MECANISMOS DE SEGURANÇA E SUA INTEGRAÇÃO COM O *KERBEROS*

Jorge Granjal e Edmundo Monteiro
Serviços de Informática e Comunicações
Departamento de Engenharia Informática, Universidade de Coimbra - PoloII
3030 Coimbra
Tel.: +351 039 790000, Fax: +351 039 701266, E-mail: jgranjal@dei.uc.pt, URL: http://www.dei.uc.pt

Sumário
Apresentação de mecanismos de segurança da Rede do DEI[1] e sua integração com o *Kerberos* [1]. Esta integração envolveu a adaptação dos *Proxies* de Aplicação da *Firewall* [2] ao *Kerberos* através da construção de um módulo de autenticações e adaptação do gestor de autenticações, *Proxies* de TELNET, RLOGIN e FTP.

1. INTRODUÇÃO

As necessidades de segurança numa Rede de computadores apresentam-se a diversos níveis e aumentam proporcionalmente à heterogeneidade dos Sistemas utilizados. Vão desde a protecção das comunicações em sessões de terminal remoto à protecção contra ataques externos utilizando uma Rede de *Firewall*.
Pretende-se que os vários mecanismos de segurança implementados interferiram o mínimo possível no funcionamento tradicional dos Serviços a que os Utilizadores se habituaram. Características como encriptação transparente das comunicações e gestão automatizada de autenticações são importantes. Neste âmbito surge a necessidade de integrar mecanismos de segurança com objectivos à partida distintos e complementares, como os descritos neste trabalho.

2. NECESSIDADES E MECANISMOS DE SEGURANÇA

Na Fig.1 identificam-se algumas das principais necessidades de segurança e produtos utilizados na sua implementação.

Necessidades	Produtos
Verificações periódicas da integridade de ficheiros.	Tripwire.
Auditorias periódicas.	Cops, Satan, Crack.
Acessos seguros a Servidores.	Kerberos e SSH.
Filtragem de acessos por Serviço.	TCP-Wrappers.
Protecção contra ataques externos.	Firewalls, FW-TK.

Figura 1 - Necessidades e produtos de segurança.

2.1 Verificações periódicas da integridade de ficheiros

Com o aumentar do número e sofisticação dos ataques contra Sistemas é importante possuir mecanismos que permitam detectar modificações não desejadas em ficheiros binários ou de configuração. Os Sistemas atacados vêm os seus binários substituídos por comandos forjados à medida dos propósitos dos seus invasores (*rootkits*), fornecendo ao administrador uma imagem falsa do estado do Sistema. O *Tripwire* [3] permitiu implementar um Sistema de verificações diárias de ficheiros de Sistema para diversos Servidores mediante o cálculo de um *checksum* para cada ficheiro e sua comparação com *checksums* anteriores.
Os resultados dos *checksums* são copiados para um Servidor principal onde são efectuadas as comparações com *checksums* armazenados anteriormente. A base de dados de *checksums* é actualizada após a instalação de novos Sistemas (a partir dos seus binários e ficheiros de configuração) e mantida num disco configurado por *hardware* como *read-only*. Os *scripts* utilizados para as operações de verificação têm a capacidade de enviar alarmes via *e-mail* ou Sistema de *paging* do DEI.

2.2 Auditorias Periódicas

Para além da constante vigilância a ficheiros de *log* e outras medidas básicas de segurança as auditorias periódicas a Servidores assumem um papel importante, permitindo detectar problemas comuns de segurança. São utilizados para esse fim produtos como o *Cops* [4], *Satan* [5] e *Crack* [6].

2.3 Acessos seguros a Servidores

Os principais mecanismos de acessos seguros a Servidores são a *Ssh* e o *Kerberos*, que passam a descrever-se.
A *Ssh* [7] permite implementar uma *Shell* e um comando de cópia de ficheiros entre Servidores utilizando encriptação em todas as comunicações. A *Ssh* pode ser utilizada em substituição do *rlogin*, *rcp*, *rsh* e *rdist*, permitindo ainda o estabelecimento de ligações X11 em modo seguro.
O *Kerberos* [1] permite autenticar Utilizadores e Serviços utilizando encriptação em todas as comunicações. Baseia-se na existência de um ou mais Servidores de confiança utilizados para autenticação e designados por *principals*. Os *principals* partilham uma chave secreta com cada Serviço *Kerberos* de modo a possíbilitar autenticação mutua. Os domínios administrativos do *Kerberos* designam-se por *realms*.
No funcionamento do *Kerberos* os *principals* utilizam *tickets* para provar a sua identidade. A Fig.2 ilustra o funcionamento básico de uma sequência de autenticação *Kerberos*. Um utilizador (A) que pretenda utilizar um dado Serviço (S) pede ao Servidor *Kerberos* um *ticket* para acesso ao Serviço. Depois de validar os *principals* (A e S) o Servidor cria um pacote de resposta que contém entre outros parâmetros uma chave secreta da sessão. Esse pacote é encriptado com a chave secreta de S e depois de acrescentados mais alguns

[1] Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

parâmetros é novamente encriptado com a chave secreta de A, após o que é devolvido a A. O utilizador A desencripta o pacote recebido e armazena-o para uso futuro.

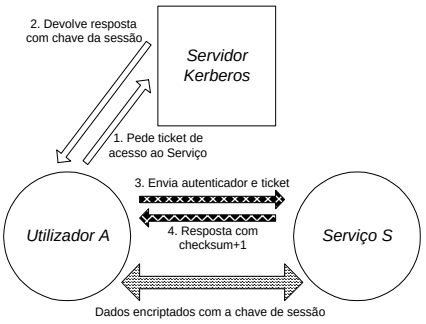

Figura 2 - Sequência de autenticação *Kerberos*.

Antes de enviar uma mensagem a S o utilizador A cria um autenticador que contém um *checksum* e encripta-o utilizando a chave secreta da sessão. Este autenticador é enviado a S juntamente com o *ticket* recebido do Servidor. S recebe esses dados e desencripta-os utilizando a sua própria chave secreta. Dado que o *ticket* contém a chave secreta da sessão com a qual o autenticador foi encriptado, S pode também desencriptar o autenticador. Para verificar a identidade de A, S compara o conteúdo do *ticket* com o do autenticador. Se os conteúdos corresponderem S considera A autenticado. Em resposta a A, S devolve o autenticador com um valor de *checksum* adicionado de 1 e encriptado com a chave da sessão. Deste modo A sabe que S teve acesso à chave da sessão ficando A e S mutuamente autenticados. As restantes comunicações respeitantes à utilização do Serviço S por A são encriptadas com a chave secreta da sessão.

2.4 Filtragem de acessos por Serviço

O *TCP-Wrappers* permite filtrar e monitorizar pedidos de acesso a Serviços utilizando *daemons* próprios em substituição dos *daemons* reais. Todas as ligações têm que ser autorizadas antes de executado o *daemon* real do Serviço. As operações de autorização decorrem de forma totalmente transparente para o Cliente. São configurados os *hosts* autorizados a aceder a cada Serviço, utilizando opcionalmente o protocolo descrito no *RFC931* [8] para comprovação da identidade do Cliente. O *TCP-Wrappers* implementa também protecção contra ataques via *name-spoofing* ou *IP-spoofing*.

2.5 Protecção contra ataques externos

Neste tipo de protecção pretende-se impedir acessos do exterior directamente a máquinas internas à Rede ou Organização (p.ex. máquinas de salas de aulas, gabinetes ou de Serviços administrativos). Esta protecção é conseguida através da configuração de uma *Firewall*, definindo listas de acesso no *router* de ligação ao exterior e utilizando *Proxies* de Aplicação para encaminhamento das ligações. Nas máquinas da rede exposta são disponibilizados os Serviços públicos. Os *Proxies* são configurados juntamente com mecanismos de autenticação e registo de acessos. Para mais informações sobre *Firewalls* consultar [9].

3. O *KERBEROS* NA REDE DO DEI

A versão do *Kerberos* utilizada na Rede do DEI encontra-se descrita em [10]. São utilizados dois *realms* cujos nomes são *student.dei.uc.pt* e *dei.uc.pt*. O primeiro é utilizado em máquinas de alunos e o segundo em máquinas administrativas e de docentes.

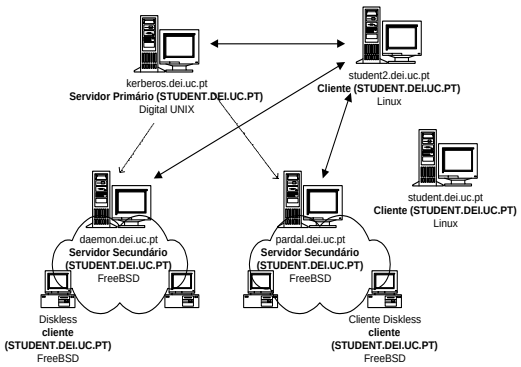

Figura 3 - *Realm student.dei.uc.pt*.

A criação de dois *realms Kerberos* visa a separação física e administrativa das bases de dados de autenticação, à semelhança do que é feito tradicionalmente para áreas de trabalho em Servidores da Rede.

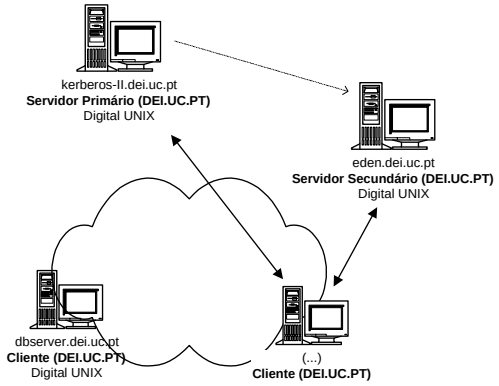

Figura 4 - *Realm dei.uc.pt*.

A comunicação entre *realms* possibilita acessos entre máquinas pertencentes a domínios diferentes. Esta comunicação é possibilitada pela criação de dois *principals* especiais: Um *principal* de nome *krbtgt.dei.uc.pt* no Servidor do *realm student.dei.uc.pt* e um *principal* de nome *krbtgt.student.dei.uc.pt* no Servidor do outro *realm*. Um Cliente que pretenda utilizar um Serviço existente noutro *realm* pede um *ticket* ao Servidor do seu próprio *realm*.
Para ambos os *realms* são utilizados Servidores secundários para maior redundância e as bases de

dados de autenticação são actualizadas nesses Servidores duas vezes por dia.
Para a integração do *Kerberos* nos Servidores substituem-se os binários de programas Clientes e Servidores pelos binários com suporte *Kerberos*, permitindo implementar os seguintes Serviços com suporte *Kerberos*: *Telnet*, *Rsh*, *Rlogin*, *Ssh*, *Scp*, *Ftp*, *Pop*, *Xdm* (para *logins* em *Clusters FreeBSD diskless*) e *Tacacs* (para *logins* em Servidores de acesso *PPP* e *RDIS*). As ligações de máquinas não pertencentes a nenhum dos *realms* (p.ex. Clientes *Windows*) a Servidores configurados com *Kerberos* podem ser efectuadas com Clientes tradicionais ou utilizando *Ssh* com suporte *Kerberos*.

4. INTEGRAÇÂO DO *KERBEROS* NA FIREWALL DO DEI

4.1 Configuração da *Firewall* do DEI

A *Firewall* do DEI apresenta uma configuração em *Screened-Subnet*. Os *Proxies* de nível de Aplicação são implementados em Sistemas *FreeBSD*.

A *subnet* exposta alberga os Servidores *Proxy* e de Serviços públicos. Regras de filtragem no *router* impedem ligações do exterior destinadas a máquinas da Rede interna e ligações da Rede Interna directamente para o exterior. Os *Proxies* de Aplicação são implementados utilizando o *TIS Firewall-ToolKit* e o *Squid*. O primeiro implementa *Proxies* de *FTP*, *TELNET*, *RLOGIN*, *RSH*, *REAL-AUDIO*, *REAL-VIDEO*, *NNTP* e oferece um *front-end* para recepção segura de *e-mail* (*SMAP*). O *Squid* implementa *Proxies* de *HTTP*, *GOPHER* e *WAIS*.

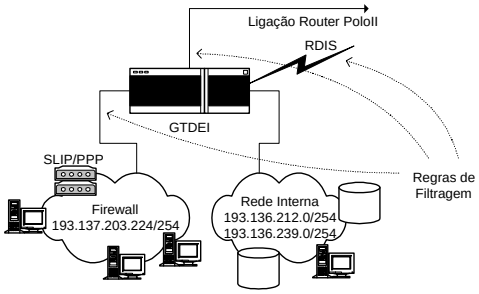

Figura 5 - Configuração da *Firewall* do DEI.

Utilizando *load-balancing* por DNS e duas ou mais máquinas *Proxy* com configurações semelhantes consegue-se distribuir equitativamente a carga de pedidos pelas várias máquinas, mediante o registo um nome (*proxy.dei.uc.pt*) com dois ou mais endereços IP. Esta solução apresenta vantagens do ponto de vista da robustez no funcionamento dos *Proxies* e facilidade de gestão. Consultar [12] para mais informações relativas ao trabalho de implementação da *Firewall*.

4.2 O *Kerberos* e a *Firewall*

Neste cenário a integração do *Kerberos* com os módulos *Proxy* do *TIS FW-TK* surge como uma necessidade no sentido de facilitar a gestão de autenticações. Tal integração permitiria utilizar nos *Proxies* de Aplicação as *passwords* já definidas nos *realms Kerberos*.

O *FW-TK* não suporta actualmente este tipo de autenticações, pelo que optou-se pelo desenvolvimento de código com o objectivo de proceder à implementação desse suporte. O código desenvolvido divide-se em duas categorias: A implementação de um novo módulo de autenticações para o *FW-TK* e a modificação de alguns programas da mesma *package* para o suporte de *Kerberos*.

Relativamente ao novo módulo para autenticações *Kerberos*, este funciona integrado no *daemon* de autenticação utilizado pelos *Proxies* do *FW-TK* (*authd*), como ilustrado pela Fig.6. Originalmente o *authd* suporta autenticações utilizando *SecurID*, *SNK*, *S/Key* ou *reusable passwords*.

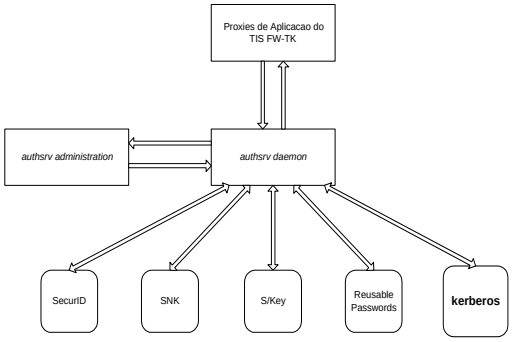

Figura 6 - Integração do módulo de autenticações *Kerberos* no *TIS FW-TK*.

O novo módulo de autenticações funciona ao mesmo nível dos módulos originais, sendo as verificações de *passwords Kerberos* efectuadas em todos os *realms* (definidos em */etc/krb.realms*). A Fig.7 esquematiza a sequência de operações necessárias à verificação de uma *password Kerberos*. Os pedidos de autenticação *Kerberos* são encaminhados por intermédio do *authd*, o *daemon* de autenticações do *FW-TK*. O módulo de autenticações *Kerberos* implementa o diálogo de autenticação com os Servidores *Kerberos* responsáveis pelos vários *realms*. Este diálogo prossegue até que a *password* seja validada ou não existam mais *realms* definidos para validação. O resultado da validação da *password* é comunicado ao *authd* como resposta à chamada da respectiva função de autenticação pertecente ao módulo de autenticações *Kerberos*.

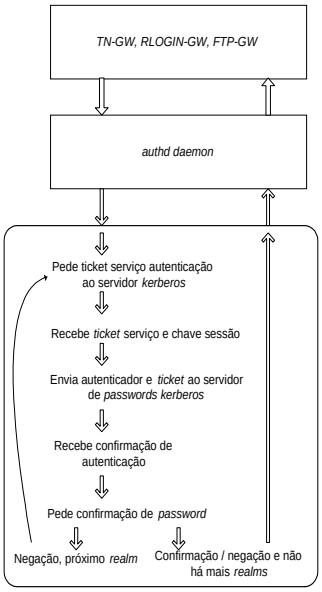

Figura 7 - Verificação de *passwords Kerberos* pelo novo módulo de autenticações.

No tocante às modificações operadas em ficheiros da *package FW-TK*, estas permitem adaptar o código do programa de gestão de autenticações (*authsrv*) e acrescentar suporte *Kerberos* aos *Proxies* de nível de Aplicação *TN-GW*, *RLOGIN-GW* e *FTP-GW* (*Proxies* de *Telnet*, *Rlogin* e *FTP*, respectivamente). A adaptação do *authserv* permite a definição na base de dados de autenticação do *FW-TK* de Utilizadores para os quais é definido o *Kerberos* como mecanismo de autenticação. A Fig.8 apresenta um exemplo de um diálogo com o *authsrv* para esse efeito:

```
authsrv# addu jorge "Jorge Costa"
authsrv# ena jorge
authsrv# proto jorge krb
authsrv# ls
Report for users in database
user    group  longname     status  proto   last
jorge           Jorge Costa  y       krb     never
```

Figura 8 - Criação de um utilizador *Kerberos*.

A implementação de suporte *Kerberos* nos *Proxies* referidos permite ligações totalmente encriptadas, quando efectuadas a partir de máquinas pertencentes a um dos *realms*.

Muitos dos acessos a partir do exterior são efectuados a partir de máquinas sem suporte *Kerberos*. Neste caso a utilização de encriptação pode ser conseguida utilizando a *Ssh* com suporte *Kerberos* e configurada via *plug-gw*. O *plug-gw* é um Proxy de utilização genérica do *FW-TK*. A configuração do *plug-gw* para o porto utilizado pela *Ssh* (porto 22) permite encaminhar as ligações para uma máquina da Rede interna onde o *sshd* tenha sido compilado com suporte *Kerberos*. O *sshd* comunica com o Servidor *Kerberos* e com o Cliente de modo a efectuar a autenticação necessária à ligação, como é ilustrado pela Fig.9.

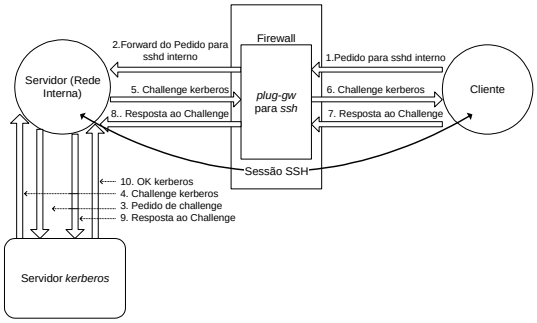

Figura 9 - Ligações utilizando *ssh* via *plug-gw*.

5. CONCLUSÕES

A integração de vários mecanismos de segurança oferece importantes vantagens funcionais e administrativas. A integração descrita neste documento do *Kerberos* com o *FW-TK* é disso exemplo, permitindo a utilização das bases de dados de autenticação *Kerberos* num novo Serviço de Rede. O *software* desenvolvido encontra-se em fase final de testes na Rede do DEI.

6. REFERÊNCIAS

[1] Miller, S.P., Neuman, B.C., Schiller, J.I. e Saltzer, J.H. "Kerberos Authentication and Authorization System", Project Athena Technical Plan, M.I.T., Dez 1987.

[2] *TIS Firewall-ToolKit*. "TIS Firewall Toolkit Overview", Trusted Information Systems, Jun 1994. Consultar *http://www.tis.com*.

[3] Kim, Gene e Spafford, Eugene. "Experiences with Tripwire: Using Integrity Checkers for Intrusion Detection", Purdue University. Technical Report CSD-TR-93-071, Feb 1994.

[4] Computer Oracle & Password System, produzido por Dan Farmer. Disponível em *ftp://coast.cs.purdue.edu/pub/tools/unix/cops.*

[5] Security Administrator Tool for Analyzing Networks, produzido por Dan Farmer. Disponível em *ftp://ftp.win.tue.nl/pub/security.*

[6] Programa para identificação de *passwords* inseguras em Sistemas UNIX, disponível em *ftp://info.cert.org/pub/tools/crack.*

[7] Ylonen, T. "The SSH (Secure Shell) Remote Login Protocol", Helsinki University of Technology, Jul 1995.

[8] StJohns, Mike. "Authentication Server", RFC931, Jan 1985.

[9] Ranum, Marcus J. "Thinking About Firewalls", proceedings of the Second World Conference on Systems Management and Security (SANSII), 1993.

[10] Danielsson, Johan e Westerlund, Assar. "Kerberos 4 from the Royal Institute of Technology in Stockholm, Sweden", Dez 1997.

[11] Squid Internet Object Cache. Documentação disponível em *http://squid.nlanr.net/Squid*.

[12] Nunes, Bruno e Barros, Nuno. "Projecto Firewall, Relatório de Estágio",D.E.I., Jul 1998.