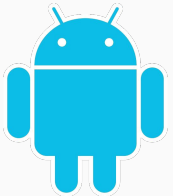
The Android Platform Security Model

René Mayrhofer
Jeffrey Vander Stoep
Chad Brubaker
Nick Kralevich

Em resumo, sobre o que fala o artigo?

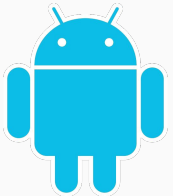
Crescente nos dispositivos android

- + 2 bilhões de dispositivos ativos mensais
- Tendência para a internet móvel
- Muitas aplicações de várias áreas
- Aplicativos críticos



O artigo aborda:

1) Motivação e definição de princípios de segurança no android
2) Definição do modelo de ameaça e como o modelo de segurança aborda
3) Explicação de como o AOSP (Android Open Source Project) impõe o modelo de segurança baseado em múltiplas camadas.
4) Identificação de lacunas abertas e potenciais melhorias



Contexto do Android

Contexto do Android

- Universo Android é imenso
- Qualquer dispositivo baseado em AOSP pode ser feito sem permissão
- Mudança demoram para serem efetuadas na maioria dos casos de uso
- Permissão de instalação por fonte arbitrárias

Contexto do Android

- Focado no usuário final
- Usuários, no geral, não são especialistas
- Qual cenário de sucesso para o android, em termos de segurança, privacidade e usabilidade?

Princípios de segurança do Android

- Safe by design
- Defense in depth
- Os princípios se aplicam a módulos, APIs, canais de comunicação e interface de todos os tipos

Estratégias abordadas:

- Isolamento e contenção
- Mitigação de exploração
- Integridade
- Correções/atualizações

Modelo de Ameaça

- Modelo de ameaça diferente do comum, desktop
- Facilmente perdidos ou roubados
- Se conectam em redes não confiáveis
- Estão muito próximos ao usuário

Deve-se supor o pior

Ameaças de acesso físico ou proximal:

T1) Dispositivos desligados sob controle físico de um adversário (potencialmente de alta sofisticação incluindo atacantes de nível estado-nação)

T2) Dispositivos bloqueados na tela sob controle físico completo de um adversário

T3) Dispositivos desbloqueados (compartilhados) na tela sob controle de um usuário autorizado, mas diferente, por exemplo abuso de parceiro íntimo

T4) dispositivos (tela bloqueada ou desbloqueada) na proximidade física adversário (com a capacidade assumida de controlar todos os canais de comunicação de rádio disponíveis, incluindo celular, WiFi, Bluetooth, GPS, NFC e FM)

Ameaças no nível da rede:

T5) Espionagem passiva e análise de tráfego, incluindo dispositivos de rastreamento dentro ou através de redes, por exemplo baseado em endereço MAC ou outros identificadores de rede do dispositivo.

T6) Manipulação ativa do tráfego de rede, p. man-in-the-middle in the transport layer security connections.

Ameaça em múltiplos níveis:

T7) Abusando APIs suportadas pelo sistema operacional com intenção maliciosa, por exemplo. spyware.

T8)Explorando bugs no sistema operacional, por exemplo kernel, drivers ou sistema serviços.

T9) abusando de APIs suportadas por outros aplicativos instalados no dispositivo.

T10)O código não confiável da Web (ou seja, JavaScript) é executado sem consentimento explícito.

Ameaça em múltiplos níveis:

T11) Imitando o sistema ou outras interfaces de usuário de aplicativos para confundir os usuários

T12)Lendo conteúdo do sistema ou de outras interfaces de usuário de aplicativos

T13) Injeção de eventos de entrada no sistema ou em outras interfaces de usuário do aplicativo.

Ameaças em múltiplos níveis:

T14) Explorando código que processa conteúdo não confiável no SO ou aplicativos, por exemplo nas bibliotecas de mídia

T15) Abusar de identificadores exclusivos para ataques direcionados (que pode acontecer mesmo em redes confiáveis), p. usando um número de telefone ou endereço de email para envio de spam ou correlação com outros conjuntos de dados, incluindo locais.

Modelo de segurança da plataforma Android

R1 - Consentimento de três partes

- Usuário, plataforma e desenvolvedor
- Os atores controlam o acesso aos dados que eles criam
- O consentimento não é apenas exigido do ator que criou um item de dados, mas de todos os atores envolvidos.
- Casos especiais

R2 - Ecossistema Aberto ao Acesso

- Não existe única loja de app
- A verificação central de desenvolvedores ou registro de usuários não é necessária
- App são livres para definir suas próprias APIs

R3 - Segurança é um requisito de compatibilidade

- Aparelhos que não estão em conformidade com o CDD e não passam no CTS não são Android
- Inicialização verificada e o atestado de chave de hardware pode ser usado para validar o firmware
- Rooting

R4 - O reset de fábrica retorna o dispositivo para um estado seguro

- Limpa/formata as partições de dados graváveis
- Retorna dispositivo para um estado que depende da integridade das partições protegidas
- Software do sistema não precisa ser reinstalado

R5 - Aplicações são objetos de segurança

- Apps Android não são agentes totalmente autorizados para ações do usuário
- Isso evita ransomwares e vazamento de dados privados
- Usuário tem menos poder, apesar de ser o usuário final

Implementação

Consentimento, exemplo

•Compartilha dados de um app para outro requer:

- consentimento do usuário através da seleção de um aplicativo de destino no compartilhar diálogo;

- consentimento da plataforma, arbitrando o acesso aos dados e garantindo que o aplicativo de destino não possa acessar outros dados além de o item explicitamente compartilhado através do mesmo link, que forma uma relação de confiança temporária entre dois aplicativos

Exemplo (...)

-consentimento do desenvolvedor do aplicativo de origem iniciando o compartilhamento com os dados (por exemplo, imagem) que eles desejam permitir aplicativo;

- consentimento do desenvolvedor do aplicativo de destino ao aceitar o compartilhamento dados;

Consentimento: Desenvolvedores

- Concedido pelo código escrito e executado pelo sistema
- Exemplo: flag no manifesto para poder debugar
- Chave de assinatura do desenvolvedor
- Side-loading está fora do modelo de segurança

Consentimento: Plataforma

- Assinatura também via código
- Atua para garantir que o sistema funciona como pretendido
- Aplicativos da própria plataforma também tem chave de assinatura

Consentimento: Usuário

- Evite pedir demais
- Solicitar de uma maneira que seja compreensível
- Prefira selecionadores e consentimento transacional em vez de ampla granularidade
- O sistema operacional não deve descarregar um problema difícil no usuário
- Fornecer aos usuários uma maneira de desfazer decisões tomadas anteriormente

Autenticação

- Função do Gatekeeper para que o sistema interaja com um usuário legítimo
- A principal autenticação é a tela de bloqueio
- Troca entre segurança e usabilidade
- modalidades de segurança, a partir do 9.0

Autenticação

- Primeira modalidade, restrita a fatores de conhecimento: por padrão, PIN, padrão e senha, fornece acesso às funcionalidades do celular
- Segunda modalidade, biometria: como íris, face
  - Não executa alguns tipos de ação, como descriptografar arquivos
  - Fallback para autenticação primária a cada 72h
- Terceira modalidade, weak biometrics: como desbloquear em certa localização ou quando conectar com bluetooth já pareado, etc
  - Todas as restrições da segunda modalidade + não acessa Keymaster
  - Fallback para autenticação primária após 4h inativas

Isolamento e contenção

- Imposição do modelo de segurança na máquina real
- Muitas das funcionalidades básicas providenciada pelo kernel Linux
- Controle de segurança se encontra na borda do processo
- Princípio da defesa em profundidade

Implementação do consentimento

- **Processos**: Discretionary Access Control (DAC)
  - Permissões rwx UNIX: grant URI permission, Intent flags
  - Inter-Process Communication (IPC): Android Interface Definition Language (AIDL)
  - Contornado por root

- **Plataforma**: Mandatory Access Control (MAC)
  - SELinux (originalmente desenvolvido pela NSA)
  - Atuação a nível de kernel
  - Não é contornado por root

Implementação do consentimento

- Permissões Android
  - Normal: permissões garantidas ao processo quando instalado **(Plataforma)**
  - Dangerous: usuário precisa conferir permissão ao processo **(Usuário)**
    - Granularidade e agrupamento
  - Signature: permissões obtidas somente por aplicativos assinados com mesma chave (**Processo)**

Sandbox: Aplicativos

- Base provida por Linux: usuários e privilégios rwx (DAC)
- User ID (UID) distinto para cada aplicativo
- Privilégios distintos para cada UID (portanto, para cada aplicativo)
- Diretório próprio para cada aplicativo, com privilégios garantidos somente para o seu UID

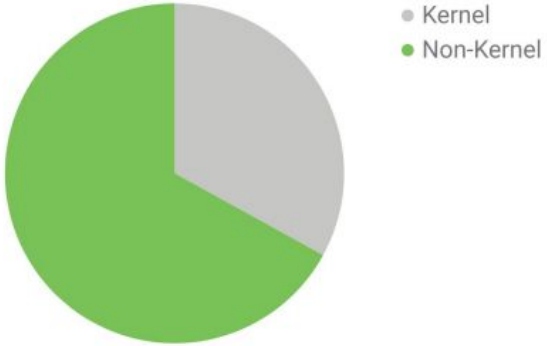
Sandbox: Sistema

- Processos do sistema considerados de alto risco também possuem UID próprio:
  - Processamento de mídia
  - Telefone
  - Wi-Fi
  - Bluetooth

Sandbox: Kernel

android

Kernel vulns in Android



Kernel accounts for ⅓ of security vulnerabilities on Android.

Data: Sep 2017 → May 2018 (Android Oreo)

Sandbox: Kernel

- Privileged eXecute Never (PXN): proíbe execução de código em userspace pelo kernel
- SELinux
- Privileged Access Never (PAN): proíbe acesso à memória de usuário pelo kernel (exceção: `copy-*-user()`)

Sandbox: Abaixo do kernel

- Última linha de defesa contra atacantes que tomem controle do kernel

- Proteção de dados críticos até mesmo no cenário mais capcioso

Sandbox: Abaixo do kernel

- Keymaster
  - Armazenamento de chaves de criptografia
  - Requer destravamento de tela por parte do usuário para acessar as chaves, mesmo sob situação de kernel comprometido
- Gatekeeper
  - Verifica autorização do usuário e comunica ao Keymaster
- Strongbox / Weaver
  - Funcionalidade idêntica a Keymaster / Gatekeeper
  - A diferença: localizado em hardware separado (proteção contra spectre, meltdown, etc)
- Protected Confirmation
  - Requer que o usuário aceite o uso de uma chave, mesmo com a tela destravada

Criptografia

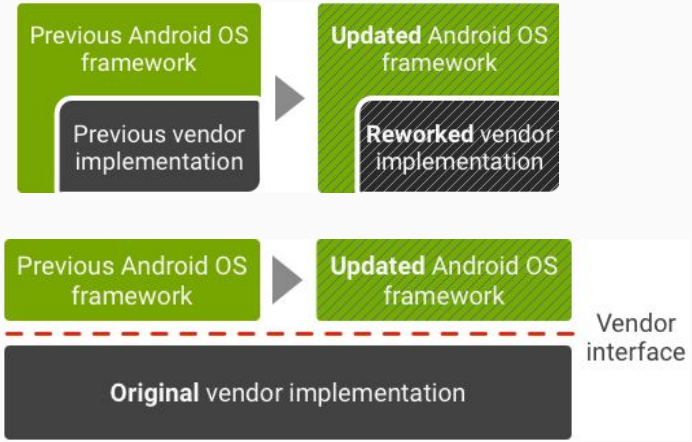
- Full Device Encryption (FDE)
  - Algumas funcionalidades críticas indisponíveis: removido em Android 10
- File Based Encryption (FBE)
- Criptografia de dados de rede
  - Posição assumida: todo tráfego de rede deve ser criptografado
  - Funcionalidades para reforçar o uso da criptografia
  - Requer opt-in explícito por parte do desenvolvedor para não usar criptografia

Patching

- Atualizações são necessárias para manter a segurança
- Dificuldades para distribuir atualizações devido ao ecossistema imenso

Patching

- Android Enterprise Recommended e acordos com OEMs: limite de 90 dias
- Project Treble: reorganização da arquitetura Android a partir de 8.0



Casos Especiais

Casos Especiais

- listing packages
- Aplicativos VPN podem monitorar/bloquear o tráfego de rede
- Backup
- Enterprise
- Proteção de redefinição de fábrica (FRP)

Conclusão

- Seguro por padrão
- Defesa em profundidade
- Equilíbrio entre usabilidade e segurança
- Alguns pontos de melhoria:
  - Ataques OEM
  - Ataques a nível de Hardware

Obrigado pela atenção!

Henrique Mariz e Lucas Lin

link para o artigo