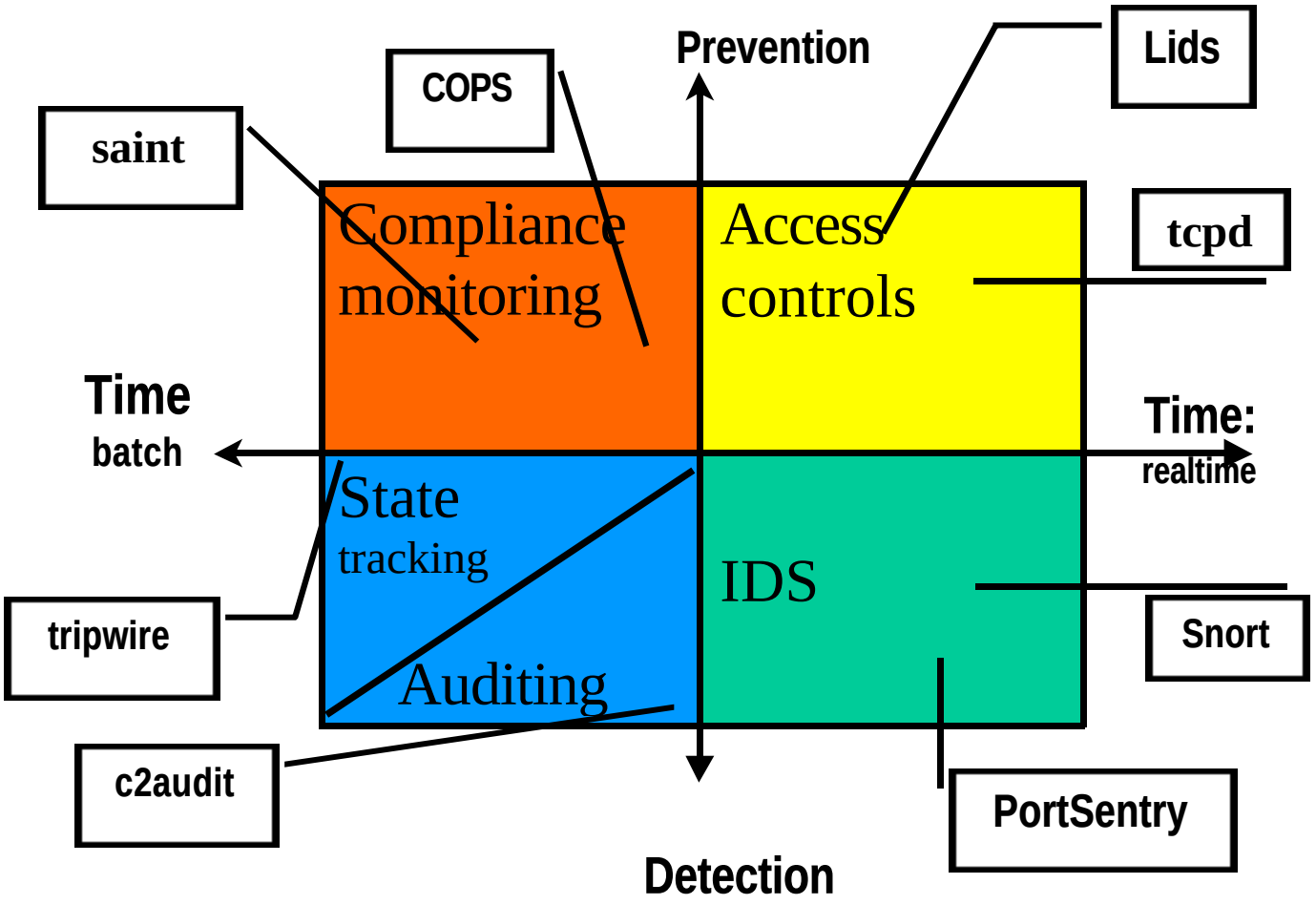
Osborne's security tool typology



www.loud-fat-bloke.co.uk

1 Introduction

I wrote this typology some five years ago, to help explain to a project team how I was going to use the old RAXCO Unix STK as a baseline checker and *bootstrap* the OS to a reasonable state – then use tripwire to monitor it. I have always been of the opinion that every one in the security community understood the nature and purpose of these tools.

I was wrong, I spoke at the e-Security conference 2002 and ComSec 2002 again this year and mentioned that in my experience that a high number of IDS project fail because people set illogical expectations on the ability of the software – that people expecting the software to catch hackers and automatically perform their incident response just didn't know what an IDS was for. **I was nearly lynched**, then Marcus Ranum stood up at conference, said basically the same thing & they all swooned.

To rub salt into the wound, ***The Register*** today points to a survey that claims that 75% of IDS projects fail, mainly for this very reason.

There should really be no need for this model – however, it has been extremely useful in the past for helping clients developing their technical architecture or their security strategy – cast your mind back to all the sites you know where they are buying yet another state or keystroke monitor in an attempt to improve Unix o/s security settings, when what they really need is a good baseline and someone to enforce it.

2 The model

The theory is based round a simple four box model. On one axis it categorises a product based on its role from primarily detection based to primarily prevention based.

On the other access, we plot the time frame it operates in – event driven/realtime as opposed to scheduled, batch mode products. This results in a series of definition

Access controls - a set of security rules on the actual operating system or network that control access.. This is a bucket for the traditional definitions of Authentication or Authorisation (which has a wider definition extended to defining what services are offered by the host or network). It is preventative and happens in realtime.

This quadrant is essential/mandatory to any architecture

Compliance monitoring – compare a set of security rules with the actual operating system settings that in our model represent the *access controls* quadrant and report differences. It is preventative and pro-active in its nature as it defines a set of services,

authentication methods and Access lists that will prevent inappropriate use. However, it operates in a batch mode

Auditing & State tracking

State tracking – Taking a verified cryptographic snapshot image of the operating system objects and reporting on deviations. It is detective and operates in a batch mode. Security input comes initially in deriving the snapshot and once a deviation occurs.

Auditing - records all actions on the operating system in realtime. Requires security input during the later review for signs of malevolent activity. Although the actions are logged in realtime, no review occurs to determine the security effect until later making it inherently batch. Automation of this log inspection process would push this into the IDS zone.

IDS - compare all actions on the operating system or network to a set of security rules and reporting where appropriate. Here the security rules are not a security policy or security baseline but almost the inverse – a set of attack signatures. It is detective but operates in realtime.

3 How do you use the model

3.1 Analysing your products

Helps you define the right tool for the job and the right functions for the tool:

- State checkers are not *Host IDS*. Any such product-placement, usually enacted by a salesman or marketing department, will result in the wrong tools for the job. IDS are event driven
- A compliance monitor checks for ideal setting and recommended patches, they can be used to enforce a security baseline or to implement one. Try this with an IDS or Statechecker.

3.2 Analysing your security regime

A security regime (i.e. security architecture) should consist of all five components. Each host should have as a bare minimum Detective and Preventative controls (at least 1 top layer quadrant and at least one bottom layer section).