BotTracer: Execution-based Bot-like Malware Detection

Lei Liu, Songqing Chen

George Mason University

Guanhuan Yan

Los Alamos National Lab

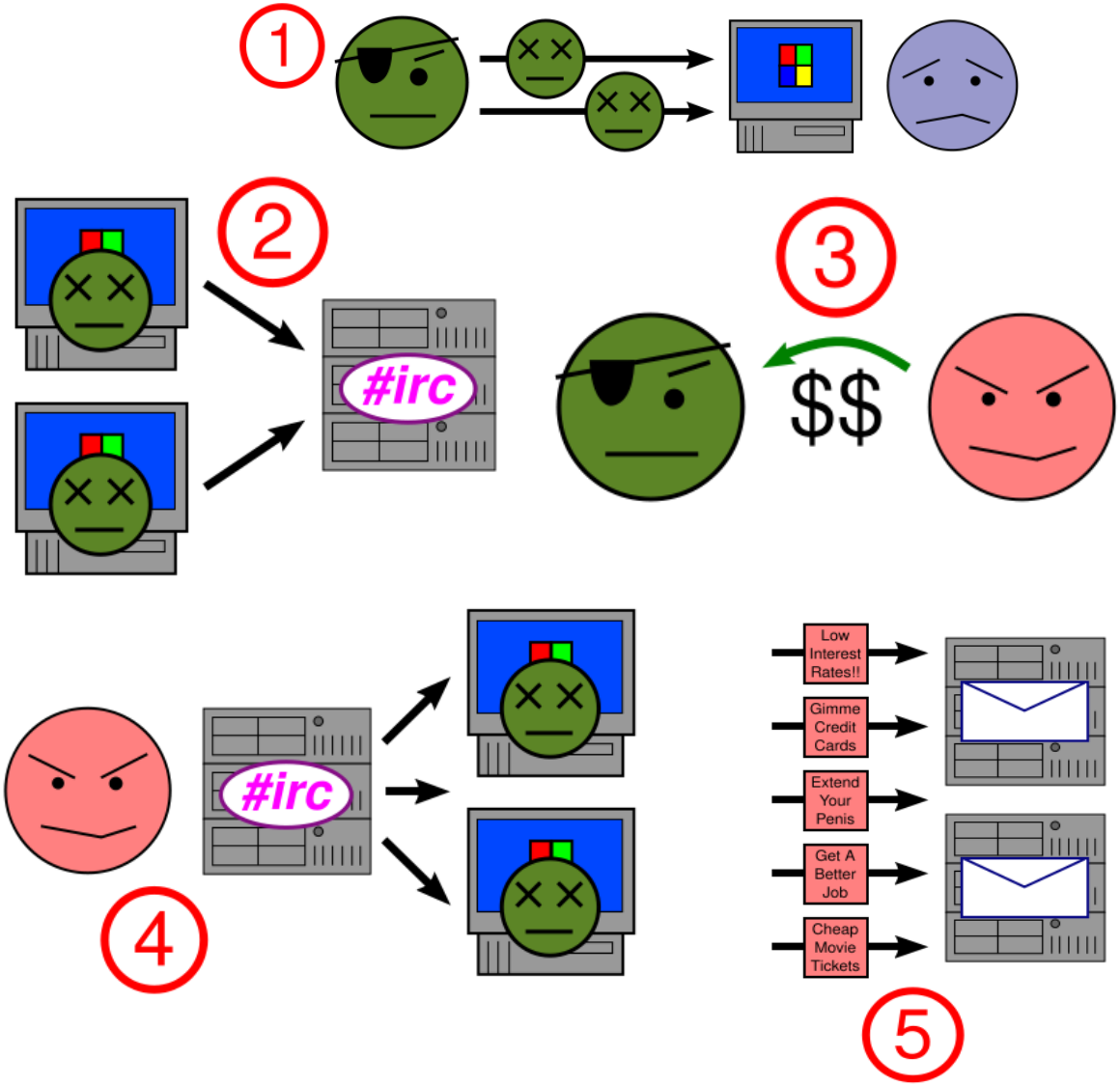
Zhao Zhang

Iowa State University

Background

- Botnets
  - A number of Compromised Internet computers (***bots***, ***zombies***)
  - Under a common ***command-and-control*** infrastructure
  - Controlled by a ***botmaster***
  - Responsible for ***spam***, ***DDoS***, ***sniffing*** and other attacks

How Spammer Uses Botnets



Growing Threats

- The Dutch police found a ***1.5 million*** node botnet [2005]
- ***1000*****'s of new bots** each day [Symantec 2005]
- ***One quarter*** of all PCs connected to the internet may become part of a botnet

Existing Solutions and Problems

- Signature
  - Unknown bots
- Identify IRC Traffic (port, content) [J. Zhuge 2007]
  - New protocols like ***HTTP***, ***FTP***
  - Encryption
- Network Traffic Patten Recognition [BotHunter]
  - New botnets architecture: ***P2P***
  - Hard to identify with legitimate traffic
- Taint Analysis [E. Stinson, 2007]
  - Performance

Neither solution captures the basic characteristics of botnets

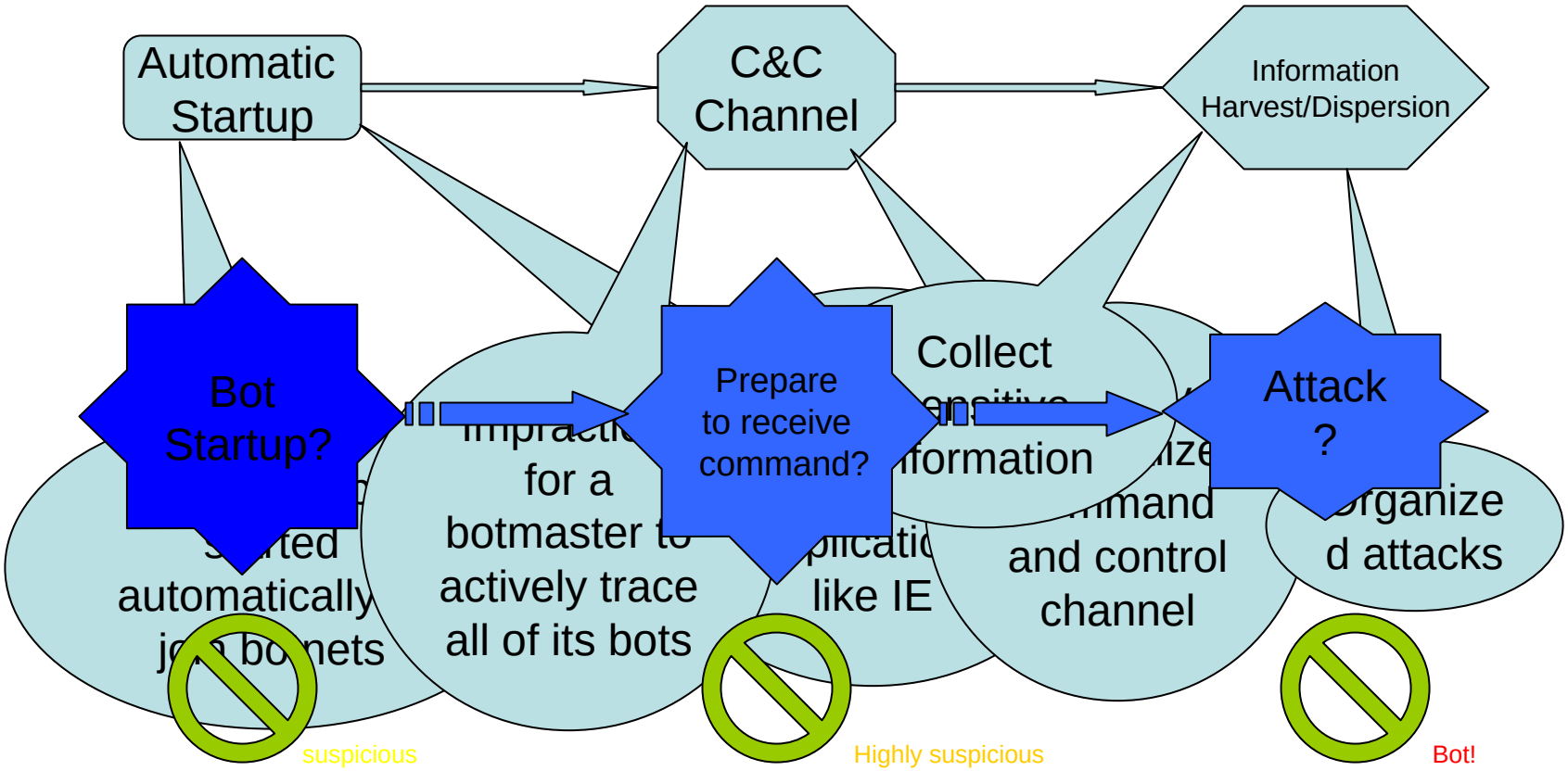
Our Contributions

- Analyze the ***basic characteristics*** of botnets
- Propose ***BotTracer*** to execute malware samples in a controlled environment and effectively capture botnets characteristics
- Implement a ***prototype system***
- Experiment based on ***real*** bot samples

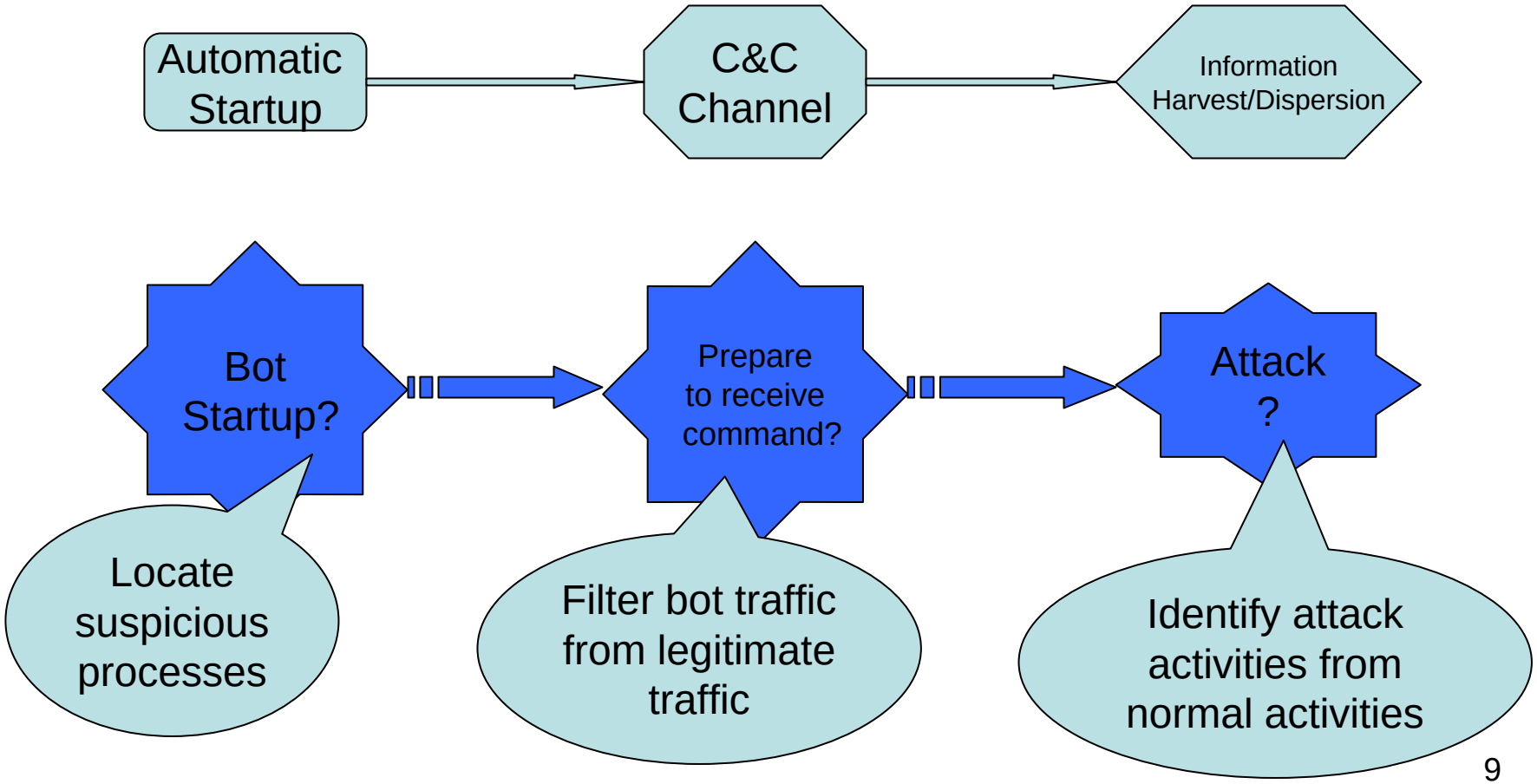
Outline

- Design Principles and Overview
- Design Issues
- Evaluations
- Conclusions

Bot Attack Phases

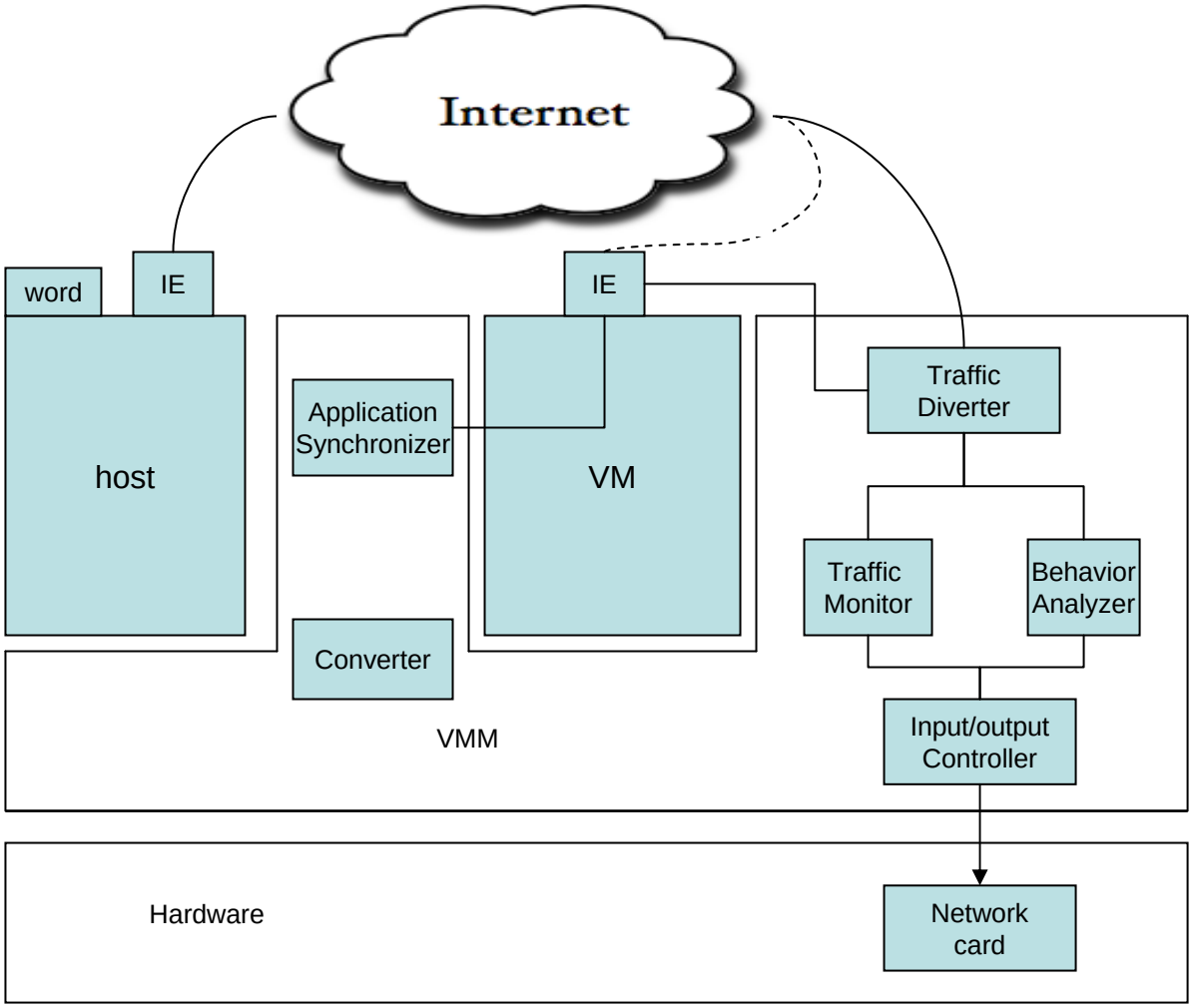


Design Goal and Challenge



BotTracer Architecture

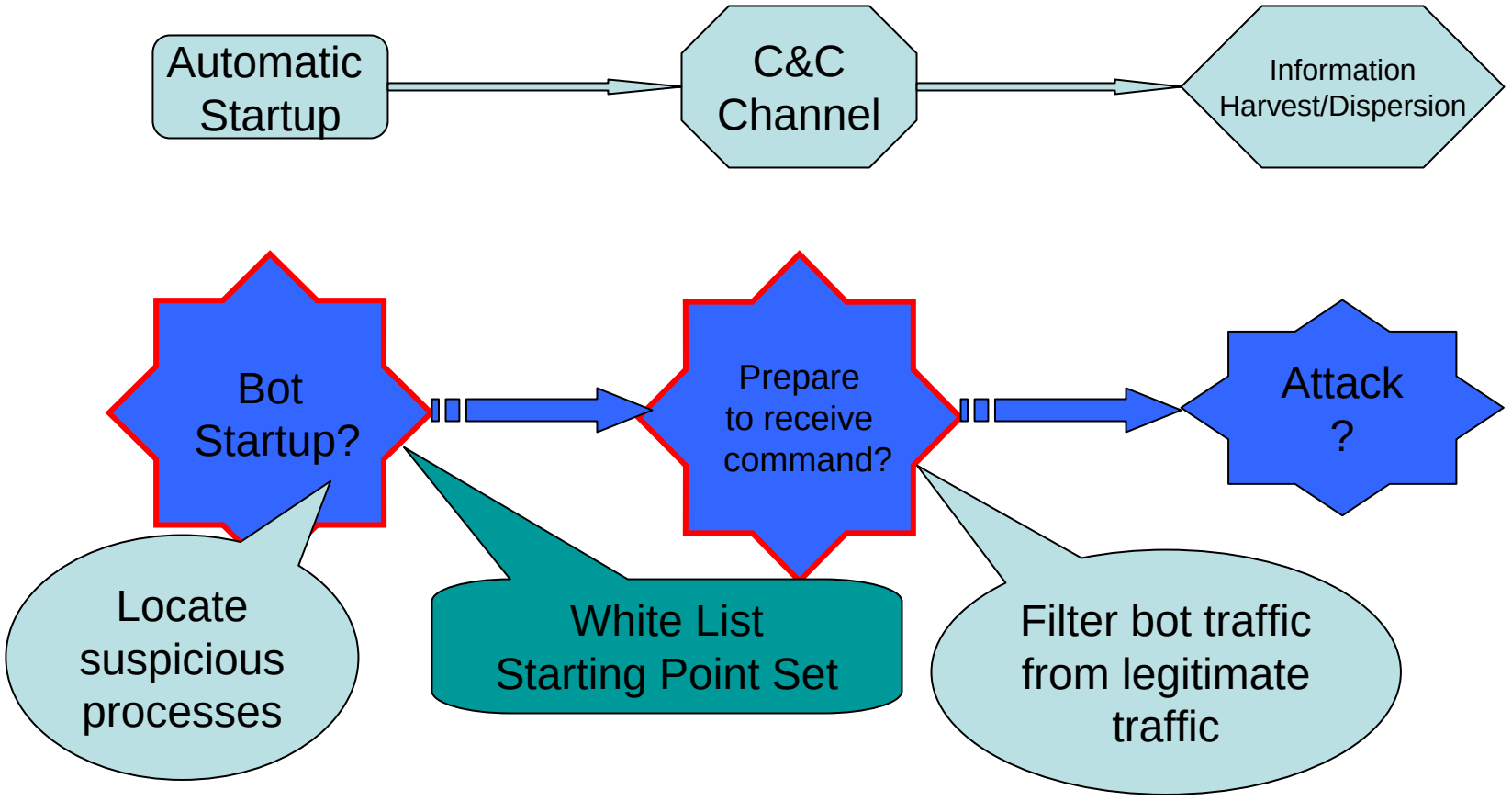
- Synchronizer
  - Static
  - Dynamic
- Traffic Monitor
  - Identify C&C channel
- Behavior Analyzer
  - Monitors suspicious process behaviors



Outline

- Design Principles and Overview
- Design Issues
- Evaluations
- Conclusions

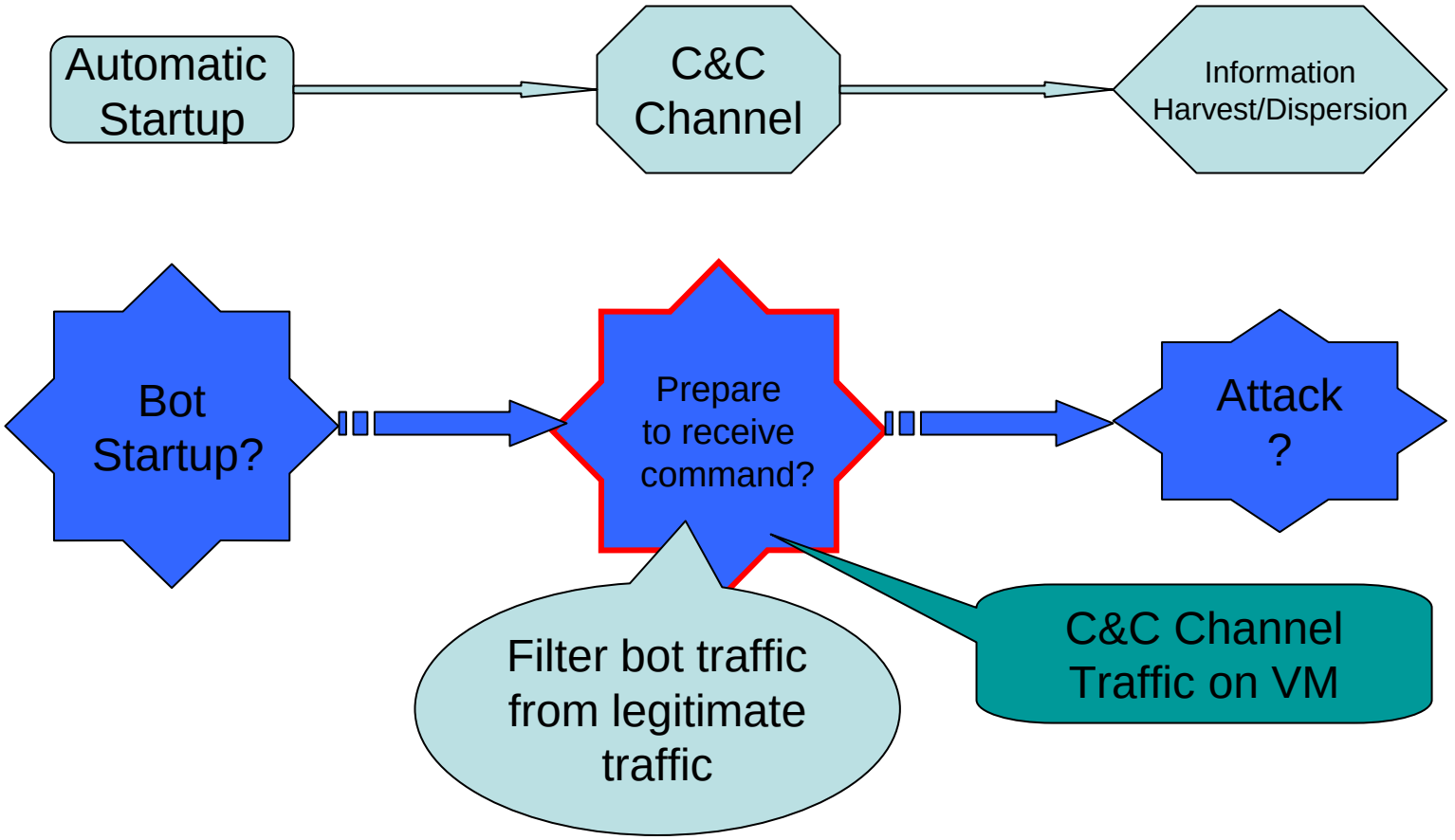
Phase I Challenge



Whitelist and Starting Point Set

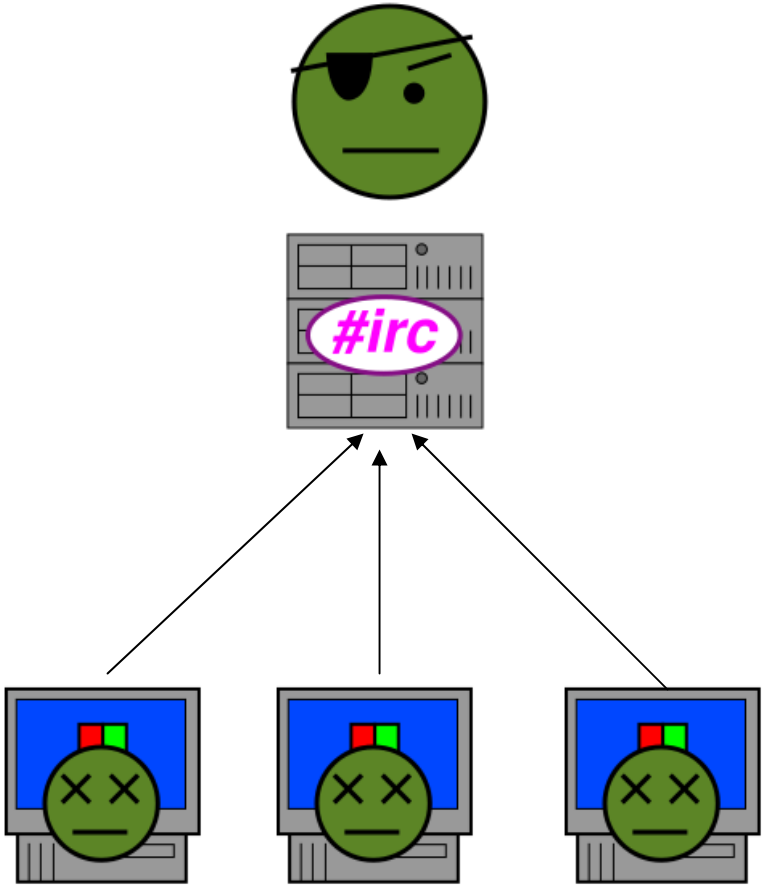
- Whitelist Legitimate Processes and Traffic
  - System daemons (services.exe)
  - Software update
  - Other known process (MSN, Yahoo)
- Disable Connections to Starting Point Set
  - Exclude unnecessary traffic
  - Functionality of original copy
  - Performance

Phase II Challenge



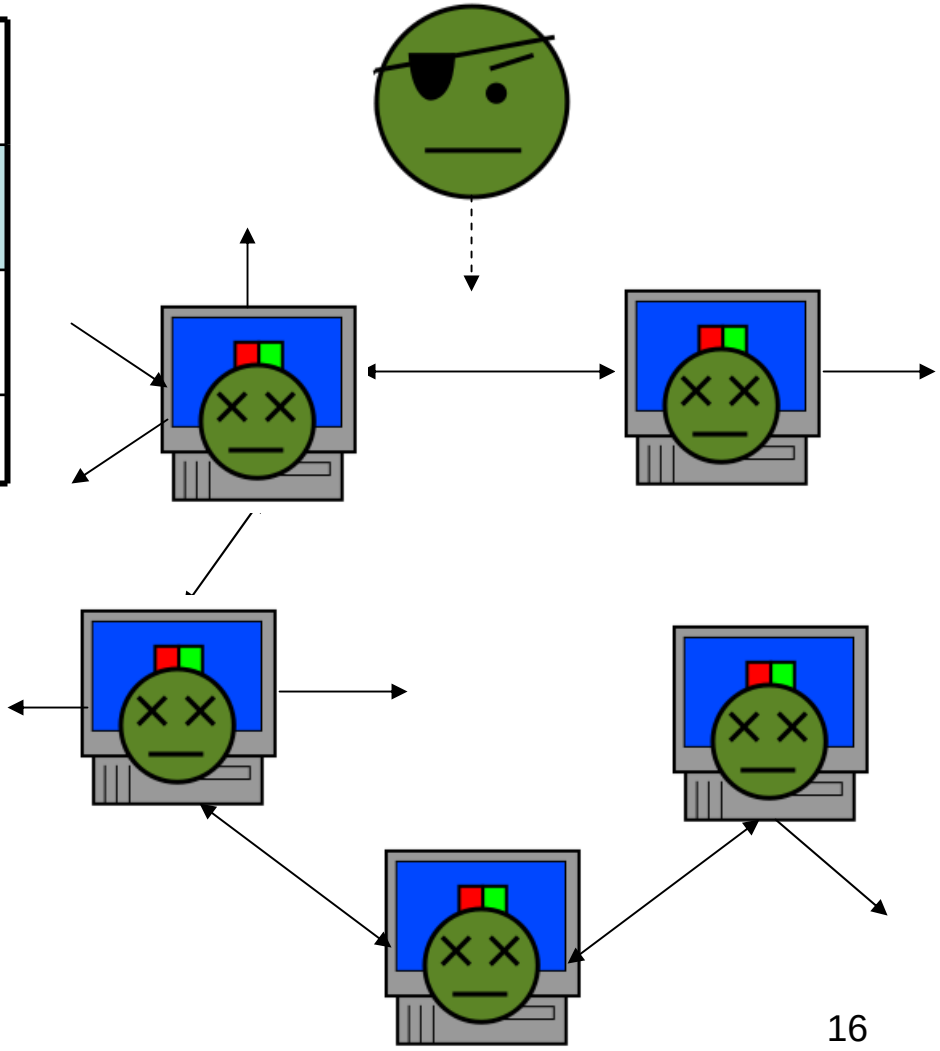
Command and Control Channel

Architecture	Centralized	IRC
	Decentralized	P2P
Type	Persistent	IRC
	Periodic/Sporadic	Web based HTTP



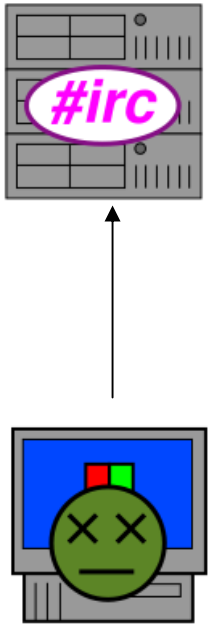
Command and Control Channel

Architecture	Centralized	IRC
	Decentralized	P2P
Type	Persistent	IRC
	Periodic/Sporadic	Web based HTTP



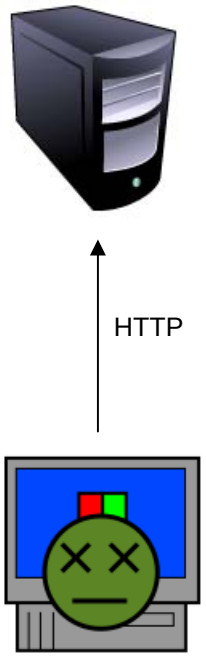
Command and Control Channel

Architecture	Centralized	IRC
	Decentralized	P2P
Type	Persistent	IRC
	Periodic/Sporadic	Web based HTTP

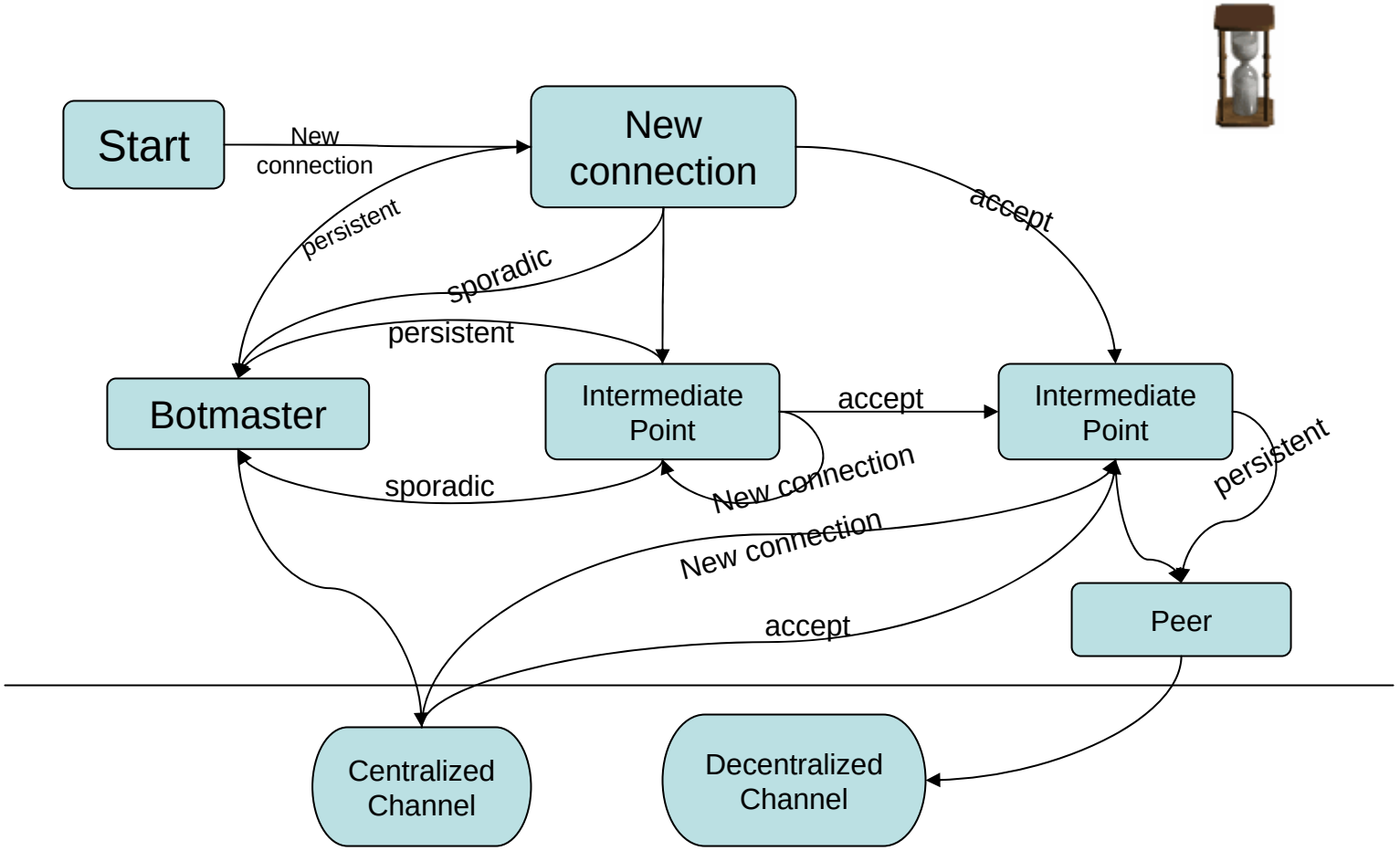


Command and Control Channel

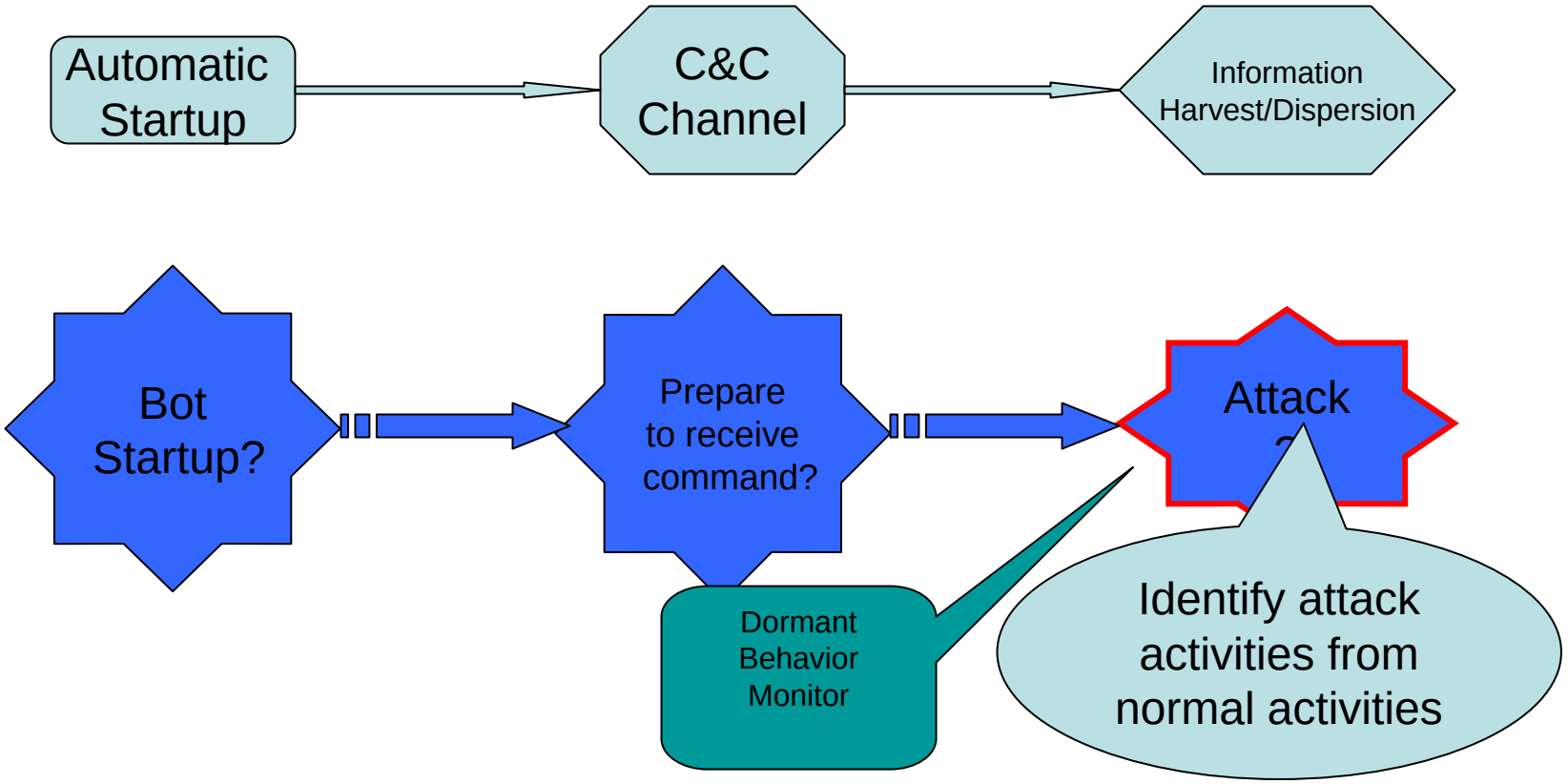
Architecture	Centralized	IRC
	Decentralized	P2P
Type	Persistent	IRC
	Periodic/Sporadic	Web based HTTP



Command and Control Channel Event Model



Phase III Challenge



An Example Dormant Profile

- IE Profile

{

program name =

C:\Program Files\Internet Explorer\iexplore.exe

starting point set = www.google.com

registry access = false

file access function = GetFileSize

file access directory =

C:\Documents and Settings\user\Local Settings

\Temporary Internet Files\Content.IE5\index.dat

……

}

Behavior Monitor

Traditional	BotTracer
Exhaustive analysis	Only on dormant status
Huge profile	Small profile
Long analysis time	Quick analysis
High false positive rate	Low false positive rate

Information Harvesting Detection

- Disk Access APIs
  - OpenFile, CreateFileMapping ……
- Memory Access APIs
  - WriteProcessMemory, ReadProcessMemory ……
- Registry
  - RegOpenKeyEx …….

Information Dispersion Detection

- Common Attacks by Botnets
  - Port scan
  - Infection attempts
  - DDoS, Spam
- BotTracer Solution
  - Connection: new connection/failure rate
  - Content: signature
  - Protocol: HTTP, SMTP

Outline

- Design Principles and Overview
- Design Issues
- Evaluations
- Conclusions

Experimental Setup

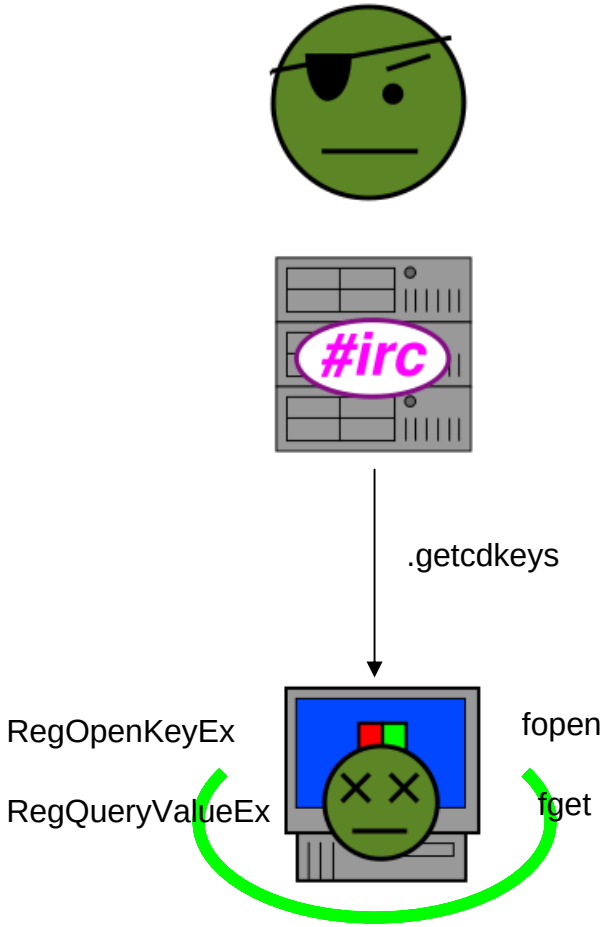
- BotTracer runs on Windows XP Professional, 2.79 GHz CPU and 2 GB RAM.
  - VMWare Workstation 5.5
  - Guest OS is cloned by converter, Windows XP Professional
  - API interceptor: Microsoft Detours
- Bot Samples
  - IRC bots and their variants:Agobot4 private, Forbot, Jrbot,Sdbot, Reptilebot, and Rxbot
  - P2P: Nugache
  - Other Protocols: Graybird

Channel Establishment Detection

Name	Alarm Time (s)	Architecture	Type
Agobot	6.532	Centralized	Persistent
Forbot	34.173	Centralized	Persistent
Jrbot	1.895	Centralized	Persistent
Reptilebot	2.719	Centralized	Persistent
Sdbot	0.953	Centralized	Persistent
Rxbot	4.409	Centralized	Persistent
Graybird	2.997	Centralized	Persistent
Nugache	1.422	Suspicious	Suspicious

Rxbot GetCDKeys

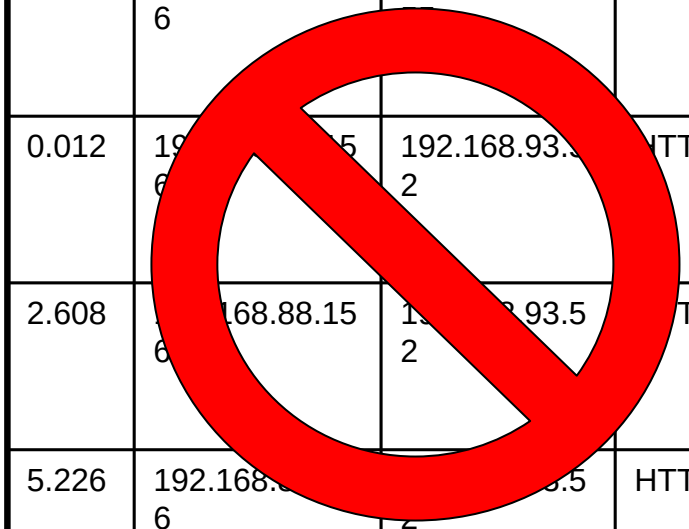
Action	API Call	Arguments
Access Registry	RegOpenKey RegQueryVal e	Software\BioWare\NWN\Neverwinter
Access Director	fopen fget	Neverwint Nights\NWN\nwncdkey.ini
...	...	...



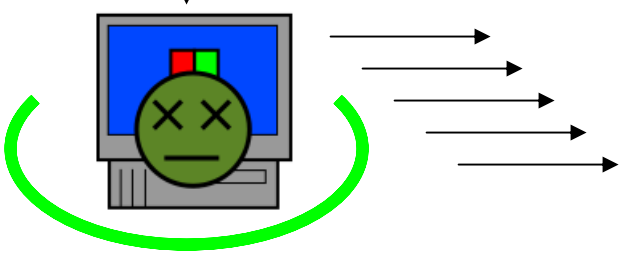
No profile/No action in profile
Alarm raised!

Agobot HTTP DDoS Attack Packets

Times	Source	Destination	Type
0	192.168.88.15 6	192.168.88.1	IRC
0.012	19 6	192.168.93. 2	ITTP
2.608	168.88.15 6	93.5 2	TP
5.226	192.168. 6	.5	HTTP
.....			



.ddos.httpflood http://www.aaaaa.com
100 www.aaa.com 2000



BotTracer Behavior Monitor

Connection rate over Threshold, alarm raised!

False Positive Test

- Microsoft Outlook Express 6
  - ***Without*** Starting Point Set
  - ***With*** Starting Point Set
  - Conclusion:Good idea to have Starting Point Set
- pcAnywhere
  - Has nearly the same functionality as Graybird
  - Different traffic pattern

Outline

- Design Principles and Overview
- Design Issues
- Evaluations
- Conclusions

Conclusions and Future Work

- Based on basic characteristics of botnets, we propose BotTracer to execute malware samples in a controlled environment to detect bot behaviors
- We have implemented and experimented on real bot samples to demonstrate its feasibility
- We need to further improve
  - How about BotTracer without VM?

Thanks

&

Questions?