Comparação de Aplicativos Livres para Android:

Primeira etapa do desenvolvimento do SECUREGENMOD.

MARINA SALLES

CAIO CANIC

CARYBÉ SILVA

MARCELO KOZUMA

MÁRCIO RIBEIRO[1]

RESUMO: Motivado pelas recentes denúncias de monitoramento da comunicação digital pela Agência de Segurança Nacional dos Estados Unidos (NSA), pela intensificação do monitoramento com finalidades comerciais por parte de empresas e pela popularização de celulares com acesso ininterrupto à internet, o projeto *SECUREGENMOD* surgiu com a pretensão de oferecer uma proteção contra a vigilância massificada dos meios de comunicação, buscando alcançar esse objetivo através do desenvolvimento de uma distribuição livre e segura do sistema operacional *Android*, a qual será baseada no *CyanogenMod*, contendo aplicações voltadas para a proteção da privacidade. Em sua primeira etapa de desenvolvimento, o projeto visa comparar e analisar aplicações compatíveis com o sistema *Android*, buscando selecionar as aplicações mais seguras e confiáveis. Nesse artigo discutiremos o processo de comparação das aplicações de troca de mensagens, armazenamento remoto e navegadores web,bem como a escolha destas pode influir positiva ou negativamente no contexto de segurança dos dados pessoais dos usuários e na proteção da privacidade.

Palavras-chave: android, privacidade, securegenmod, segurança.

ABSTRACT: Motivated by recent reports of monitoring of digital communication by the US National Security Agency (NSA), the intensification of monitoring for commercial purposes by companies and the popularization of mobile phones with uninterrupted access to the Internet, *SECUREGENMOD* project aims to offer a protection against mass surveillance of the communication media, seeking to achieve its goal through the development of a free and secure distribution of the *Android* operating system, which will be based on *CyanogenMod*, containing applications aimed at the protection of privacy. In its first stage of development, the project aims to compare and analyze applications compatible with the *Android* system, trying to select the most secure and reliable ones. In this article we will discuss the comparison process of messaging, remote storage and web browsers applications, and how the choice of these can influence positively or negatively in the context of users' personal data security and privacy protection.

Key-words: android, privacy, securegenmod, security.

RESUMEN: Teniendo por motivo las recentes denuncias de espionaje de las comunicaciones digitales llevada a cabo por la Agencia de Seguridad Nacional de Estados Unidos(NSA), por la vigilancia intensificada con fines comerciales por parte de empresas y la popularización de los teléfonos móviles con acceso ininterrumpido a Internet, el proyecto *SECUREGENMOD* vino con la intención de ofrecer una protección contra el espionaje masivo de los medios de

[1]Escola de Artes Ciências e Humanidades da Universidade de São Paulo

comunicación, tratando de lograr este objetivo mediante el desarrollo de una distribución libre y segura del sistema operativo *Android*, que es basada en *CyanogenMod* y contiene aplicaciones para ofrecer protección y privacidad. En su primera etapa de desarrollo, el proyecto objetiva comparar y analizar las aplicaciones compatibles con el sistema *Android*, tratando de seleccionar las aplicaciones más seguras y fiables. En este artículo vamos a discutir el proceso de comparación de las aplicaciones de mensajería, almacenamiento remoto y navegadores web y la elección de estos que influyen positiva o negativamente en el contexto de seguridad de los datos personales de los usuarios y la protección de la privacidad.

Palabras clave: android, privacidad, securegenmod, seguridad.

INTRODUÇÃO

Em julho de 2014 o Ministério da Defesa anunciou a implementação de um sistema seguro de comunicação digital em 14 unidades da administração pública federal. Desenvolvido pela *Serpro* (Serviço Federal de Processamento de Dados), o software chamado de Expresso V3, é uma resposta às recentes denúncias de monitoramento da comunicação digital pela Agência de Segurança Nacional dos Estados Unidos (NSA)[2]. A denúncia feita por Edward Snowden, funcionário terceirizado da agência, trouxe a público o esquema de monitoramento de comunicações entre bilhões de pessoas em todas as partes do mundo (Snowden 2013). Apesar de atingir diretamente milhões de cidadãos, a resposta do governo brasileiro à vigilância em massa foi a proteção restrita aos órgãos da administração pública.

Além de ser alvo da vigilância por parte de órgãos públicos de espionagem, a sociedade também sofre o monitoramento de empresas atrás principalmente de hábitos de consumo. A popularização de celulares com acesso intermitente a internet facilitou imensamente o cruzamento de dados sobre indivíduos: suas mensagens, ligações, locais por onde trafega e sítios que visita na web.

O projeto *SECUREGENMOD* pretende oferecer uma proteção à sociedade contra a vigilância dos meios de comunicação. Para tanto, pretende desenvolver uma distribuição segura e livre de um sistema operacional para celular. O sistema será baseado no *CyanogenMod*, uma distribuição livre do sistema *Android*, e conterá aplicativos que respeitem a privacidade dos usuários, bem como que protejam o anonimato e a segurança da comunicação. A primeira etapa do projeto, descrita neste artigo, consiste na comparação de aplicações compatíveis com o sistema *Android*, buscando selecionar os aplicativos mais seguros e confiáveis, que deixem o usuário menos vulnerável a monitoramento e roubo de dados efetuados tanto por terceiros quanto pelas próprias prestadoras de serviço dos aplicativos.

[2] http://www.brasil.gov.br/defesa-e-seguranca/2014/04/defesa-implanta-novo-sistema-de-comunicacao-digital

METODOLOGIA

Primeiramente foram selecionadas treze categorias de aplicativos: armazenamento remoto, navegadores de GPS, gerenciadores de arquivo, notas, tocadores de música, gravadores de som, teclados virtuais, visualizadores de documentos, navegadores de internet, mensageiros, exibidores de vídeo, leitores e gerenciadores de e-mail e comunicadores VoIP. Para cada uma dessas categorias foram considerados todos os aplicativos disponíveis na loja de aplicativos livre *F-Droid*[3], além dos aplicativos mais populares na loja *Google Play*[4] totalizando mais de 50 aplicativos. Os aplicativos foram comparados de maneira qualitativa em relação a usabilidade, funcionalidades, permissões de uso e licença.

O projeto foca em aplicativos com licenças livres, por uma série de razões. Além de garantir as liberdades do usuário em executar, distribuir, estudar e modificar o programa, softwares livres tendem a ser mais seguros. Primeiramente, uma aplicação segura não deve contar com o sigilo de nada que não seja fácil de ser alterado (Diffie 2003). A possibilidade de ler o código-fonte de uma aplicação permite a verificação de quais dados estão sendo enviados para servidores remotos e permite, mas não garante, que a aplicação seja livremente auditada. Por fim, vulnerabilidades tendem a ser resolvidas mais rapidamente em softwares livres (Wheeler, 2004).

Por padrão uma aplicação Android só pode ter acesso a uma parte limitada dos recursos do sistema, o qual fica responsável por gerenciar o acesso de tais aplicações a recursos que são protegidos por uma política de segurança baseada em permissões. O sistema *Android* e seus parentes livres, como o *CyanogenMod*, funcionam sobre o *kernel* do *Linux*. Em um sistema *Linux* os usuários funcionam em ambientes isolados incapazes de frustrar ou limitar um ao outro. Para garantir o isolamento das aplicações, o sistema *Android* cria usuários diferentes para cada aplicação. Tais usuários (em que as aplicações são executadas) têm permissões restritas para o uso de hardwares específicos[5].

O sistema garante que, para acessar determinados componentes do

[3] https://f-droid.org/

[4] https://play.google.com/store

[5] Veja https://source.android.com/devices/tech/security/

dispositivo (placa de rede, *bluetooth*, GPS, câmera etc.), a aplicação precisa requerer uma permissão de maneira explícita em um arquivo chamado "*manifest*". Dessa maneira, é possível facilmente verificar quais partes de hardware cada aplicativo é capaz de acessar.

Quando solicitada a instalação de tais aplicações, o sistema informa ao usuário as permissões requeridas pela aplicação e pergunta se deve continuar a instalação. Se o usuário concordar em continuá-la, o sistema entende que o usuário concordou em dar todas as permissões requeridas àquela aplicação. É preciso ressaltar que no sistema Android não é possível instalar um aplicativo com apenas parte das permissões exigidas; por exemplo, se um aplicativo de navegação na internet exige acesso aos dados pessoais do usuário, não é possível usar o aplicativo e ao mesmo tempo proteger seus dados de serem acessados por ele, logo, se um aplicativo instalado requer determinada permissão, ele terá essa permissão disponível a todo instante. Dessa forma, é recomendado avaliar criticamente a necessidade de conceder cada permissão a cada aplicativo antes de optar pela instalação do mesmo. Os usuários podem comparar aplicações e suas necessidades de acesso, identificando possíveis aplicações mal-intencionadas ou que exijam permissões não condizentes com suas características. Em (Lin et. al. 2014) os autores compararam milhares de aplicativos quanto a expectativa dos usuários em relação a suas permissões. Vários aplicativos que consideramos, porém, não foram considerados por esses autores por não estarem disponíveis na loja *Google Play*.

Além dos critérios mencionados acima (acesso livre ao código e permissões) consideramos diversos critérios menos objetivos como a frequência de atualizações. Como cada categoria de aplicação pode possuir um específico modelo de ameaças e consequentemente requerem medidas específicas de mitigação, adotamos critérios específicos para cada uma delas. O trabalho completo, disponível em https://gpopai.usp.br/wiki, contém uma comparação detalhada de várias categorias de aplicativos. Na seção apresentaremos o motivo das escolhas do *CyanogenMod* como base para o desenvolvimento da distribuição e do *F-Droid* como loja de aplicativos padrão, na seção seguinte selecionamos três categorias como modelo de comparação e concluímos o trabalho na última seção.

A ESCOLHA DA DISTRIBUIÇÃO E DA LOJA DE APLICATIVOS

CyanogenMod

Seguindo a metodologia descrita na seção anterior, optamos por distribuições do sistema Android preocupadas com a liberdade e a segurança do usuário e, portanto, consideramos duas possibilidades: o *CyanogenMod*[6] e o *Replicant*[7].

O *CyanogenMod* é uma distribuição baseada no projeto original do Android. Diferente das versões instaladas pelos fabricantes, essa distribuição não possui aplicativos de difícil remoção (*"Bloatwares"*). Durante sua instalação, é necessário obter *firmwares* proprietários da distribuição original do celular. Por esse motivo o *CyanogenMod* não pode ser considerado totalmente livre. O *Replicant*, por sua vez, não depende de tais partes proprietárias. Por outro lado, o *Replicant* é compatível com uma gama muito estreita de aparelhos o que dificultaria bastante sua adoção no Brasil. Julgamos que o *CyanogenMod* oferece o melhor equilíbrio entre liberdade, segurança e compatibilidade.

F-Droid

Sendo uma loja de aplicativos livres, o *F-Droid* era a escolha natural como padrão para o *SECUREGENMOD*. Mesmo assim, vale destacar outras características que consideramos importantes:

- Sua política de uso respeita a privacidade do usuário. Não é necessário cadastro ou identificações para poder realizar o download de aplicativos e ele nenhuma forma de rastreamento é feita. A única informação enviada ao servidor é a identificação da versão do *F-Droid*.
- Seus aplicativos são compilados do código fonte sempre que possível, o que os faz livremente auditáveis.
- Aplicativos que dependem de bibliotecas ou *plugins* proprietários são compilados sem essas partes.
- Aplicativos que enviam informações a servidores remotos são destacados, e o usuário é avisado na página de instalação do aplicativo.

[6] http://www.cyanogenmod.org/

[7] http://www.replicant.us/

• Há suporte para adição de outros repositórios fora dos oficiais, permitindo assim que aplicações provenientes de outros grupos possam ser distribuídas por meio de seus servidores.

COMPARAÇÃO DE APLICATIVOS

Armazenamento Remoto

Aplicativos de armazenamento remoto guardam pastas e arquivos em servidores remotos. Consideramos as seguintes aplicações para comparação: *Google Drive*, *Dropbox*, *SpiderOak*, *Wuala* e *OwnCloud*. Além dos critérios apresentados na seção Metodologia, para esta categoria analisamos a segurança da transmissão das informações entre o cliente (aparelho de celular) e o servidor remoto. As quatro primeiras são aplicações comerciais. O modelo de negócio delas é a venda de espaço para armazenamento em seus servidores. Todas elas oferecem uma quantidade variada de espaço gratuito e cobram valores variados para aumentar o espaço de armazenamento. Assim, devemos considerar não apenas a aplicação a ser instalada no aparelho de celular como também o servidor que se comunica com a mesma.

O *Dropbox* e o *Google Drive* são as aplicações mais populares na categoria. A comunicação entre cliente e servidor é feita de maneira segura (criptografada), porém a chave de criptografia fica no servidor. No caso do *Dropbox*, a aplicação é fechada (não possui código aberto) em ambas as pontas, tanto no cliente quanto no servidor. O *Google Drive*, por sua vez possui cliente livre (licença *Apache*), mas servidor fechado. O projeto *PRISM* da NSA, cujas informações foram vazadas e publicadas em uma reportagem do jornal inglês *The Guardian* (Greenwald & Macaskill 2013), mencionam a empresa Google como parceira e o *Dropbox* como possível futuro parceiro. De fato, a arquitetura de ambas favorece este tipo de espionagem.

Diferente dos dois anteriores, o *SpiderOak* e o *Wuala* oferecem criptografia dos dados no lado do cliente (*end-to-end*). Ou seja, as chaves de criptografia de ambos aplicativos não são enviadas para o servidor. As informações enviadas dessa forma não estão acessíveis mesmo aos administradores do servidor remoto. O *Wuala* é fechado em ambas as pontas, já o *SpiderOak* possui cliente livre.

O *OwnCloud* não oferece serviço de criptografia *end-to-end*, mas possui

cliente e servidor livres. Sendo livre do lado do servidor, ele dá aos usuários a liberdade para abandonar um servidor caso desconfiem que sua privacidade está sendo violada, e migrem para outro servidor com o mesmo serviço instalado. Existem ferramentas, como o *BoxCryptor* e o *Cryptonite* que poderiam ser usadas para criptografar as informações do lado do cliente antes de enviá-las para o servidor. A primeira dessas ferramentas é, porém, fechada e a segunda ainda está em uma versão instável.

Mensageiros

Mensageiros correspondem ao grupo de aplicativos capazes de enviar, receber e gerenciar mensagens entre usuários. Por tratar-se de uma categoria de aplicativos que lida direta e necessariamente com a comunicação interpessoal, analisamos principalmente mensageiros que se propõe a proteger os dados dos usuários. Para isso, tomamos como ponto de partida o material comparativo de mensageiros seguros disponível pela EFF[8] e incluímos os demais mensageiros presentes no *F-Droid*. Os aplicativos que se destacaram são: *ChatSecure*, *Telegram*, *Wickr*, *TextSecure* e *Kontalk*. Há também uma variedade de mensageiros que se utilizam de transmissão *bluetooth*, gerando uma rede local, e de estruturas de chat como o IRC (*Internet Relay Chat*), porém estes não possuíam entre suas preocupações a proteção dos dados pessoais dos usuários e não serão considerados nessa avaliação. Apesar da popularidade, não serão considerados aplicativos como *Facebook Messenger* e *Whatsapp*, pois estes não se propõem a proteger os dados pessoais dos usuários, por mais que o *Whatsapp* tenha se comprometido a utilizar um protocolo de criptografia ponta-a-ponta (Greenberb 2014), tal não aparenta ter sido implementado.

O XMPP é um protocolo livre implementado por muitos aplicativos de mensagens síncronas e, também, por uma ampla gama de servidores para esses clientes, permitindo aos usuários optarem pelos serviços que mais confiam. Porém, apesar do protocolo ser livre, ele ainda não implementa métodos de criptografia ponta-a-ponta, o que faz com que os servidores sejam pontos únicos de falha (“*single points of failure*”). Uma alternativa é o uso do protocolo OTR (*Off-the-Record Messaging*) que criptografa as mensagens na camada de aplicação para, então, enviá-las aos destinatários pelos servidores

[8] https://www.eff.org/secure-messaging-scorecard

XMPP, que implementam encriptações na camada de transporte. O protocolo OTR implementa muito mais que criptografia ponta-a-ponta, ele garante a autenticação dos usuários com negação plausível (“*plausible denyability*”) e “*foward secrecy*”. Negação plausível significa que quem recebe a mensagem tem garantia sobre o emissor dela, mas não pode usar isso como prova para um terceiro. *Forward secrecy* garante que, mesmo que alguém intercepte a chave de uma das partes, será capaz de ler apenas conteúdo daquela mensagem e não das mensagens anteriores.

O *Kontalk*, utiliza criptografia *OpenPGP* (*Open Pretty Good Privacy*) que garante a autenticidade e segurança das informações constantes nas mensagens, mas se limita ao utilizar chaves de longo período de duração e pela impossibilidade de se rejeitar a autoria de uma mensagem (Borisov, Brewer, & Goldberg 2004).

Os demais aplicativos utilizam protocolos diversos com características próprias. O *Wickr* possui código fechado e promete encriptar as mensagens localmente e ponta-a-ponta com *foward secrecy*, porém, como ambos os códigos, do sistema e do servidor, são fechados, não há forma fácil de confirmar. O *Telegram* destaca-se dentre os demais pela popularidade e por possuir um protocolo de criptografia próprio, o código de seu cliente é aberto, porém o do servidor é fechado, e como a criptografia das mensagens limita-se a sessões síncronas de "conversa segura", as conversas entre os usuários normalmente são visíveis ao servidor (Couprie 2013).

Por último o *TextSecure* se destaca por ser, além de um mensageiro assíncrono de cliente e servidor livre, um gerenciador de SMS que, se for definido como gerenciador padrão, pode criptografar as mensagens de texto armazenadas no telefone, além disso apenas o *TextSecure* e o *ChatSecure* possuem atualmente as funcionalidades de criptografia e autenticação ponta-a-ponta, "*forward secrecy*" e "*plausable deniability*". Além desses recursos, o *TextSecure* possui uma funcionalidade baseada no sistema do que foi por eles apelidado de "*future secrecy*" (Marlinspike 2013) que permite manter a consistência do fluxo de mensagens ao mesmo tempo que gera novas chaves a todo o momento, impossibilitando o comprometimento da privacidade de todo o registro de mensagens assim como de eventuais mensagens futuras, caso alguma chave decifrada.

Navegadores de Internet

Navegadores de internet (*browsers*) são aplicativos que permitem acessar sites e páginas na internet. Comparamos os seguintes aplicativos, escolhidos por sua popularidade: *Orweb*, *Mozilla Firefox*, *Lucid*, *Tint*, *Lightning*, *Zirco*, *Chrome*, *Opera Mini*, *UC Browser* e *Dolphin*. Na comparação consideramos três diferentes modelos de ameaças: a) os próprios navegadores enviarem dados pessoais do usuário para servidores externos; b) o construção de um perfil do usuário por um site não autorizado; e c) o monitoramento por parte da navegação por um terceiro.

Ataques do tipo (a) podem ser mitigados restringindo os dados a que o aplicativo tem acesso. Os navegadores livres *Orweb* e *Lucid*, por exemplo possuem permissão apenas para funções indispensáveis para um navegador, como acessar a internet livremente e armazenar arquivos no cartão de memoria. Assim, os dados e arquivos pessoais do usuário estão protegidos pelo sistema operacional, especialmente nas versões modernas do *Android*, em que há restrição de acesso aos arquivos no cartão de memória. O aplicativo *Zirco* requer, além disso, apenas acesso aos favoritos e histórico de navegação, e o aplicativo *Opera* requer essa permissão e mais a de criação de atalhos; podemos dizer que esses aplicativos acessam apenas dados pessoais criados por eles mesmos. Todos os outros navegadores requerem acesso a dados pessoais de outras fontes, como GPS.

Ataques do tipo (b) são feitos através de *cookies* HTTP, chamados coloquialmente de *cookies*. *Cookies* são arquivos de texto que são recebidos pelo navegador ao acessar sites na internet, e que podem ser enviados de volta para esses sites no próximo acesso. Esses arquivos contém um atributo "domínio", que define para que site o *cookie* deve ser enviado. Dessa forma, se o navegador guarda um *cookie* com o domínio "exemplo.com", ele só irá enviar esse *cookie* para um site que esteja sob o domínio "exemplo.com". Cookies são usados para identificar unicamente o usuário, mantendo uma sessão aberta através de páginas diferentes no mesmo site, e guardando as informações de navegação do usuário por esse site, inclusive preferências e "carrinhos de compras" usados em sites de compras. Assim, eles são uma parte fundamental do funcionamento moderno da internet.

Quando o site que cria o *cookie* possui um domínio diferente do especificado, dizemos que se trata de um "*third-party cookie*" ou *cookie* de terceiros. Em contraste, se o domínio especificado no *cookie* é o mesmo do site acessado, dizemos que ele é um "*first-party cookie*". *Cookies* de terceiros são usados para rastrear o usuário na sua navegação através de diversos domínios. Cada vez que o navegador recebe um *cookie* de terceiros, ele guarda a informação de em qual site o usuário esteve. Eventualmente o usuário pode acessar um site com o mesmo domínio dos *cookies* coletados, e então o navegador enviará todos os *cookies* para esse site. *Cookies* de terceiros põem em risco a privacidade dos usuários na medida em que facilitam o rastreamento e a coleta de informações sobre os mesmos. Por esse motivo, um dos nossos critérios de avaliação utilizado em nossa análise foi a possibilidade de bloquear todos os *cookies* de terceiros, de preferência por padrão.

Todos os navegadores avaliados aceitam todos os *cookies* por padrão. Dos navegadores livres avaliados, apenas *Firefox* e *Orweb* permitem algum controle sobre quais *cookies* serão aceitos: todos os outros oferecem apenas a opção de bloquear todos os *cookies*, o que torna esse bloqueio pouco prático. *Firefox* oferece três opções: aceitar todos os *cookies* (selecionada por padrão), aceitar apenas *first-party cookies*, e bloquear todos os *cookies*; nota-se que o aplicativo para Android oferece menor controle que o oferecido pela versão para PC, que inclui uma lista de exceções, ou lista negra. *Orweb* também oferece três opções: aceitar todos os *cookies* (padrão), aceitar apenas dos domínios na lista-branca, e recusar todos. *Orweb* não permite tratamento diferenciado para *cookies* de terceiros a lista branca se refere ao domínio registrado no *cookie*, independente do site de origem. Dessa forma, qualquer domínio listado recebe carta-branca para rastrear o usuário através da web. Além disso, atualmente *Orweb* contém um defeito na manutenção da lista-branca que pode comprometer o funcionamento do mecanismo.

O modelo de ataque (c) deve ser mitigado por meio de mecanismos de navegação anônima. Isso pode ser conseguido combinando determinados navegadores com o *Orbot* ou alguma outra aplicação de acesso a rede TOR.

Seguimos com uma descrição crítica dos navegadores, ressaltando suas fraquezas específicas.

Os aplicativos *UC Browser* e *Dolphin Browser* são aplicativos de código

fechado, e constam nesta lista apenas para fins de comparação. Ambos os aplicativos exigem uma grande quantidade de permissões delicadas. Em particular, ambos são aplicativos que podem iniciar automaticamente quando o dispositivo é ligado, e que têm acesso a dados pessoais do usuário (inclusive, no caso do *UC*, a arquivos anexados em e-mails), assim como a dados do sistema. Considerando que os aplicativos de navegação na internet podem se conectar à internet livremente, essa combinação de permissões abre a possibilidade de que informações sobre o usuário sejam enviadas sem que o mesmo perceba. Além disso, uma aplicação com tantas permissões se torna uma falha de segurança em potencial, especialmente quando é de código fechado. Em suma, consideramos esses aplicativos potencialmente vulneráveis a ameaças do tipo (a). O navegador *Opera*, em comparação, exige apenas permissões diretamente relacionadas com a função de navegação, e é de código aberto, o que nos leva a considerá-lo um aplicativo mais seguro.

Chrome é o navegador com maior número de usuários, por ser um aplicativo padrão nos sistemas Android mais recentes. Se assemelha bastante ao *Firefox*, tanto pelas funcionalidades quanto pelas permissões de acesso. O fato de ser um aplicativo parcialmente aberto, além de apresentar poucas configurações de gerenciamento da privacidade, mina as nossas confianças no serviço, por mais que este se apresente, como o *Firefox*, com uma imensa gama de extensões disponíveis para proteger a privacidade do usuário.

Lightning, *Lucid*, *Tint* e *Zirco* são aplicações de softwares livres, com poucas permissões e facilmente configuráveis. O navegador *Lightning* também pode ser configurado para se conectar através de *Orbot*. Por outro lado, tanto *Lightning* quanto *Tint* são aplicações com acesso à localização GPS, uma informação que pode identificar o usuário. O navegador *Zirco*, além de ser a mais leve das aplicações avaliadas, não tem acesso a GPS; a permissão não-essencial que essa aplicação requer é acesso ao histórico de navegação e favoritos. A desvantagem dessas aplicações em relação a outras mais populares, como *Firefox*, é a falta de extensões.

Os melhores candidatos na nossa análise foram os navegadores *Firefox* e *Orweb*. *Orweb* é uma aplicação livre voltada inteiramente para preservar o anonimato na navegação. Para usá-la é preciso instalar a aplicação Orbot, que redireciona o tráfego de internet pela rede TOR, com o objetivo de ocultar o IP

e a localização do usuário. Além disso, *Orweb* mascara as configurações de navegação ao interagir com servidores na internet, tornando-se mais dificil de identificar. Porém, *Orweb* falha no seu tratamento de *cookies*, como explicado anteriormente.

Firefox é o navegador mais popular dentre os livres disponíveis. Ele tem uma comunidade grande e ativa de desenvolvedores e usuários, com uma grande quantidade de extensões desenvolvidas para ele, inclusive extensões que aumentam a segurança do usuário, o que o torna um navegador interessante especialmente para o usuário que se dispõe a configurá-lo cuidadosamente. Por outro lado, o *Firefox* envia para seu servidor informações a respeito das ações do usuário. É possível configurar o navegador para não enviar esses dados, mas como estamos buscando privacidade por padrão, consideramos esse um ponto negativo. O controle de *cookies* do navegador deixa a desejar, mas atende o requisito mínimo de bloquear *cookies* de terceiros, ainda que não por padrão. Com o uso de extensões e configurações não-padrão, o *Firefox* pode se tornar um navegador bastante seguro.

CONCLUSÃO

Em resposta à descoberta de que o Brasil está sendo vigiado pela NSA, o governo brasileiro implementou medidas adicionais de segurança, incluindo um sistema unificado de comunicação online criptografada a ser utilizado em todos os órgãos governamentais. Por outro lado, a população em geral continuará vulnerável a espionagem e monitoramento. Se quisermos nos proteger da vigilância, é preciso que todos tenhamos acesso a um sistema de comunicações e uso da internet que seja seguro por padrão.

A contribuição do projeto *SECUREGENMOD* é o desenvolvimento de uma distribuição baseada no sistema *CyanogenMod* para celulares, que seja segura contra vigilância em massa por padrão. Como primeira etapa do projeto comparamos aplicativos de diversas categorias quanto as suas permissões, licença, funcionalidades e usabilidade para selecionar aqueles que serão incluídos na distribuição. Focamos principalmente em aplicações livres que podem ser auditadas por qualquer um. O produto desta primeira etapa do projeto estará em breve disponível em um sítio na web.

REFERÊNCIAS BIBLIOGRÁFICAS

Snowden, E. (2013). *Ein Manifest für die Wahrheit*, In Der Spiegel. Recuperado de http://www.spiegel.de/spiegel/print/d-119402581.html

Diffie, W. (2003). *Risky business: Keeping security a secret. ZDNet.* Recuperado de http://www.zdnet.com/news/risky-business-keeping-security-a-secret/127072

Wheeler, D. A. (2004). *Secure Programming for Linux and Unix HOWTO.* Recuperado de http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/index.html

Lin, J., Liu, B., Sadeh, N., & Hong, J. I. (2014). Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings. In Proceedings of *Symposium on Usable Privacy and Security*.

Greenwald, G., & Macaskill, E. (2013). *NSA Prism program taps in to user data of Apple, Google and others.* The Guardian. Recuperado de http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data

Greenberb, A. (2014) *Whatsapp Just Switched on End-to-End Encryption for Hundreds of Millions of Users.* Recuperado de http://www.wired.com/2014/11/whatsapp-encrypted-messaging/

Borisov, N., Brewer, E. & Goldberg, I. (2004). *Off-the-Record Communication,or, Why Not To Use PGP*. Recuperado de https://otr.cypherpunks.ca/otr-wpes.pdf

Couprie, G. (2013). *Telegram, AKA "Stand back, we have Math PhDs!"*. Recuperado de http://unhandledexpression.com/2013/12/17/telegram-stand-back-we-know-maths/

Marlinspike, M. (2013). *Advanced cryptographic ratcheting. WhisperSystems*. Recuperado de https://whispersystems.org/blog/advanced-ratcheting/