Symptoms-Based Detection of Bot Processes

Jose Andre Morales Erhan Kartaltepe
Shouhuai Xu Ravi Sandhu

MMM-ACNS – St Petersburg, Russia 2010

Introduction

- Botnets (centralized & P2P): spam distribution, DoS, DDos, unauthorized FTP, etc.
- Bot masters lease their botnets = $$$$$$$
- Current research focuses on detecting infected bot machines but not the actual process on that machine
- This is good for botnet identification but for disinfection, process information is mandatory

Introduction - 2

- We attempt to fill this gap by identifying the actual bot process running on compromised machines with behavior based detection of bot/malware symptoms
- We study the execution behavior of known bot samples and attempt to distinguish characteristics exclusive to a bot and/or malware process
- We partition the behaviors into symptoms as basis of detection algorithm: Bot network behavior, Unreliable provenance and Stealth mechanisms
- Use data mining algorithms along with logical evaluation of symptoms to detect bots

Contributions

- The process-based identification of:
  - Bot network behavior, Unreliable provenance, Stealth mechanisms:
- A formal detection model based on non-trivial use of established data mining algorithms (C4.5).
  - Generate and evaluate detection models. Results show our methodology has better detection accuracy for both centralized and Peer-to-Peer (P2P) bots than a straightforward use of established data mining algorithms.

Observed Behaviors

- B(P) Bot Network: tcp, udp, icmp, dns usage
- U(P) Unreliable provenance: process self replication and dynamic code injection, & verified digital signature
- S(P) Stealth mechanisms: lacking a GUI & no user input to execute
- Analyzed in real time

Bot Behavior Symptoms

- DNS/rDNS highly used by bots to:
  - Locate active remote hosts, harvest new IP addresses
  - Successful DNS/rDNS should connect, failed should not
  - Bots may depend on DNS for botnet activity
- B1: Failed connection attempt to the returned IP address of a successful DNS query.
- B2: IP address in a successful DNS activity and connection. This is considered normal behavior.
- B3: Connection attempt to the input IP address of a failed reverse DNS query.

Unreliable Provenance Symptoms

- Most malware lack digital signatures, self replicate and dynamically inject other running processes with malicious code
- U1: Standalone executable's static file image does not have a digital signature.
- U2: Dynamic code injector's static file image does not have a digital signature.
- U3: Creator of process's static file image does not have a digital signature.

Stealth Mechanism Symptoms

- Malware execute in “silent” mode requiring no user interaction: no GUI & no user input
- S1: Graphical user interface. A process executing without a GUI
- S2: Human computer interface. A process executing without reading keyboard or mouse events is considered to have a stealth mechanism.

Evaluation

- Four symptom evaluations to predict a bot: Bot(P) -> T or F
- Bot( ) constructed by function f as follows:
  - f0: established data mining algorithm → J48
  - f1: B(P) or (U(P) and S(P))
  - f2: B(P) and (U(P) or S(P))
  - f3: B(P) and U(P) and S(P)
- F3 most restrictive requiring all three symptoms present to identify a bot
- Evaluations partially based on J48 classification trees

Data Collection – Training Set

- Vmware workstation: XP-SP2; Windows network monitor, sigcheck, various hooking techniques, 20 bot & 62 benign processes
- 4 active bots: virut, waledac, wopla & bobax
- 5 inactive bots: nugache, wootbot, gobot, spybot & storm
- 41 benign applications
- Bots executed for 12 hour period, results drawn from post analysis of log files
- Benign data collected on two laptops 12 hour period: FTP, surfing, P2P, instant messaging and software updates
- Bots and benign samples executed multiple times

Data Collection – Test Set

- Test data collected on 5 laptops
  - Minimal security
  - No recent malware scans
  - 8 to 12 hours
- Post scan malware analysis revealed two bot processes
  - Cutwail bot: servwin.exe
  - Virut bot: TMP94.tmp
- Cutwail bot not part of training set
- Test set consisted of 34 processes including 2 bot processes, the rest were assumed benign
- Several benign processes not part of training set

Bot Predictions

Process Name	Bot Network Activity Behavior				Unreliable Provenance				Stealth Behavior			Bot Prediction			
	b_1	b_2	b_3	$B(P)$	u_1	u_2	u_3	$U(P)$	s_1	s_2	$S(P)$	f_0	f_1	f_2	f_3
svchost.exe	N	0	N	F	N	N	N	F	N	N	T	F	F	F	F
googletalk.exe	N	2	N	F	N	N	N	F	Y	Y	F	F	F	F	F
firefox.exe	N	5	N	F	N	N	N	F	Y	Y	F	F	F	F	F
cutftp32.exe	Y	1	N	T	Y	N	N	T	N	N	F	F	T	T	F
firefox.exe	N	44	N	F	N	N	N	F	Y	Y	F	F	F	F	F
svchost.exe	N	0	N	F	N	N	N	F	N	N	T	F	F	F	F
servwin.exe	Y	0	Y	T	Y	N	N	T	N	N	T	T	T	T	T
Framework Services.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F
iexplore.exe	N	126	N	F	N	Y	N	T	Y	Y	F	T	F	F	F
firefox.exe	N	49	N	F	N	Y	N	T	Y	Y	F	T	F	F	F
rundll32.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F
firefox.exe	N	67	N	F	N	N	N	F	Y	Y	F	F	F	F	F

Bot Predictions

firefox.exe	N	7	N	F	N	N	N	F	Y	Y	F	F	F	F	F
iexplore.exe	N	54	N	F	N	N	N	F	Y	Y	F	F	F	F	F
firefox.exe	N	45	N	F	N	N	N	F	Y	Y	F	F	F	F	F
firefox.exe	N	10	N	F	N	N	N	F	Y	Y	F	F	F	F	F
SshClient.exe	N	1	N	F	Y	N	N	T	Y	Y	F	F	F	F	F
BitLord.exe	Y	1	N	T	Y	N	N	T	N	N	F	F	T	T	F
Acrobat.exe	N	1	N	F	N	N	N	F	Y	Y	F	F	F	F	F
Thunder5.exe	Y	13	N	T	N	N	N	F	Y	Y	F	F	T	F	F
Thunder Minisite.exe	N	7	N	F	N	N	N	F	Y	Y	F	F	F	F	F
Thunder5.exe	Y	24	N	T	N	N	N	F	Y	Y	F	F	T	F	F
wmplayer.exe	Y	17	N	T	N	N	N	F	Y	Y	F	F	T	F	F
setup_wm.exe	N	1	N	F	N	N	N	F	Y	Y	F	F	F	F	F

Bot Predictions

chrome.exe	N	3	N	F	N	N	N	F	Y	Y	F	F	F	F	F
TMP94.tmp	N	3	Y	T	N	Y	N	T	N	N	T	T	T	T	T
Google Update.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F
Google Update.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F
chrome.exe	N	28	N	F	N	N	N	F	Y	Y	F	F	F	F	F
Adobe_ Updater.exe	N	2	N	F	N	N	N	F	Y	Y	F	F F	F F	F F	F F
gup.exe	N	1	N	F	N	N	N	F	Y	Y	F	F	F	F	F
Tvanst.exe	Y	1	N	T	Y	N	N	T	N	N	F	F	T	T	F
msfeeds															
sync.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F
zclientm.exe	N	1	N	F	N	N	N	F	N	N	T	F	F	F	F

Table 2. Test Set: Decision Tree and Bot Process Predictions

Prediction Results

- f0: simplistic use of J48 classifier; 2 FP, 0 FN.
- f1: least restrictive; 6 FP, 0 FN.

  B(P) or (U(P) and S(P))
- f2: more restrictive; 3 FP, 0 FN

  B(P) and (U(P) or S(P))
- f3: most restrictive; 0 FP, 0 FN

  B(P) and U(P) and S(P)

Discussion

- FP were a mix of browsers, FTP, video streamers, P2P & torrent clients
- Both bots in test set detected by all 4 functions. The different functions f only served to eliminate FP
- F3 gave the best results by eliminating all FP, suggesting a high restriction can improve results in bot detection
- F1 & F2 with weaker restrictions produced more false positives but may be applicable in detecting non-bot malware
- Symptoms B1, B2, U1, U2 & S1 used in final bot prediction; S1 most dominant with 13 processes
  - Several benign samples were system services running in background

Conclusion

- Presented 3 sets of symptoms usable in detecting bot processes
- Enhances current research which focuses most on bot machines
- Results drawn from real time data collection
- Most restrictive evaluation most suitable for bot detection, but combining with less restrictive may detect broader range of bots and non-bot malware
- Future Work: identify more symptoms, test with kernel based bots and implement automated detection techniques

THANK YOU !

QUESTIONS?

вопрос