

A Host-Based Behavioural Correlation for Bots Detection

Yousof Al-Hammadi, Uwe Aickelin

Intelligent Modelling and Analysis - IMA
School of Computer Science
The University of Nottingham

27-10-2009

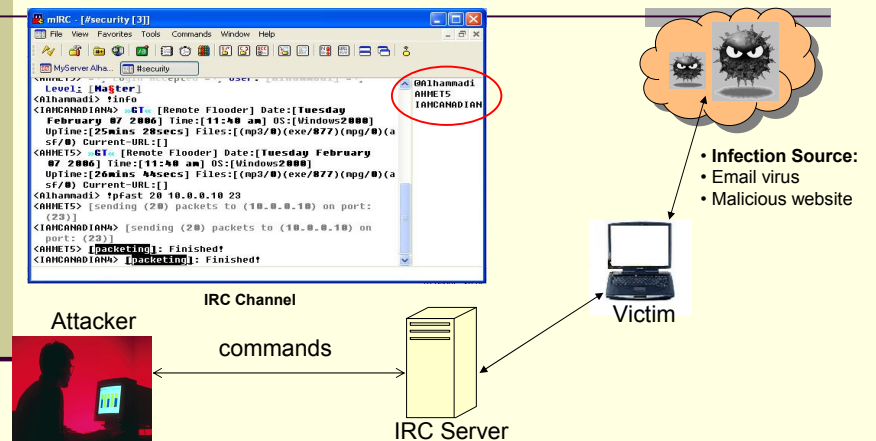
Outline

- Background
 - Bots
 - Methods of Infection
 - Motivation
 - Existing Botnet/Bots detection
- Objectives
- IRC Bots Detection
 - SRC
 - DCA for Detecting IRC bots
 - Results
- P2P Bots Detection
 - Background
 - DCA for Detecting P2P Bots
 - Results
- Conclusion and Future Work

Background

- Good IRC Bots
 - IRC protocol
 - Good Bots
 - To mimic the channel Operators
 - Provide services to the clients
 - Respond automatically to activities
- Malicious IRC Bots
 - Malicious program that is loaded in user machine without his/her knowledge
 - Responds to various instructions generated by the attacker
 - Relation between IRC and Bot

Methods of Infection



Motivation

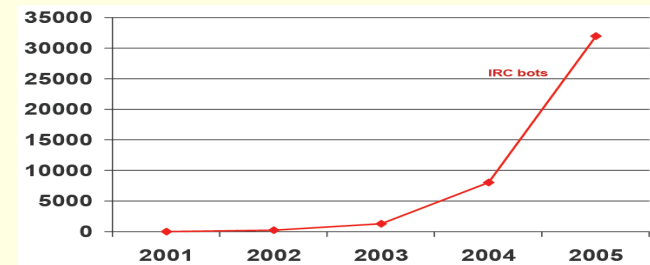
- Data regarding bots is rarely available
- Insufficient research in detecting a single bot
 - Signature-based vs Anomaly-based detection
 - Network-based vs Host-based Detection
- Serious threats
 - DDoS
 - Spamming
 - Keylogging
 - Spread Malwares
- Different commands and control (C&C)

5

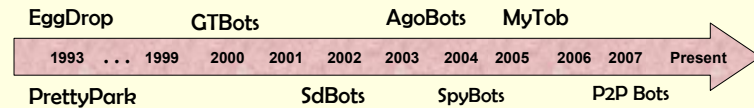


Motivation – continue ...

- large number of bots, growing exponentially



■ Source: McAfee



6



Existing Botnet Detection

- Botnet Taxonomy
 - Barford et al. (2006): classify botnet based on
 - Host control mechanism
 - Propagation
 - Exploit
 - Malware delivery
 - Deception and Hiding transmitted data
 - Trend Micro (2006): classify botnet based on
 - Attacking Behaviours
 - C&C structures (centralized, decentralized)
 - Communications mechanism (IRC, HTTP, P2P)
 - Rallying Mechanism
 - Activities and Evasion Techniques
 - Abu Rajab et al. (2007): determine botnet sizes
 - Botnet Infiltration by joining C&C channel
 - DNS redirection and queries

7



Existing Botnet Detection

- Honeypots
 - Freiling et al. (2005):
 - Collect information to infiltrate botnet network
 - Data Control: to limit # of outbound connections and BW
 - Data Capture: Snort, tcpdump, reverse engineering
 - Cooke et al. (2007):
 - Use proxy device as Data Control
 - Detect C&C communications by:
 - Monitoring standard ports
 - Payload Analysis
 - Look for behaviours that differ from human

8



Existing Botnet Detection

■ Network-Based

- Strayer et al. (2006): identify C&C
 - Filter traffic that unlikely to be part of botnet
 - Use machine learning to classify IRC flows based on network statistics
 - Keep long active botnet flows
 - Correlate flows which have common similarities
- Goebel and Holz – Rishi (2007):
 - Monitor network traffic looking for suspicious IRC nicknames
- Karasaridis et al. (2007):
 - Identify hosts with suspicious behaviour
 - Get data flows from these hosts
 - Analyse data flows to identify connections to controller

9

Existing Botnet Detection

■ Network-Based

- Gu et al. (2007, 2008, 2008):
 - BotHunter – correlate alerts from different network IDS
 - Inbound and Outbound port scan
 - Inbound exploit
 - Binary download and execution
 - C&C communications
- BotMiner - detect botnet by clustering
 - Hosts with similar communications traffic (extract network stats)
 - Hosts with similar Activity traffic (scan, exploit, spam, download)
 - Apply clustering algorithms to hosts which have similar comm. And activities traffic
- BotSniffer - correlate similar activities bet. Bots within similar time window
- Monitoring Engine and Correlation Engine

10

Existing Botnet Detection

■ Host-Based

- Stinson and Mitchell (2007)
 - Identify local execution commands from remote execution commands
 - Monitor processes who received data over network through function calls arguments.
 - No correlation between events
 - Problem: encrypted traffic or data manipulation

11

Objectives

■ To detect an individual IRC bot on a machine

- Monitoring selected API function calls executed by the bot
 - Communication functions
 - File access functions
 - Keyboard status functions
- Each function call represents one activity
 - Idea: to correlates different activities of the process
- Correlation Methods
 - Spearman's Rank Correlation - SRC
 - Dendritic Cell Algorithm - DCA

12

Detecting Bot – SRC

- Purpose
 - Technique to test the relationship between two variables (activities)
- Null hypothesis
 - There is no relationship between the two activities
- How it works
 - Rank the two data (the higher value in each column is ranked as '1')
 - Find the difference in the ranks
 - Square the differences and sum them $\sum d^2$
 - Calculate the coefficient (Rs)

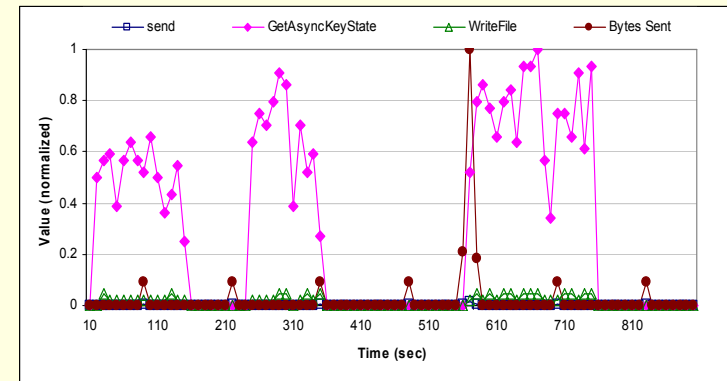
$$R_s = 1 - \frac{6 * \sum d^2}{n^3 - n}$$

- Interpretation
 - The closer Rs to (+1) or (-1) the stronger the likely correlation

13

Spearman's Rank Correlation – SRC

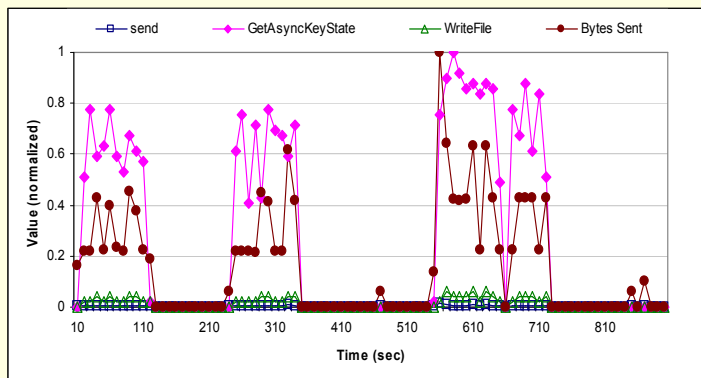
- Keylogger is activated and no data is transmitted



14

Spearman's Rank Correlation – SRC

- Keylogger is activated and data is transmitted



15

Spearman's Rank Correlation – SRC

- E1 - Inactive bot
- E2 - Active bot
 - No keylogger
- E3 - Keylogger activated no data is transmitted
 - E3.1: Long sentences
 - E3.2: Short sentences
- E4: Keylogger activated
 - E4.1: Long sentences
 - E4.2: Short sentences
- E5: IRC
 - Normal conversation

Experiments	SRC(GetAsyncKey, Bytes Sent)		SRC(GetAsyncKey, WriteFile)		Keylog. Activity existence	API Detection confidence
	with zeros (S1)	without zeros (S2)	with zeros (S1)	without zeros (S2)		
E1	0.863	0.671	1.000	1.000	No	N/A
E2	0.648	0.498	0.967	0.897	No	N/A
E3.1	0.509	0.183	0.559	0.172	Yes	Weak
E3.2	0.423	-0.003	0.928	0.618	Yes	Medium
E4.1	0.506	0.189	0.560	0.089	Yes	Weak
E4.2	0.927	0.579	0.957	0.663	Yes	Strong
E5	0.594	0.499	0.983	0.958	No	N/A

16

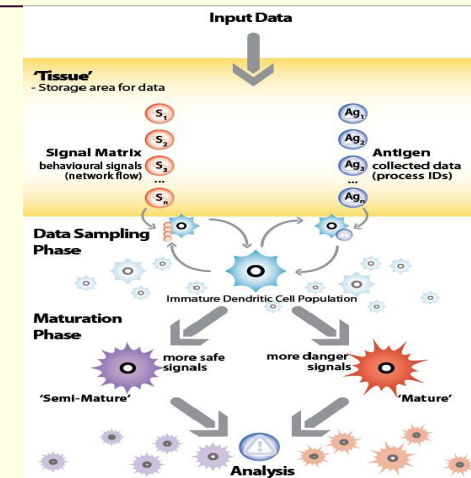
DCA – Dendritic Cell Algorithm

- Three stages
 - Filtering
 - Signals processing
 - Correlation
 - Correlating multiple input data streams (Signals with Antigens)
 - Classification
 - Normal/Abnormal
- Antigen – data to be classified
 - API function calls represented by process ID
- Signals – data used to perform classification
 - S1
 - Keyboard status functions
 - S2
 - How fast the bot responds to the attacker commands
 - S3
 - Time difference between two consecutive communication functions

17



DCA Algorithm



18



DCA – MCAV/MAC

- MCAV – Mature Context Antigen Value
 - The probability of the process being anomalous
 - MCAV of process x is given by:

$$MCAV(x) = \frac{A(x)}{A(x) + N(x)}$$

- Problem: small number of antigen leads to miss-classification
- MAC – MCAV Antigen Coefficient
 - MAC of x is given by:

$$MAC(x) = \frac{MCAV(x) * Antigen(x)}{\sum_{i=1}^n Antigen(i)}$$

19



Results

- E1
 - Inactive processes
- E2.1 (keylogger)
 - A: GetAsyncKeyState
 - B: GetKeyboardState
- E2.2 (flood)
 - A: SYN attack
 - B: UDP attack
- E2.3 (keylogger + flood)
 - A: SYN + GetAsyncKeyState
 - B: SYN + GetKeyboardState

20



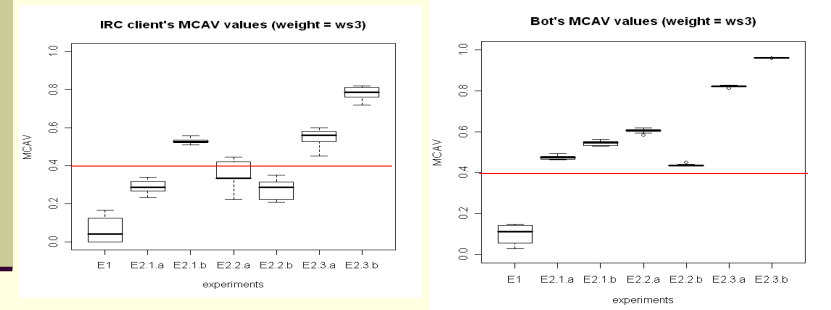
Results

Experiment	Process	Output Antigen	mean		Mann-Whitney P-Value	
			MCAV	MAC	MCAV	MAC
E1	bot	35	0.0978	0.0578	0.1602*	0.0202
	IRC	24	0.0625	0.0255		
E2.1.a	bot	1329.7	0.4736	0.4542	0.0002	0.0002
	IRC	59	0.2881	0.0122		
E2.1.b	bot	1296.2	0.5441	0.3098	0.0089	0.0002
	IRC	464.9	0.5284	0.1077		
	cmd	8.9	0.7889	0.0031		
	Notepad	239.4	0.6916	0.0726		
E2.2.a	Wordpad	268.8	0.8286	0.0977	0.0002	0.0002
	bot	19206.3	0.6047	0.6038		
E2.2.b	IRC	18	0.3441	0.0003	0.0002	0.0000
	cmd	9.8	0.2889	0.0002		
E2.3.a	bot	5790.5	0.4360	0.4346	0.0002	0.0000
	IRC	19	0.2772	0.0009		
E2.3.b	bot	41456	0.8218	0.8214	0.0002	0.0000
	IRC	20.5	0.5480	0.0003		
	bot	22446	0.9598	0.9461		
	IRC	59.1	0.7802	0.0021		
	cmd	9.7	0.6300	0.0003		
	Notepad	23.1	1.0000	0.0010	0.0001	0.0002
	Wordpad	233.6	0.8801	0.0090	0.0002	0.0002

21



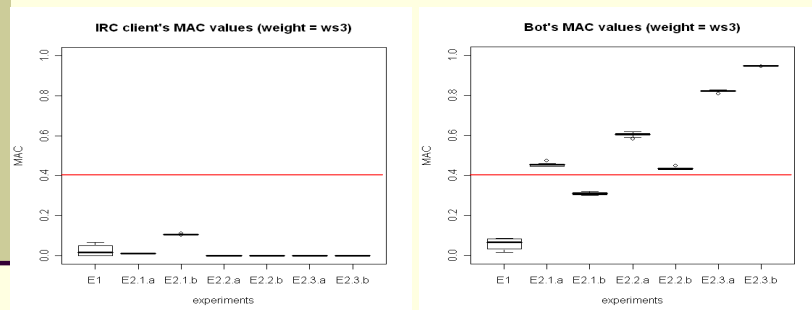
Results – MCAV for IRC client and Bot



22



Results – MAC for IRC client and Bot



23



P2P Bots

- **Background**
 - Napster
 - Gnutella
- **Malicious P2P Bots**
 - Phatbot
 - Connect to Gnutella cache servers to find other peers
 - Uses WASTE P2P protocol
 - Peacomm
 - Uses Overnet P2P protocol
 - Overnet implement DHT based on Kademia protocol
 - Node share information with other peers which are close to them
 - Sinit
 - Nugache

24



Existing Methods for P2P Bots Detection

- **Schoof and Koning (2007):**
 - Some P2P bots communicate on a fixed port
 - Some bots traffic are encrypted making traffic analysis a difficult task
- **Holz et al. (2008)**
 - Extract connection information
 - Join the botnet and receive commands
 - Inject commands into the botnet
 - Mitigate Storm bot
 - Reverse engineer to identify functions which generates keys

25



DCA for Detecting P2P Bots

- **Similar to IRC bots Detection but:**
 - Different signals
 - S1: DU, RST, FCA
 - S2: number of packets/sec
 - S3: Time difference between two consecutive communication functions
- **P2P Bots Scenarios**
 - PhatBot
 - Idle (PhatE1)
 - Information gathering (PhatE2.1)
 - Attack (PhatE2.2)
 - Normal
 - PhatE3 (no bot)
 - Peacomm
 - Idle with no user activity (PmE1)
 - Idle with user activity (PmE2)

26



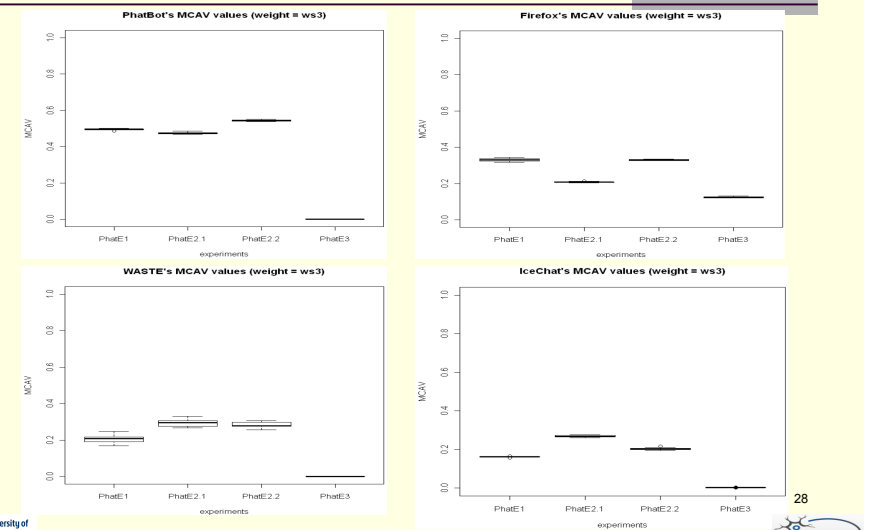
DCA for Detecting P2P Bots

Experiment	Process	Output Antigen	mean		Mann-Whitney P-Value	
			MCAV	MAC	MCAV	MAC
PhatE1	Phatbot	4362.3	0.4949	0.2065		
	Firefox	3524.9	0.3312	0.1115	0.00018	0.00018
	Icechat	1846.9	0.1621	0.0285	0.00001	0.00018
	WASTE	178	0.2049	0.0072	0.00001	0.00017
PhatE2.1	Phatbot	5615.5	0.4743	0.1724		
	Firefox	7394.2	0.2082	0.0996	0.00018	0.00018
	Icechat	1870.8	0.2685	0.0324	0.00018	0.00001
	WASTE	147.3	0.2943	0.0026	0.00018	0.00017
PhatE2.2	Phatbot	6630.7	0.5440	0.2201		
	Firefox	6337.1	0.3298	0.1275	0.00018	0.00018
	Icechat	2474.5	0.2032	0.0305	0.00018	0.00018
	WASTE	180.5	0.2836	0.0030	0.00018	0.00017
PhatE3	Phatbot	0	0.0000	0.0000		
	Firefox	4240.1	0.1239	0.0827	N/A	N/A
	Icechat	1630.3	0.0030	0.0007	N/A	N/A
	WASTE	0	0.0000	0.0000	N/A	N/A
PmE1	Peacomm	134985.2	0.5651	0.5554		
	Firefox	2000	0.3737	0.0054	0.00018	0.00016
	Icechat	47.2	0.1067	0.0000	0.00018	0.00006
PmE2	Peacomm	136521.6	0.5014	0.4851		
	Firefox	2695.3	0.3996	0.0076	0.00018	0.00017
	Icechat	1421.9	0.3378	0.0033	0.00018	0.00016

27



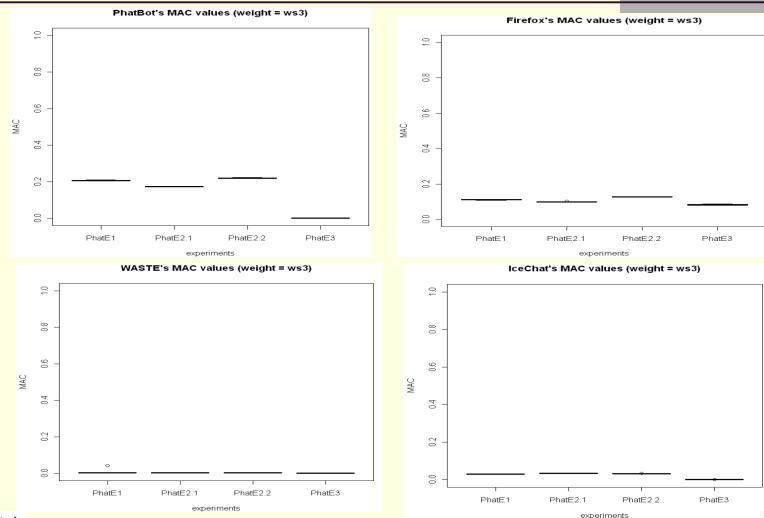
DCA for Detecting P2P Bots –PhatBot MCAV



28



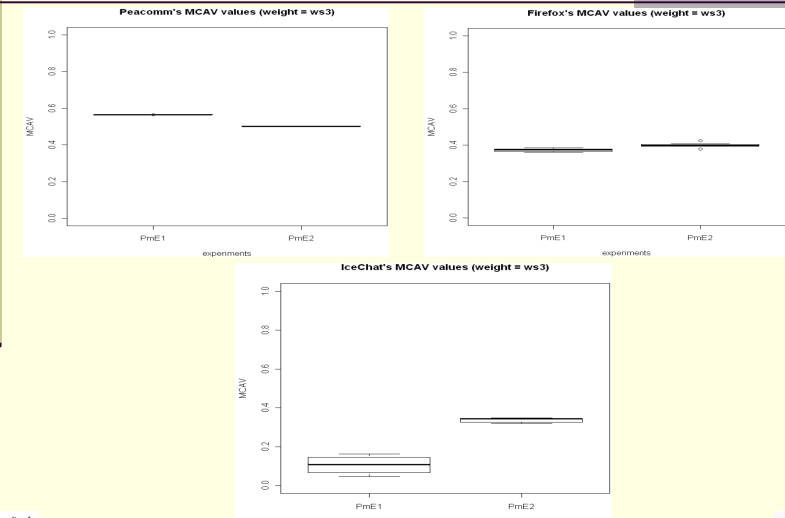
DCA for Detecting P2P Bots –PhatBot MAC



29



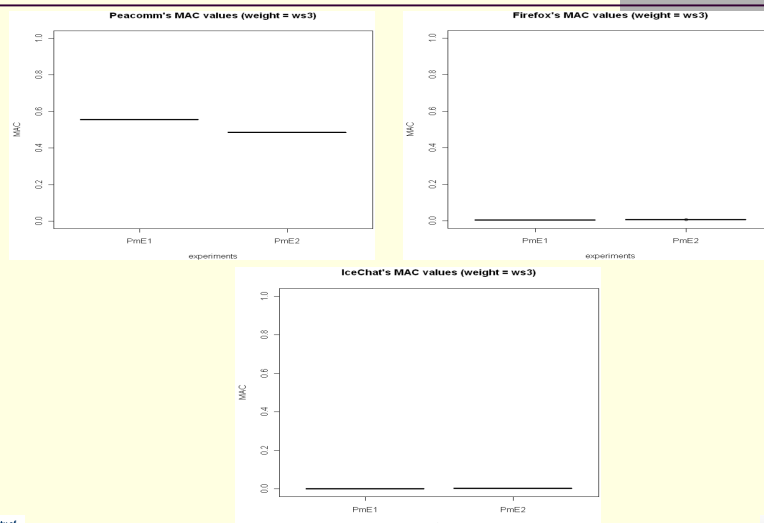
DCA for Detecting P2P Bots –Peacomm MCAV



30



DCA for Detecting P2P Bots –Peacomm MAC



31



Conclusion and Future Work

- Bots are the new threat to the Internet community
- Correlate different activities by monitoring API function calls
- Use simple and intelligent way of correlating activities to detect the bot
 - DCA enhance the detection performance by reducing the number of false alarms
- P2P bot
 - Apply DCA to detect P2P bots
 - DCA successfully detect Bots with different C&C structures
- Future works
 - More P2P bots with advanced features should be examined
 - Adaptive Signal Selections (ASS)
 - Combine behaviour-based detection with signature-based

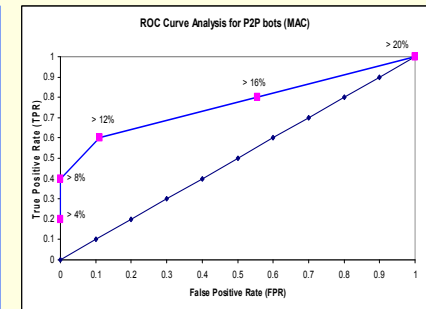
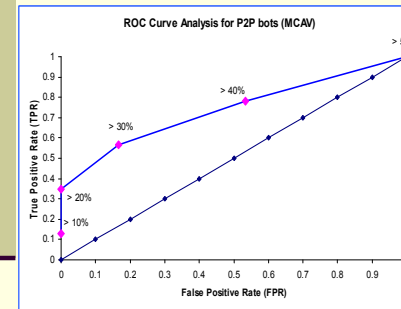
32



Thank You

■ Any Questions!

ROC Curve Analysis



Signal Weight values

	Signal	WS1	WS2	WS3	WS4	WS5
L (csm)	S1	2	4	4	2	8
	S2	1	2	2	1	4
	S3	2	6	3	1.5	0.6
N (semi)	S1	0	0	0	0	0
	S2	0	0	0	0	0
	S3	1	1	1	1	1
A (mat)	S1	2	8	8	8	16
	S2	1	4	4	4	8
	S3	-3	-12	-6	-6	-1.2