Prometheus: Um Serviço de Segurança Adaptativa

Marcos Pirmez[1], *Luci Pirmez[1], Luis Fernando Rust da Costa Carmo[1],

Flávia C. Delicato[2], Paulo F. Pires[2], Edson Barbosa de Sousa[1]

[1] Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro,
Rio de Janeiro, RJ – Brasil.
[2] Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte- Natal, RN - Brasil.

marcos.pirmez@gmail.com {luci,rust,edsonsouza}@nce.ufrj.br
{flavia.delicato, paulo.pires}@dimap.ufrn.br

Abstract. *We propose Prometheus, an Adaptive Security Service that provides facilities to dynamic adaptation of security controls used to mitigate risks and to dynamic adaptation of adaptation policies. These adaptations are driven by context changes and guided by predefined policies. The main contribution of this work is to provide runtime adaptation of security controls and of adaptation policies for ubiquitous applications executing in mobile devices, thus ensuring availability, confidentiality and integrity of these applications.*

Resumo. *Esse trabalho apresenta Prometheus, um Serviço de Segurança Adaptativa que provê funcionalidades para: (i) adaptação dinâmica dos controles de segurança utilizados para mitigar riscos e (ii) adaptação dinâmica das próprias políticas de adaptação. Tais adaptações são guiadas por mudanças no contexto de execução e norteadas por políticas de adaptação previamente definidas. O principal diferencial deste trabalho é a realização de adaptações em tempo de execução de controles de segurança e de políticas de adaptação durante a execução de aplicações ubíquas em dispositivos móveis de forma a garantir a disponibilidade, confidencialidade e integridade dessas aplicações.*

1. Introdução

O avanço tecnológico das redes sem fio e a recente proliferação de dispositivos portáteis (celulares, laptops, PDAs, etc.), aliados ao aumento de popularidade da computação móvel, levaram ao surgimento de aplicações e ambientes de computação ubíquos [Weiser 1991]. Esses ambientes possuem características que originam uma série de desafios ligados ao desenvolvimento e à execução de aplicações com requisitos de segurança em um dispositivo móvel. O primeiro desafio está relacionado ao desenvolvimento de aplicações ubíquas. Aplicações ubíquas devem levar em conta as constantes mudanças dos requisitos dos usuários, da aplicação e da própria rede em seu estado de execução a fim de contornar ou amenizar o impacto dessas alterações em sua operação. Essas mudanças fazem com que as aplicações ubíquas necessitem realizar alterações em seu comportamento para se adaptar às novas condições de execução. Para tal, essas aplicações precisam ser cientes de seu contexto de execução. Um sistema é *ciente de contexto* ([Hoh 2006]), se ele usar o contexto para prover informações ou serviços ao usuário. *Contexto*, segundo [Springer 2006], é qualquer tipo de informação acerca do ambiente de execução de uma aplicação que possa

* Parcialmente Financiado pelo CNPq

ser usada para aprimorar seu comportamento fazendo-a operar de maneira adaptativa, personalizada, autônoma e flexível. Como não é desejável que o funcionamento de aplicações ubíquas seja interrompido para realizar adaptações nas próprias aplicações, é necessário que tais adaptações sejam feitas de forma dinâmica e transparente. Dessa necessidade, surge o conceito de adaptação dinâmica, que consiste em alterar o comportamento da aplicação, isto é, do código que implementa as funcionalidades providas para os usuários, durante sua execução de forma transparente ao usuário ([Zhang 2004], [Zhang 2005], e [Zhang 2006]).

Um segundo desafio relaciona-se à execução de aplicações ubíquas, particularmente no tocante a questões de segurança. Trata-se de como gerenciar os requisitos de disponibilidade, integridade e confidencialidade da aplicação em presença de um ambiente altamente dinâmico e heterogêneo que faz com que controles de segurança sejam diferentes conforme o contexto corrente. As propriedades dinâmicas e heterogêneas dos ambientes relacionadas ao contexto de segurança se referem aos requisitos da aplicação, da rede e dos dispositivos. Observa-se na literatura que a importância de prover requisitos de segurança adaptativos não é exatamente uma novidade ([Hinton 1999]; [Izquierdo 2007]). Entretanto, a proliferação de redes sem fio e seu uso por dispositivos portáteis impõem um novo patamar no tratamento da segurança. Uma mudança no ambiente de execução que acarrete um aumento do risco de incidentes de segurança (ex: aumento de vulnerabilidades), é razão suficiente para implicar um acréscimo de controles de segurança no próprio dispositivo (comportamento pró-ativo) de forma a garantir um ambiente seguro para as aplicações ubíquas. Um outro desafio relacionado à execução de aplicações ubíquas consiste em tratar eventos que não foram originalmente considerados durante o projeto de um serviço de segurança. Não é possível, em tempo de projeto, antecipar todas as respostas que um serviço de segurança ciente de contexto pode dar frente à ocorrência de eventos ainda não previstos, uma vez que novas vulnerabilidades são descobertas todos os dias.

Para superar esses desafios, todo o alicerce de sustentação dos requisitos de segurança das aplicações deve ser adaptado segundo o ambiente de execução, levando-se em conta propriedades dinâmicas e heterogêneas que reflitam os diferentes ambientes computacionais envolvidos. Isso envolve uma manipulação em tempo real de: (i) Controles de Segurança (mecanismos utilizados para mitigar os riscos encontrados em um contexto); e (ii) as próprias Políticas de Adaptação usadas para determinar quais são os controles de segurança adequados para cada contexto. O presente trabalho enfoca especificamente questões relativas à automatização das adaptações necessárias à sustentação dos requisitos de segurança das aplicações ubíquas. Esse trabalho apresenta um Serviço de Segurança Adaptativa, denominado Prometheus. O Prometheus busca adaptar em tempo de execução o conjunto de ***controles de segurança*** e as ***próprias políticas de adaptação*** durante a execução de aplicações ubíquas nos dispositivos móveis, de forma a garantir a disponibilidade, a confidencialidade e a integridade dessas aplicações. As adaptações são disparadas por mudanças de contexto e norteadas por políticas de adaptação previamente definidas. Assim, controles de segurança são definidos para cada contexto operacional e políticas de adaptação são determinadas para definir quais controles de segurança devem ser usados para um dado contexto. Uma forma de adaptar políticas de adaptação e controles de segurança em tempo de execução é permitir que tais políticas e controles sejam trocados devido a mudanças no contexto operacional. Vale ressaltar que a adaptação das próprias políticas de adaptação permite ao Prometheus tratar a imprevisibilidade em tempo de projeto de mudanças ambientais, por exemplo, ameaças que não existiam no momento da construção da aplicação.

No Prometheus foi adotada uma abordagem baseada em componentes uma vez que o modelo de componentes de software define abstrações que o tornam bastante adequado para a construção de sistemas dinamicamente adaptáveis [Canal 2006], minimizando os impactos da alteração na própria aplicação e nos controles de segurança e políticas de adaptação.

O artigo está estruturado da seguinte forma. A seção 2 apresenta os trabalhos relacionados. A seção 3 descreve a arquitetura do Prometheus. As seções 4 e 5 apresentam respectivamente o Gestor de Segurança Adaptativa e o Gestor de Política de Adaptação. A seção 6 descreve o estudo de caso desenvolvido para validar a proposta. A seção 7 apresenta a análise dos resultados. Por fim, a seção 8 contém as conclusões.

2. Trabalhos Relacionados

A proposta de [Hinton 1999] consiste em acrescentar controles de segurança gradativamente ao sistema toda vez que o mesmo estivesse sob algum tipo de ataque. Esta proposta se caracteriza por uma análise reativa face à ocorrência de falhas. Hoje, com a proliferação de redes sem fio, a segurança junto às empresas passa a ter um papel mais importante. Na abordagem empregada no Prometheus, uma mudança de contexto com um aumento do risco de incidentes de segurança já é suficiente para implicar um acréscimo de controles de segurança de forma a garantir um ambiente seguro para as aplicações. Ou seja, adota-se uma abordagem pró-ativa.

Em [Elkhodary 2007] são apresentados quatro tipos de abordagem para serviços de segurança adaptativos, porém todas são restritas ao nível de aplicação, diferentemente da abordagem empregada no Prometheus. Neste trabalho também é apresentada uma interessante taxonomia para classificação das abordagens adaptativas. Seguindo tal classificação, Prometheus é baseado em componentes e é sensível a contexto; sua reconfiguração é integral; e a resolução de conflitos, é autônoma e interativa (a resolução de conflitos está fora do escopo desse trabalho).

É proposto em [Izquierdo 2007] um serviço de criptografia adaptativo específico para Ecossistemas digitais, levando em conta as limitações de capacidade de processamento e requisitos de segurança de cada um desses ecossistemas. A proposta se concentra no uso de um modo de criptografia orientado a bloco (OCB) não encadeado, de forma a permitir que apenas blocos específicos de informação mais sensíveis sejam codificados. Ecossistemas digitais se adéquam perfeitamente como um caso de uso do serviço provido pelo Prometheus, onde a abordagem orientada a contexto adotada permite indexar diferentes níveis de criptografia. Já em [Chigan 2005] é proposto um *framework* para provisionamento de serviços de segurança auto-adaptativos em função da disponibilidade de recursos. Esse *framework* é capaz de empregar diferentes combinações de protocolos seguros para satisfazer diferentes necessidades de segurança de diferentes aplicações. Já o trabalho de [Doomun 2007] apresenta uma análise dos principais protocolos seguros passíveis de serem usados em uma rede IEEE 802.11i, usando modelos rigorosos de consumo de energia. Esses dois trabalhos fazem uso de monitores de recursos para inferir quanto à necessidade de adaptação da segurança, sendo que o primeiro é orientado a desempenho e o segundo é orientado a energia. A abordagem proposta no presente trabalho também faz uso de monitores, mas sugere o seu uso em conjunto com técnicas de caracterização do contexto para uma melhor avaliação da segurança necessária.

Em suma, o Prometheus trata de questões que não foram abordadas nos trabalhos levantados na área de segurança, tais como o comportamento adaptativo relacionado aos controles de segurança e as políticas de adaptação segundo um contexto operacional, e de estratégias de adaptação cientes ao contexto voltadas para requisitos de segurança.

Adicionalmente, esses trabalhos também não implementam controles de segurança e políticas de adaptação cujos códigos podem ser dinamicamente carregados, o que é uma importante vantagem, principalmente considerando dispositivos com recursos limitados.

3. Descrição do Prometheus

A arquitetura do Prometheus (Figura 1) consiste de um conjunto de componentes que provêem um serviço de segurança adaptativa ciente de contexto. A implementação desses componentes é feita por classes Java. O ***Ambiente de Segurança*** se refere ao conjunto de componentes do Prometheus, os quais são configurados dinamicamente de acordo com o contexto operacional e com os requisitos de segurança das aplicações. O ***Repositório de Requisitos*** de segurança consiste de um arquivo XML que contém uma relação de nomes de aplicativos catalogados no Prometheus, sendo que para cada aplicativo há uma lista de requisitos de segurança. *O* ***Repositório de Políticas de Adaptação (RPA)*** possui a relação de todos os adaptadores do acervo do Prometheus que satisfaçam aos requisitos de segurança. Cada adaptador possui cinco atributos: seu nome, lista dos requisitos de segurança satisfeitos por esse adaptador; tempo esperado para executar a adaptação, o tamanho do código do adaptador e a energia consumida em sua execução. A versão mais atual do RPA é mantida no servidor e replicada nos dispositivos. Qualquer alteração na versão do RPA mantida no servidor é interpretada pelo Prometheus como uma mudança no contexto de execução, e tratada de forma a garantir a consistência das versões nos dispositivos. O ***Gestor de Carga (GCarga)*** é responsável por obter instâncias do código executável das classes que implementam os componentes do Prometheus a partir do nome plenamente qualificado dessas classes. Quando solicitada a carga de uma classe na máquina virtual Java, o *GCarga* procura o código em um repositório local ao dispositivo. Se não for possível encontrá-lo, é feita uma busca no repositório remoto. A função do *GCarga* é eliminar a necessidade de manter no dispositivo todo o acervo de componentes, reduzindo a necessidade de espaço para armazenamento, o que é uma grande vantagem para os dispositivos limitados tipicamente usados em ambientes de computação ubíqua. Adicionalmente, tal funcionalidade traz a vantagem de prover maior flexibilidade ao sistema, já que novos componentes podem ser incorporados ao Prometheus em tempo de execução. Os ***Adaptadores*** são componentes capazes de executar modificações no Ambiente de Segurança para que um ou mais requisitos de segurança sejam atendidos.

O ***Simulador de Contexto (SC)*** simula o funcionamento de um *middleware* de provisão de contexto ([Sacramento 2004]; [Day 2000]). O *SC* foi criado para abstrair do Prometheus a complexidade de interagir com um middleware de contexto real, dessa forma simplificando todo o seu desenvolvimento. Seu objetivo é simular a ocorrência de mudanças nas informações de contexto e informar essas mudanças ao ***Provedor de Informações de Contexto (PIC)***. O *PIC* é o ponto de acoplamento entre o ***Gestor de Contexto de Outros (GCO)*** e o middleware de Contexto. A cada evento de mudança de informação de contexto, o *PIC* informa o *GCO* acerca da informação de contexto alterada. O *GCO* mantém uma ***Tabela de Informações de Contexto (TIC)*** contendo o estado de todas as informações contextuais, sejam elas referentes ao contexto de rede e do dispositivo, bem como referentes à versão do Repositório de Políticas de Adaptação e do Repositório de Requisitos de Segurança. O *GCO,* ao receber as informações atualizadas de contexto, atualiza a *TIC* e notifica o ***Gestor de Segurança Adaptativa*** de que houve alteração na *TIC*. O módulo ***Interceptador,*** por sua vez, é responsável por interceptar o início e o final de execução de aplicações controladas pelo Prometheus e chamar o ***Gestor de Contexto da Aplicação (GCA).*** O *GCA* trata do início e do final da execução de aplicativos. Antes que uma aplicação inicie sua execução, o *GCA* é invocado pelo Interceptador com o nome da

aplicação. O *GCA*, por sua vez, envia ao ***Gestor de Segurança Adaptativa (GSA)*** os requisitos dessa aplicação, obtidos do Repositório de Requisitos de segurança, para que este adéqüe o ambiente de segurança de forma que a aplicação possa ser iniciada. Se por algum motivo o *GSA* não puder efetuar as adaptações, ocorre um erro e a aplicação não é iniciada. Da mesma forma, ao final da execução de cada aplicação, o *GCA* submete os requisitos de segurança da aplicação que está sendo encerrada ao *GSA*, que fará os procedimentos necessários para ajustar o ambiente de segurança.

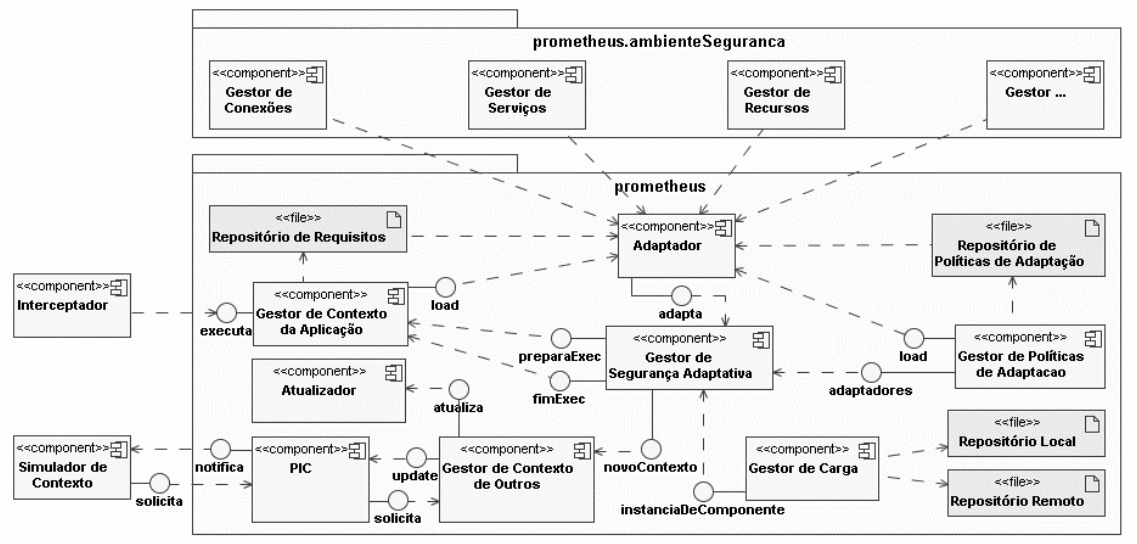

Figura 1 - Diagrama de Componentes do Prometheus

O ***Gestor de Segurança Adaptativa (GSA)***, componente central do Prometheus, é responsável por determinar, em tempo de execução, a política e os controles de segurança adequados ao contexto operacional, bem como por definir novas políticas de adaptação. O detalhamento de sua descrição se encontra na Seção 4. O ***Gestor de Política de Adaptação (GPA)*** é o componente responsável por selecionar um conjunto de ***Adaptadores*** que satisfaça aos requisitos de segurança das aplicações em uso, tornando o ambiente seguro para as mesmas. O *GPA*, através de uma máquina de inferência cujo funcionamento é detalhado na Seção 5, seleciona os adaptadores adequados para reagir à mudança do contexto, ou dos requisitos de segurança. Os critérios utilizados na escolha dos adaptadores dizem respeito aos requisitos de segurança das aplicações em uso e englobam, dentre outros: (i) a restrição no uso de serviços específicos (portas) e no uso simultâneo de serviços; (ii) a necessidade de acesso a determinados sítios; (iii) o uso de protocolos seguros de transporte (*SSL*), de rede (*IPSEC*) ou de enlace (IEEE 802.11i); (iv) o uso de um tipo especifico de rede (pública ou corporativa); e (v) a prioridade de um aplicativo específico em relação a outros aplicativos em uso no dispositivo. Além disso, são considerados ainda: (i) os *requisitos do dispositivo* **-** englobam o consumo de energia e o gasto de memória, ao serem aplicadas adaptações; e (ii) os *requisitos de rede* - englobam informações sobre os tipos de redes sem fio disponíveis no dispositivo e qual é o protocolo de segurança no nível de enlace adotado em cada uma dessas redes. Requisitos de rede e de dispositivos são estaticamente configurados no Prometheus.

O ***Gestor de Conexões*** busca proteger as informações em trânsito contra a modificação/divulgação por uma entidade não autorizada, ou seja, garantir a ***integridade e a confidencialidade*** da informação. Para garantir tais requisitos, adaptadores podem ser acionados dependendo do nível de segurança oferecido pelo ambiente onde o dispositivo está localizado (hostil - rede pública - ou não hostil - rede corporativa) e do nível de segurança oferecida pela rede (segura – em termos de presença de algoritmo de criptografia

- ou insegura). Os adaptadores acionados criam um sistema de túneis providos através de uma sessão SSH (*secure shell*) [Ylonen 2006] até o servidor, provendo a criptografia mínima para garantir os requisitos de segurança da aplicação e estabelecendo a melhor relação entre criptografia e consumo de recursos, tais como memória e energia. Quanto ao funcionamento do *Gestor de Conexões*, este aguarda que as aplicações se conectem aos servidores. A cada estabelecimento de conexão entre uma aplicação e seu servidor, o *Gestor de Conexões* intercepta a conexão e cria um sistema de túneis até o servidor usando o algoritmo de criptografia adequado.

O ***Gestor de Recursos*** é responsável pela gerência dos recursos dos dispositivos. Um requisito básico de segurança diz respeito à garantia de que determinado recurso esteja sempre "disponível", garantindo assim ***a disponibilidade*** da aplicação. Por exemplo, um adaptador pode ser acionado quando o dispositivo tiver pouca ***energia residual disponível*** (ultrapassa limiar definido)**,** desativando uma ou mais redes ou diminuindo a potência de transmissão, com o intuito de economizar energia e permitir que as operações da aplicação com o servidor corporativo sejam finalizadas. Um segundo adaptador pode ser acionado ao tomar conhecimento que o dispositivo possui pouca ***memória disponível*** (ultrapassa limiar definido), liberando memórias de dados não atualizados recentemente (por exemplo, pelo sistema de túneis), permitindo manter sempre memória disponível para o aplicativo corporativo. O ***Gestor de Serviços*** é responsável pela proibição e pela liberação de funcionalidades como o uso de portas TCP, programas de BATE-PAPO, MP3, entre outros. O ***Atualizador*** de Políticas de Adaptação e Requisitos é responsável por atualizar os Repositórios de Requisitos de segurança e o de Políticas de Adaptação inserindo, respectivamente, novos requisitos e novas políticas de adaptação. A atualização de políticas de adaptação e requisitos é tratada como uma mudança no contexto, ou seja, uma mudança de contexto ocorre todos os dias às 2 horas da manhã.

4. Descrição do Gestor de Segurança Adaptativa (GSA)

A execução do *GSA* é feita numa linha de processamento (*thread*) separada de outros componentes do Prometheus, pois o tempo para processar as adaptações pode ser significativo. A Figura 2 ilustra o diagrama de seqüência denotando as trocas de mensagens realizadas durante a execução de aplicações com o Prometheus.

O *GSA* recebe notificações do ***Gestor de Contexto de Aplicação (GCA)*** e do ***Gestor de Contexto de Outros (GCO)***. No caso do *GCO*, o *GSA* recebe notificações de alteração *na Tabela de Informações de Contexto (TIC)* indicando a ocorrência de modificações de contexto de rede ou do dispositivo. Em seguida, o conjunto de requisitos de segurança de redes e do dispositivo (denominado de CRO), é comparado com os valores da *TIC* para obter um ***Conjunto de Requisitos Não Atendidos*** (***CRNA***). No caso do *GCA*, o *GSA* recebe a solicitação para executar uma nova aplicação ou a notificação de final de execução de uma aplicação, juntamente com o conjunto de requisitos de segurança dessa aplicação. Em caso de solicitação ***de início de execução***, o *GSA* constrói uma nova ***Aplicação Representativa (AR),*** levando em conta os requisitos de segurança da "nova" aplicação. Define-se uma *AR* como sendo uma aplicação que representa, em termos de segurança, todas as outras que estão em execução no dispositivo. Assim, uma *AR* é representada por um conjunto de ***requisitos de segurança representativos*** (***RSRs***) onde um *RSR* é calculado de forma a atender ao requisito de segurança em questão para todas as aplicações em uso. Os procedimentos para construir uma AR são descritos ainda nessa seção. Em seguida, é verificado se existem requisitos de segurança conflitantes. Se existir, um erro é gerado. Caso contrário, o valor de cada *RSR* da *AR* é comparada com o valor do requisito correspondente da TIC para obter um Conjunto de Requisitos Não Atendidos (*CRNA*).

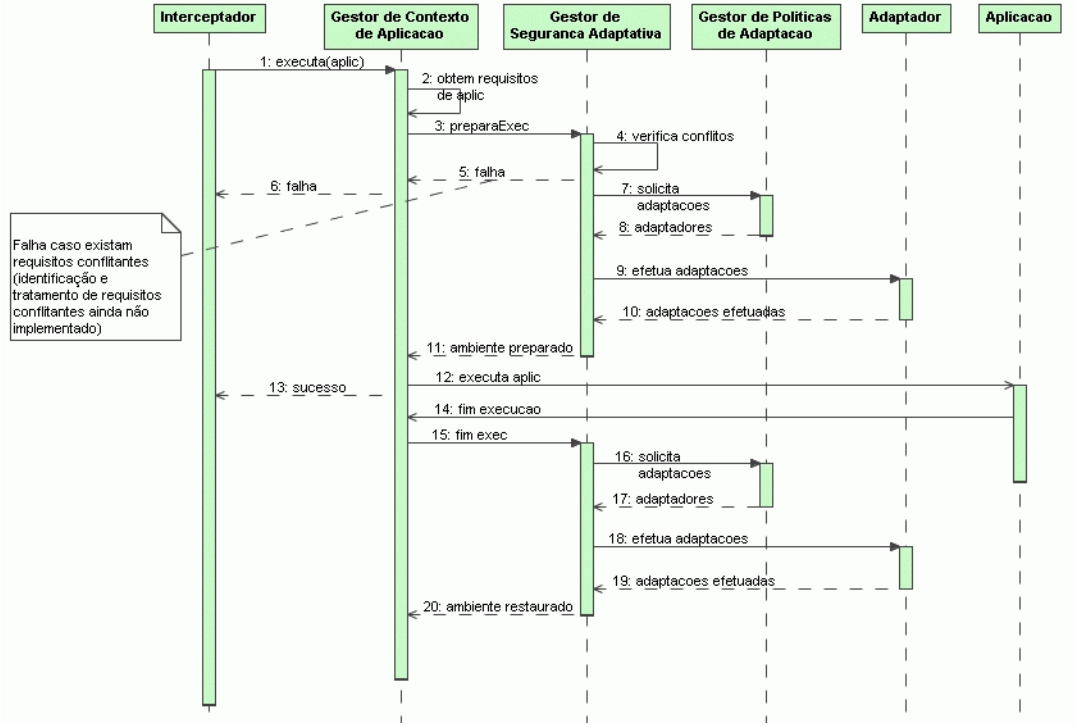

Figura 2 – Diagrama de Seqüência do contexto de Aplicação

Após obter o *CRNA*, tanto para o caso disparado pelo ***GCO*** como pelo ***GCA***, o ***GSA*** solicita ao Gestor de Políticas de Adaptação (*GPA*) uma lista de nomes de Adaptadores que atendem ao *CRNA*. Para cada nome da lista retornada, o *GSA*, através do Gestor de Carga, obtém uma instância do adaptador. Os Adaptadores retornados pelo ***GPA*** são executados pelo ***GSA***, ou seja, o método *adapta()* de cada adaptador é chamado para efetuar a adaptação. O resultado da adaptação é o ajuste do ambiente de segurança de modo a atender aos requisitos de segurança das aplicações. No caso específico de ***início de execução*** de aplicação, o ***GSA*** retorna o controle ao Gestor de Contexto da Aplicação (*GCA*), informando que a execução da nova aplicação pode ser realizada. Em caso de indicação de ***final de execução***, o *GSA* constrói uma nova aplicação representativa (*AR*) retirando os requisitos da aplicação cuja execução está sendo finalizada. Como visto, uma AR é representada pelo conjunto de ***requisitos de segurança representativos*** (***RSRs)***. A construção de cada ***RSR*** deve refletir os anseios sob o prisma de segurança de todas as aplicações em uso para o requisito em questão. A seguir, a título de exemplo, são descritas estratégias adotadas no Prometheus para a construção de um ***RSR*** de uma ***AR***. Na ***primeira estratégia***, têm-se RSR caracterizados pela proibição quanto ao uso de um recurso. Por exemplo, o valor do ***RSR*** (binário) relacionado à proibição quanto ao uso do serviço de MP3, denominado de ***RS1***, é obtido através do produto dos valores do requisito ***R1*** correspondente, relacionados a essa proibição (0 – proibido, 1 permitido), de cada uma das aplicações em uso. Em um segundo exemplo, o valor do ***RSR*** relacionado à proibição quanto ao uso do microfone, denominado de ***RSR2***, é obtido através do produto dos valores do requisito ***R2*** correspondente relacionados a essa proibição de cada uma das aplicações em uso. Um outro exemplo, o valor ***RSR*** relacionado à proibição quanto ao uso de um conjunto de portas TCP, denominado de ***RSR3***, é obtido através do produto dos valores do requisito ***R3*** correspondente relacionados a essa proibição, de cada uma das aplicações em uso. Na ***segunda estratégia***, têm-se ***RSRs*** caracterizados pela necessidade quanto ao uso de recursos. Por exemplo, o valor do ***RSR*** relacionado à necessidade quanto ao uso de um conjunto de portas, denominado de ***RSR4***, é obtido através da união dos valores do requisito

correspondente relacionados a essa necessidade, ***R4***, de cada uma das aplicações em uso. Em um segundo exemplo, o valor do ***RSR*** relacionado à necessidade quanto ao uso de um conjunto de endereços (URL), denominado de ***RSR5***, é obtido através da união dos valores do requisito correspondente relacionados a essa necessidade, ***R5***, de cada uma das aplicações em uso. Na ***terceira estratégia***, tem-se, por exemplo, um requisito de segurança ***R6*** de uma aplicação que está relacionado ao nível de criptografia exigido no uso do protocolo de transporte SSH pela aplicação. Nesse artigo assumiu-se que o protocolo já está em execução e permite utilizar três níveis de criptografia: fraca (DES), média (3DES), ou forte (AES). O ***RSR6*** está relacionado ao nível de criptografia exigida no uso do protocolo de transporte SSH pelas aplicações em execução. O valor de ***RSR6*** é obtido ao adotar-se o maior valor (algoritmo mais seguro) dentre os valores ***R6*** especificados para cada um das aplicações em execução.

5. Gestor de Política de Adaptação (GPA)

O ***GPA*** é responsável por escolher quais adaptadores devem ser executados de forma a tornar o ambiente seguro. Para tal, o GPA utiliza o Mecanismo de Seleção de Adaptadores que faz uso de um conjunto ***I de requisitos*** de segurança; do acervo ***J de adaptadores*** do Prometheus (*RPA*); e do conjunto ***A de aplicações*** controladas pelo Prometheus. Cada dispositivo móvel possui (i) um subconjunto de aplicações de A em uso em questão, e (ii) um subconjunto dos requisitos I de segurança, os quais podem, ou não, ser atendidos pelo contexto corrente (*TIC*). Os procedimentos que descrevem o funcionamento do mecanismo de seleção de adaptadores são agrupados em etapas detalhadas a seguir: (i) escolher adaptadores potenciais; (ii) calcular Função utilidade; e (iii) selecionar adaptadores.

5.1. Escolher Adaptadores Potenciais

Etapa responsável por identificar o subconjunto de adaptadores potenciais, do acervo de adaptadores do Prometheus, capazes de atender aos requisitos do *CRNA*. Para tal, essa etapa faz uso do Repositório de Políticas de Adaptação.

5.2. Calcular Função Utilidade

O cálculo da função utilidade é necessário apenas quando existe mais de um adaptador capaz de satisfazer a um dado requisito. Para o cálculo da ***Função Utilidade*** F_j adotou-se uma abordagem orientada à segurança: é dada prioridade aos adaptadores que tenham *maior relevância* do ponto de vista de segurança e, ao mesmo tempo, tenham *um menor impacto* para o dispositivo, ou seja, gerem o menor consumo de energia ou o menor tempo gasto para efetuar a adaptação. A ***relevância*** I_j de um adaptador j, se comparada com os outros adaptadores escolhidos na primeira etapa, depende do conjunto de requisitos de segurança não atendidos (*CRNA*) pelo contexto corrente e que podem ser atendidos por esse adaptador. Definem-se adaptadores relevantes (do ponto de vista de segurança) como sendo aqueles que atendem a mais de um requisito não atendido pelo contexto corrente. Assim, a relevância I_j de um adaptador j é estimada da seguinte forma: (i) calcula-se a diferença entre o número de elementos do *CRNA* e o número efetivo de requisitos atendidos por esse adaptador, e que não são atendidos pelo contexto corrente; e (2) essa diferença é dividida pelo número de elementos no *CRNA*. Atribui-se uma ***penalidade*** P_j a um adaptador *j* para cada requisito atendido pelo adaptador e não requisitado, estimada como segue: (i) calcula-se a soma do número de requisitos do *CRNA* com o número de requisitos atendidos pelo adaptador e não requisitados; e (ii) obtém–se P_j através da divisão do número de requisitos atendidos pelo adaptador e não requisitados, pela soma calculada em (i).

Com ***relação ao impacto*** do uso do adaptador para o dispositivo, busca-se por um adaptador com menor consumo de energia e maior rapidez no processo de adaptação. A execução de um adaptador é um dos fatores que afetam o consumo de energia W_j do dispositivo. A ***energia residual*** U de um dispositivo é definida como sendo a quantidade de energia atual que o dispositivo possui em um dado instante de tempo. Assim, o termo $(U - W_j)$ denota a energia final do dispositivo no caso de ser aplicado o adaptador j (energia inicial U menos o consumo de energia do dispositivo no processo de adaptação W_j). Em suma, busca-se escolher um adaptador j para a qual o termo $(U - W_j)$ que denota a energia final do dispositivo, em caso dele aplicar o adaptador j, seja maximizado. No que se refere à ***Duração Relativa da Adaptação (DRA)***, busca-se escolher um adaptador j no qual o tempo gasto para efetuar a adaptação D_j seja o menor dentre os adaptadores selecionados na etapa 1. Para tal, define-se a ***Duração Máxima de Adaptação (DMA)*** como sendo o maior tempo gasto para efetuar uma adaptação dentre os adaptadores do acervo do Prometheus. Assim, DRA é definido como sendo $1 - (DMA - D_j)$. Em suma, busca-se escolher um adaptador j para a qual o termo $1 - (DMA - Dj)$ seja maximizado. Logo, a ***Função utilidade*** $\boldsymbol{F_j}$ (Fórmula I) pode ser representada por:

$$F_j = a(I_j - P_j) + b(U - W_j) + q(1 - (DMA - D_j)) \text{ onde } \alpha + \beta + \theta = 1 \qquad \text{(I)}$$

O resultado final da etapa descrita na presente seção é um subconjunto de adaptadores potenciais com suas respectivas utilidades.

5.3. Selecionar Adaptadores

A meta dessa etapa do mecanismo de seleção é selecionar os melhores adaptadores fazendo uso da função utilidade obtida na etapa anterior. Em outras palavras, busca-se uma solução que maximize a função utilidade $\boldsymbol{F_j}$ de forma a escolher os melhores adaptadores possíveis dentre o conjunto de adaptadores obtidos na primeira etapa (melhores valores de utilidade), garantindo um ambiente seguro e, ao mesmo tempo, o uso eficiente dos recursos do dispositivo. Define-se uma função ***FMax*** (Fórmula II) como: $FMax = Max\sum F_j$. (II)

Devido à existência de múltiplos adaptadores que atendem a um mesmo requisito de segurança não atendido e, ao mesmo tempo, para evitar escolher adaptações redundantes, a segunda e a terceira etapas são executadas repetidamente até que não exista nenhum requisito no conjunto de requisitos não atendidos (CRNA). Assim, para cada rodada, após escolher o adaptador que maximize a função utilidade, são retirados do CRNA os requisitos satisfeitos. Adicionalmente, para cada rodada, são eliminados os adaptadores redundantes do conjunto de adaptadores obtidos na etapa 1 que satisfazem o mesmo requisito de segurança que é atendido pelo adaptador escolhido nessa rodada mas não satisfazem outros requisitos ainda não atendidos.

6. Estudo de Caso

Com o intuito de ilustrar a importância do Prometheus, nesse trabalho é descrito o caso de uso de aplicações do Grupo Capivara, que exigem segurança e cujo acesso à base central, localizada no servidor corporativo, pode ser efetuado por dispositivos móveis. Para agilizar os processos internos da empresa, o Grupo Capivara equipou seus melhores vendedores com *smartphones* com os quais eles podem ter acesso aos bancos de dados da empresa de qualquer local. Para garantir a segurança, o Prometheus foi instalado nesses *smartphones*. O Grupo definiu os requisitos de segurança para cada um dos seus aplicativos. Esses aplicativos podem ser executados em um dispositivo móvel, que pode estar localizado na empresa do Grupo, na empresa do Cliente, no trajeto do vendedor para a empresa Cliente ou

na casa do vendedor. A cada inicio de execução de uma aplicação, ou ao seu término, e a cada mudança do tipo de rede (corporativa ou não), o Prometheus deve realizar adaptações de forma a tornar o ambiente seguro. Os aplicativos referenciados nesse trabalho são: CLIENTES e VENDAS. O aplicativo CLIENTES fornece, entre outros, as visitas que um vendedor deve realizar por dia. Já o VENDAS permite que um vendedor registre suas vendas, dentre outras funcionalidades. Um outro aplicativo usado é o BATE-PAPO. Os requisitos de segurança dessas aplicações estão listados na Tabela 1.

Tabela 1 - AR e seus RSRs

R	AP1 CLIENTES	AP2 VENDAS	AP3 BATE-PAPO	*RSR*	AR (AP1 e AP2)	AR (AP1 e AP3)	AR (AP2 e AP3)	AR (AP1 AP2 e AP3)
R1	-	-	1	***RSR1***	-	1	1	1
R2	-	-	1	***RSR2***	-	1	1	1
R3	-	200	-	***RSR3***	200	-	200	200
R4	443	443	200	***RSR4***	443	443,200	443,200	443,200
R5	146.164.40.50	146.164.40.50	200.150.144.4	***RSR5***	146.164.40.50	146.164.40.50 200.150.144.4	146.164.40.50 200.150.144.4	146.164.40.50 200.150.144.4
R6	Média	Média	Baixa	***RSR6***	Média	Média	Média	Média

O vendedor sênior João foi um dos contemplados com um *smartphone*. Num dia típico de um vendedor, João, ao chegar à empresa, aciona o aplicativo CLIENTES para verificar quais são as visitas do dia. Usando o aplicativo, João consulta a base de dados de clientes através da conexão à rede WI-FI da empresa, e verifica que precisa visitar a empresa Glostora, seu melhor cliente. João sai do prédio com seu *smartphone* e pega um táxi. Ao se afastar do prédio o *smartphone* perde a conexão com a rede WI-FI e se conecta à Internet através de uma rede EVDO. Ao chegar à Glostora, João realiza a venda de um lote importante de mercadorias. Para registrar a venda, ele aciona em seu *smartphone* o aplicativo VENDAS. Após efetuar o registro da venda, João aciona o aplicativo BATE-PAPO, mas recebe uma mensagem de erro informando que deve encerrar o aplicativo VENDAS se deseja usar o BATE-PAPO. João encerra o VENDAS, e tenta novamente executar o BATE-PAPO, desta vez com sucesso.

Antes de ser iniciada a execução do CLIENTES acionado por João, o Prometheus: (i) obteve os requisitos de segurança dessa aplicação; (ii) verificou quais desses requisitos não foram atendidos pelo contexto de segurança do *smartphone*; (iii) selecionou os adaptadores; (iv) efetuou as adaptações necessárias no contexto; e (v) executou a aplicação. Os requisitos da aplicação CLIENTES, conforme a Tabela 1, são: necessidade do uso da porta 443 remota (R4); necessidade de acessar o endereço 146.164.40.50 do servidor corporativo (R5); e uso de criptografia média para acesso ao servidor corporativo (R6). Na Tabela 1 são ilustrados os valores dos requisitos de segurança para todas as possíveis Aplicações Representativas (representa as aplicações em uso) que o vendedor João executou em um dado instante.

A caminho da empresa Glostora, a troca de rede foi efetuada de forma transparente para o aplicativo CLIENTES. Ao detectar a mudança de uma rede para outra, o Prometheus registrou a mudança no contexto de segurança (rede corporativa x rede pública) e selecionou um adaptador. O adaptador selecionado mudou o nível de criptografia usado pelo aplicativo CLIENTES de média para alta. Em seguida, João executou o aplicativo VENDAS em seu *smartphone*. Os requisitos de segurança do VENDAS são: proibição do uso da porta 200 (R3); necessidade do uso da porta TCP remota 443 (R4); uso do endereço 146.164.40.50 do servidor corporativo (R5); e a exigência de criptografia média para conexões ao servidor corporativo (R6). Para satisfazer aos requisitos dessa aplicação,

apenas o adaptador para o requisito R3 foi selecionado uma vez que os requisitos R4, R5 e R6 já estavam atendidos. Para que a execução do VENDAS possa ser iniciada, o Prometheus efetuou essa adaptação. Após efetuar o registro da venda, João acionou o aplicativo BATE-PAPO. Os requisitos de segurança do BATE-PAPO são: uso do programa de MP3 (R1); uso do microfone do dispositivo (R2); uso da porta 200 (R4); uso do endereço 200.150.144.4 do serviço de BATE-PAPO (R5); e a exigência de criptografia baixa para conexões ao servidor corporativo (R6). Ao examinar os requisitos da aplicação BATE-PAPO, o Prometheus detectou um conflito entre o requisito R4 (uso da porta 200) e o Requisito Representativo R3 (proibição do uso da porta 200). O Prometheus então negou a execução do BATE-PAPO, enviando uma mensagem de erro. João após encerrar o VENDAS, executou novamente o BATE-PAPO, desta vez com sucesso, já que não havia mais o conflito de requisitos.

7. Análise dos Resultados

No âmbito do presente trabalho, o sistema proposto foi implementado na linguagem Java com o auxílio da ferramenta ECLIPSE. Para capacitar a carga dinâmica foi estendida a funcionalidade do carregador de classes (*ClassLoader*) disponível na JVM. Os experimentos do presente trabalho foram realizados num notebook Compaq Presario V2607CL conectado a uma rede sem fio com um único roteador do tipo 802.11g, o qual está conectado à internet, possibilitando assim que o dispositivo móvel possa se conectar a um servidor corporativo. Os experimentos foram configurados de modo ao Prometheus atender a uma ou mais aplicações em uso em um dispositivo móvel. O dispositivo móvel ao ser ligado inicia a execução do Prometheus. O Gestor de Segurança Adaptativa solicita ao Simulador de Contexto: (i) quais são as interfaces do dispositivo móvel e seus endereços e (ii) que o mesmo monitore todas as informações de contexto referentes ao dispositivo em questão. Nos experimentos realizados, as informações de contexto analisadas são: início e final de execução de aplicação, requisitos de segurança da aplicação, requisitos do dispositivo e requisitos de rede. Em seguida, o Gestor de Segurança Adaptativa lê o Repositório de Requisitos contendo a lista de aplicativos para os quais o Prometheus deve garantir segurança e seus respectivos requisitos de segurança.

Quanto às métricas usadas na avaliação do Prometheus, estas são detalhadas em seguida. A métrica ***Tempo para Selecionar Adaptadores (TSA***) para os requisitos de segurança exigidos é definida como sendo o tempo decorrido entre o instante que o Gestor de Segurança Adaptativa (*GSA*) solicita a adaptação ao Gestor de Política de Adaptação (*GPA*) e o instante seguinte ao envio da lista de adaptadores pelo *GPA*. O ***Tempo de Execução da Adaptação (TEA)*** é definido como sendo o tempo decorrido entre o instante que o Gestor de Segurança Adaptativa solicita ao Adaptador a execução das adaptações e o instante seguinte ao componente Adaptador retornar o controle de execução ao Gestor de Segurança Adaptativa. A métrica ***Tempo de Reação (TR)*** é definida como sendo o tempo de reação do Prometheus a uma adaptação acionada devido a uma mudança de contexto. Por exemplo, quando um dispositivo móvel passa de um ambiente seguro (rede corporativa) para um não seguro (rede pública), é necessário que o Prometheus reaja a essa mudança mudando o algoritmo de criptografia usado pelo túnel SSH para um mais robusto de forma a garantir os requisitos de confidencialidade e integridade da aplicação. A mudança do algoritmo é feita após uma negociação entre o Prometheus e o servidor SSH, efetuada através da troca de mensagens. Assim, ***TR*** é definida como sendo o tempo gasto para efetuar a negociação. É importante mencionar que durante o tempo de reação, os dados da aplicação podem ficar expostos a eventuais ataques. A métrica ***Volume de Dados Trafegados Durante a Reação (DR)*** mede o volume máximo de dados recebidos desde que

foi solicitada a troca do algoritmo de criptografia até que o novo algoritmo tenha sido adotado. Exceto os adaptadores relacionados ao Gestor de Conexão, todos os outros adaptadores analisados nesse trabalho não envolvem nenhuma negociação com o servidor (troca de mensagens) para que a adaptação possa ser realizada. Portanto, os valores do *TR e DR,* para esses outros adaptadores, são iguais a zero.

Nesses experimentos, para cada medida, foram feitas pelo menos 30 amostragens e computados a média, o desvio padrão e o intervalo de confiança de 95%. As medidas de tempo foram obtidas ao se calcular a diferença dos valores adquiridos ao invocar o método *currentTimeMillis()* no final e no início do que se deseja medir.

7.1. Tempo para Selecionar Adaptadores (TSA)

Nesse experimento é avaliado o tempo gasto para selecionar adaptadores que satisfaçam aos requisitos não atendidos, onde esse número de requisitos é variado de 1 a 8. Repetiu-se o teste quando o número de adaptadores que satisfazem a um mesmo requisito é variado de 1 a 3. Na Figura 3 a curva (a) mostra o *TSA* quando o número de requisitos varia entre 1 e 8 e têm-se 1 requisito satisfeito por adaptador. Observa-se que para os diferentes valores de número de requisitos, os valores do *TSA* não são significativos. Como esperado, obtém-se que o *TSA* é uma função linear que depende do número de requisitos não atendidos (*CRNA*). Na mesma figura, são ilustradas mais duas outras curvas quando existe dentre o conjunto de adaptadores potenciais um adaptador que atende a mais de um requisito. Observa-se que (i) o *TSA* tende a diminuir quando existe dentre o conjunto de adaptadores potenciais adaptadores que atendam a mais de um requisito; e (ii) o tempo para executar o algoritmo de seleção é muito pequeno (da ordem de microssegundos). Como o tempo de execução do algoritmo de seleção é muito menor que a resolução do relógio, foi necessário executar o algoritmo 100.000 vezes para fazer a medida.

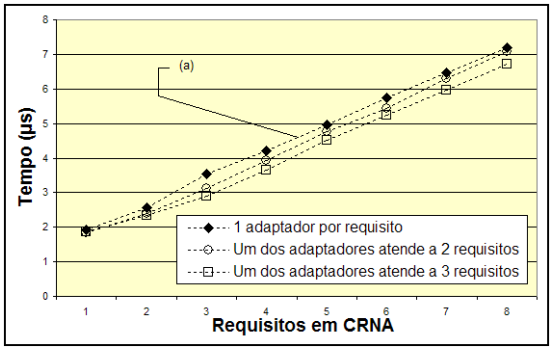

Figura 3 - TSA

7.2. Tempo de Execução da Adaptação (TEA)

Nesse experimento avaliou-se o tempo de execução apenas dos adaptadores relacionados ao Gestor de Conexão por esse envolver negociação com o servidor. Assim, a métrica *TEA* relacionada à adaptação que garante o estabelecimento de conexões TCP Seguras (*TCON*) através do mecanismo de túneis SSH é definida como sendo a soma dos tempos de Estabelecimento de Sessão SSH (*TSessão*), de estabelecimento do Túnel (*TTunel*), e de Conexão TCP através do túnel (*TTTCP*). *TSessão* é definido como sendo o tempo de conexão com o servidor SSH mais o tempo de autenticação e *TTunel* como sendo o tempo do cliente SSH solicitar ao servidor SSH a criação de um túnel e o cliente SSH receber a confirmação do pedido. Já *TTTCP* é o tempo para se estabelecer uma conexão TCP através de um túnel.

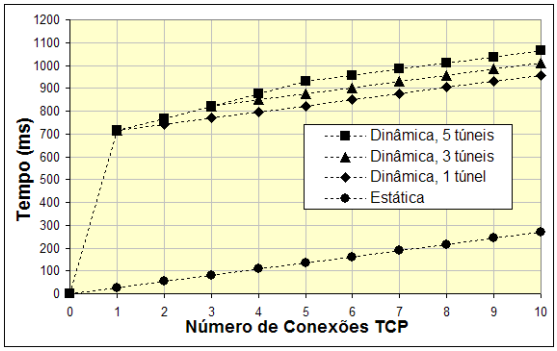

Figura 4 - TEA

Quanto ao momento de se estabelecer um túnel, definem-se duas estratégias: sob demanda ou estática. Na ***estratégia sob demanda***, os três passos envolvidos no cálculo no estabelecimento de conexões TCP Seguras através do mecanismo de túneis SSH são executados quando a aplicação solicita a primeira conexão TCP com o servidor corporativo. Os recursos (memória, *socket*, etc) não são alocados enquanto não houver necessidade de comunicação. Constata-se na Figura 4 que tal estratégia, se comparada com o estabelecimento de conexão TCP normal (*baseline*), impõe um retardo importante no estabelecimento dessa conexão. Com o intuito de reduzir tal retardo, adotou-se no presente trabalho a ***estratégia estática***. Nessa estratégia, os dois primeiros passos envolvidos no cálculo do estabelecimento de conexões TCP Segura são processados na inicialização do Prometheus de forma que os recursos permanecem alocados para o túnel, mesmo que o túnel não esteja sendo utilizado. A diferença entre o atraso gerado por essa estratégia e o estabelecimento de conexão TCP normal (*baseline*) é mínima, conforme a Figura 4, não impactando no tempo de resposta da aplicação.

7.3. Tempo de Reação

Através dos experimentos realizados, observou-se que na troca de algoritmo de criptografia: (i) o tempo de reação ou tempo de exposição da aplicação a eventuais ataques não é instantâneo; e (ii) durante essa troca de mensagens é possível que alguns dados trafeguem ainda criptografados com o algoritmo antigo. Nas condições de tráfego intenso, o tempo de reação é em média 620 ms, com picos de 1000 ms e, em condições de tráfego mínimo, esse tempo é de 272 ms. Adicionalmente, observou-se que em condições de tráfego intenso até três blocos de dados (contendo um total de 30.000 bytes) foram recebidos criptografados com o algoritmo antigo de criptografia. Já em condições de tráfego mínimo, todos os dados foram criptografados segundo o algoritmo de criptografia corrente.

8. Conclusão

Através de experimentos, constatou-se que, com o uso do Prometheus, o ambiente de segurança em dispositivos móveis é capaz de adaptar-se de forma dinâmica sem necessidade de injetar código nas aplicações. As aplicações são interceptadas apenas no início de sua execução para verificar se o ambiente do ponto de vista de segurança está propício para recebê-la e no seu término para relaxar a segurança quando não se faz mais necessária a garantia de certos requisitos de segurança. O tempo de intervenção do prometheus não é perceptível pelos usuários conforme se constatou nos experimentos realizados. Ou seja, a segurança adaptativa pôde ser provida pelo Prometheus sem onerar a aplicação, os seus desenvolvedores e seus usuários.

Referências

Canal, C., Murillo, J.M., Poizat, P. (2006) “Software Adaptation”. L’Objet, 12(1), 2006. Special Issue on WCAT’04.

Chunxiao Chigan Leiyuan Li Yinghua Ye (2005) “Resource-aware self-adaptive security provisioning in mobile ad hoc networks”, Anais da IEEE 2005 Wireless Communications and Networking Conference, WCNC 2005, pps 2118 - 2124 Vol. 4, USA.

Dey, A. et al (2000) “The Context Toolkit: Aiding the Development of Context-Aware Applications”. Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6, 2000.

Doomun, M et al (2007) “Adaptive IEEE 802.11i Security for Energy-Security Optimization”, Anais da The Third Advanced International Conference on Telecommunications, AICT 2007, pps 38 - 38, Mauritius.

Elkhodary et al (2007) “Survey of Approaches to Adaptive Application Security”, Anais do International Workshop on Software Engineering for Adaptive and Self-Managing Systems, 2007, ICSE Workshops SEAMS '07, pps 16 - 16, USA.

Hoh, S et al (2006) “Context-aware systems, 2006 - a primer for user-centred services”. BT Technology Journal, Volume 24, Issue 2 (April 2006), pp. 186 – 194, Kluwer Academic Publishers Hingham, MA, USA.

Hinton, H. et al (1999) “SAM: Security Adaptation Manager”, Anais da 15th Annual Computer Security Applications Conference, ACSAC '99, pps 361 – 370, Scottsdale, AZ.

Izquierdo Antonio et al (2007) “Providing Security for Digital Ecosystems with Adaptative Encryption”, Digital EcoSystems and Technologies Conference, DEST '07. pps 329 – 333, Australia.

Sacramento, V. et al., MoCA (2004) “A Middleware for Developing Collaborative Applications for Mobile Users”, ACM/IFIP/USENIX International Middleware Conference.

Springer T et al (2006) “Middleware Support for Context-Awareness in 4G”, International Workshop on Wireless Mobile Multimedia archive, Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks Environments, pps 203 – 211, IEEE Computer Society

Ylonen, T (2006) RFC 4251 The Secure Shell (SSH) Protocol Architecture.

Weiser M. (1991) “The Computer for the Twenty-First Century” Scientific American.

J. Zhang et al (2004) “Adding safeness to dynamic adaptation techniques”, in Proceedings of the ICSE 2004 Workshop on Architecting Dependable Systems, 2004.

J. Zhang et al (2005) “Enabling safe dynamic component-based software adaptation”, in Architecting Dependable Systems III, Springer Lecture Notes for Computer Science (A. R. Rogério de Lemos, Cristina Gacek, ed.), Springer-Verlag, 2005.

J. Zhang et al (2006) “Model-Based Development of Dynamically Adaptive Software” IEEE International Conference on Software Engineering (ICSE06), China, May 2006. IEEE.