How to migrate a massive environment to Check Point and automate your activities to succeed

Federico Meiners, CCSM, CRISC, PMP

MOLTEN MINDS

tecsystem

About me

Cybersecurity architect & Level 3 support

> Certifications: CCSM, PMP, CRISC, ITIL, GRCP

First Check Point version -> R77.30

> More than 50 Check Point deployments (Banks, Oil & Gas, ISPs, Healthcare, among others)

SOC Deployments, advisory, F5, Arbor, RSA

> CheckMates (Contribution is key)

The customer

> Big ISP & Datacenter company

> Government

> MSSP

> Key player: Cybersecurity

The project

x2 23500 appliances

60 VS (ASA to CHKP)

6 months

Next stage: Maestro

Automation

It doesn't have to be complex

Don't automate the whole thing

Less errors, less time, more **predictability**

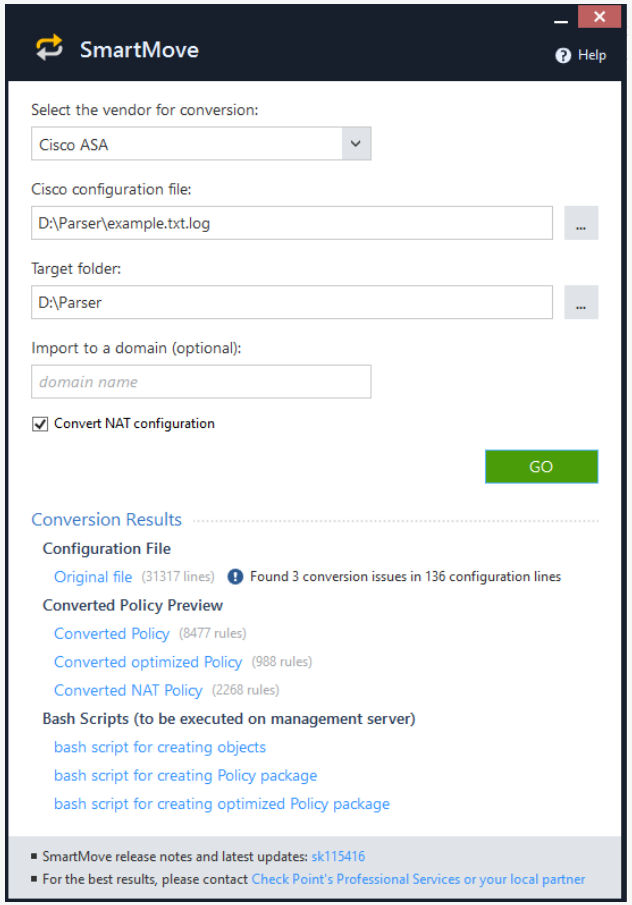
Automate whenever possible – SmartMove (sk115416)

From 59778 rules to 6027 **(90%)**

> Beware: Tool limitations

Optimized policy vs Normal policy

> Play safe!



Smart Move – Considerations for large MSSP deployments

> Avoid repeated names - Don’t worry for repeated IP

Host_ / Network_ / Interface_Outside

> Make them unique – Avoid chain reactions

> Check the error logs

> If possible, script

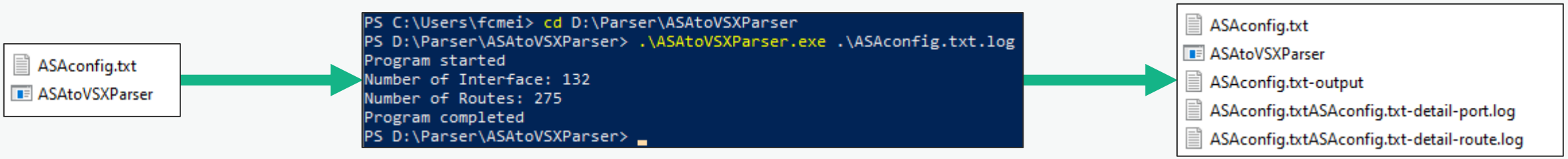
```
sed -i 's/network_/network_FILENAME_/gI' FILENAME_objects.sh
sed -i 's/network_/network_FILENAME_/gI' FILENAME_policy_opt.sh
```

Manual: 2 minutes
Automated: 1 second
99% optimization

Rules are ok. Now what? - ASAtoVSX parser

> Created by Molten Minds (Daniel Azar)

Developed in Golang - Open source



CheckMates

ASAtoVSX - Translate your running config to vsx_util and more! CheckMates post

Outputs- ASAtoVSX parser

vsx_util output

```
1   transaction begin
2   add interface name bond3 ip 172. netmask 255.
3   add interface name bond1 ip 10. netmask 255.
4   add interface name bond1 ip 10. netmask 25
406 add route destination 192. netmask 255.255.255.255 next_hop
407 add route destination 192. netmask 255.255.255.0 next_hop 10.
408 add route destination 192. netmask 255.255.254.0 next_hop 10.
409 transaction end
```

Interfaces JSON

```
{
    "ID": "1.
    "Name": "IN
    "Description":
    "Shutdown": false,
    "SecurityLevel": 30,
    "IPAddres": "172
    "Netmask": "255.255.128.0"
},
```

Routes JSON

```
{
    "Name": "IN
    "NextHop":
    "IPAddres":
    "Netmask": "255.255.224.0"
},
```

Key lessons from this project

Technical

Pre/Post checks

IPS Tunning

Max interfaces vs Max VS (sk99121)

#R. ASA	Interfaces
156	8
8477	132
184	10

Project management

Setbacks

Lack of knowledge

Team mindset

Technical decisions

Customer understanding

Predictable & Repeatable

Numbers

4 min

VS deployment

5-15 min

Objects and rules

10-30 min

Rulebase tunning

1:30 min

push policy duration

Automation 33 mins

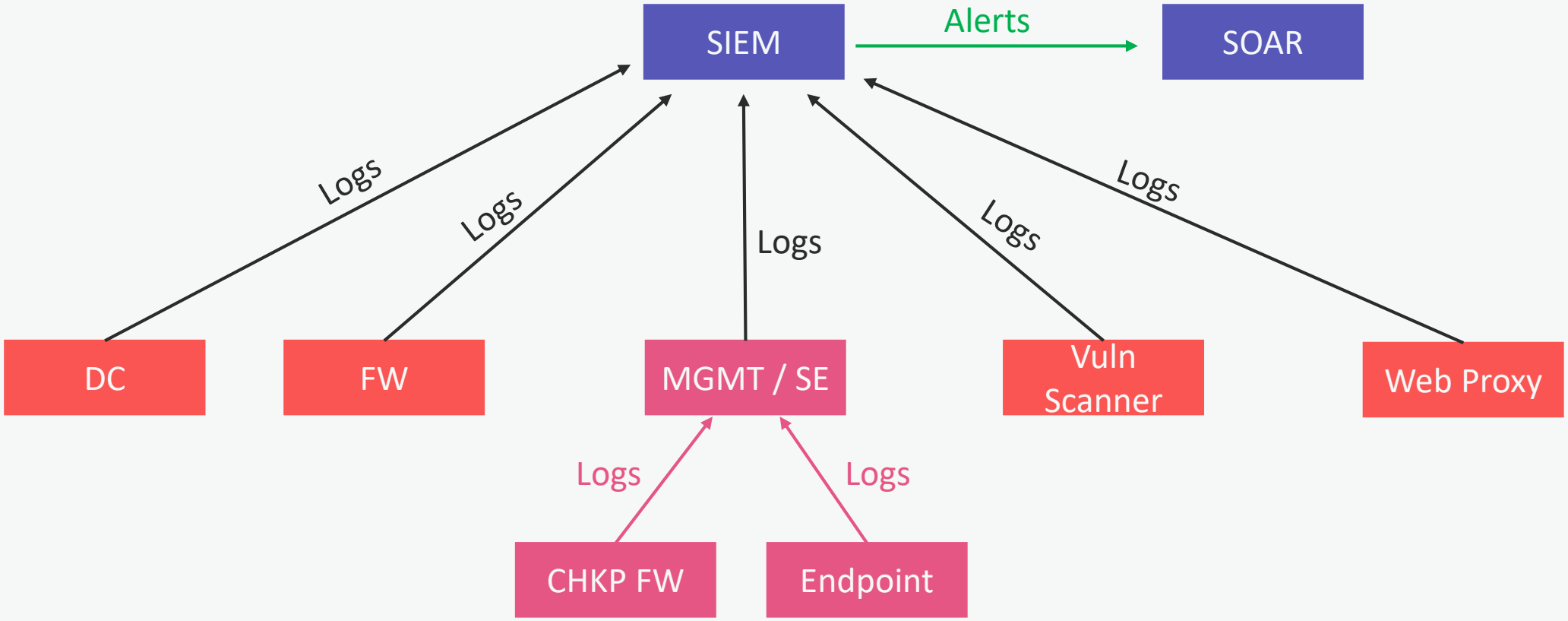
Manual 27 hours

98% less time

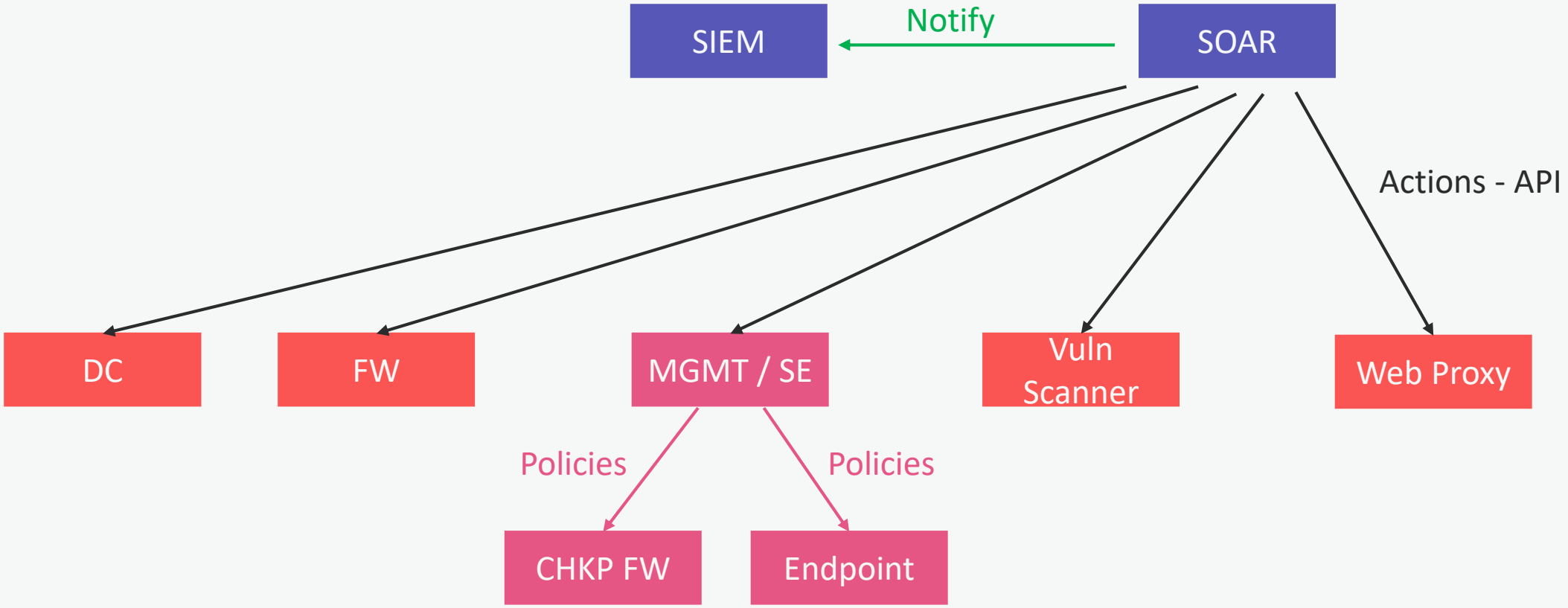
What about now?

SOAR

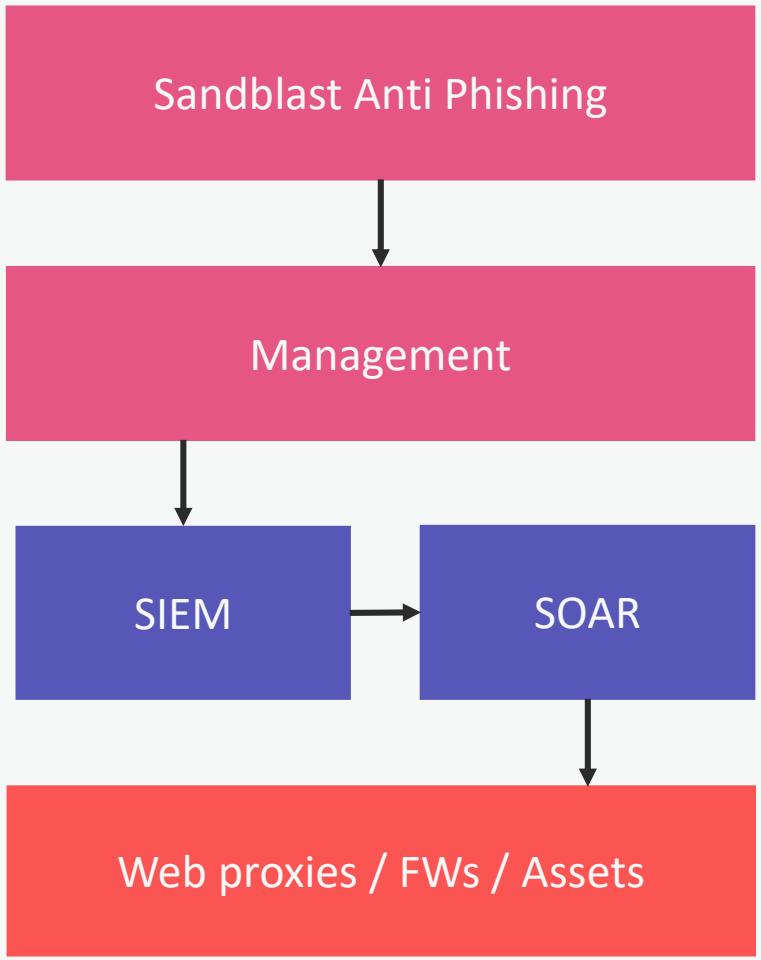
SOAR – How does it work? Part 1

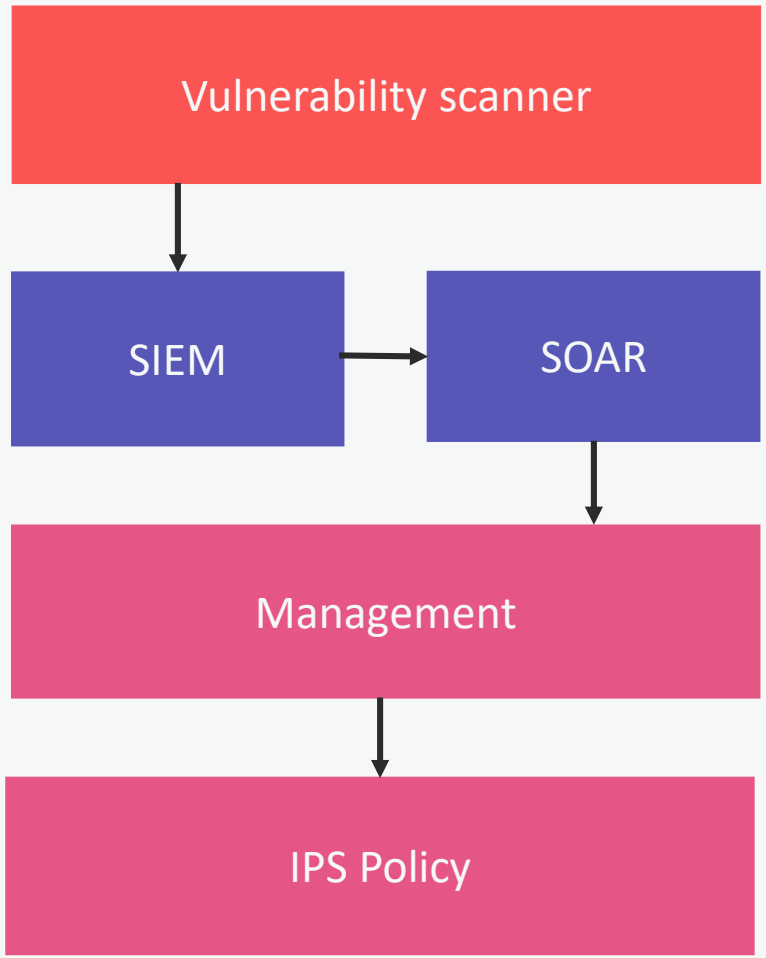


SOAR – How does it work? Part 2



SOAR – Check Point use cases





SOAR – Summary

Premises	Benefits	Warnings
SecOps	+ ROI	High skills required
Multi vendor integration	- Incident response time	Cybersecurity maturity
Automatic actions	Leverage your security	Dependency

The end – Special thanks

Check Point SE´s

Lucas Garcia & Alejandro Botter

Check Mates

Tim Hall - Kaspars Zibarts - Michael Endrizzi

Customer and CPX Staff

For the given opportunity

#cpstop

(the end)

Mail fcmeiners@gmail.com
LinkedIn https://www.linkedin.com/in/federicomeiners/
CheckMates FedericoMeiners