Fênix Firewall System
Um Firewall Pessoal Sensível ao Contexto para Dispositivos Móveis

Marcos Alves T. de Azevedo, Vagner Sacramento

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
INF - Instituto de Informática
UFG - Bloco IMF I, sala 239 - Campus II - Samambaia
Caixa Postal 131 - CEP 74001-970 - Goiânia - GO

marcos@digitalsec.com.br, vagner@inf.ufg.br

Resumo. *A segurança de dispositivos móveis está se tornando rapidamente parte integrante de uma estratégia completa de segurança. O uso de um firewall pessoal que ofereça mecanismos de defesa em tempo real contra ameaças emergentes que atingem* smartphones *e outros dispositivos móveis é imprescindível contra invasões, vazamento de dados e perda de produtividade. A arquitetura de Firewall aqui proposta vai além dos sistemas tradicionais de firewall, por se utilizar da localização do usuário para carregar as políticas de segurança de acordo com suas preferências. Dessa forma, a arquitetura toma medidas de segurança baseado em políticas pré-definidas, gerenciando as preferências de usuário segundo a localização e alertando o usuário sobre potenciais perigos.*

1. Introdução

Na sociedade em que vivemos a necessidade de comunicação e informações em tempo real é cada vez maior. Para atender essa necessidade, a disseminação das redes sem fio e a popularização de dispositivos móveis estão tornando o meio em nossa volta cada vez mais pervasivo. No uso diário, os dispositivos móveis podem guardar informações pessoais preciosas ou informações comerciais sigilosas, que podem estar expostas a acessos não autorizados por pessoas maliciosas ou meros curiosos.

A facilidade de mobilidade proporcionada pelos dispositivos móveis os torna ainda mais vulneráveis a ataques, pois não há uma fronteira bem definida do perímetro que representa uma ameaça para o usuário final. O dispositivo móvel de um usuário pode ser alvo de ataques em diferentes ambientes e circunstâncias, por exemplo, no seu trabalho, no shopping, em uma festa, etc. Isto ocorre, principalmente, porque muitos dos dispositivos móveis tais como Smartphones têm suas interfaces de rede sem fio (e.g., Bluetooth, WiFi) habilitadas por padrão, e nem sempre os usuários sabem desabilitar tais interfaces ou restringir o acesso a um serviço executando em seu dispositivo. Tudo isto representa ameaças a privacidade dos usuários e a confidencialidade das informações armazenadas em seu dispositivo móvel.

Para auxiliar os usuários a configurar suas políticas de segurança em função da sua localização corrente, é proposto neste trabalho um Firewall pessoal chamado *Fênix Firewall System*. Este firewall pode impedir o acesso de pessoas não autorizadas e evitar que informações sejam extraídas através da exploração de serviços que estão em execução. O Fênix Firewall System foi projetado como um firewall pessoal que, quando instalado e configurado no dispositivo móvel, aumenta o nível de segurança e o controle do usuário, bloqueando ou permitindo as conexões entrantes ou saintes e notificando o usuário de conexões, portas abertas ou serviços suspeitos que estejam em execução no dispositivo móvel.

2. Visão Geral da Arquitetura

O Fênix é um firewall pessoal que carrega as políticas de segurança e preferências do usuário de acordo com a sua localização corrente. Através de uma interface, o usuário pode criar suas localizações para os ambientes que deseja, utilizando as coordenadas oferecidas pelo sistema *GPS*[1] integrado no seu dispositivo móvel. Assim, quando o usuário voltar a estas localizações já mapeadas, as políticas de segurança e preferências definidas por ele anteriormente serão carregadas automaticamente. Por exemplo, o usuário pode definir uma política de segurança para a sua residência, outra para o seu trabalho e outra para momentos de lazer.

Embora grande parte dos dispositivos móveis atuais ofereçam o recurso de *GPS* integrado, seu uso é geralmente restrito a ambientes *outdoor*. Para solucionar este o problema, o Fênix permite que o usuário crie referências abstratas, ou seja, o usuário cria manualmente uma localização, associa as políticas de segurança e suas preferências a esta localização e depois a seleciona sempre que necessário. Uma explicação mais aprofundada quanto à arquitetura do Fênix Firewall será feita nas próximas seções deste artigo, mas de forma bem sucinta o funcionamento do Fênix Firewall pode ser representado através do fluxograma 1:

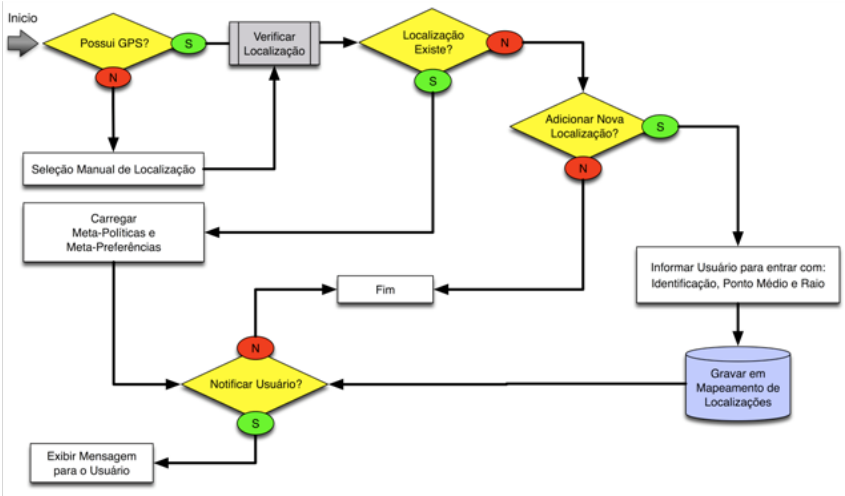

Figura 1. Fluxograma de Funcionamento do Fênix Firewall.

O fluxograma acima, apresenta as decisões que o *Fênix Firewall* toma nas situações de (I) Carregamento de política de segurança e preferências do usuário com base na localização corrente; (II) Mapeamento de uma nova localização e (III) Notificações, tendo o dispositovo móvel *GPS* embutido ou não.

3. Arquitetura Embarcada

A arquitetura embarcada ilustrada na Figura 2 apresenta todos os componentes que implementam os serviços providos pelo Fênix. Como os dispositivos móveis possuem recursos limitados, sendo a bateria um dos mais importantes e escasso, cada módulo da arquitetura embarcada, apresentados na Figura 2, foi projetado visando uma melhor utilização dos recursos computacionais disponíveis.

Os módulos são partes de código que são carregados somente quando solicitados por algum aplicativo ou dispositivo e descarregadas da memória quando não são mais

[1] Sistema de Posicionamento Global (GPS).

usados. Este recurso é útil por 2 motivos: Evita a construção de um *App Kernel* grande (estático) que ocupe grande parte da memória com todos os *drivers* compilados e permite que os módulos do Fênix Firewall ocupem a memória somente quando forem necessários.

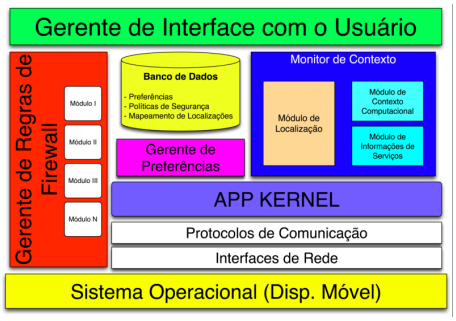

Figura 2. Arquitetura embarcada do Fênix Firewall

3.1. App Kernel

O *App Kernel*, núcleo do Fênix Firewall, foi projetado utilizando o paradigma de *exo-kernel* [Silberschatz et al. 2005]. *Exokernel* é um paradigma de implementação em que existe um *kernel* simples que faz a gerência dos recursos do sistema e um conjunto de bibliotecas que implementam a abstração de um sistema operacional. Assim sendo, os módulos do Fênix podem utilizar diretamente os recursos do sistema operacional instalado ou podem utilizar as bibliotecas de software fornecidas pelo *App Kernel*. A adoção deste paradigma, atribui mais flexibilidade ao programador, tornando o desenvolvimento do Fênix Firewall mais aberto e fácil de ser portado para outras plataformas de hardware, já que o Fênix e seus módulos estão fracamente acoplados ao Sistema Operacional do dispositivo móvel hospedeiro (*host*).

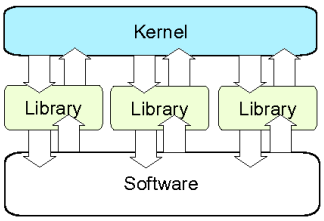

Figura 3. Diagrama de interação do Exokernel.

As funções atribuídas ao *kernel* do Fênix firewall são:

1. **Carregamento dinâmico de módulos sob demanda:** A gerência de memória do Fênix consiste em administrar partes da memória em uso e as que não estão, alocando memória livre quando os processos e módulos precisarem, liberando memória ocupada quando os processos forem finalizados.
2. **Entrada e saída com interfaces de comunicação**: Capacidade de registrar as conexões (*log*) e filtrar pacotes, permitindo ou negando as conexões.
3. **Habilitar e desabilitar as interfaces de rede:** O Fênix, por executar no modo privilegiado (*root*), interage através de chamada ao sistema (*syscall*) ao Sistema Operacional do dispositivo móvel habilitando ou desabilitando as interfaces de comunicação (ex.: interface serial, wireless, bluetooth etc) de acordo com a política de segurança que é carregada.

4. **Criar uma camada de abstração *(API)*:** Conjunto de funções acessíveis somente por programação, e que permitem utilizar características do software menos evidentes. Por exemplo, funções para criar janelas, exibir mensagens para o usuário no padrão do Fênix, utilizar os recursos de *GPS* do dispositivo móvel dentro dos padrões adotados pelo Fênix. Através desta abstração o Fênix Firewall tem sua implementação baseada em padrões pré-estabelecidos pelas interfaces de programação e pode ter suas funcionalidades estendidas através de novos módulos e plugins.

3.2. Monitor de Contexto

O *Monitor de Contexto* é o componente do Fênix Firewall responsável pelo mapeamento das localizações, por coletar as informações do contexto computacional do dispositivo móvel e por coletar informações sobre os serviços em execução e portas de conexão abertas no dispositivo móvel. Este componente se subdivide em três módulos: o *Monitor de Localização*, o *Monitor de Contexto Computacional* e o *Monitor de Informações de Serviços*, que são melhor explicados abaixo:

- **Monitor de Localização:** É o módulo responsável por inferir a localização do usuário. É através deste módulo que o usuário obtém as informações necessárias para mapear uma certa localização. É também este módulo que notifica o *App Kernel* quando o usuário entra em uma localização já mapeada, passando a identificação da localização para que as políticas de segurança e preferências do usuário sejam carregadas. O monitor de localização poderá utilizar-se do recurso de *GPS* quando disponível no dispositivo móvel, ou o usuário poderá utilizar-se do mapeamento manual, onde o próprio usuário cria e seleciona seus mapeamentos sem o auxílio de *GPS*.
- **Monitor de Contexto Computacional:** Responsável por coletar as informações do sistema, tais como: interfaces disponíveis, interfaces ativas e inativas, nível de bateria, quantidade de memória (em uso e disponível), módulos do Fênix em uso, frequência do processador, versão do sistema operacional, modelo, fabricante e outras informações pertinentes. Estas informações são importantes para que o Fênix Firewall possa, por exemplo, identificar se o dispositivo móvel possui ou não *GPS* integrado ou descarregar os módulos do Fênix que não estão sendo utilizados.
- **Monitor de Informações de Serviço:** Módulo que irá interagir com o sistema operacional e identificar os serviços que estão em execução no dispositivo, as conexões ativas e as portas abertas no Sistema Operacional do dispositivo móvel que estão ouvindo por conexões. Através destas informações o Fênix poderá não só alertar o usuário quando um determinado serviço entra em execução ou porta de comunicação aberta, como pode também prevenir de forma pró-ativa as tentativas de invasões por possível *backdoor* ou *trojans* que o usuário inadvertidamente tenha instalado no dispositivo móvel e que passarão a tentar estabelecer conexões ilícitas, afim de enviar informações do usuário ao atacante remoto.

3.3. Banco de Dados

O banco instalado no dispositivo móvel como parte integrante do Fênix Firewall irá armazenar as tabelas *"Políticas"*, *"PreferenciaUsuário"*, *"Mapeamento"* e os dados necessários para o funcionamento do Fênix Firewall.

1. **Tabela *"Políticas"*:** Onde são armazenadas as políticas de segurança que o usuário atribui para cada uma das localizações mapeadas.

2. **Tabela *"Mapeamento"***: Nesta tabela são armazenados todos os mapeamentos de localizações. Os mapeamentos por *GPS* que possuem uma Identificação, Latitude e Longitude, e os mapeamentos manuais onde só existe a Identificação da Localização. A Identificação da Localização é uma palavra ou frase de no máximo 15 caracteres incluindo espaço e sem caracteres especiais (Ex.: !, @, $, # etc).
3. T**abela *"PreferenciaUsuário"*:** Nesta tabela são armazenadas as preferências do usuário. Como preferências do usuário podemos citar a escolha da interface de rede padrão e o nível de notificação desejado.

3.4. Gerente de Regras de Firewall

Quando o usuário entra em uma localização já mapeada, um evento é disparado pelo *Monitor de Contexto*, notificando o *App Kernel*. O *Gerente de preferências* então é acionado pelo *App Kernel* para consultar a tabela *Política de Segurança* no banco de dados e verificar se existe alguma Política de Segurança associada àquela localização. Caso exista alguma Política associada àquela localização, o *Gerente de Preferências* repassa estas informações no formato de um *Meta-Dado* para que o *Gerente de Firewall* interprete e implante as regras. Um exemplo de uma política de segurança armazenada no formato de *Meta-Dado* no banco de dados pode ser vista no trecho código 3.1:

```
1  <?xml version='1.0'?>
2
3  <!-- Aceita e registra os pacotes vindos do endereco IP '192.196.1.0/24',
4       na interface 'eth0' direcionados a porta TCP '80'-->
5
6  <append>
7    <rule direction='in' source-ip='192.196.1.0/24' interface='eth0'>
8      <tcp destination-port='80' />
9      <accept />
10     <log />
11   </rule>
12 </append>
```

Código 3.1: Exemplo de uma Política de Segurança no formado de Meta-Dado (*XML*).

4. Politicas de Segurança Baseada em Localização

Muitos *Smartphones* vem dotados com uma interface *GPS* embutida, contudo por utilizar satélites para a inferência, a localização real do usuário pode se tornar pouco confiável ou até mesmo inviável para ambientes *indoor*. Isso se dá devido aos pontos de obstrução tais como paredes e outras interferências eletromagnéticas que podem prejudicar a qualidade do sinal *GPS*. Pensando nisso, duas formas de definir a localização do usuário podem ser utilizadas no Fênix Firewall:

4.1. Localização Manual

Embora muitos *Smarthphones* possuam uma interface *GPS* embutida, ainda assim alguns outros podem não ter esse recurso disponível. Em uma situação ainda mais frustante, o sinal *GPS* estaria indisponível e o usuário não teria conectividade para que o Fênix inferisse a localização do usuário e carregasse as políticas de segurança e devidas preferências. Em situações como essa, o usuário poderá lançar mão da localização manual, onde através de um *Menu*, conforme apresentado na Figura 4, o usuário consegue listar as localizações e assim carregar suas preferências e políticas de segurança armazenadas no *Banco de Dados*.

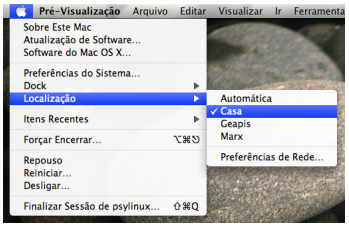

Figura 4. Seleção Manual de Localizações Mapeadas.

4.2. Ambientes Outdoor

Para ambientes outdoor poderá ser utilizado o recurso de *GPS*, quando presente no dispositivo móvel. No mapeamento realizado por *GPS* o usuário portando seu dispositivo móvel irá, através do menu do Fênix Firewall, selecionar a opção "*Mapear Nova Localização*" e em seguida será orientado a caminhar até o centro da localização que deseja mapear e pressionar "*OK*". Em seguida o usuário é solicitado a caminhar até a mais distante extremidade da localização que está sendo mapeada e pressionar "*OK*" novamente.

Nesta fase inicial do projeto, o Fênix Firewall é capaz de trabalhar apenas com áreas circulares. Dessa forma, o Fênix Firewall com as informações obtidas através do processo descrito anteriormente irá obter as posições (Latitude e Longitude) do *Ponto Médio* e *Raio*, gerando assim uma localização definida pelo usuário.

Ao finalizar o mapeamento, o usuário é solicitado a entrar com uma identificação para a nova localização, sendo que esta identificação deve estar dentro dos padrões apresentados na Subseção 3.3. A partir de então, o usuário poderá criar e associar uma política de segurança a nova área mapeada. No caso do usuário entrar em uma localização já mapeada, o *Módulo de Localização* irá identificar a localização e informar ao *App Kernel*, que por sua vez irá consultar o *Gerente de Preferências* quais as políticas e preferências associadas aquela localização e efetivá-las no *Gerente de Regras de Firewall*.

5. Conclusão

O Fenix Firewall System apresenta uma arquitetura modular, flexível e apropriada para ser instanciada em um dispositivo móvel. A arquitetura do Fenix apresenta boas contribuições comparada às propostas dos trabalhos correlatos [Qiu et al. 2004], [Chaokai 2007] e [Tan et al. 2007], pois o firewall proposto leva em conta a facilidade de mobilidade dos usuários para prover serviços de segurança aos mesmos. Através do firewall proposto, o usuário poderá associar politicas de segurança à localidades do seu interesse. Além disto, o Fenix oferece serviços de controle de acesso de um Firewall convencional controlando acesso a interfaces de rede, portas e serviços executando no dispositivo móvel.

Como trabalhos futuros, pretendemos desenvolver um protótipo para um dispositivo iPhone explorando as facilidades e o poder de controle e configuração de regras no Kernel do S.O. deste dispositivo, que segue o padrão do IPFilter, comum em plataformas BSD.

Referências

Chaokai, H. (2007). Design and implementation of a personal firewall based on ndis intermediate drivers. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*.

Cremonini, M., Damiani, E., di Vimercati, S. D. C., and Samarati, P. (2004). An xml-based approach to combine firewalls and web services security specifications. *Universitá di Milano - Italy*.

Hager, C. T. R. (November, 2004). Context aware and adaptive security for wireless networks. *Virginia Polytechnic Institute and State University*.

Qiu, Y., Zhou, J., and Bao, F. (2004). Design and optimize firewall for mobile networks. *IEEE Vehicular Technology Conference*.

Silberschatz, A., Galvin, P. B., and Gagne, G. (2005). *Operating system concepts. 7.ed.* Hoboken: Wiley.

Sinha, A. and Chandrakasan, A. (San Jose, November, 2001.). Energy efficient real-time scheduling. *International Conference on Computer Aided Design (ICCAD)*.

Snekkenes, E. (2001). Concepts for personal location privacy polices. *Norwegian Computing Center*.

Tan, H. C., Zhou, J., and Qiu, Y. (2007). A mobile firewall framework - design and implementation. *WCNC 2007 - IEEE Communications*.