Use of NetFlow/IPFIX Botnet Detection Tools to Determine Placement for Autonomous VMs

Razvan-Ioan Dinita

OWASP

The Open Web Application Security Project

OWASP
The Open Web Application Security Project

- 7+ years of programming experience (PHP, JS, C/C++, Java, C#, Bash, Scala)
- Open Source web development
- Web/Server Development focus
- Keen on ensuring app security
- PhD in Cloud Computing – *expected early 2015*
- Lecturer at ARU, Cambridge

Anglia Ruskin University
Cambridge Chelmsford Peterborough

Outline

OWASP
The Open Web Application Security Project

- What?
- The Cloud
- Test Bed Overview
- A Software Approach – AMDS
- Botnets
- NetFlow/IPFIX Overview
- Botnet Detection Module
- Experimental Design and Results
- Conclusion

OWASP
The Open Web Application Security Project

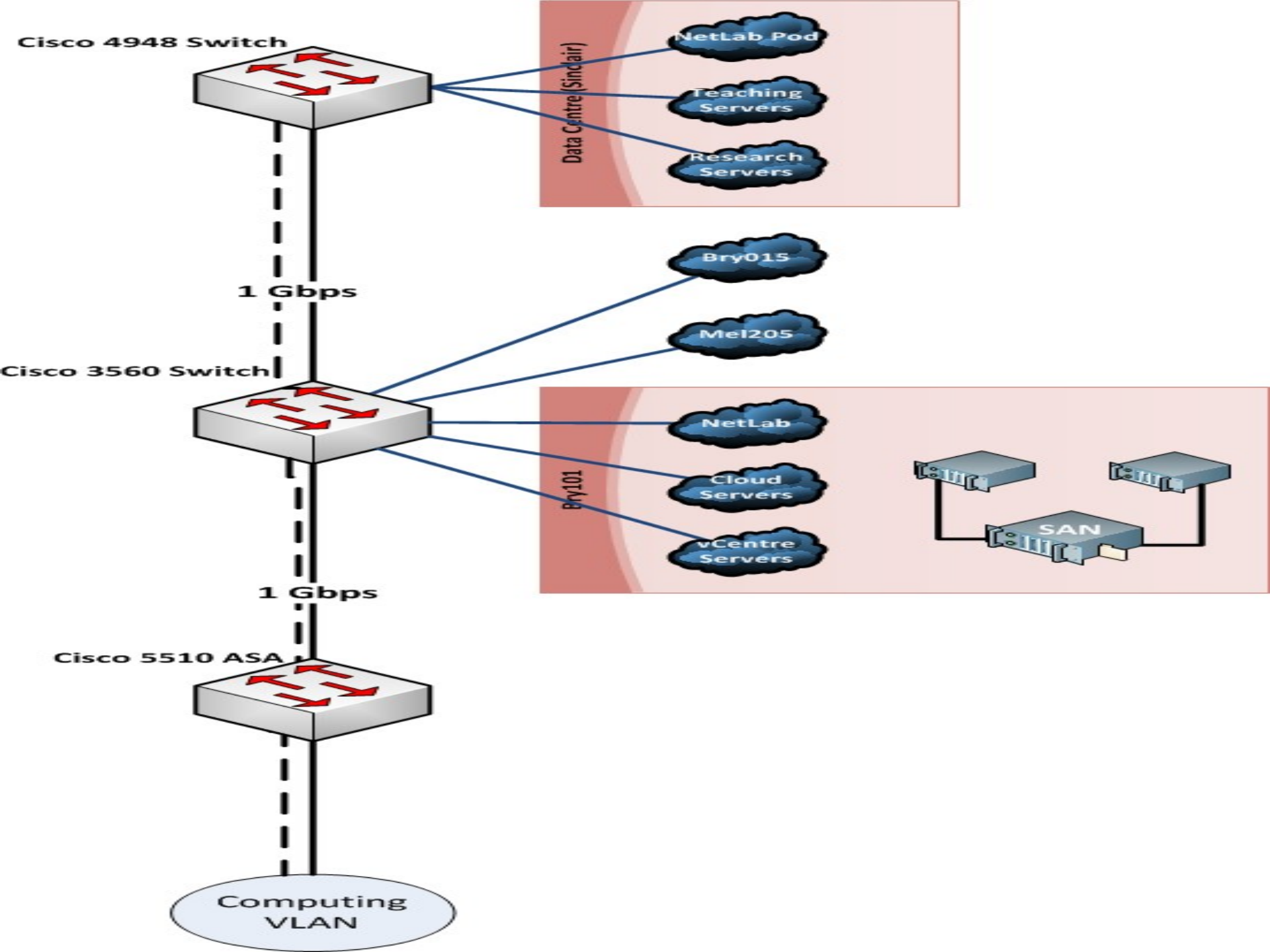
- Autonomous software-based Botnet Detection
  - The test bed cloud infrastructure
  - Autonomous Management Distributed System
  - Botnets and NetFlow/IPFIX
  - Botnet Detection module design
  - Experiment design
  - Experimental results
  - Conclusion

OWASP
The Open Web Application Security Project

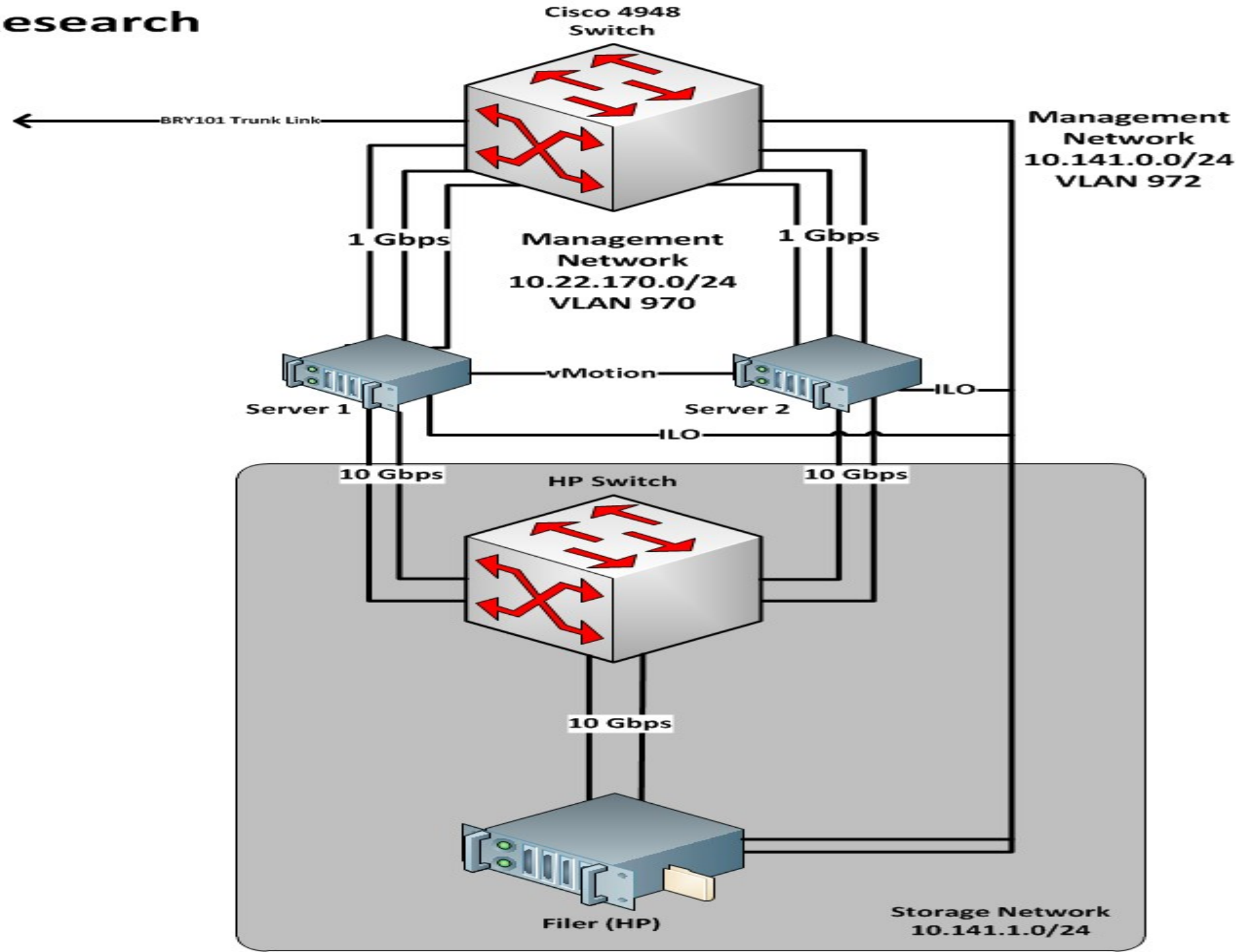
- Hype word
- Hosting Reloaded
- SaaS (G Apps), PaaS (App Engine), IaaS (AWS, Azure)

OWASP
The Open Web Application Security Project

- Hardware based in 4 different locations
- Fast 1Gbps (external) and 10Gbps (local) connections
- 7 servers (Dell R710), 3 Storage Area Networks, Back-up server, Multiple Routers and Switches, Integrated Light-Out


Cisco 4948 Switch
Data Centre (Sindair)
NetLab Pod
Teaching Servers
Research Servers
1 Gbps
Bry015
MeI205
Cisco 3560 Switch
Bry101
NetLab
Cloud Servers
vCentre Servers
SAN
1 Gbps
Cisco 5510 ASA
Computing VLAN

Research


Cisco 4948
Switch
BRY101 Trunk Link
Management
Network
10.141.0.0/24
VLAN 972
1 Gbps
Management
Network
10.22.170.0/24
VLAN 970
1 Gbps
vMotion
Server 1
Server 2
ILO
ILO
10 Gbps
HP Switch
10 Gbps
10 Gbps
Filer (HP)
Storage Network
10.141.1.0/24

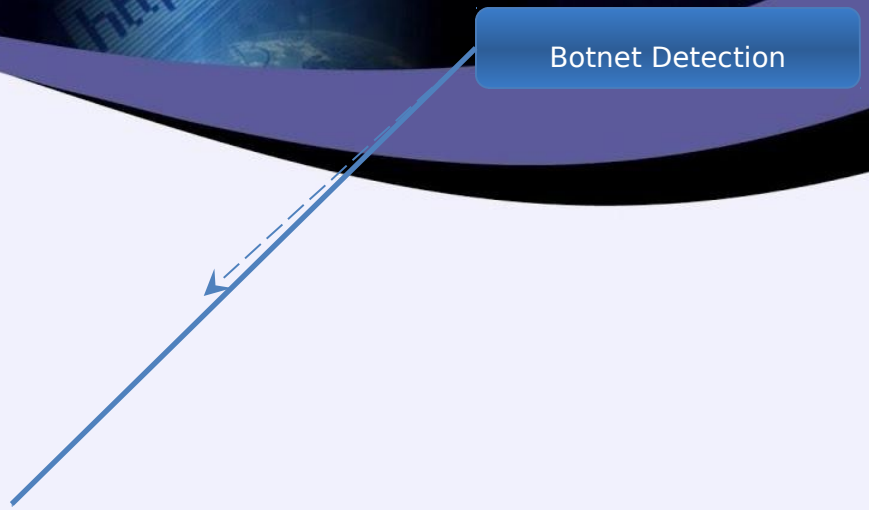
OWASP
The Open Web Application Security Project

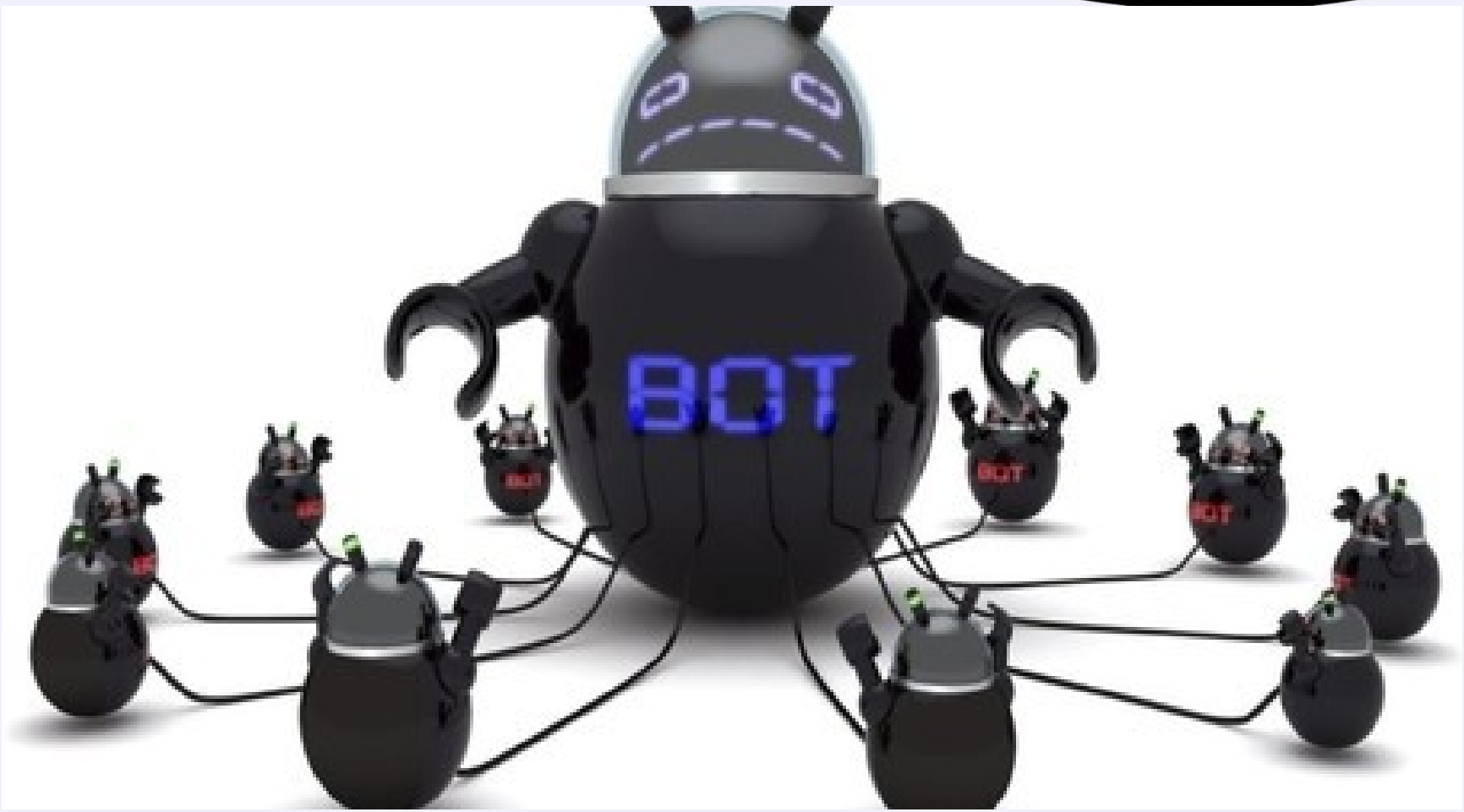
- Autonomous Management Distributed System
- Modular design, highly extensible
- Makes use of Java vSphere APIs
- Highly scalable (load balance)
- Resides inside a Linux based VM
- Plugs directly into existing infrastructure

OWASP
The Open Web Application Security Project

- Built using Scala
- Scala (Akka) vs Java
  - Native thread management
  - Built-in fault tolerance
  - More with less
  - Native support for Java

A Software Approach - AMDS




BOT

OWASP
The Open Web Application Security Project

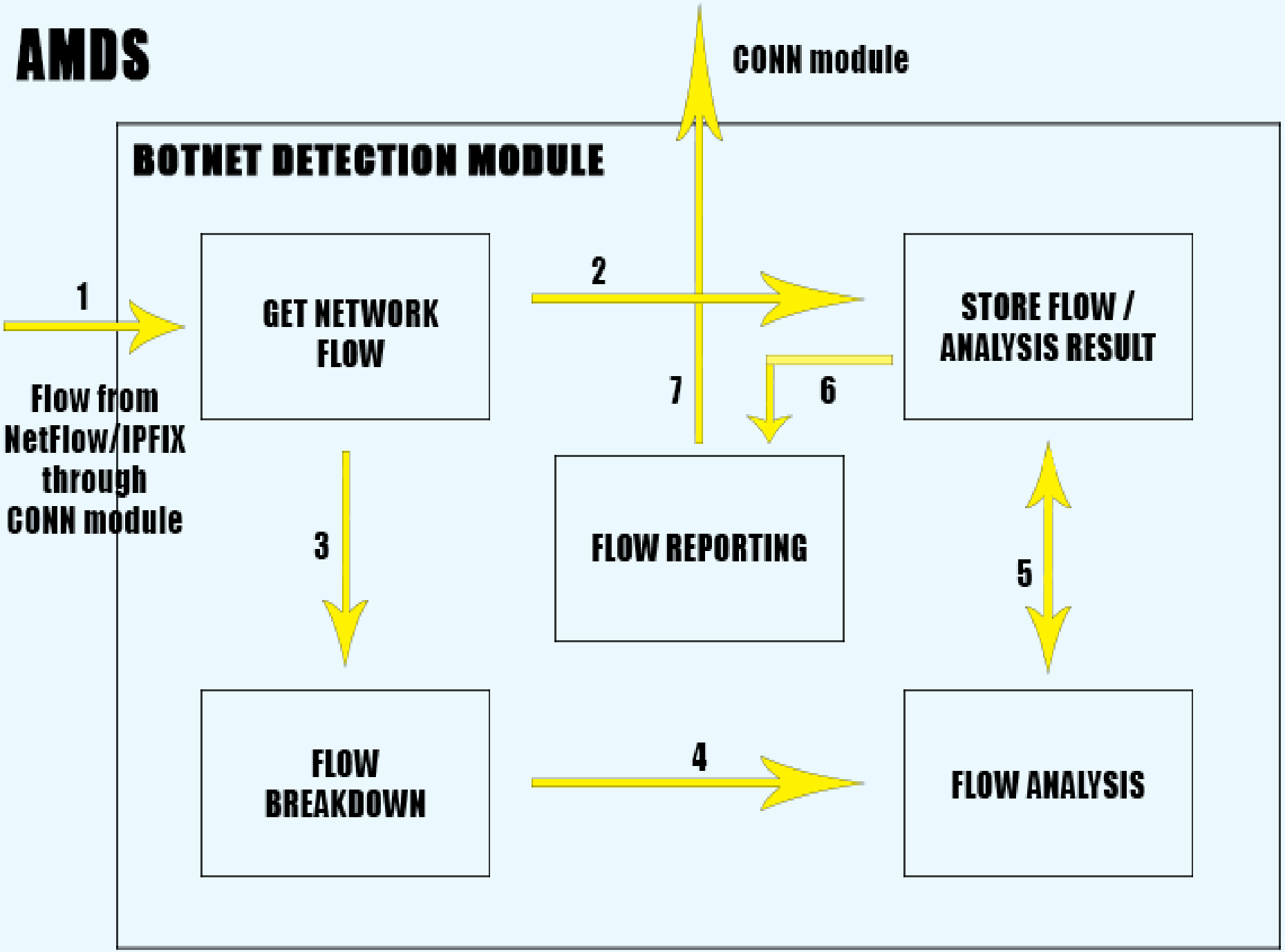
- Internet Access Monitoring: Peering & Traffic
- IETF Standard for Data Sampling and Export
- Security DDOS Monitoring Tool
- Flow timers, timing of network traffic types
- Who, what, where, when in the network

OWASP
The Open Web Application Security Project

- General data transport protocol
- Based on NetFlow version 9
- Flexible flow key (selection)
- Flexible flow export – TEMPLATE BASED
- Efficient data representation

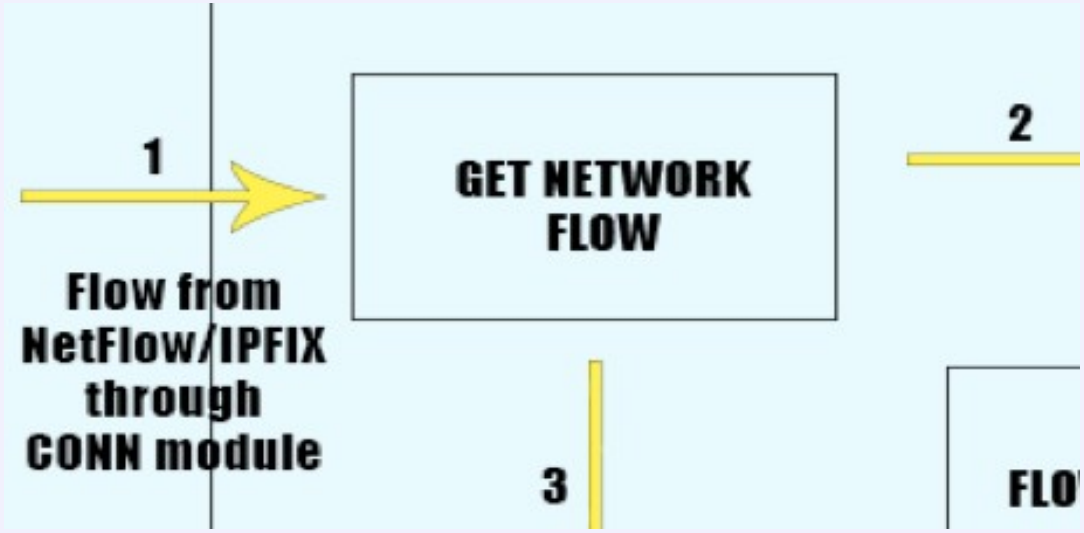
OWASP
The Open Web Application Security Project

- Bolted onto AMDS
  - Access to Data Centre (DC) management
- Ability to instantly react to threats
  - Lockdown DC
  - Restrict access
  - Relocate sensitive VMs to secure part of the DC

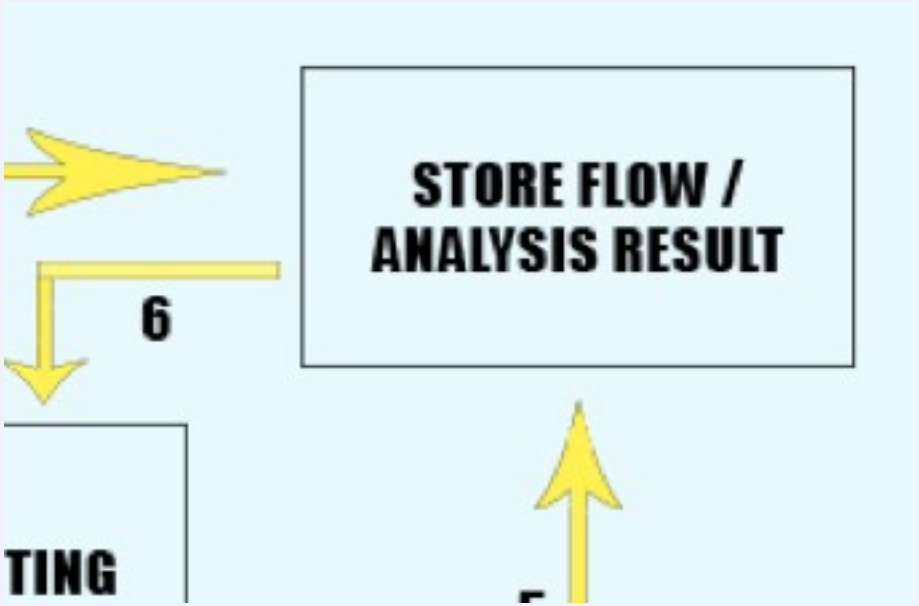

AMDS
CONN module
BOTNET DETECTION MODULE
1
Flow from NetFlow/IPFIX through CONN module
GET NETWORK FLOW
2
STORE FLOW / ANALYSIS RESULT
7
6
FLOW REPORTING
3
5
FLOW BREAKDOWN
4
FLOW ANALYSIS

OWASP
The Open Web Application Security Project

- Interfaces with the outside through AMDS Connection module
- Requests and Accepts NetFlow/IPFIX Flows
- Passes them on to storage and breakdown

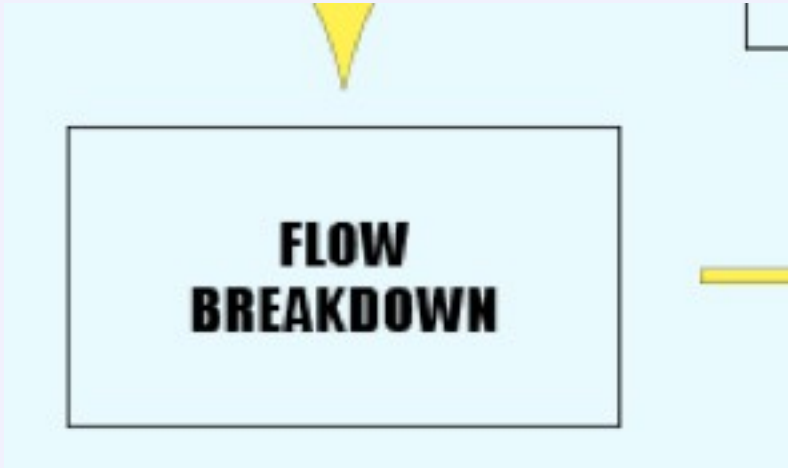


- Long term *local* storage
- Stores
  - Raw Flows
  - Flow Analysis Results
- Responds to statistics queries



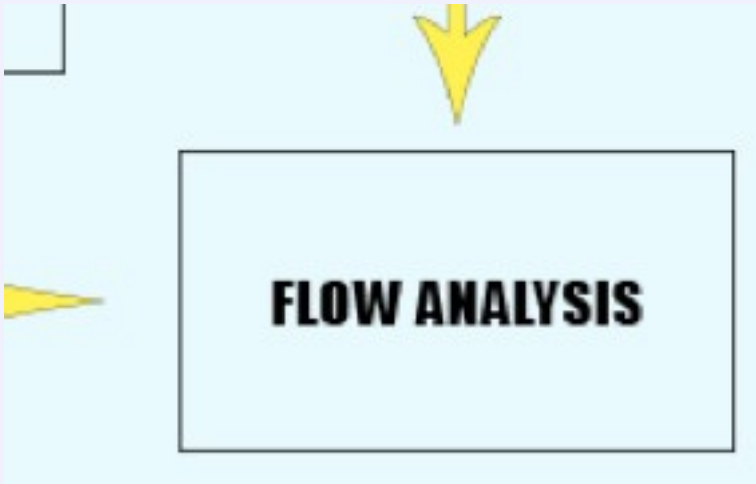
OWASP
The Open Web Application Security Project

- Deals with raw Flow information
- Extracts key Flow components
- Looks for
  - packet size, IP addresses and ports for both packet source and destination, class of service, device interface, protocol type
- Passes results to analysis component



OWASP
The Open Web Application Security Project

- Embodiment of heuristic detection algorithm
- Malicious behaviour detection through network traffic/Flow analysis
- Compares current Flow to past Flows
- Flags inconclusive results for further comparisons
- Passes results to storage



OWASP
The Open Web Application Security Project

- Refines a client model from Flow data
- Considers
  - packet size, IP addresses and ports, class of service, device interface, protocol type
- Access pattern-based detection

OWASP
The Open Web Application Security Project

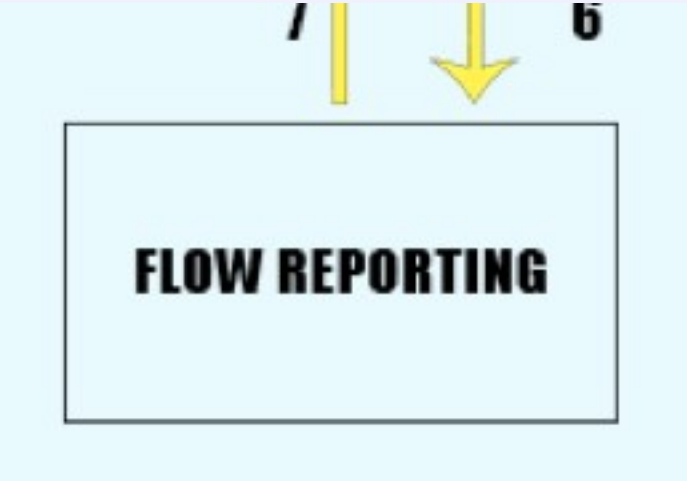
- Also uses a TCP work weight

$$w = (S_s + F_s + R_r)/T_{sr}$$

- S_s = SYNS + SYNACKS
- F_s = FINS
- R_s = RESETS
- T_{sr}= total number of packets
  - Closer to 100% -> anomaly

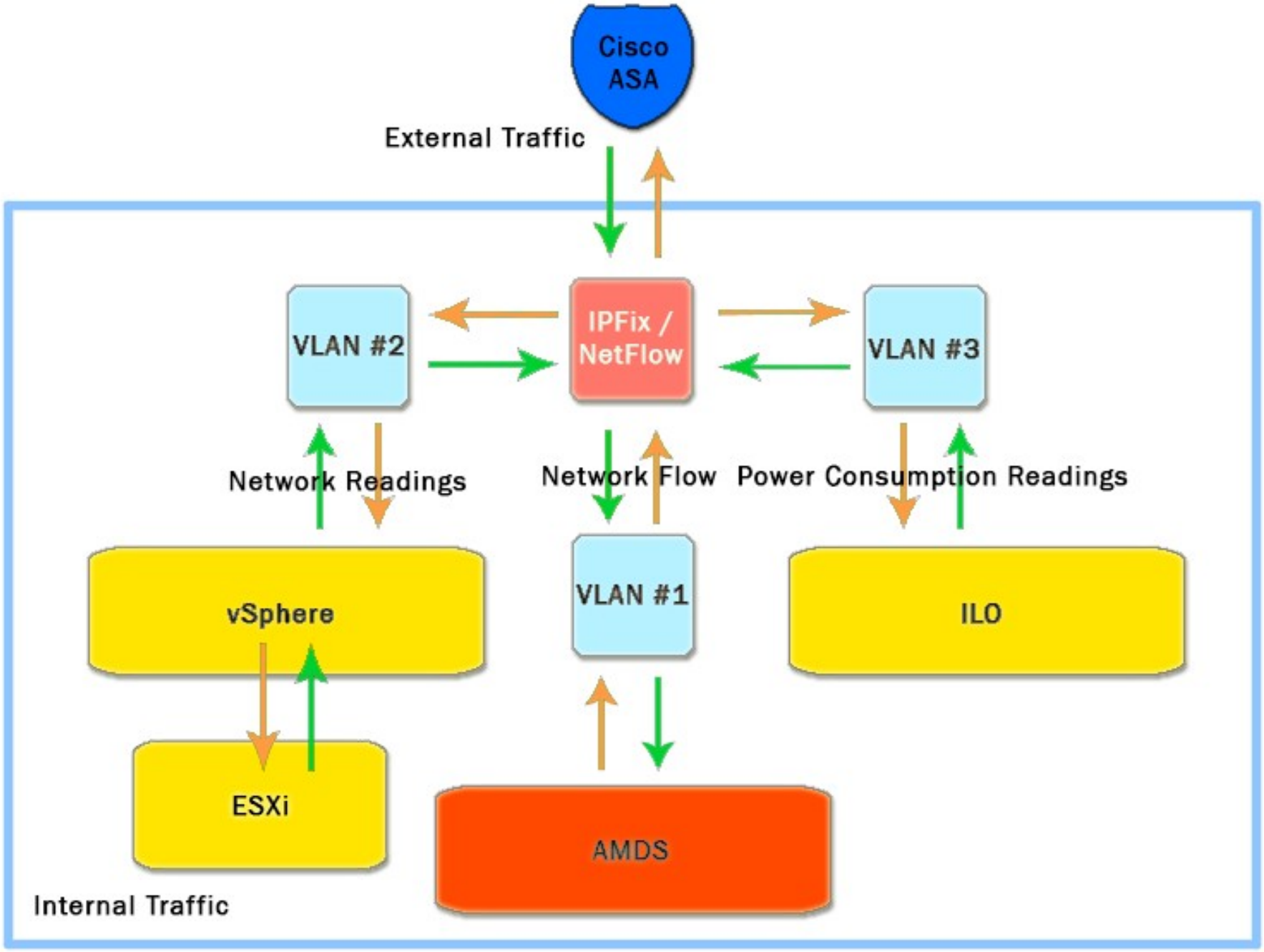
OWASP
The Open Web Application Security Project

- UI / Admin contact point
- Retrieves flow statistics
- Provides module activity overview

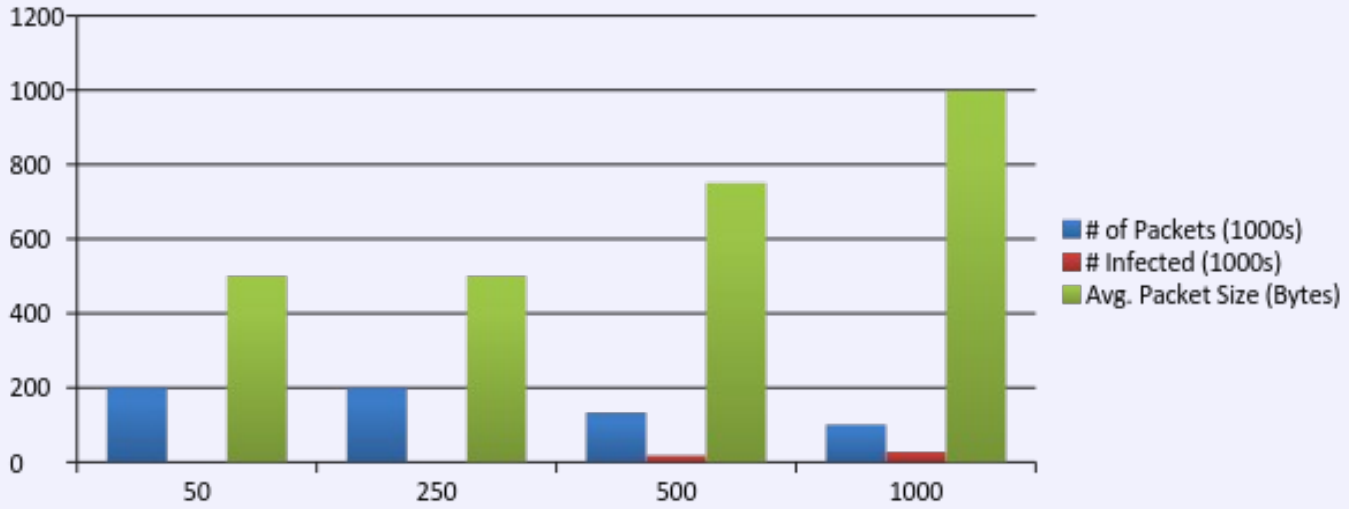


OWASP
The Open Web Application Security Project

- Sample 10% of all network data flow using IPFIX / NetFlow.
- Sort collected samples into logical groups based on parameters such as data packet Size, Source, Destination, and Commands
- Data packet sample size was set at 10% of all traffic at the point of collection
- Average data packet size ranged between 500 and 1000 bytes
- Infected (Botnet) packets have been used randomly starting with Sample #500


Cisco ASA
External Traffic
VLAN #2
IPFix / NetFlow
VLAN #3
Network Readings
Network Flow
Power Consumption Readings
vSphere
VLAN #1
ILO
ESXi
AMDS
Internal Traffic

Sample #	# of Packets (1000s)	Avg. Packet Size (B)	# Infected (1000s)	# Detected (1000s)	Detection Rate (%)
50	200	500	0	0	0
250	200	500	0	0	0
500	133	750	18	5	28
1000	100	1000	28	12	43



OWASP
The Open Web Application Security Project

- Botnet Detection abstract module design
  - Various programming languages
- The more it runs, the better it gets!
- Builds towards a comprehensive botnet detection model based on network Flows
- Real-time reaction to threats

OWASP

The Open Web Application Security Project

Thank you!

Questions, please?