



18 Slides
30 Minutos

Agenda

- 1)O que é API?
- 2)Motivações
- 3)Princípios
- 4)Esquema
- 5)Detalhamento
- 6)Padrões Diversos
- 7)Próximos Passos

O que é API?

É um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços

Motivações

Permitir que a expansão do SCIFI vá além do desenvolvimento em java de seu código

Princípios

- KISS
- Complexidade tem custo e deve mostrar seu valor
- Transparência tem valor
- Delphi, a metodologia
- ...
- Catedral e o Bazar

Esquema V12

#	NAME	SCRIPT	RETURN VALUE	RETURN STRING	OBSERVATIONS
1	SCIFI-VERSION	scifi-version.sh		12	SCIFI Used to identify SCIFI equipment and its version
2	SCIFI-SUBVERSION	scifi-subversion.sh		0	Subversion. Ex: 12.x
3	SCIFI-USERS	scifi-users.sh	# of connected users	# of connected users	AP - # of clients Server - scan MRTG bases and consolidate this numbers
4	SCIFI-TYPE	scifi-type.sh	Device version	AP - AP identification SERVER – 'CONTROLLER'	For auto-detection and auto-configuration
5	SCIFI-COORDINATES	scifi-coordinates.sh		latitude,longitude,altitude Ex: -22.894442,-43.118939,12	Altitude in meters above sea level
6	SCIFI-TAGS	scifi-tags.sh	# of Tags	Comma separated tags	For automatic build/rebuild of MRTG and NAGIOS' groups
7	SCIFI-CONNECTED2	scifi-connected2.sh	# of direct connected under this	IP or equipment name above this	For automatic build/rebuild of NAGIOS' map
8	SCIFI-NEIGHBORHOOD	scifi-neighborhood.sh	# of neighbors	Comma separated MACs	For future security analysis
9	SCIFI-UPTIME	Scifi-upptime.sh		0# of minutes since up	Uptime

SCIFI-VERSION

- Objetivo
 - Dizer a versão da API e dar um rótulo SCIFI
- Script
 - scifi-version.sh
- Retornos Atuais/Prováveis
 - Numero – 12
 - Texto - SCIFI
- Observações:
 - A versão encontra-se em scifi-version.txt

SCIFI-SUBVERSION

- Objetivo
 - Dizer a subversão da API
- Script
 - scifi-subversion.sh
- Retornos Atuais/Prováveis
 - Numero – 0
 - Texto - 0
- Observações:
 - A subversão encontra-se em scifi-subversion.txt

SCIFI-USERS

- Objetivo
 - Dizer a o nº de usuários
- Script
 - scifi-users.sh
- Retornos Atuais/Prováveis
 - Número – 0
 - Texto - 0
- Observações:
 - Nos Aps
 - Consulta o nº de usuários no AP
 - No servidor
 - Consolida os números coletados pelo MRTG

SCIFI-TYPE

- Objetivo
 - Identificar o tipo do equipamento
- Script
 - scifi-type.sh
- Retornos Atuais/Prováveis
 - Número – versão do dispositivo
 - Texto – vindo do arquivo scifi-type.txt
- Observações:
 - Nos Aps
 - Retorna um texto que identifica o tipo do AP
 - No servidor
 - Retorna “Controller”

SCIFI-COORDINATES

- Objetivo
 - Diz a localização do dispositivo
- Script
 - scifi-coordinates.sh
- Retornos Atuais/Prováveis
 - Número – 0
 - Texto – Ex: -22.894442,-43.118939,12
- Observações:
 - Busca dados em scifi-coordinates.txt

SCIFI-TAGS

- Objetivo
 - Permitir uma estrutura flexível de classificação
- Script
 - scifi-tags.sh
- Retornos Atuais/Prováveis
 - Número – # de tags
 - Texto – Ex: “POE,SDC”
- Observações:
 - Busca dados em scifi-tags.txt

SCIFI-CONNECTED2

- Objetivo
 - Em qual dispositivo o equipamento está?
- Script
 - scifi-connected2.sh
- Retornos Atuais/Prováveis
 - Número – # de equipamentos conectados abaixo
 - Texto – Equipamento conectado acima
- Observações:
 - Equipamento acima vem de scifi-connected2.txt
 - Número - não implementado ainda

SCIFI-NEIGHBORHOOD

- Objetivo
 - Quem está ao redor do AP?
- Script
 - scifi-neighborhood.sh
- Retornos Atuais/Prováveis
 - Número – # de vizinhos
 - Texto – lista de MACS separados por espaço
- Observações:
 - Lista vem de scifi-neighborhood.txt

SCIFI-UPTIME

- Objetivo
 - Tempo em que o dispositivo está funcionando ininterruptamente
- Script
 - scifi-uptime.sh
- Retornos Atuais/Prováveis
 - Número – 0
 - Texto – # de minutos
- Observações:
 -

Padrões Diversos

- Informações do SCIFI ficam em /etc/scifi
- Logs em /var/log/scifi
- Estruturas sendo adaptadas são preservadas por links simbólicos
- Esforço em unificar arquivos e programas evitando duplicidade

Próximos Passos

- 1) Discutir a API
- 2) Selecionar os implementáveis de curto prazo para a próxima versão e deixar o resto para a seguinte
- 3) Fazer os testes
- 4) Fazer o piloto na ND
- 5) Colocar em produção
- 6) Voltar ao item 1



Caio
Cosme
Daniel
Fernando
Glauco
Helga
Schara