

SCRIPTING EM PYTHON PARA ANDROID



Automatizando tarefas
Aprimorando e testando segurança mobile
Agendando execução pelo cron do busybox

WORKSHOP
T3chn0l0g1as de S3GUR4NÇA
da 1nf0rmaçã0

Fatec
Americana

Rafael Sanches Rocha

Os personagens



- Sintaxe simples
- Agilidade na codificação
- Rápida prototipação
- Variedade de módulos
- Maturidade em Infosec



- Mobilidade
- Discrição
- > 80% do mercado mobile
- > 1.4bi dispositivos
- Open Source

Unindo tudo





Passos

1 - Rooting

2 - Instalar SuperSU (Play Store)

3 - Configurar > Segurança > Fontes Desconhecidas

4 - SL4A

<https://github.com/kuri65536/sl4a/releases>

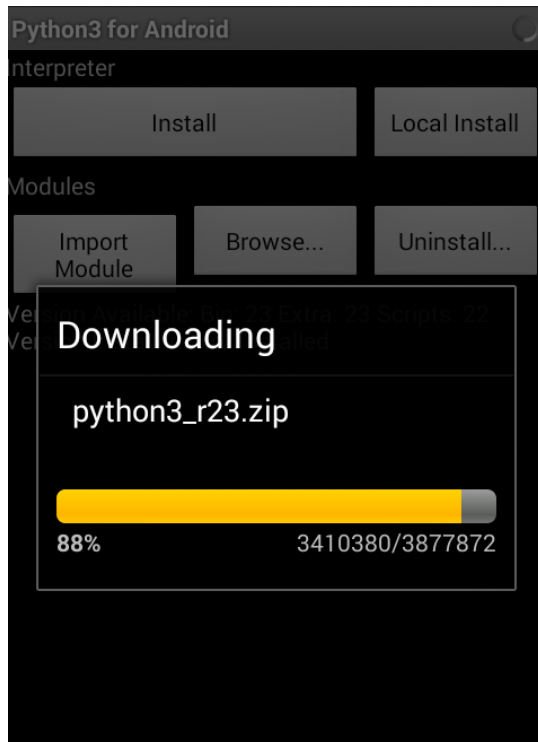
5 - Python4Android

<https://github.com/kuri65536/python-for-android/releases>

Preparando o ambiente

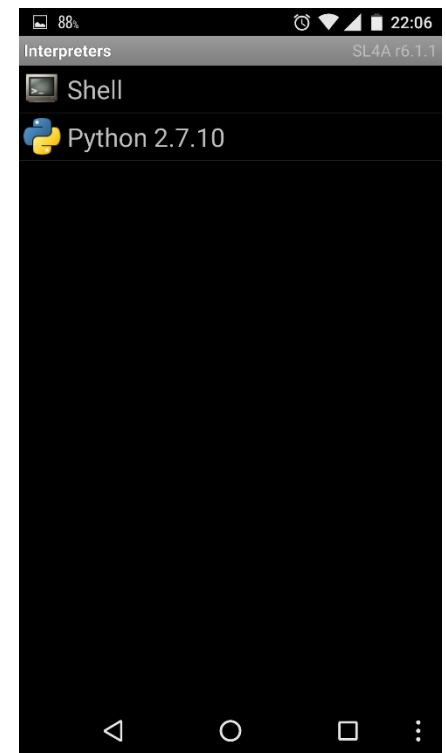
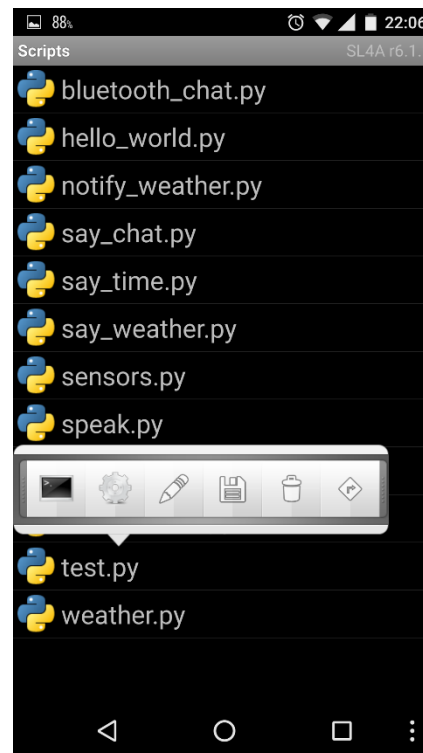
Py4A

Instalar Python



SL4A

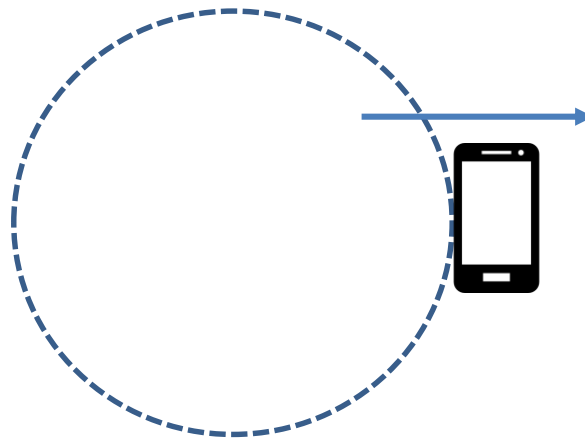
Menu > View > Interpreters > Python 2.7.x



Script 1 – Alarme de Perímetro

Proposta: Disparar um alarme quando o smartphone sair de uma área pré-determinada

Função que havia anteriormente no Avast Anti-Theft, mas não há mais



Código e execução: <EMULADOR>

Script 2 – Navegação Segura

Proposta: Simular um módulo de web protection de alguns antivirus, procurando por links possivelmente maliciosos na página

 Trust me, i'm a good link!
href="http://phishing.com"

Código e execução: <EMULADOR>

Script 3 – Scanner de Redes Vulneráveis

Proposta: Verificar por redes vulneráveis, armazenando dados da rede e localização



Código e execução: <EMULADOR>

Automatizar execução dos scripts

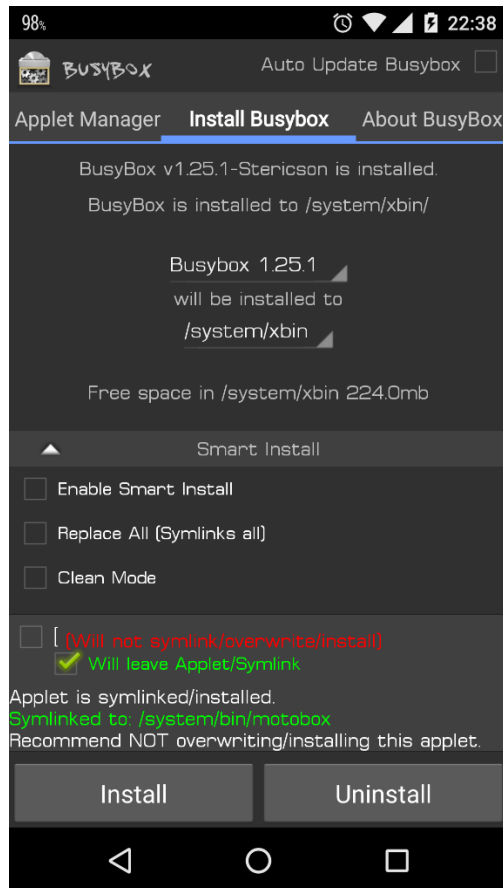
Mais interessante seria programar a execução automática dos scripts, ao invés de depender sempre de um controle manual

Existem apps para isso, mas, por vezes são pagos (TASKER), por outras não funcionam bem...

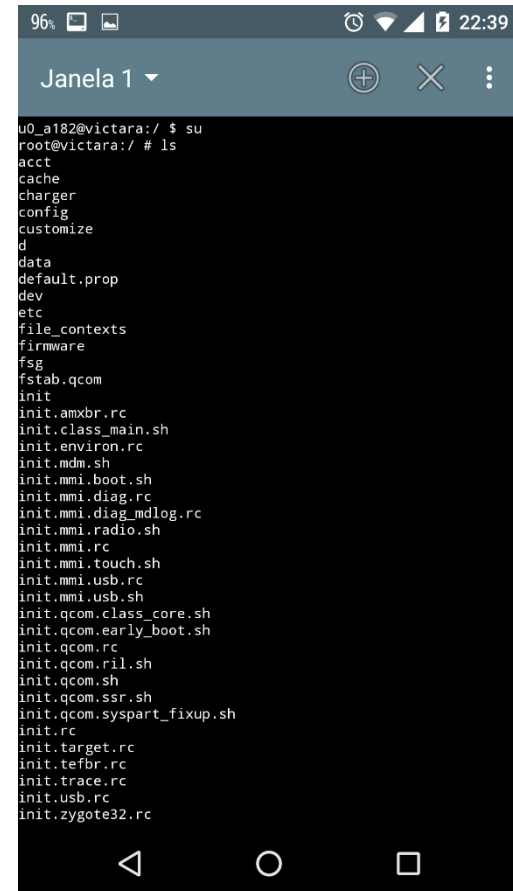
Solução: Ambiente unix rodando agendador de tarefas (CRON)

Apps necessários

Busybox



Terminal Emulator for Android



Habilitando Python no shell

- Permitir rodar o python diretamente no emulador de terminal
- Configurar variáveis de ambiente necessárias...
Já há um script para isso!
<http://forum.xda-developers.com/showpost.php?p=47607547&postcount=17>
- Observações:
 - Salve como “python” em /system/bin
 - Verifique o diretório correto em seu smartphone (/mnt/sdcard; /mnt/storage/; etc...)
 - Indicar a versão instalada!
\$PY4A/lib/python2.7/lib-dynload
 - Se o arquivo foi criado no Windows, retirar os “^M” ao final de cada linha (usar editor vi)

Criando a crontab

- Crie em um diretório permanente:
`#touch /data/crontab/root`



Ex: `0 0 * * * python bkp.py`

Editando a crontab


```
#vi /data/crontab/root
```

Modo de Comando



```
- /var/spool/cron/crontabs/root [Modified] 1/1 100%
```

Modo de Edição



```
* * * python /mnt/shared/AndroidSF/helloworld.py
```

```
I /var/spool/cron/crontabs/root [Modified] 1/3 33%
```

Tecla I => entra modo de edição → Adicione as linhas desejadas



SHIFT + Z + Z => Salva e Sai

:q! => Sai sem salvar

← Tecla ESC => volta ao modo de comando

No emulador, pressione o botão VOL+, e depois a letra E

Habilitando o CRON

```
#entra como root
$su -

mount -o remount,rw /
mount -o remount,rw /system

#criar /etc/passwd
echo "root:x:0:0:root:/root:/bin/sh" > /system/etc/passwd

#criar link usado para encontrar o shell system/bin/sh
ln -s /system/bin/ /bin

#criar diretorio onde cron busca a crontab
mkdir -p /var/spool/cron/

#link entre o arquivo real, salvo permanentemente, e o diretorio em que é buscado
#assim, sempre apos rodar "crontab -e", não perderá o link
ln -s /data/crontabs/ /var/spool/cron/

#iniciando o cron
crond
```

- Porém, precisaríamos executar isso toda vez que o sistema reiniciasse

Habilitando o init.d

- Preparando suporte para o init.d, onde scripts inicializam com o sistema
 - Criar diretórios, rodar scripts do sistema, setar permissões...
- Já há um script para isso!
- <http://www.theandroidhow.com/2014/06/how-to-enable-initd-support-on-android.html>

Colocar este script em algum diretório, e executar:

```
# sh /<caminho>/term-init.sh
```

Agora, dentro de /system/etc/init.d, é possível colocar um script com os comandos da etapa anterior, que irá rodar na inicialização



Recapitulando

1-Habilitamos o python para rodar pelo shell do terminal
(para o cron poder executar scripts em python)

2-Vimos como criar uma crontab

3-Criamos os arquivos e diretórios necessários para o cron, e linkamos com o nosso arquivo crontab

4-Habilitamos o init.d para que essa configuração anterior seja feita automaticamente em toda inicialização de sistema

Agora podemos colocar scripts .sh para inicializar com o sistema, ou scripts .py para serem inicializados conforme determinado na crontab!



Concluindo

Alternativa, rápida, para programação nativa do Android

Pratique as instalações e configurações primeiro em um emulador (Genymotion – procure a mesma imagem de seu modelo de smartphone)

Dica: Faça clones pelo VirtualBox!

Muita informação (ideias e soluções) na internet (learn how to search in english)

StackOverflow FTW!

Infelizmente, não fornece a mesma robustez do que o desenvolvimento em seu computador



Obrigado

Contato

rafael.si.123456@gmail.com

Github

<https://github.com/R4fa>

