SmartAuth: User-Centered Authorization for the Internet of Things

Joceleide Dalla Costa Mumbelli

Patrik Bedin Schettert

Introdução

- Tecnologias de Internet das coisas estão levando a uma nova era da automação residencial
  - Exemplos: Samsumg SmartThings, Google Weave e Brillo e Apple Home Kit
- Framework que utiliza cloud
- Permite a interação entre dispositivos
- Em geral comandados por aplicativo
- Problemas de segurança

Introdução

- Problemas no SmartApp (Samsung)
- Mais privilegio do que o necessário
- Ao atribuir permissão para uma função especifica, é garantido acesso ilimitado a todo o dispositivo e seus eventos
- Apps podem tirar proveito
- Sistemas de permissão por usuário geralmente não funcionam
- Dificultados de entendimento do usuário(exemplo acelerômetro)

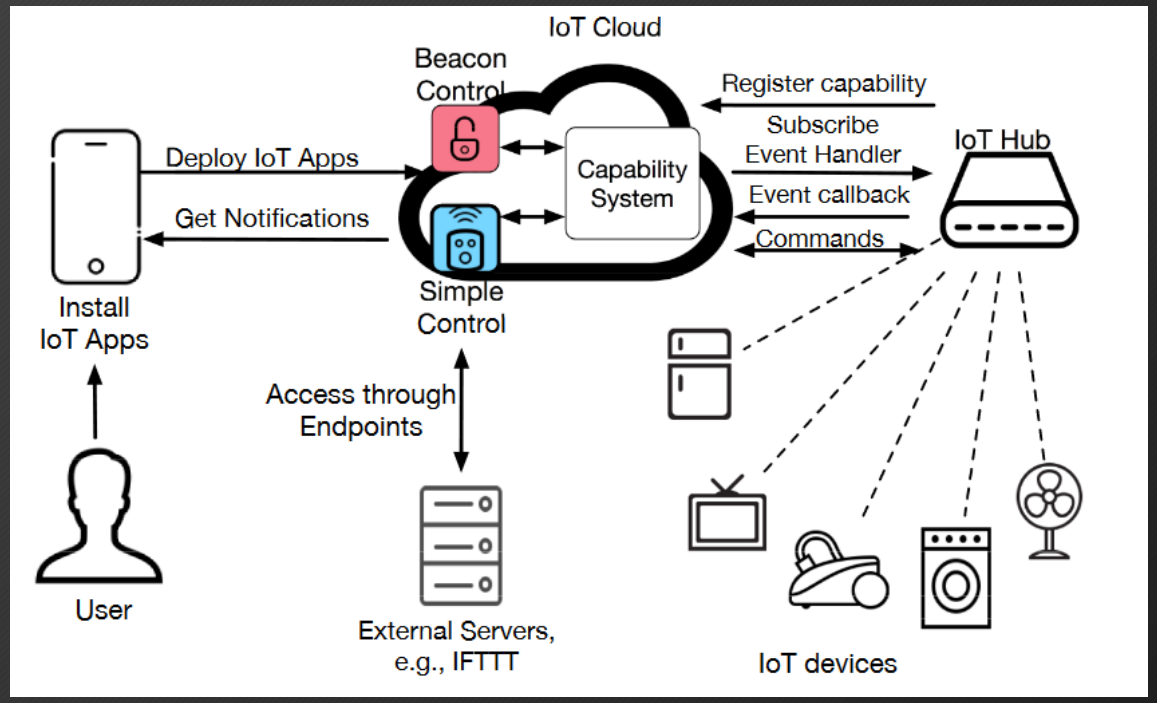
Objetivo

- Gap entre o que o usuário acredita que o app faz e o que realmente ele faz.
- Conceito de privacidade baseada em contexto
  - Exemplos: informação a respeito da saúde do indivíduo.
  - Aplicativo IoT monitoramento de temperatura

Proposto um novo sistema de autorização centrado no usuário para plataformas IoT

Contexto

- Automação residencial- receita de $100b até 2020
- Apps de terceiros podem acessar e controlar dispositivos
- Permite o cloud do fabricante a interagir com seus dispositivos localmente
- Atualmente usa o conceito de capacidades



Contexto

- Utiliza analisadores de linguagem natural(NLP)
- Identifica a descrição dos apps, contexto, substantivos e verbos
- Ferramenta Word2Vec
- Apps utilizam capacidades que não são claras para o usuário

- 16.7% dos apps para Samsumg SmartThings apresentam overprivilege

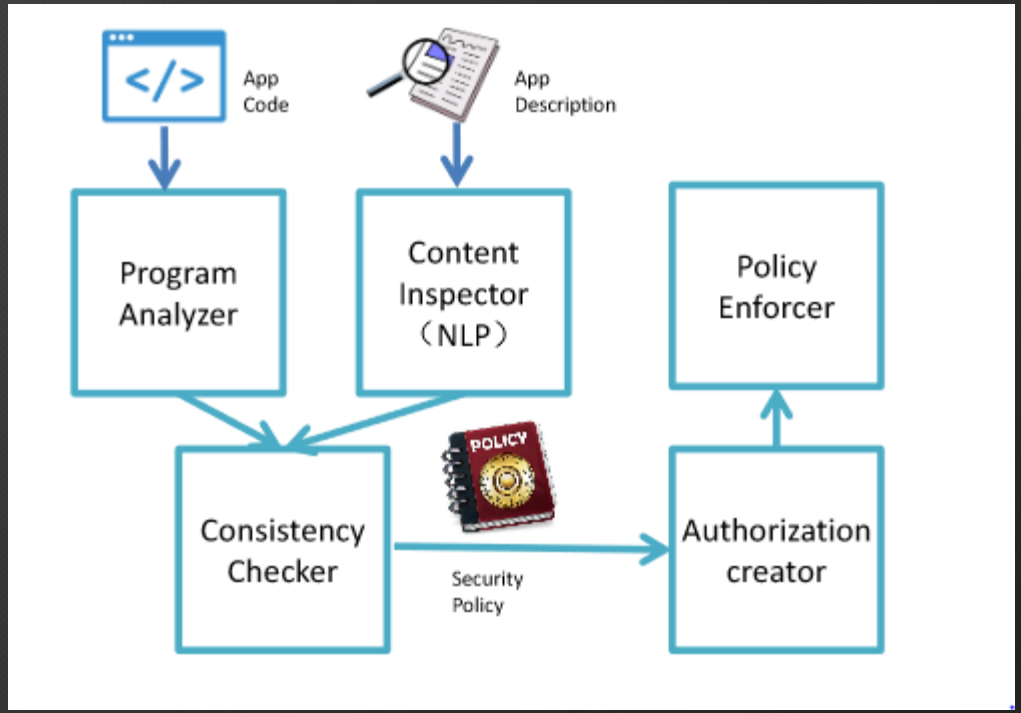
- 27 apps fazendo comunicação remota sem consentimento.
- Problema: roubar informações ou ganhar acesso indevido

SMARTAUTH

- Menos privilégios
- Específico para IoT – multi-dispositivos, baseado em contexto, operações automáticas
- Usável – evitar confusão do usuário
- Leve – não pode prejudicar performance
- Compatível - ser aplicável em diferentes plataformas

SMARTAUTH

- Analisador de código – extrai semântica criando o conjunto de privilégios que o app utiliza
- Inspetor de contexto – verifica por NPL os privilégios da descrição do app.
- Verificador de consistência- compara os dois, identifica discrepâncias
- Interface de autorização – mostra ao usuário as discrepâncias e permite a ele a decisão
- Policy Enforcer – implementa as politicas selecionadas



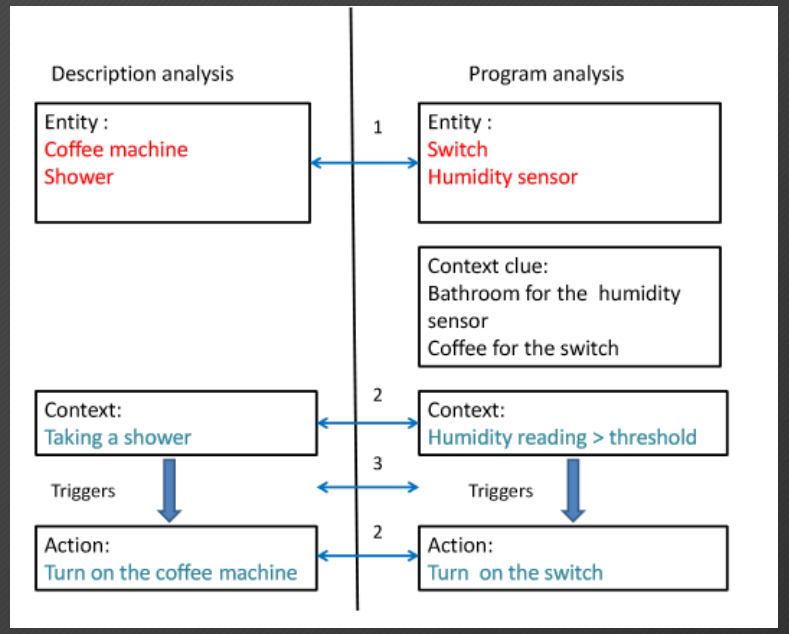
Modelagem e Implementação

- Coletado código fonte de 180 Samsumg SmartThings apps
- Analisa o código utilizando Abstract Syntax Tree(AST)
- Extraídas as “capacidades” pesquisando pelo termo.
- Analisa os comandos e atributos associados a cada requisição
- Analisa acesso remoto (envio de dados ou funcionamento como webserver). Groovyx.net.http, createAcessToken
- Comentários do código

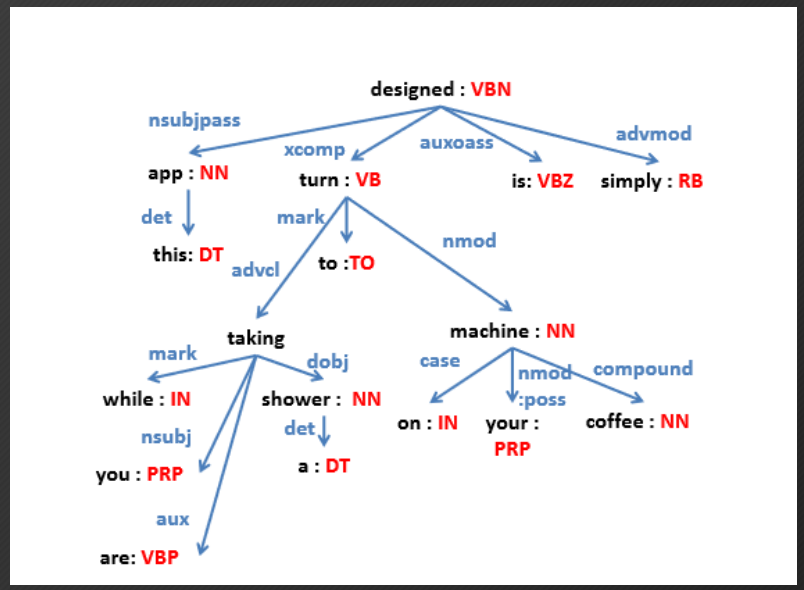
Avaliação da descrição

- NPL para extrair a politica de segurança da descrição em texto do app

1) Realiza correlação de entidades “Bathroom e “”coffe”
2) Correlaciona contexto e ação
3) Correlaciona politicas da descrição com da analise do programa



Café depois do banho



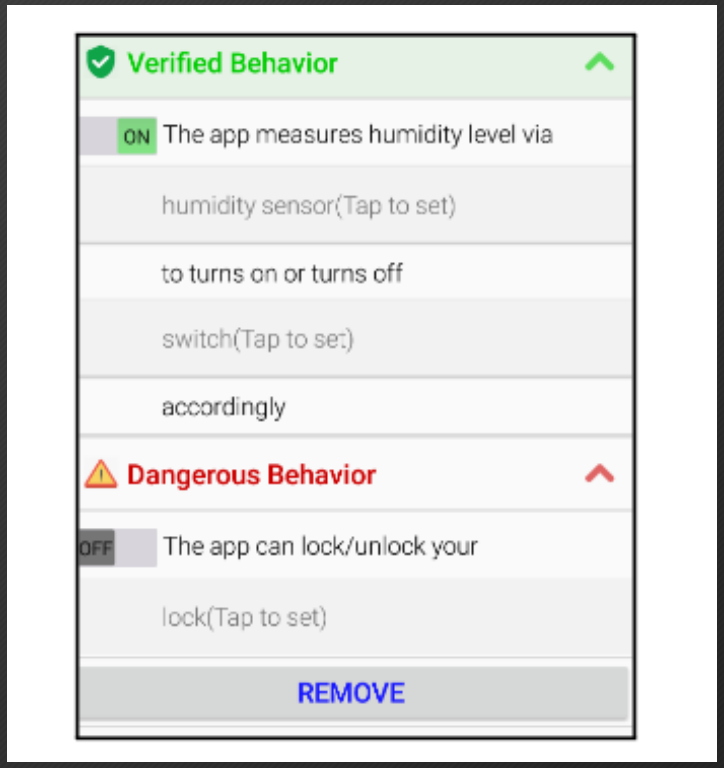
Análise NPL para aplicativo com descrição “This app is designed simply to turn on your coffe machine while you are taking a shower”

Interface de Autorização

- Pesquisa com usuários
- Levantamento de modelos mentais dos usuários
- Usuários se importam mais com funcionalidade do app (66% se importam muito) e privacidade (57% se importam muito).
- Diferentes percepções de risco (3.28 para abrir porta e 1.87 para verificar o nível de bateria)
- 69% acreditam que “capacidade” em automação residencial é mais sensível que permissões no Android

Interface de Autorização

- Usuários leem mais a descrição do que o processo de permissão
- Permissões presentes na descrição – garantia automática
- As discrepâncias são mostradas em um interface



Humidity alert app

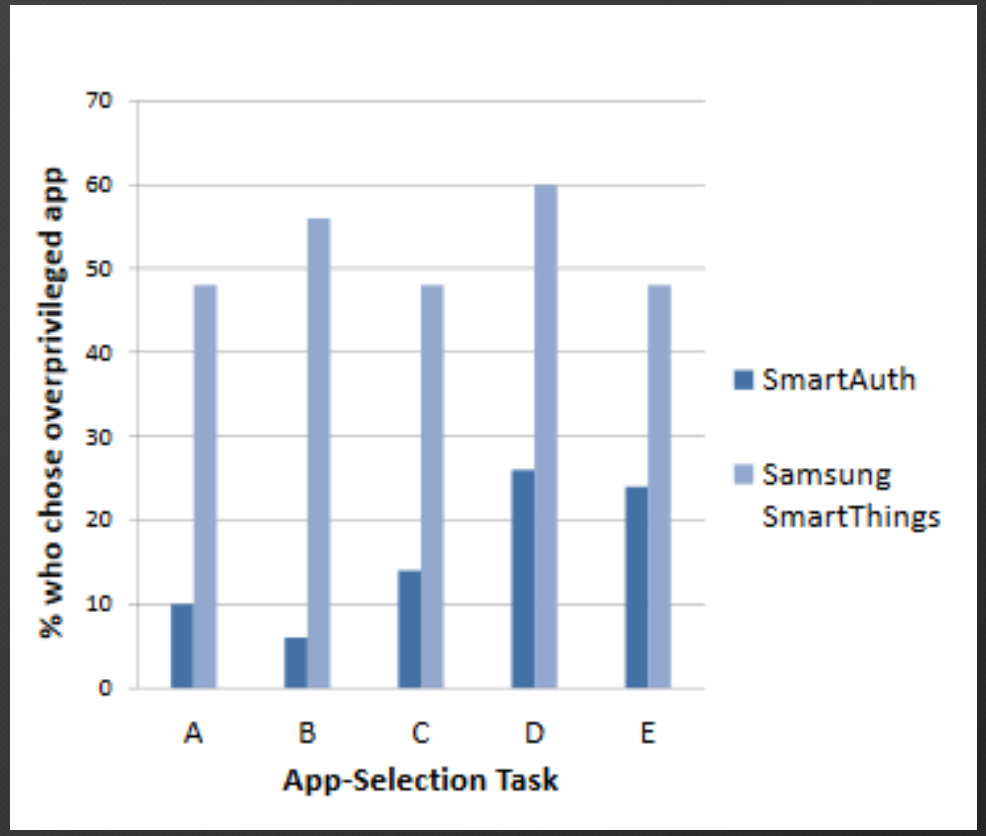
Garantia de Politica

- Bloqueia comandos não autorizados
- Bloqueia acesso a atributos não autorizados
- Módulo transparente para o app original
- Funcionamento:
  - Comando ou atributo -> verifica o banco de dados de autorização
    - Se autorizado : envia o comando para cloud
    - Else : retorna mensagem de erro para o app

Avaliação

- 100 participantes
- Usuarios de smartphone e conhecimento sobre automação residencial

- Tarefas A e B – privilegio potencialmente perigoso (abrir porta)
- Tarefas C-E – privilegio potencialmente menos perigoso (ler temperatura)



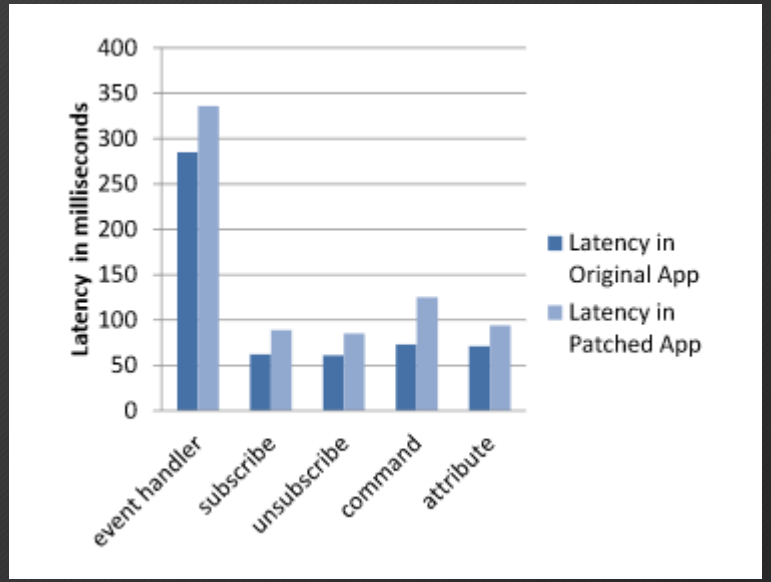
Avaliação

- Perguntas:

- “I feel that the app interface explains thoroughly why the app can access and control these sensors/doors” – SA 4.06, ST 2.4
- “I feel confidente to make decision wheter or not to install the app after Reading de interface”- SA 4.12, ST 2.46
- “It is difficult to find the information from interface” – SA 2.72, ST 3.56

Diferença de performance

- Testados 180 apps open-source
- Pré- processamento (analise de código, NPL, etc) – 10.42seg
- Run-time – gráfico
- Nenhum dos 180 apps deu “crash” quando aplicado o patch
- Relacionado ao acesso remoto, apenas 6 apps sofreram com o patch



Limitações

- Desenvolvedor pode usar métodos personalizados ou nomes proprietários para enganar a analise
- Pode modificar a descrição do aplicativo para que o SmartAuth identifique erroneamente um comportamento malicioso
- Adicionar capacidade para lidar com dispositivos externos
- Sempre terá uma opção de app alternativa ?
- Assumir que o usuário sempre le a descrição?

Conclusão

- Identificado gap entre o que os usuários pensam de um aplicativo e o que ele realmente faz.
- Nova abordagem de autorização
- Avaliação desse abordagem mostrou resultados eficientes comparados ao cenário atual.

SmartAuth: User-Centered Authorization for the Internet of Things

Joceleide Dalla Costa Mumbelli

Patrik Bedin Schettert