Information Operations Immunity Style

www.immunityinc.com

Agenda

- Scenario
- Problems of scale when hacking
  - Client-sides
- Immunity's PINK Framework
- Trojaning hard targets
  - Immunity Debugger Parasitic Infection

Scenario

- Modeling attack on high value target
- Long scale attack
- Wide internal scope

IO simulation vs. Pen-test

- Modern pen-test is compressed timescale.
- IO is not. Time passes, collection occurs.
- Collection over time gives clear picture of the network, people and data.
- No need for blind network scans or random break-ins. First learn where to go.
- Exploit trust!

Soft direct approach

- Did not start with client-sides
  - client-sides are somewhat blind
  - detection is much easier for smart opponent
  - hard to clean up after them
- Recon
  - Intense versioning on mail server
  - One box only
  - No class-C scan
  - No port scan of that one box

Soft direct approach - II

- Audit 3rd party AV-SPAM product on MTA Gateway. Easier task than to look into core OS components.
- Extensive file format parsing proven by many researchers to be badly implemented.
- AV on gateways has to be hi-avail, which means watchdogs and intensive exception-handling. Memory corruptions handled or process restarted.
  - Gives unlimited exploitation trial.

Soft direct approach - III

- Model your target in lab.
- Vmware vs. Real Iron
- Language detection is key (CANVAS)
- Extensive modeling of your target in lab cuts down the exploit development time by half.
- AV products vague about restarts and crashes. Makes attempts less suspicious.
- Almost all breaks DEP and SafeSEH. Most compiled with Borland = insecure heap metadata.

Why not the web server

- Web server was on some random other ISP
  - Dry content without useful logic
  - hard targets are just that – HARD
  - Even if we broke into the web server, no guarantee of anything useful there
  - Apache + IIS only players
    - Hard to audit – large investment

Recon results

- MTA Gateway
  - No big corporation can run without SPAM/Malware filter
  - Guaranteed to exist
  - Hard to fingerprint – Protocol response is the best way (now in CANVAS)

Audit results

- Heap overflow in unpacking (quite common)
- Exploitation vector:
  - Email attachment
  - Could be send to void user
  - Scanned no matter what, than discarded
  - Not much trace left even after failed exploitation
  - DEP disabled, Watchdog restarts process

Custom Payload

- First a MOSDEF shell
- Than custom dll payload for email collection
- Hooks API within the AV process to get a copy of the scanned email
- Stores email in achieve file for later collection
- Custom DLL injected into memory (MS detours!) also PE header of a random product DLL
- Also scans email content for keyword to callback MOSDEF shell to encoded IP

Further breach

- Email collection over long period leads to further breach
- DMZ to internal LAN cross over is simple with acquired intelligence
- Exploiting trust is trivial at this point
- Now you know which internal box is high value

Further breach - II

- Broke into PDC with DNS msrpc exploit
- Obtained domain admin hash
- Installed executable remotely to high value target using the admin hash (CANVAS)
- Recently accessed files folder had soft-links to high value content but files were not on the HDD
- USB drive!

Breaching the Air-Gap

- USB drive a goes between segmented development network and the Internet network
- Error logs from 3rd party product is emailed to the support group
- Error logs are generated in the same folder with the high value content so high value content travels with them!
- USBDumper comes to mind! - http://www.secuobs.com/news/07062006-sstic_usbdumper.shtml

Breaching the Air-Gap – II

- Modified USBDumper for in-memory injection
- Same DLL injection trick
- Added file tracking and disk free space tracking
- Once again, time passes.
- Eventually partial access to high value "segmented" data
- Breach vector: Simply a tainted USB drive

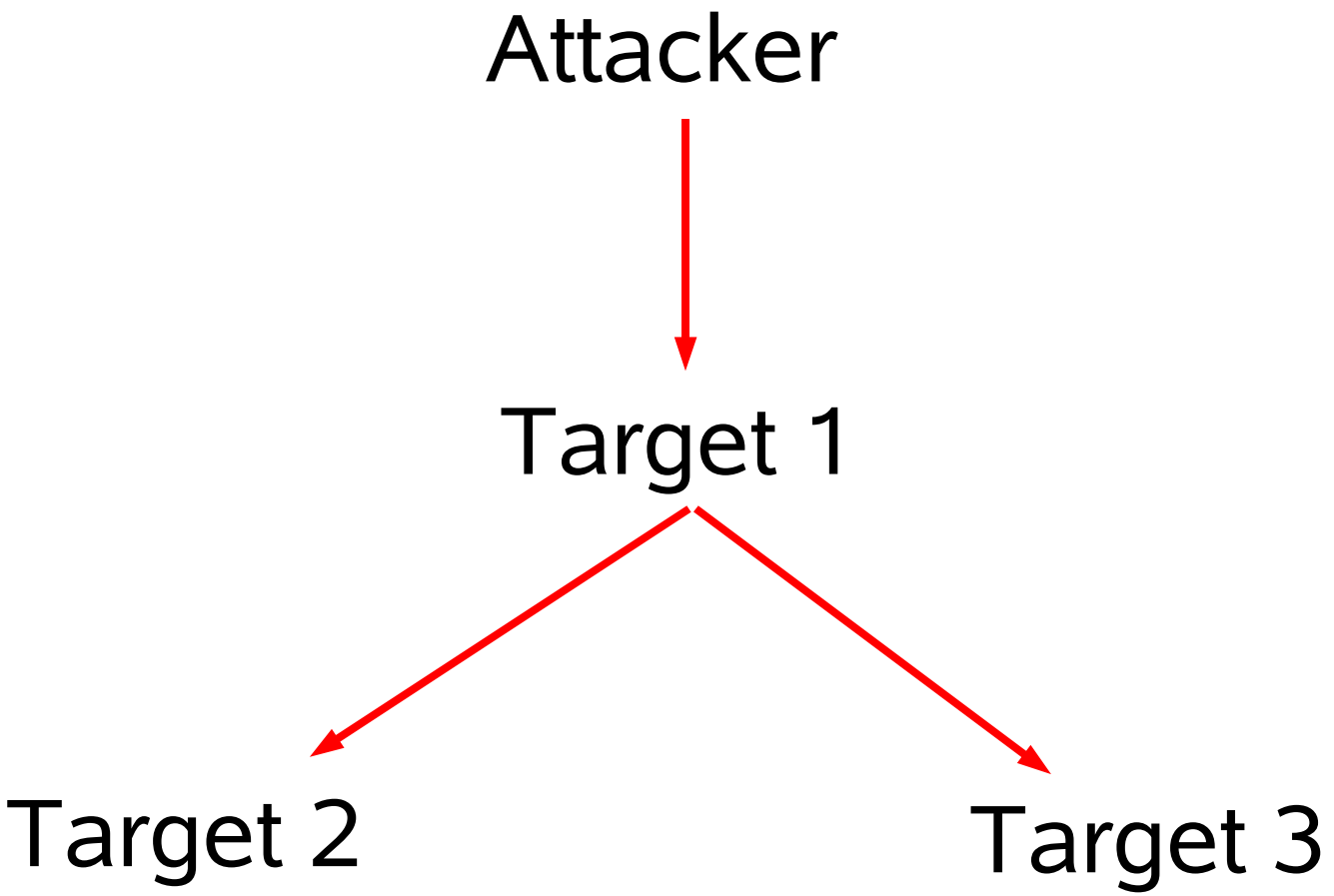
Agenda

- ~~Scenario~~
- Problems of scale when hacking
  - Client-sides
- Immunity's PINK Framework
- Trojaning hard targets
  - Immunity Debugger Parasitic Infection

Targets are ephemeral

- Time
  - Your workstation turns on and off as you come to work
- Location
  - Your laptop travels across network security boundaries
- Configuration
  - Your server is upgraded, reconfigured, network infrastructure changes around it

Command and Control in most hacking platforms is a tree



IMMUNITY

Networks are not trees

- A fully connected graph is what we want
  - Self routing with some human input
- This is a hugely expensive solution
  - Management costs
  - Development costs
    - Need to emulate TCP over thousands of protocols
    - Those who don't use TCP are doomed to re-implement it...

Building and storing routing tables is a hard problem

- Harder for us due to covertness
- We don't want any node to have a larger picture of all the other owned nodes than it absolutely has to
- Automatic solutions are possible, but for now, manual operation of routing is easiest

Scalability problems

- Management of one hundred ants is easy
  - Picture of thirty million ants
- A good client-side vulnerability can be used to own a quarter million boxes a day
- Future work involves self-directed worms

Asymmetric attack means we need to not have a rack of machines

- Portable C&C
- Scalable C&C
- Covert C&C
- Immunity's PINK infrastructure solves these problems

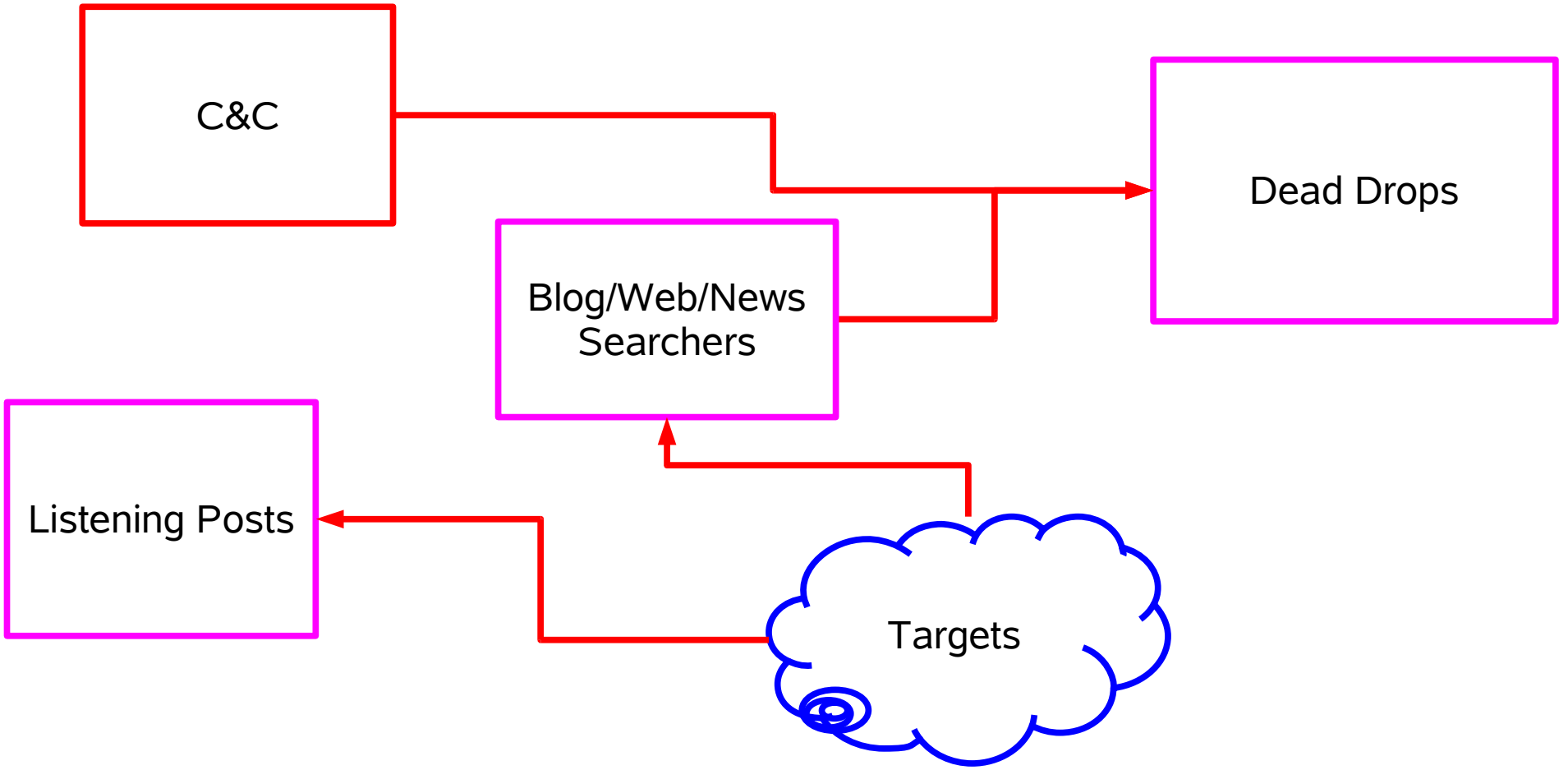
Current Botnet C&C technology

- IRC
- HTTP to single server
- Fast-Flux of DNS Servers
- Storm P2P protocols

Covertness or Reliability?

- P2P is reliable, not covert
  - Requires chatty communications on the network
  - Difficult to pass through strict proxies
  - Easily fingerprintable

PINK C&C Framework



IMMUNITY

Blogsearch

- Blog searching is the current best parasitic host protocol for PINK
  - Almost instantaneous responses
  - Easy to find hosts for our blogs
  - Lots of signal to hide in
- Any search engine will do though

IMMUNITY 

PINK DEAD DROPS

- <Cover Text>
- <TRIGGER>
- <base 64><RSA Encrypted/Signed Command></base64>
- <END TRIGGER>
- <More Cover Text>

Each Target is looking for multiple triggers

- Goal is to divide our targets into manageable sets
  - Per Country
  - Per Company
  - Per Domain
  - Per Time-of-exploit
  - etc
- “All hosts from immunityinc.com” please contact listeningpost.my.com using HTTP MOSDEF on port 443
- All target.com's please deliver any .xls with “Payroll” string to email address bob@example.com

Signed and Encrypted payloads prevent replay attacks with removal kits

- Triggers need to be signed with time-based key as well
- Making triggers strings of random words makes it hard for search engines to filter our requests

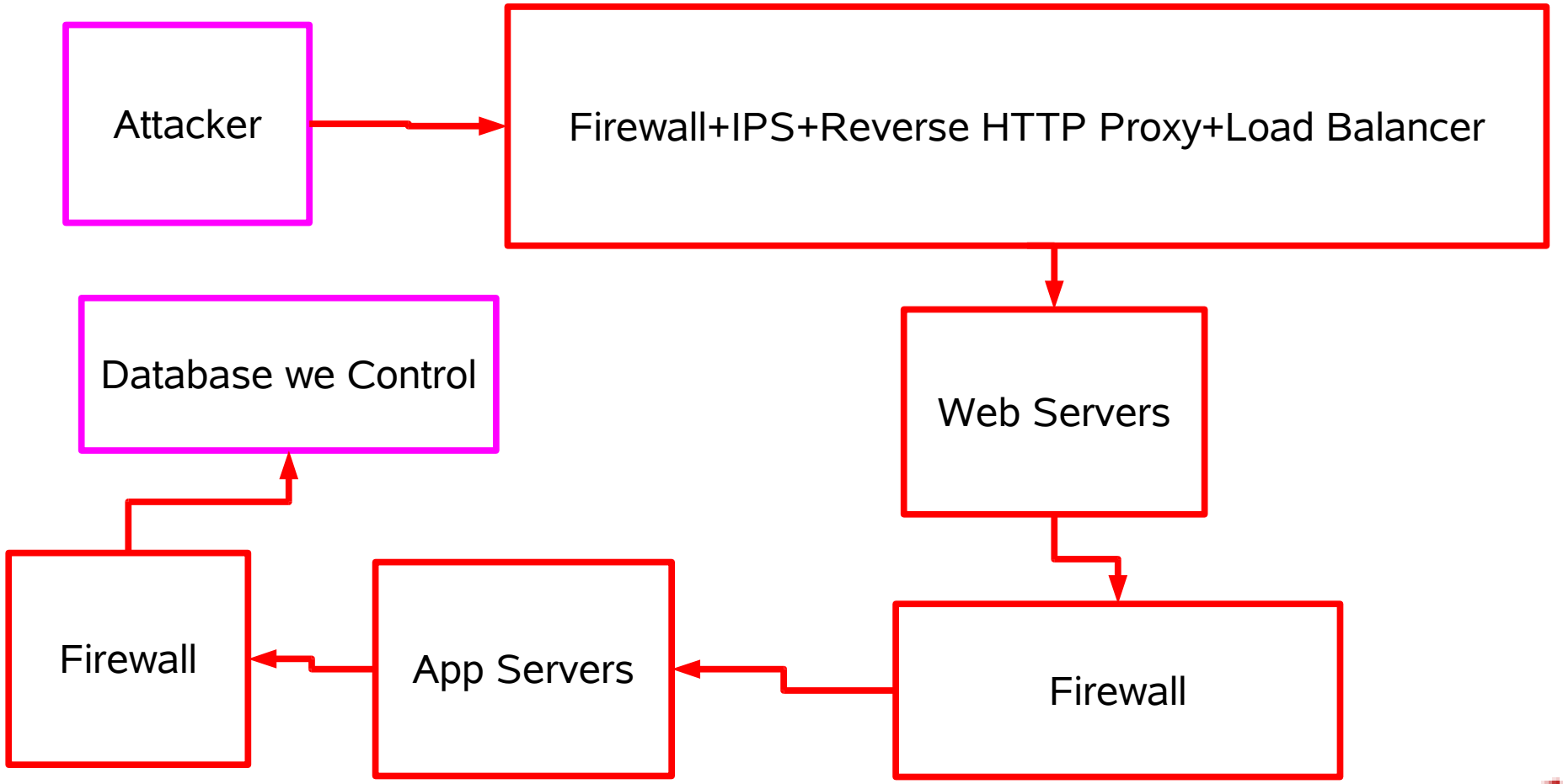
Client-side conclusions

- Currently in Beta-testing state – pushing out to CANVAS shortly
- Parasitic C&C is:
  - Nearly impossible to detect and monitor
  - Easily re-targetable to any search engine or search option on a web page
  - Does not require expensive infrastructure to maintain

Servers and hard targets

- Servers may not be able to contact us via HTTP
- Need way to connect to stationary targets behind firewalls and application proxies covertly
- Each target is different!
- Example target: MS SQL Server 2005 in strict DMZ tier

Every web application is a unique snowflake



IMMUNITY

Custom automatic backdoors

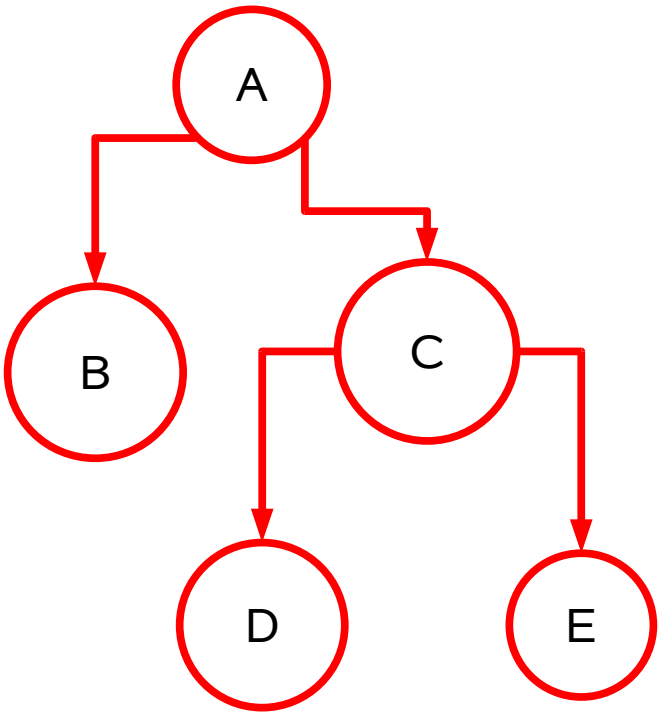
- Use Immunity Debugger to analyze target .exe/.dll
- Send traffic to it and trace where our triggers are seen
- Create custom patch to PINKize target .dll and write this to disk and memory
- Box is now trojaned in a way that does not require direct connectivity!

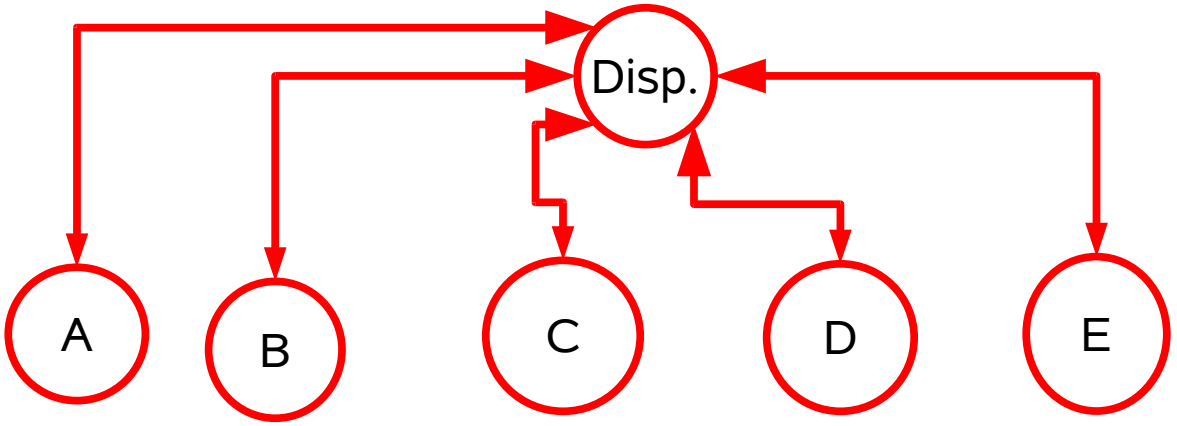
Why Immunity Debugger?

- Includes built in analysis engine
- Full Python scripting API can do both dynamic and static analysis
- Send data to the server and then see what API it triggers
- Mutate our parasite to look statistically like the target program
- Trojan in memory or on disk or both

Avoiding Structural BinDiff

- Change all CALL opcodes to point to our dispatcher
- Have dispatcher send hooked API's to our code instead





IMMUNITY

Overall Conclusions

- Botnets and trojans will be extremely difficult to find and analyze in the near future.
- Nascent market shift to automated incident response as part of vulnerability analysis faces ongoing challenges as attackers build one-time custom-use trojans

IMMUNITY