Разграничение доступа и минимизация ущерба от атак с помощью сильного принципа наименьших привилегий

с.н.с. ВМК МГУ, к.ф.-м.н.
Д. Ю. Гамаюнов

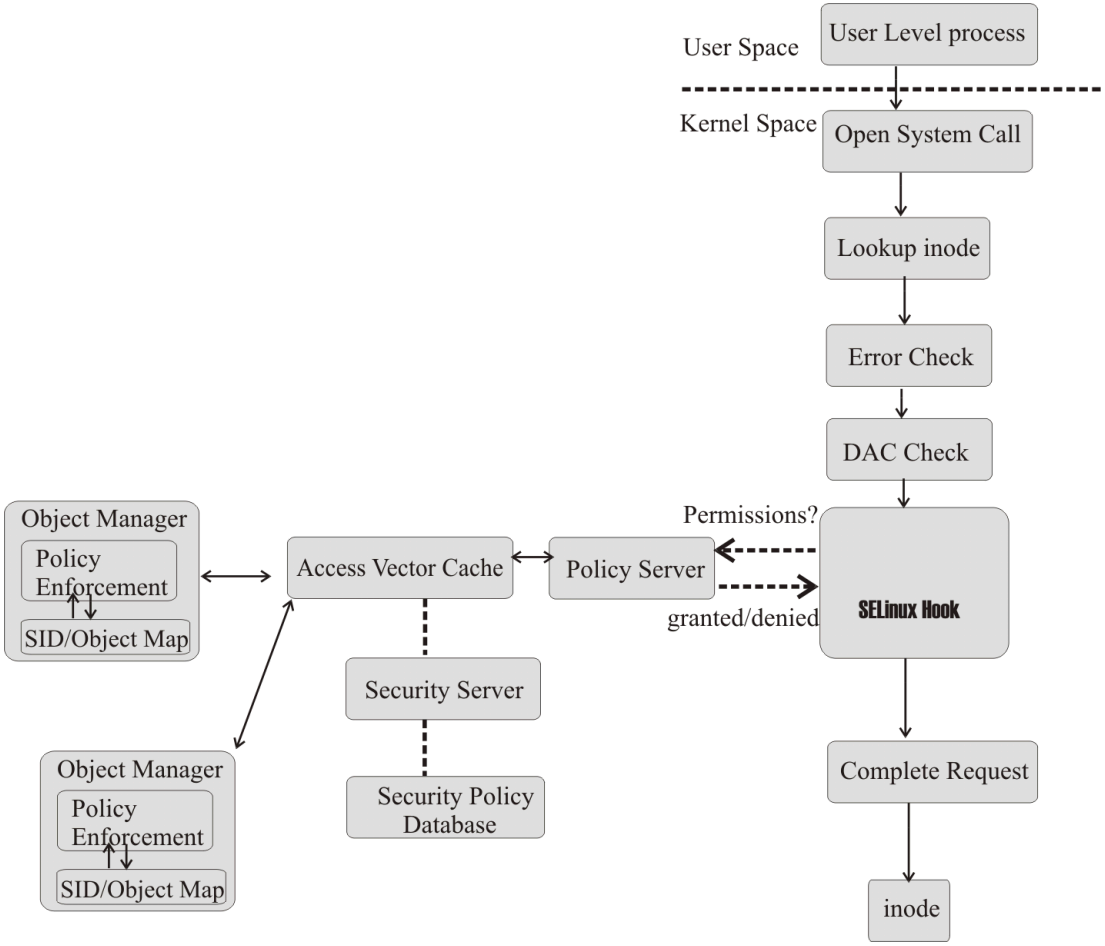
Минимизация ущерба от атак

- Сетевые черви
- Направленные атаки на уязвимости «порчи памяти»
- Drive-by-downloads
- Ключевые области:
  - Безопасность мобильных платформ
  - Безопасность серверного сетевого ПО

Принцип наименьших привилегий

- **Приложение должно иметь ровно столько полномочий, сколько ему требуется для корректной работы**
  - Код атаки выполняется в виртуальном адресном пространстве атакованного приложения
  - Выполняется с правами атакованного приложения
  - Ограничение прав приложения по доступу к ресурсам до необходимого минимума снижает возможный вред от успешного использования уязвимости
- Примеры: SELinux, AppArmor, Tomoyo

Security Enhanced Linux (SELinux)



Усиление принципа наименьших привилегий

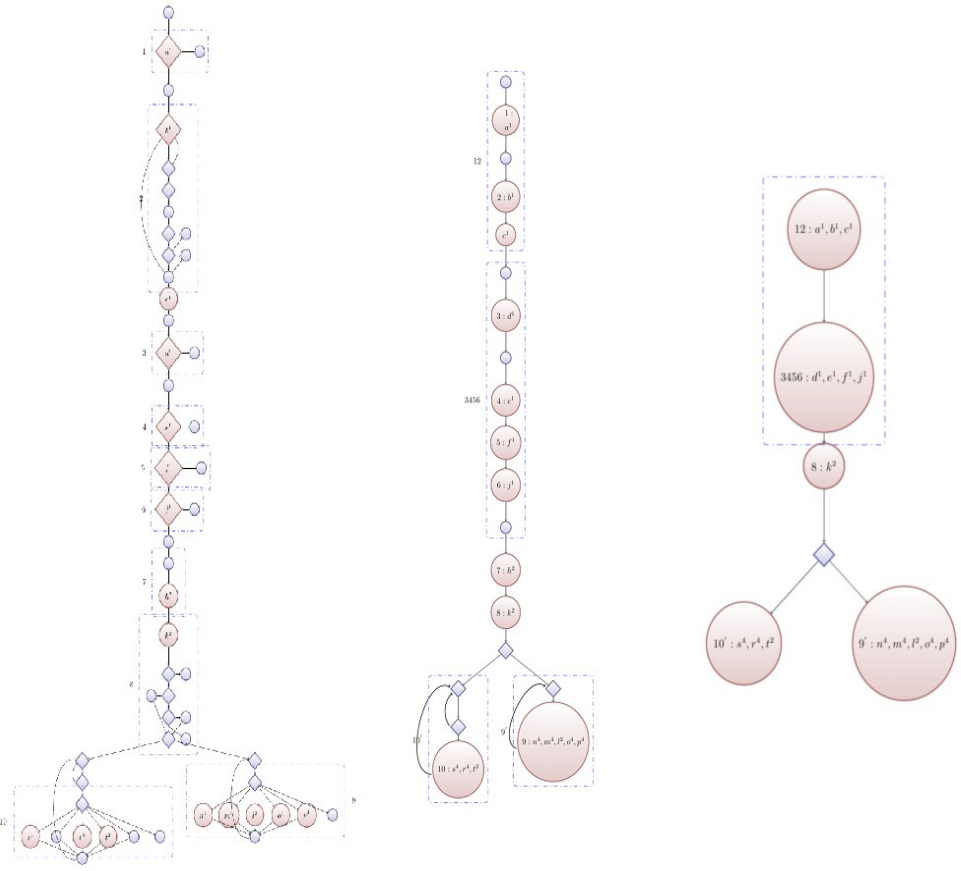
- **Добавим время – набор привилегий меняется, в зависимости от потребностей приложения в данный момент времени**
- Автоматическое построение модели поведения приложения
  - Комбинация статического и динамического анализа приложения
  - Построение срезов CFG и минимизация набора привилегий для каждого среза
  - Построение модели поведения приложения в форме контролирующего автомата и набора профилей SELinux
- Использование модели для контроля поведения в режиме реального времени
  - Механизм легковесных контрольных точек с помощью программных брейкпоинтов для обнаружения смены состояния (подсистема ядра Utrace/Uprobes)
  - Смена контекста SELinux при каждом переходе состояния в соответствии с моделью

Декомпозиция задачи

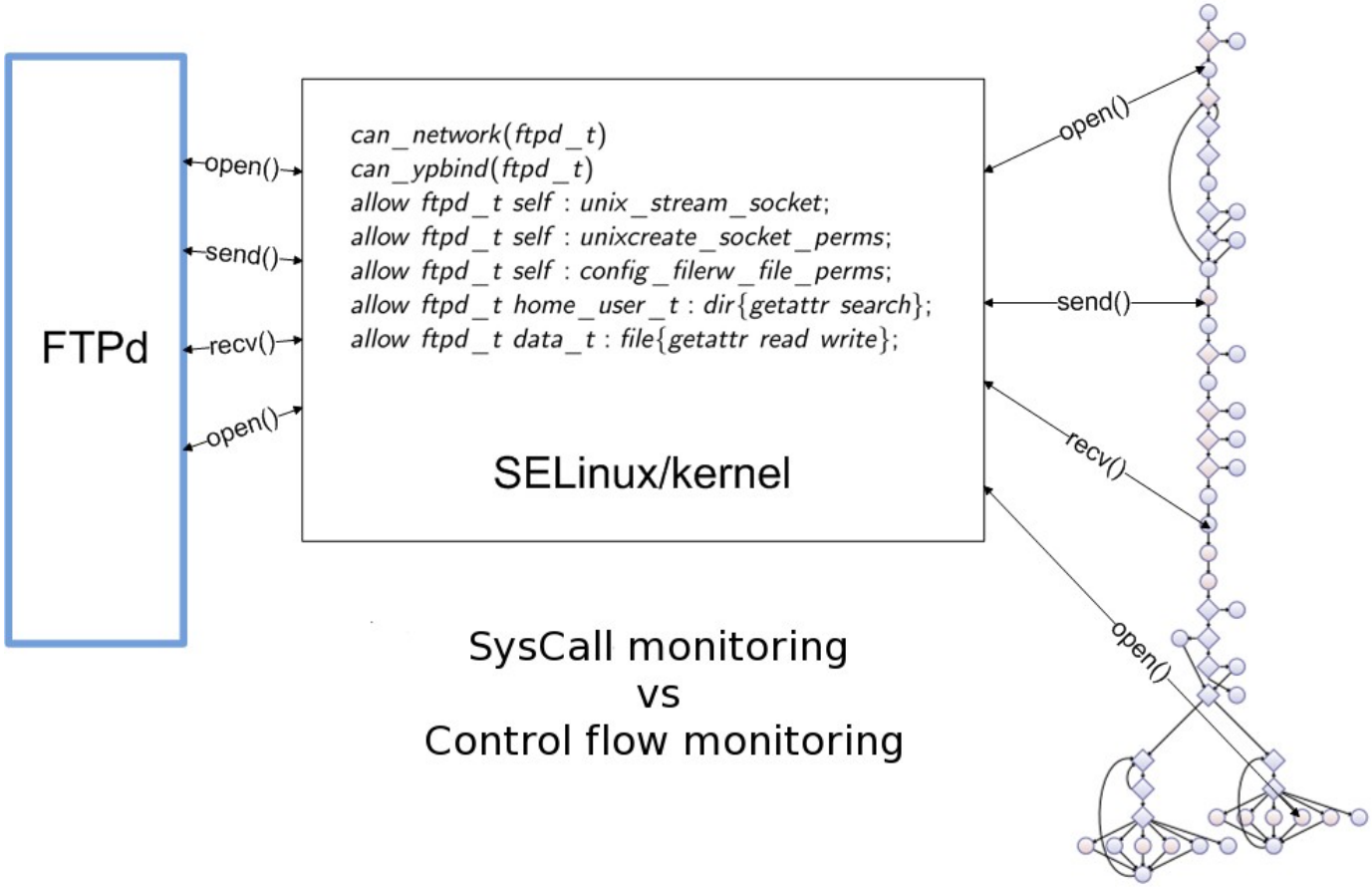
- Подзадача 1 – **построение срезов программы**
  - Разбиение CFG программы в на множество непересекающихся блоков: число привилегий в каждом блоке строго меньше, чем число доступных привилегий в начальном профиле SELinux для приложения
- Подзадача 2 – **построение модели нормального поведения**
  - Построение модели нормального поведения приложения в форме ДКА, символами алфавита для которого являются системные вызовы и контрольные точки
- Подзадача 3 – **мониторинг поведения**
  - На стороне ядра ОС выполняется анализ параметров системных вызовов и прохода контрольных точке в режиме реального времени на основе сопоставления с автоматной моделью
  - При изменении состояния модели изменяется контекст безопасности для приложения

Пример – сервер miniftpd

- Построение срезов miniftpd CFG
- Построение автоматной модели
- Разметка контрольными точками – для каждого среза точка входа и точки выхода
- Построение профиля SELinux для каждого среза



Результат – минимум привилегий в каждом состоянии



Безопасность мобильных платформ

Сильный принцип наименьших привилегий в SE Android

Разграничение привилегий приложений – актуальная проблема

- Использование традиционных методов аутентификации ограничено
- Приложения при установке запрашивают избыточно много привилегий
- Частое появление троянских программ в Market

Проект SE Android

- Портирован код SELinux
- Портировано значительное число политик SELinux
- Контроль поведения приложений расширен до многопоточных приложений
  - Каждый поток контролируется независимо
  - Переключение контекстов SE Linux для каждого потока индивидуально
- Накладные расходы на дополнительный контроль поведения – 3-4% CPU time

Спасибо за внимание!

Контактная информация:
Денис Юрьевич Гамаюнов
E-mail: gamajun@cs.msu.su
Телефон: +7 (495) 932-88-58