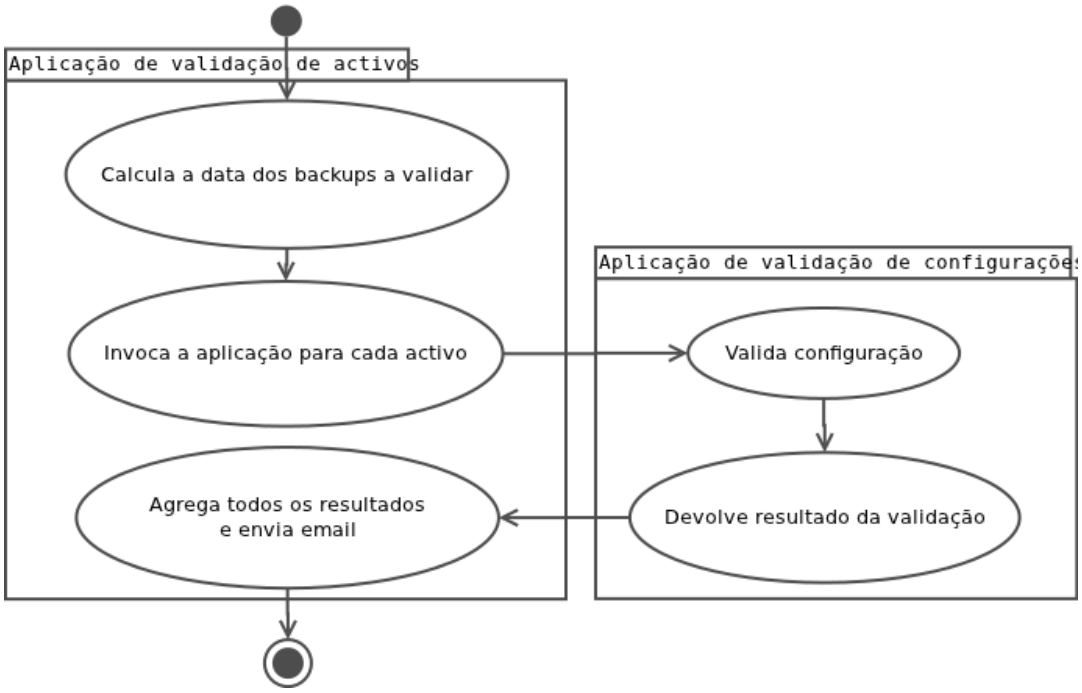

Figura 6: Fluxo do sistema de validação dos backups de configurações

1 Arquitectura do sistema

1.1 Objectivos

O objectivo deste sistema, é a simplicidade de processos. O conjunto de aplicações irão funcionar localmente, acedendo aos backups das configurações.

1.2 Arquitectura Geral

A aplicação é composta por 2 módulos que se encontram a vermelho na Figura 7:

- Camada de apresentação – duas páginas web, desenvolvidas em html e php que comunicarão com o middleware através de chamadas ao sistema operativo.
  Para integração destas com a aplicação de validação de configurações, será usada uma ferramenta já existente – diff2html (http://diff2html.tuxfamily.org/ consultado a: 03-05-2012). Esta trata-se de um programa escrito em Python que transforma os dados de saída do comando diff numa página html. Um terceiro componente, será um analisador de backups que será responsável por validar todos os backups de configurações ao fim do dia.
- Aplicação de validação de configurações – desenvolvida em python que receberá um parâmetro (caminho para o ficheiro de configuração) e será a responsável pela análise dos ficheiros de configuração, a fim de determinar se estão consoante os procedimentos.

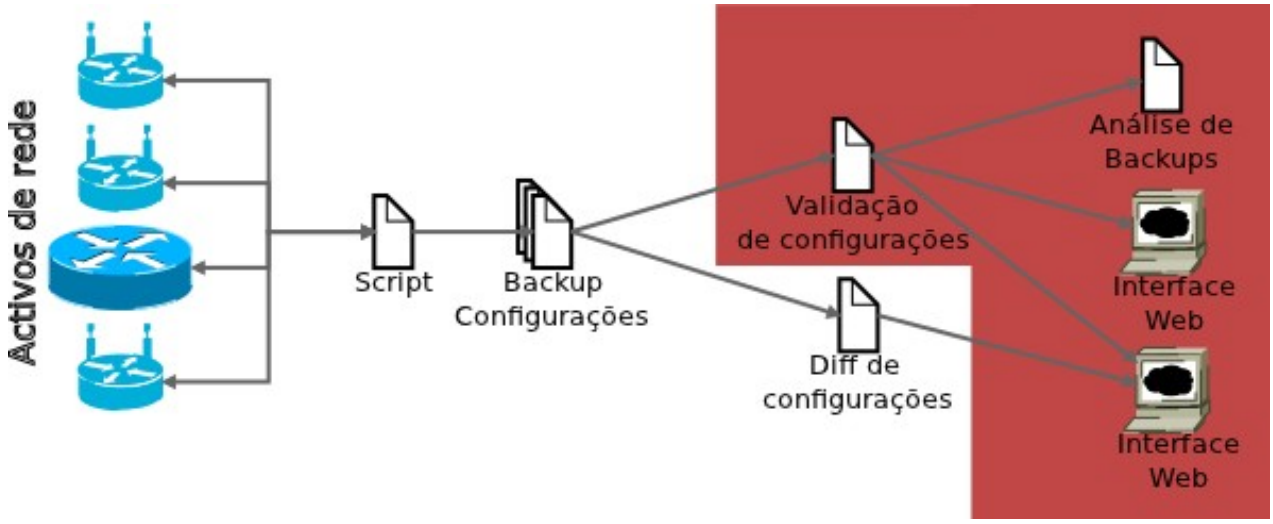

Figura 7: Arquitectura geral do sistema. A vermelho, os componentes a implementar.

1.3 Visão Geral

A solução encontrada para a implementação do sistema é composta por três camadas (Figura 8):

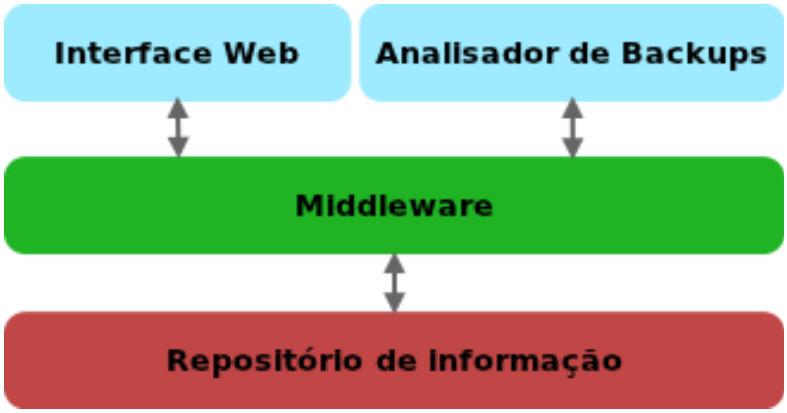

Figura 8: Modelo de três camadas usado no sistema.

A camada superior, camada de apresentação, terá dois tipos de componentes: duas interfaces web responsáveis pela interacção com o utilizador e um componente responsável por analisar todos os backups. A camada Middleware, valida os backups dos activos de rede e devolve o resultado à camada de apresentação. Por último, o repositório de informação, são os backups dos ficheiros de configuração.

Manual de Instalação

Sistema de detecção de alteração de configurações

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

O presente documento pretende ser uma descrição pormenorizada da instalação e configuração do sistema.

Requisitos mínimos do sistema:

- Sistema operativo fedora
- php 5.1.6
- python 2.4

Tempo: 20 minutos

Instalação:

A instalação do presente sistema é bastante simples, bastando copiar 7 ficheiros para pastas distintas. O procedimento segue a seguir:

```
ssh <ip da maquina responsável pelo backup de configurações dos pontos de acesso>
# Ir até à pasta dos dos backups e criar uma nova directoria
cd /(pasta ou caminho)
mkdir comparador
cd comparador
# Copiar os ficheiros analisa_backup_proc.py, check_procedure.py,
# diff2html.py, diff2html_procedure.py e regularExpressions.py
wget http://endereço.do.alojamento.do.analisa_backup_proc.py
wget http://endereço.do.alojamento.do.check_procedure.py
wget http://endereço.do.alojamento.do.diff2html.py
wget http://endereço.do.alojamento.do.diff2html_procedure.py
wget http://endereço.do.alojamento.do.regularExpressions.py

# Alterar o caminho para os ficheiros de configuração,
# nos ficheiros diff2html.py e diff2html_procedure.py
vi  diff2html.py
# procurar por FILE
/FILE
# alterar a variável para o caminho pretendido e guardar
vi diff2html_procedure.py
# procurar por FILE
/FILE
# alterar a variável para o caminho pretendido e guardar
# testar: deverá devolver o help
python diff2html.py
python diff2html_procedure.py

# Ir até à pasta do servidor web
cd /var/www/html
# Criar duas pastas
mkdir conf
mkdir diff
cd conf
wget http://endereço.do.alojamento.do.index.php.do.conf
cd ../diff
wget http://endereço.do.alojamento.do.index.php.do.diff

# Alterar o caminho para o ficheiro python, nos ficheiros das interfaces web
```

vi ../conf/index.php
Procurar por process
/process
Alterar a variável para o caminho do ficheiro python e guardar
vi ../diff/index.php
procurar por process
/process
Alterar a variável para o caminho do ficheiro python e guardar

Para testar, aceder ao endereço e seguir o manual de utilizador
<ip da maquina responsável pelo backup de configurações dos pontos de acesso>/diff
<ip da maquina responsável pelo backup de configurações dos pontos de acesso>/conf

Manual de Utilizador

Sistema de detecção de alteração de configurações

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1. Passos para utilização do sistema de validação de backups de configurações com diferenciador de versões

Os passos para a correcta utilização da aplicação são os que se seguem. Todos os campos são de preenchimento obrigatório e devem ser preenchidos na ordem a seguir exposta (Figura 1):

- No campo 1, pode-se escolher os dias e meses a comprar. Para os alterar, pode-se escrever nas caixas de texto ou clicar nos botões de '+' e '-'
- De seguida insere-se o endereço IP do ponto de acesso (campo 2)
- Por fim activa-se a comparação com o botão 'Comparar'
- Depois de aparecer o resultado, pode-se enviar um email com o link para a aplicação. (campo 5)

Figura 1: Página inicial da aplicação que diferencia duas configurações e valida a mais recente (os números a vermelho fazem parte da legenda para explicação da utilização)

O resultado é facilmente interpretável e pode ser visto um exemplo na Figura 2. A verde pode-se ver as linhas modificadas. A azul as linhas adicionadas ao ficheiro da Data2, a vermelho as linhas removidas do ficheiro da Data1. A cor de laranja, são visíveis as linhas diferentes do procedimento.

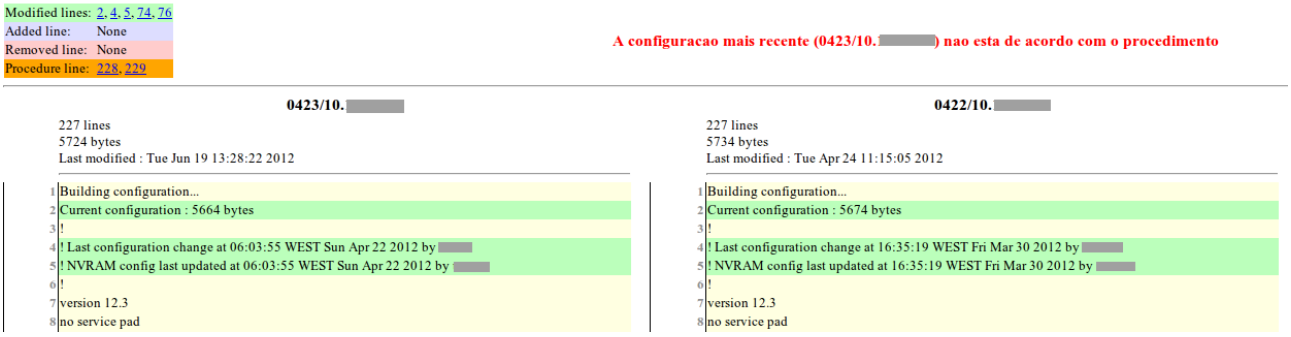

Figura 2: Resultado da execução da aplicação

2. Passos para utilização do sistema de validação de backups de configurações

Os passos para a correcta utilização da aplicação são os que se seguem. Todos os campos são de preenchimento obrigatório e devem ser preenchidos na ordem a seguir exposta (Figura 3):

- No campo 1, pode-se escolher os dias e meses a comprar do ficheiro a validar. Para os alterar, pode-se escrever nas caixas de texto ou clicar nos botões de '+' e '-'
- De seguida insere-se o endereço IP do ponto de acesso (campo 2)
- Por fim activa-se a validação com o botão 'Comparar'
- Depois de aparecer o resultado, pode-se enviar um email com o link para a aplicação. (campo 5)

Figura 3: Página inicial da aplicação que diferencia duas configurações e valida a mais recente (os números a vermelho fazem parte da legenda para explicação da utilização)

O resultado é facilmente interpretável e pode ser visto um exemplo na Figura 2. Nesta interface só serão realçadas as linhas diferentes do procedimento a laranja.

Especificação de Testes

Sistema de detecção de alteração de configurações

Tiago Martins
tjmart@student.dei.uc.pt
GSIIC - UC

1 Introdução

Este documento tem como objectivo definir um conjunto de testes de modo a verificar se a aplicação apresenta o comportamento inicialmente previsto.

2 Testes

2.1 Funcionais

Os testes funcionais foram realizados na interface web e na linha de comando dependendo do componente. A especificação dos mesmos estão a seguir:

Código Teste	teste01
Componente	Sistema antigo e validação da configuração mais recente
Condição a testar	Detectar diferença entre ficheiros
Dados de entrada	2 datas diferentes e um endereço IP que tenha sofrido alterações
Saída	Apresenta linhas alteradas, adicionadas e eliminadas realçadas

Código Teste	teste02
Componente	Sistema antigo e validação da configuração mais recente
Condição a testar	Detectar diferença entre ficheiros
Dados de entrada	2 datas diferentes e um endereço IP que não tenha sofrido alterações
Saída	Não realça nada pois os ficheiros são iguais

Código Teste	teste03
Componente	Sistema antigo e validação da configuração mais recente
Condição a testar	Validar configuração mais recente
Dados de entrada	2 datas diferentes e um endereço IP que não esteja de acordo com o procedimento
Saída	Realça as diferenças em relação ao procedimento na configuração mais recente

Código Teste	teste04
Componente	Sistema de validação de uma configuração
Condição a testar	Validar a configuração
Dados de entrada	Data e um endereço IP que não esteja de acordo com o procedimento.
Saída	Realça as diferenças em relação ao procedimento na configuração

Código Teste	teste05
Componente	Sistema de validação de uma configuração
Condição a testar	Validar a configuração
Dados de entrada	Data e um endereço IP que não esteja de acordo com o procedimento.
Saída	Configuração sem alterações

Código Teste	teste06
Componente	Sistema de validação dos backups de todas as configurações
Condição a testar	Validar o conjunto de configurações e enviar email
Dados de entrada	Accionar o sistema na linha de comando
Saída	Email enviado ao gestor com o relatório

2.2 Performance

Nos testes de performance, foram avaliados os tempos de execução e o consumo de memória. Estes testes foram realizados directamente na interface web, recorrendo a ficheiros de configuração real.

Os resultados obtidos podem ser vistos de seguida em forma de tabela sendo que as médias e desvios padrão correspondem a um conjunto de 10 medições. Na Tabela 1 pode-se ver a memória consumida e tempo de processamento da interface web de verificação de inconsistências.

Memoria	Tempo (ms)	
(Megabytes)	média	desvio padrão
6,79	103	4

Tabela 1: Inconsistências com os procedimentos

Na Tabela 2 são apresentados os resultados para o sistema completo.

Memoria	Tempo (ms)	
(Megabytes)	média	desvio padrão
6,87	134	3

Tabela 2: Diff de versões e inconsistência com os procedimentos

Na Tabela 3 são apresentados os resultados para o sistema que irá analisar os backups todos os dias (realizado na linha de comando por não ter interface web).

Memoria	Tempo (ms)	
(Megabytes)	média	desvio padrão
9,25	15680	8802

Tabela 3: Analisa todos os backups