Tolerância a Intrusões num Sistema em Grid *

Luis Sardinha Nuno Ferreira Neves Paulo Veríssimo

Departamento de informática
Faculdade de Ciências
Universidade de Lisboa
{sardinha,nuno,pjv}@di.fc.ul.pt

Resumo

Os sistemas em Grid suportam a execução de aplicações que necessitam de recursos consideráveis quer na capacidade de processamento, como no armazenamento de dados ou na largura de banda para comunicação. Necessariamente, este tipo de infra-estruturas é caracterizado pelo seu elevado custo, o que tem levado a que diversas organizações coordenem esforços e cheguem a acordos para partilharem alguns dos seus sistemas computacionais. Para facilitar esta cooperação, foram criadas ferramentas que interligam as diversas organizações, entre elas o Globus Toolkit, considerado um standard *de facto* nesta área.

Neste artigo é descrita uma arquitectura e um conjunto de protocolos que permitem que intrusões maliciosas possam ser toleradas no Globus Toolkit. Os recursos que normalmente são geridos por esta ferramenta podem ser muito tentadores para um adversário (e.g., *hacker*) pois podem ser usados, por exemplo, para quebrar palavras-passe com ataques de força bruta ou lançar ataques distribuídos de negação de serviço. Além disso, na área empresarial podem usar igualmente estas fragilidades para crimes de espionagem ou terrorismo industrial.

De modo a conseguir tolerar tais ataques, optou-se por desenvolver um sistema de replicação que permita que, mesmo que um adversário mal intencionado consiga controlar uma parte do sistema, o restante continue ser capaz que fazer progresso de forma correcta. Foram definidos para comunicação entre o servidor Globus e os recursos da organização três protocolos com características diferentes: um onde não é usada criptografia, um segundo baseado apenas em criptografia simétrica e um terceiro que usa criptografia híbrida.

Palavras chave: Segurança em Grid, Tolerância a intrusões, replicação, Globus.

*Este trabalho foi suportado parcialmente pela FCT através do projecto POSI/CHS/39815/2001 (COPE), e do Laboratório de Sistemas Informáticos de Grande-Escala (LASIGE).

1 Introdução

Os sistemas em Grid são caracterizados por possuírem um elevado número de recursos computacionais. Devido ao custo destas infra-estruturas é necessário coordenar recursos de várias organizações, tornando exequível o suporte de aplicações com requisitos consideráveis. Esses recursos podem ser tão variados como máquinas paralelas, sistemas de armazenamento de grande capacidade ou algo mais específico como um microscópio electrónico. Normalmente, estes sistemas, que muitas vezes se encontram separados fisicamente, estão ligados através de redes com grande capacidade de largura de banda, para possibilitar a troca de dados.

A experiência actual mostra que é muito difícil, ou até mesmo impossível, construir um sistema complexo completamente seguro. Como os recursos deste tipo de aplicações são muito atractivos, podemos prever que estes sejam alvo de ataques provenientes da comunidade *hacker* (ver exemplo em [2]). O seu considerável poder computacional permite a uma pessoa mal intencionada quebrar palavras-passe com ataques de força bruta, utilizar os sistemas de armazenamento para guardar ferramentas ou outro tipo de informação, ou usar a grande largura de banda da rede para executar ataques de negação de serviço. Além disso, ataques nas infra-estruturas de organizações empresariais poderão resultar em actos de espionagem ou terrorismo industrial.

Consequentemente, os sistemas em Grid devem ser pensados para a eventualidade de alguém se conseguir introduzir na organização e controlar um ou mais recursos. Neste artigo é proposta uma solução baseada na tolerância a intrusões, cujo objectivo não é tanto prevenir que o sistema seja penetrado, mas sim forçar os adversários a despenderem a um esforço significativo para obterem resultados importantes, idealmente a um esforço superior à sua capacidade de ataque. Devido a isso, temos de analisar cuidadosamente os potenciais adversários e tentar prever quais as suas capacidades.

O Globus Toolkit é um sistema desenvolvido para ambientes em Grid, e a versão corrente disponibiliza um conjunto de serviços que podem ser usados na construção de aplicações ou outras ferramentas de programação [4]. Graças ao seu sucesso, é considerado um standard *de facto* para este tipo de computação.

A arquitectura do Globus Toolkit coloca uma máquina na "fronteira" da rede de cada organização cuja função é servir de interface entre os computadores externos e os recursos internos que são cedidos para a execução de trabalhos em Grid (iremos chamar a esta máquina o *servidor Globus*). Este servidor faz, entre outras coisas, o mapeamento de um utilizador externo num local. Este mapeamento é necessário porque todas as políticas e mecanismos de segurança da organização estão preparados para lidarem apenas os utilizadores locais. Deste modo, se um utilizador externo quer aceder a um recurso interno, este normalmente tem primeiro de obter uma credencial local que o autoriza a executar a operação desejada.

Com vista a evitar qualquer falha de segurança, o Globus Toolkit foi analisado e implementado com muito cuidado. No entanto, mesmo que assumamos que o Globus é completamente seguro, o que é muito difícil de provar, um adversário pode, mesmo assim, explorar qualquer outra vulnerabilidade que o servidor possa possuir e, por exemplo, executar um ataque de personificação mudando os ficheiros de configuração do mapeamento. Caso isso aconteça, o atacante pode atribuir a si próprio, ou a outra pessoa que ele deseje,

qualquer uma das identidades normalmente geridas pelo Globus Toolkit. Logo, o atacante passa a ter acesso a todos os recursos utilizados por este sistema apenas penetrando numa máquina!

Neste artigo é explicado como tolerar intrusões no Globus Toolkit versão 3. A solução consiste na replicação do processo de autenticação e autorização de maneira a que mesmo que o servidor Globus seja comprometido, os restantes recursos não o fiquem de uma forma imediata. A replicação bizantina[9] possibilita que o sistema continue a fornecer o serviço, mesmo que uma parte das réplicas seja corrompida. No entanto, cada uma das réplicas tem de ter implementações diferentes sob pena de conterem as mesmas vulnerabilidades e de um ataque conseguir atingir o seu objectivo simultaneamente em todas as réplicas. Para evitar isso, vamos assumir que cada réplica irá ter um sistema operativo e uma implementação do mecanismo de replicação distinta das restantes.

Para tolerar alguns tipos de ataques à comunicação entre o servidor Globus e os recursos, tal como a personificação do servidor Globus ou a adulteração das mensagens, foram criados 3 protocolos distintos: um primeiro no qual é feita a comunicação entre o servidor Globus e os recursos em claro; um segundo onde são apenas usadas chaves simétricas (baseado no protocolo do Kerberos [6]) e um terceiro onde são utilizadas chaves assimétricas para proceder à autenticação mútua entre o servidor globus e os vários recursos, sendo a comunicação dos dados protegida com criptografia simétrica (baseado no SSH [1]).

O artigo encontra-se organizado da seguinte forma: na secção 2 descrevemos superficialmente o Globus Toolkit e a sua última versão (Globus Toolkit 3). Na secção 3 é apresentada a nossa solução para o problema. Os 3 protocolos para a comunicação entre o servidor Globus e os recursos são explicados na secção 4. A secção 5 descreve os aspectos relacionados com a gestão do sistema. Na secção 6 mencionamos alguns aspectos relativos à implementação. E finalmente, concluímos na secção 7.

2 Globus Toolkit

O Globus Toolkit pode ser usado para interligar várias organizações físicas de maneira a criar uma organização virtual distribuída, escalável e dinâmica. Esta organização virtual é composta por indivíduos que pretendam partilhar e usar os seus recursos de uma forma dinâmica e coordenada. A terceira versão do Globus Toolkit (GT3) é uma implementação do Open Grid Services Infrastructure (OGSI) [10], uma especificação técnica dos conceitos descritos no Open Grid Services Architecture (OGSA) [4]. O OGSA define uma arquitectura standard para aplicações baseadas em Grid, (genericamente chamadas de Grid Services), uma extensão aos Web Services.

As versões anteriores do Globus usavam na comunicação um conjunto de protocolos e portos específicos (potencialmente criando algumas dificuldades com as anteparas de segurança das diversas instituições), que deram origem a alguns problemas de portabilidade. Na versão actual, optou-se assim pela não especificação dos protocolos. No entanto, a implementação corrente é baseada em SOAP[11]. O SOAP disponibiliza formas de transmissão de envelopes XML que encapsulam o tráfego normal. Os pacotes são

depois enviados pelo protocolo HTTP.

O Web Services Description Language (WSDL) [12] é usado para descrever os métodos de um Web Service e fornece um modo de expressar as assinaturas das operações e associações a protocolos. O suporte da segurança ao nível da mensagem é baseada em WS-Security [5], XML-Signature [13] e XML-Encryption.

O núcleo da infra-estrutura de segurança do Globus é baseado no serviço de autenticação e autorização do Java, permitindo que os *Grid Services* se mantenham independentes dos mecanismos de segurança.

3 Sistema de autenticação e autorização tolerante a intrusões

No sistema de autenticação e autorização tolerante a intrusões (ITAAS) os diversos serviços possuem a flexibilidade para definir os seus requisitos de segurança. Esta definição é conseguida a partir de uma aproximação declarativa que o GT3 dispõe. No entanto, os serviços não estão directamente dependentes da implementação dos protocolos de segurança, permitindo assim, que estes possam ser modificados sem que haja a necessidade de alterar os vários serviços.

Quando um pedido necessita de ser validado, o núcleo do GT3 verifica se o certificado associado ao pedido foi emitido por alguma autoridade de confiança. Verifica também na sua configuração local se existe um mapeamento entre o identificador do utilizador externo e um interno. Para tornar este ficheiro escalável o administrador pode agrupar vários identificadores externos no mesmo utilizador local. Por exemplo, todos os utilizadores da organização A são associados ao utilizador org_A.

Caso um adversário consiga penetrar o servidor Globus, é possível que execute diferentes tipos de ataques maliciosos, tais como, mudar as políticas de segurança de um serviço ou alterar os ficheiros de configuração de modo a poder mapear qualquer pessoa em quase qualquer utilizador local (apenas aqueles que são geridos pelo Globus) sem que os sistemas internos à organização se apercebam do problema. Depois disso, o atacante pode usar a infra-estrutura local para executar tarefas nos recursos acessíveis ao Globus com as mesmas permissões que qualquer utilizador autorizado.

Para prevenir este tipo de ataques, replicou-se o sistema de autenticação e autorização com um protocolo capaz de tolerar intrusões. Um adversário terá de penetrar $f+1$ do total de $n \geq 3f+1$ máquinas que executam a autenticação e autorização para comprometer este sub-sistema (sendo f um parâmetro configurável pelo administrador do sistema). Esta solução foi desenvolvida de acordo com especificação OGSI para que qualquer organização consiga comunicar com o servidor Globus sem se aperceber que este contém algumas diferenças.

3.1 Cliente ITAAS no Globus Toolkit

Os módulos responsáveis pelas várias fases no tratamento dos pedidos, e respectivas respostas, são definidos num ficheiro de configuração no servidor Globus. Para que estes possam ser substituídos sem que seja necessário alterar qualquer código fonte, optámos por criar novos módulos responsáveis pela autorização e autenticação.

Em termos gerais, os novos módulos apenas têm de diferente a forma de adquirir a informação de autenticação e autorização. Os módulos originais iam ler à configuração local toda a informação necessária para decidir se o pedido podia ser satisfeito ou não. No novo módulo essa informação é pedida a uma biblioteca externa ao Globus Toolkit denominada de *cliente ITAAS*. Esta biblioteca tem como função comunicar com os servidores ITAAS, um conjunto de réplicas que formam um sistema de replicação bizantina, e retornar ao módulo Globus que a invocou alguns dados, como a validade da identificação do requerente ou o certificado ITAAS (ver Figura 2). O certificado ITAAS é constituído por um conjunto de dados: o URI do serviço, identificador do utilizador local com permissão para executar a tarefa, a validade do certificado, e as assinaturas dos servidores ITAAS.

Esta abordagem é, de certo modo, semelhante à delegação de certificados do Globus Toolkit. No Globus Toolkit o cliente assina um certificado do servidor Globus para que este possa executar tarefas em outras organizações – o sistema ITAAS autoriza o servidor Globus a realizar tarefas nos recursos dentro da organização. Qualquer uma destas formas de delegação de direitos tem o risco de dar privilégios a um adversário que controle o servidor Globus. Para limitar a sua utilização abusiva, os certificados contêm datas de validade muito curtas.

4 Protocolos tolerantes a intrusões

Nesta secção iremos começar por mostrar como os protocolos de comunicação entre o servidor Globus e os recursos irão comportar-se no ponto de vista funcional. De seguida, serão descritos os três protocolos e, por fim, faremos uma pequena discussão em relação às suas principais diferenças, vantagens e desvantagens.

4.1 Funcionalidade

Nesta secção é descrito como o sistema se comportará no ponto de vista funcional. Para uma melhor compreensão, os números na Figura 1 ilustram os seguintes passos do protocolo:

1. O cliente externo (iremos chamar cliente a qualquer utilizador ou serviço a executar tarefas em nome de um utilizador) contacta um serviço a executar no servidor Globus. O cliente pode estar localizado em qualquer lugar desde que tenha acesso à Internet. No pedido, o cliente envia um certificado Globus que indica os seus privilégios.

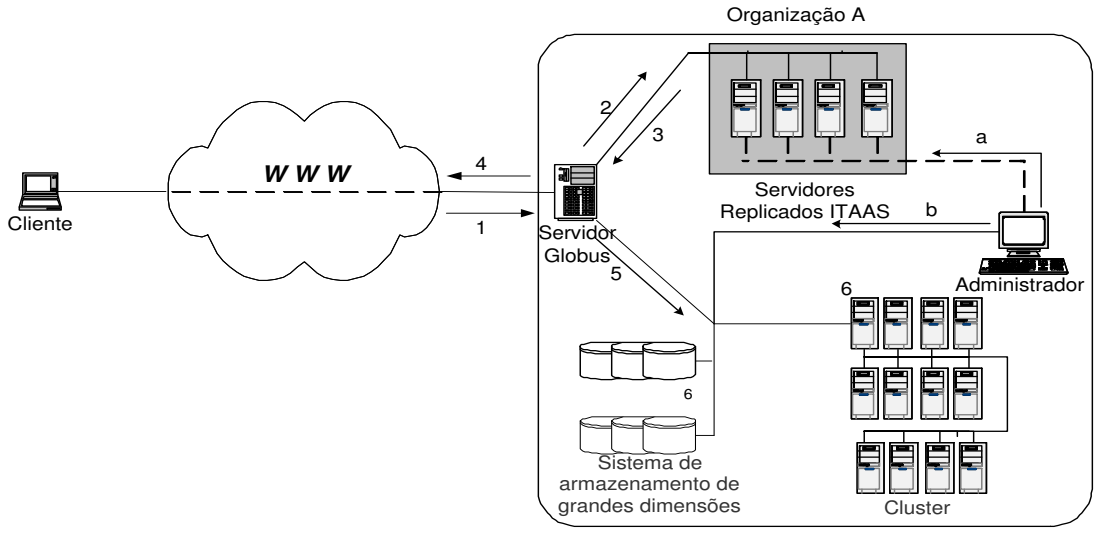

Figura 1: Um exemplo de uma arquitectura ITAAS

2. O servidor Globus faz algumas validações iniciais e envia um pedido de autenticação e autorização ao cliente ITAAS, que por sua vez o passa aos servidores replicados (os servidores ITAAS).

3. Cada réplica dos servidores ITAAS consulta os seus ficheiros de configuração e determina se o cliente poderá aceder aos recursos que requereu. Em caso afirmativo, envia uma resposta contendo informação referente ao tipo de permissão que o requerente do serviço pode ter.

   Quando o cliente ITAAS recebe $f + 1$ respostas iguais e correctamente assinadas, cria um certificado ITAAS que devolve ao Globus Toolkit. A definição de igualdade das respostas das réplicas não deve ser entendida de uma forma estrita, mas apenas que os dados que irão ser colocados no certificado ITAAS são idênticos.

   Para limitar o uso abusivo destes certificados, cada um possui uma data de validade que terá de ser obrigatoriamente inferior ou igual à validade do certificado Globus que foi enviado originalmente pelo cliente.

4. Após a confirmação dos servidores ITAAS de que o cliente pode utilizar o serviço requerido, é dada uma resposta afirmativa para que se possa iniciar a transferência dos dados necessários à execução do serviço.

5. O servidor Globus contacta os recursos da organização para enviar as sub-tarefas do serviço em causa. Como o servidor Globus pode estar comprometido por um adversário, os vários recursos apenas disponibilizam os seus serviços mediante a apresentação do certificado ITAAS, para terem a certeza que o pedido foi feito por um cliente autorizado.

6. Os recursos executam o pedido.

4.2 Protocolo com comunicação em claro

Esta secção descreve o protocolo que oferece menores garantias de segurança, mas que tem a vantagem de ser o mais simples. Nas secções seguintes iremos apresentar outros dois protocolos, mas apenas indicaremos os dados adicionais ou diferenças em relação aos anteriores.

Neste protocolo assume-se que a rede interna da organização é suficientemente segura para que não seja necessária a utilização de comunicação cifrada entre o servidor Globus e os vários recursos. Esta opção pode ser útil para organizações que utilizam recursos muito dependentes do desempenho e que confiam na sua rede interna.

A Figura 2 esquematiza todo o fluxo de informação no sistema. Para uma melhor compreensão, a enumeração do protocolo tem uma correspondência directa com os números da figura.

1. O cliente para que possa executar o serviço, necessita primeiro de se autenticar com o servidor Globus. Esta tarefa é conseguida através da troca de um conjunto de mensagens definidas num protocolo do Globus Toolkit.

2. A validação das mensagens de autenticação é efectuada com auxílio dos servidores ITAAS. Para que essa verificação aconteça, o pedido do certificado ITAAS tem de conter o conjunto de mensagens que foram transmitidas entre o cliente e o servidor Globus.

   Juntamente com as mensagens, os servidores ITAAS devem ainda receber as credenciais necessárias para poderem avaliar se o cliente pode ou não utilizar o serviço requerido.

3. O protocolo de comunicação entre o servidor Globus e os servidores ITAAS não é definido neste momento, de modo a que possa ser utilizado um qualquer protocolo de replicação tolerante a faltas Bizantinas que garanta autenticidade, privacidade, e coerência entre as diferentes réplicas. Na secção 6 é descrito brevemente o protocolo que é usado na nossa implementação.

4. Os servidores ITAAS devolvem como resposta o URI do serviço, o identificador do utilizador local que deverá executar o trabalho, a validade destes dados e a assinatura do servidor. O cliente ITAAS após ter recebido pelo menos $f + 1$ respostas iguais devidamente assinadas, cria um certificado cujo conteúdo é a resposta dos servidores e as suas assinaturas.

5. Neste passo, o servidor Globus contacta os diversos recursos para realizar as operações requeridas pelo cliente. A forma como é efectuada a autenticação entre o servidor Globus e o recurso baseia-se nas relações de confiança usadas, por exemplo, entre sistemas Unix nos serviços de rsh e rlogin com a adição do certificado ITAAS que é enviado ao recurso juntamente com o pedido de ligação.

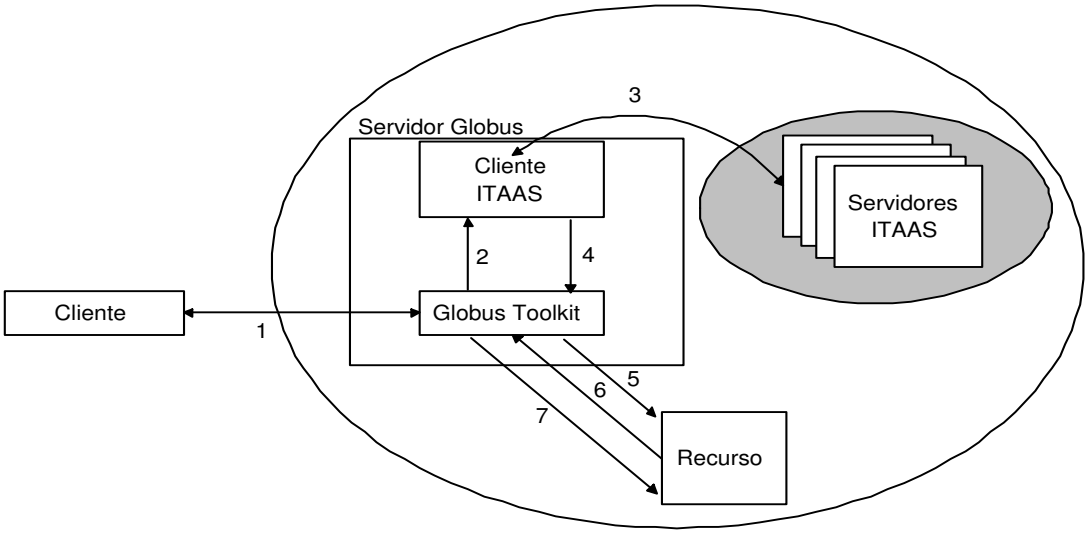

Figura 2: Esquema do protocolo

6. O recurso quando recebe um pedido vai verificar se o endereço do emissor é de confiança através da consulta do sistema de gestão de nomes da organização e pela verificação do certificado ITAAS recebido e estabelece a ligação se tudo estiver bem, caso contrário a ligação é cancelada.

7. O servidor Globus inicia a transmissão dos dados necessários para a execução do serviço.

4.3 Protocolo Kerberizado

Este protocolo é baseado no sistema de autenticação distribuída do Kerberos [6] e num protocolo que transforma certificados X509 em *tickets* Kerberos [7]. É assumido que todos os recursos partilham uma chave simétrica com os servidores ITAAS e que estes têm a capacidade de acordarem uma chave de sessão aleatória. De seguida, apresentam-se as principais alterações ao protocolo descrito na secção anterior.

No passo 2 da figura 2, torna-se necessário especificar adicionalmente qual o recurso que se pretende utilizar. Uma vez que os recursos partilham chaves diferentes com os servidores ITAAS, o servidor Globus terá de fazer um pedido por cada um dos recursos que venha a ser preciso aceder durante a execução do serviço requerido pelo cliente. Caso se saiba à partida a lista dos recursos necessários, esta comunicação pode ser optimizada através do envio da lista numa só mensagem.

A resposta devolvida pelos servidores ITAAS contém para além da informação indicada no protocolo anterior, uma chave de sessão a utilizar com o recurso e um *ticket*. Este *ticket*, que se encontra cifrado com a chave que o recurso partilhada com os servidores ITAAS, contém: o URI do serviço, o identificador utilizador local e a validade, bem como a chave

de sessão que vai ser usada na comunicação entre o servidor Globus e o recurso. Tal como foi definido anteriormente, a comunicação entre o cliente ITAAS e os servidores ITAAS é efectuada por um canal que garanta as propriedades de autenticidade e confidencialidade.

A forma como o servidor Globus se autentica com os recursos sofre as seguintes alterações. No passo 5, é enviado o *ticket* e um número aleatório cifrado com a chave de sessão enviada pelos servidores ITAAS. No passo 6, o recurso responde com o número decrementado em 1 unidade também cifrado com a chave de sessão. Neste momento é estabelecido um canal de comunicação seguro entre o servidor Globus e o recurso. No passo 7 é verificado se o servidor globus tem as permissões para executar a tarefa que deseja através do certificado ITAAS, e são também transmitidos os dados da tarefa.

4.4 Protocolo com chaves públicas

Este protocolo é baseado na autenticação RHostsRSA do SSH [1]. Um protocolo que utiliza os certificados numa fase inicial com o objectivo de estabelecer uma chave de sessão de forma a ser usada na comunicação dos dados entre o servidor Globus e o recurso.

Neste caso deixa de ser necessário enviar aos servidores ITAAS a identificação do recurso que o servidor Globus deseja utilizar. Os quatro primeiros passos do protocolo são idênticos aos do protocolo definido em 4.2.

Os passos 5 e 6 são o protocolo de autenticação RhostRSA do SSH. No passo 7 e após a autenticação mútua entre o servidor Globus e o recurso é enviado o pedido de execução da tarefa juntamente com o certificado ITAAS.

4.5 Discussão

A dificuldade que um adversário tem para conseguir comprometer todo o sistema tornou-se significativamente maior do que no sistema inicial. Após uma intrusão bem sucedida no servidor Globus, o *hacker* não terá acesso directo a nenhum dos recursos da organização porque não possui, nem tem capacidade de gerar, os certificados ITAAS, uma vez que é necessária a cooperação dos servidores ITAAS para os criar.

No pior caso, consegue apropriar-se das credenciais ITAAS que se encontrem armazenadas na máquina penetrada. No entanto, estas credenciais estão apenas associadas a um único utilizador, a um serviço e têm um tempo de validade relativamente reduzido (valor especificado na própria credencial), o que faz com que tenham um interesse limitado para o adversário. Se o adversário quiser usar os recursos da organização terá de continuar a atacar mais máquinas, quer os servidores ITAAS quer os restantes recursos da organização. Ele conseguirá criar as suas próprias credenciais apenas quando comprometer $f + 1$ dos servidores ITAAS.

O protocolo definido na secção 4.2, baseado nas relações de confiança do rsh, pode ser uma opção com alguma segurança, já que, apesar de não cifrar a sua comunicação e de ter algumas vulnerabilidades conhecidas, verifica através do sistema de nomes que o pedido vem mesmo do servidor Globus e valida a veracidade do pedido com certificado ITAAS. Este protocolo foi pensado para as organizações que possam assumir que a rede

entre o servidor Globus e os recursos é segura (ex. toda a infra-estrutura de rede utilizada está dentro de salas de acesso restrito). Desta forma consegue-se limitar a proveniência de possíveis ataques (necessariamente do servidor Globus) bem como o tempo disponível para iniciar o ataque.

No entanto, se assumirmos que não confiamos na rede interna ou que existe a necessidade de confidencialidade e/ou autenticidade na comunicação entre o servidor Globus e os vários recursos, então é necessário utilizar um dos outros dois protocolos. Um deles é baseado apenas em criptografia simétrica e no serviço kerberos, e o outro usa criptografia assimétrica na autenticação e simétrica na comunicação.

O protocolo descrito na secção 4.3 necessita que os servidores ITAAS partilhem uma chave simétrica com todos os recursos utilizados pelos serviços do Globus e que tenham igualmente conhecimento de quais os recursos que cada serviço pode utilizar. Se o número de recursos e serviços disponíveis forem reduzidos, este protocolo poderá ser uma boa opção já que, não necessita de utilizar chaves públicas. Com este sistema obtemos um bom nível de segurança, podendo adicionalmente utilizar os servidores ITAAS para fazer uma auditoria de utilização do recursos. No entanto, esta opção é de difícil manutenção, uma vez que sempre que é alterado algum serviço ou recurso, os servidores ITAAS terão de ter conhecimento desse facto. Este protocolo é viável se estas alterações forem esporádicas, caso contrário é preferível utilizar o protocolo definido na secção 4.4. Outro problema deste protocolo é a necessidade de partilhar chaves simétricas por mais do que 2 entidades. Devido a esse problema, sempre que se detecte que uma das réplicas esteve controlada por um adversário torna-se necessário a alteração de todas as chaves. Durante o espaço de tempo entre a ocorrência de um ataque bem sucedido e a alteração das chaves (um processo potencialmente moroso), existe a possibilidade do adversário conseguir ler as mensagens trocadas entre o servidor Globus e os recursos. No entanto, continua sem conseguir criar certificados ITAAS.

O terceiro protocolo é baseado no RHostRSA do SSH. Este protocolo necessita de certificados de chave pública para autenticar mutuamente os recursos e o servidor Globus. É um protocolo de mais fácil administração do que o anterior, já que os servidores ITAAS não necessitam de saber quaisquer detalhes acerca dos recursos ou serviços utilizados. No entanto, este protocolo é potencialmente o mais pesado computacionalmente das três soluções aqui apresentadas e é necessário existir uma infra-estrutura de gestão de chaves públicas dentro da organização.

5 Manutenção do Sistema

O administrador do sistema apenas tem de executar duas tarefas de manutenção nos servidores ITAAS (ver figura 1):

a *Actualização da informação acerca dos utilizadores externos:* ocasionalmente, a informação relativa aos utilizadores externos tem de ser actualizada nos servidores ITAAS (ex: um novo utilizador é criado). O administrador envia estas actualizações para todas as replicas através de uma mensagem assinada. Assumimos que a chave

privada da assinatura é apenas conhecida pelo administrador, e que está protegida contra roubo (sendo colocada num chip de um cartão, por exemplo). Caso o protocolo for o descrito na secção 4.3, é adicionalmente necessário que o administrador actualize os servidores acerca dos recursos que podem ser utilizados por cada serviço bem como da gestão de chaves simétricas partilhadas entre os servidores ITAAS e os diferentes recursos da organização.

b *Actualização de informação acerca dos servidores ITAAS:* Um recurso só tem a capacidade de validar a informação produzida pelos servidores ITAAS se ele os conhecer, isto é, se conhecer as suas chaves públicas. Esta informação terá de ser actualizada de um modo seguro sempre que essa informação se altera. Para alcançar este objectivo, estas actualizações são assinadas com a chave privada do administrador.

O administrador pode adicionar segurança ao sistema com custo de acrescentar mais servidores ITAAS. Este compromisso entre a segurança e o seu custo não tem de ser feito de um modo estático, já que é relativamente simples adicionar mais servidores. Para isso basta copiar o estado de um servidor correcto e actualizar a informação acerca dos servidores ITAAS.

6 Implementação

Um protótipo da arquitectura tolerante a intrusões foi desenvolvido para a versão corrente do Globus Toolkit. A implementação dos servidores replicados ITAAS utiliza uma biblioteca genérica de replicação denominada BASE [8]

6.1 BASE

A implementação dos servidores ITAAS foi baseada no BASE. Esta implementação fornece-nos um conjunto de serviços que facilitam o desenvolvimento de servidores replicados. Com o BASE, é possível replicar serviços que executam computações arbitrárias mas que sejam deterministas, isto é, que produzem o mesmo resultado quando processam a mesma sequência de operações. Quando esta condição não é verificada, torna-se necessário a criação de interfaces de conformidade e conversões de estados abstractos.

O núcleo da implementação BASE é um protocolo para uma gestão segura de serviços replicados [3]. Este protocolo garante que todas as replicas recebem os mesmos pedidos vindos dos clientes e que os processam pela mesma ordem. As réplicas correctas executam os pedidos de uma forma determinística. Sempre que correctas produzem as mesmas respostas. Por isso, um cliente pode usar um algoritmo de votação simples para remover as respostas erradas geradas por réplicas maliciosas (i.e. que são controladas por um adversário). O protocolo devolve respostas correctas se o número de replicas maliciosas for f sendo que f é inferior a um terço do número total de máquinas ($n \geq 3f + 1$), o que, para este tipo de sistemas, é óptimo.

6.2 Grid RSH

Para testar o protocolo apresentado foi implementado um serviço cuja funcionalidade é uma *shell* remota mas disponibilizada através de um *Grid Service*. Esta aplicação tem basicamente três intervenientes: o cliente, o serviço no servidor Globus e o servidor de GridRSH nos recursos disponíveis.

O cliente é um programa que utiliza o Globus Toolkit 3 para enviar pedidos ao serviço *GridRSH*. Este cliente poderia ser usado em qualquer servidor Globus não tendo qualquer conhecimento das melhorias apresentadas. Tem o objectivo de contactar o serviço *GridRSH*, autenticar-se convenientemente e enviar o identificador do recurso juntamente com o input necessário. Este identificador pode não traduzir directamente um recurso interno, no entanto, o servidor globus terá de mapear este identificador para um recurso específico.

O servidor Globus contém o serviço *GridRSH*, que ao receber um pedido cria dinamicamente uma instância do serviço de forma a ser possível que vários clientes possam utilizar este serviço autonomamente.

O URI do recém criado serviço é enviado ao cliente para que este se autentique e comunique com ele. Após a autenticação bem sucedida entre o cliente e o servidor Globus e a verificação dos servidores ITAAS é dado ao serviço um certificado ITAAS. O servidor Globus envia um pedido ao servidor GridRSH e caso o pedido seja aceite, a transmissão de dados vindos do cliente iniciar-se-á. Durante o normal funcionamento do serviço, o servidor Globus servirá de encaminhador das mensagens entre o cliente e o recurso.

O servidor GridRSH localizado no recurso é muito parecido com o rsh comum. Nesta implementação não é verificado se o porto, utilizado pelo emissor, é privilegiado (para evitar dar privilégios de *root* ao servidor Globus) e é feita a verificação do certificado ITAAS de modo a garantir a legitimidade do pedido.

7 Conclusão

O artigo explica como é que intrusões podem ser toleradas nos sistemas Grid que utilizam o Globus Toolkit. São descritas três variantes dum protocolo baseado na replicação segura do sistema de autenticação e autorização do Globus, sendo uma destas variantes baseada no Kerberos e outra no SSH. Um protótipo da solução proposta é brevemente apresentado, tendo este sido desenvolvido para o Globus Toolkit versão 3 usando a biblioteca de replicação BASE.

Referências

[1] D. Barrett and R. Silverman. *SSH: The Secure Shell.* O'Reilly, June 2001.

[2] B.Krebs. Hackers strike advanced computing networks. *Washingtonpost*, April 2004. http://www.washingtonpost.com/ac2/wp-dyn/A8995-2004Apr13.

[3] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *Proceedings of the USENIX: Symposium on Operating Systems Design and Implementation*, February 1999.

[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. In *Global Grid Forum*, June 2002.

[5] Microsoft IBM and VeriSign. Web services security language(ws-security), April 2002. http://www-106.ibm.com/developerworks/webservices/library/ws-secure/.

[6] J. Kohl and B. Neuman. The kerberos network authentication service. Technical report, MIT Project Athena, June 1991.

[7] O. Kornievskaia, P. Honeyman, B. Doster, and K. Coffman. Kerberized credential translation: A solution to web access control. In *Proceedings of the 10th USENIX Security Symposium*, 2001.

[8] R. Rodrigues, M. Castro, and B. Liskov. BASE: Using abstraction to improve fault tolerance. In *Proceedings of the 18th ACM Symposium on Operating System Principles*, October 2001.

[9] F. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. In *ACM Computing Surveys*, December 1990.

[10] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, and P. Vanderbilt. Open grid services infrastructure (ogsi), June 2003. https://forge.gridforum.org/projects/ogsi-wg/document/Final_OGSI_Specification_V1.0/en/1.

[11] W3C. Simple object access protocol (soap) 1.1, May 2000. http://www.w3.org/TR/soap/.

[12] W3C. Web services description language (wsdl) 1.1 - standard recomendation, March 2001. http://www.w3.org/TR/wsdl.

[13] IETF/W3C XML Signature WG. Xml-signature syntax and processing - standard recomendation, February 2002. http://www.w3.org/Signature/.