Information Operations Immunity Style

www.immunityinc.com

IMMUNITY

Agenda

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

Real Life Scenario

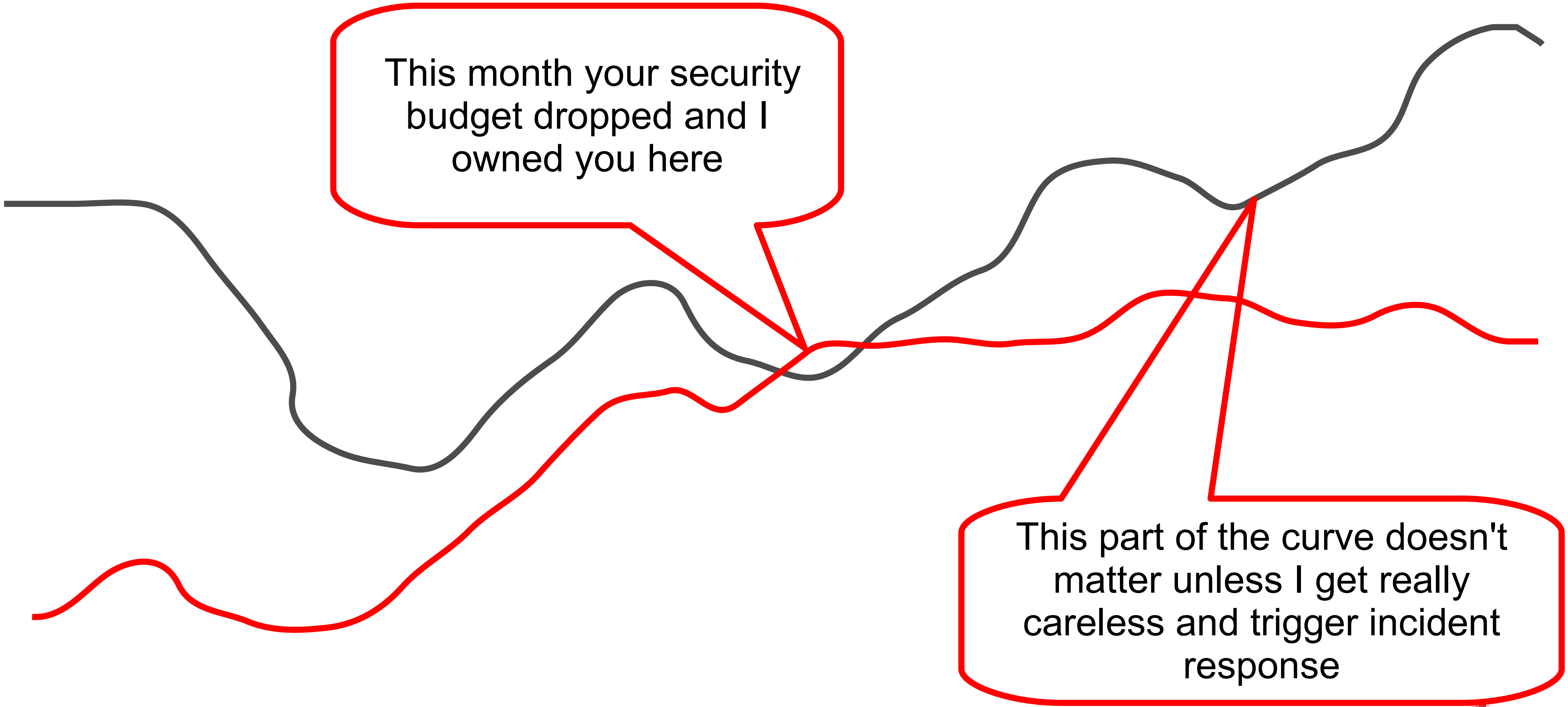
- Modeling attack on high value target
- Long time scale operation
- Wide internal scope
- A different kind of contract than pen-testing
- Immunity calls this “Information Operation (IO)”

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!

IMMUNITY

Your Network vs Your Attacker



This month your security budget dropped and I owned you here

This part of the curve doesn't matter unless I get really careless and trigger incident response

IMMUNITY

Model of attacker

- Guaranteed to exist
  - Web server
  - MTA server
  - DNS server
  - Border Routers, FW / VPN
  - Endpoints (unknown internal networks)

Not the web server

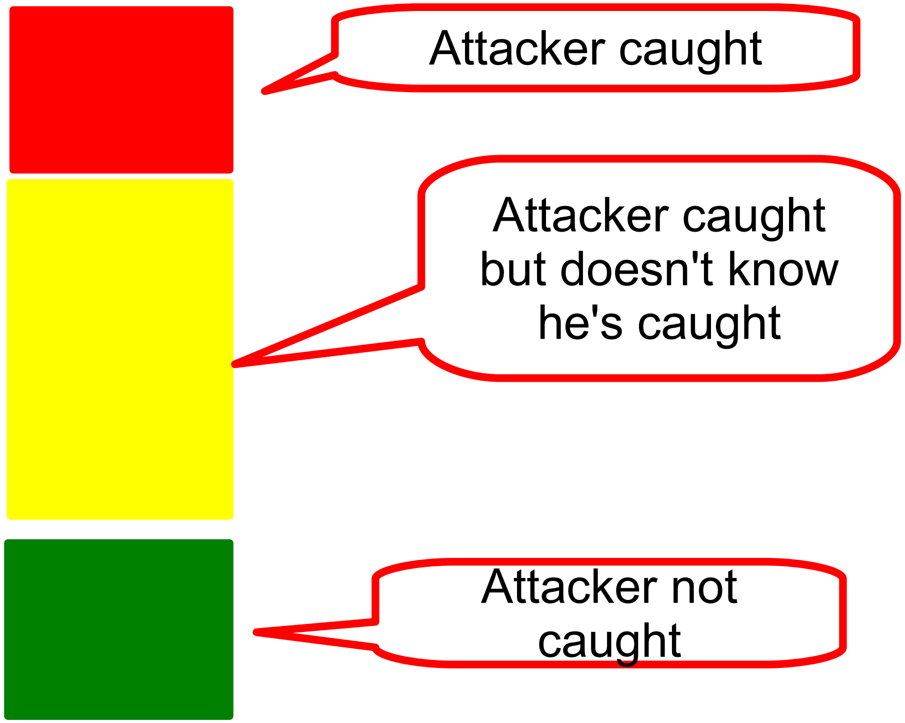
- Web server was on some random other ISP
  - Dry content without useful logic
  - Hard targets are just that – HARD
  - Even if we broke into the web server, no guarantee of anything useful there
  - Apache + IIS only players
    - Hard to audit – large investment

Not the infrastructure

- Routers
  - Embedded device exploitation is fun but
    - Costly lab setup
    - Hard to get it right for all potential firmware
    - Might not detect exact hardware (mips vs. ppc)
- VPN
- Firewall

Not the endpoint

- Did not start with client-sides
  - client-sides are somewhat blind
  - detection is much easier for smart opponent
  - hard to clean up after them



The MTA

- Intense versioning on mail server
- One box only
- No class-C scan
- No port scan of that one box
- MTA Gateways
  - No big corporation can run without SPAM/Malware filter
  - Hard to fingerprint – Protocol response is the best way (now in CANVAS)

Soft direct approach - I

- 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.

IMMUNITY

Soft direct approach - II

- Model your target in lab.
- VMware vs. Real Iron
- Language detection might be an issue
- 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 AV breaks DEP and SafeSEH. Most compiled with Borland = insecure heap metadata. Do not use /GS.

Audit results

- Heap overflow in unpacking (quite common)
- Alex Wheeler independently discovered the issue as well. Vendor patches available
- 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 by product, Watchdog restarts process

Custom Payload

- First a MOSDEF shell (CANVAS)
- Than custom backdoor DLL for email collection
- Inject custom DLL into memory (MS detours) and write into the PE header
- DLL hooks API within the AV process to get a copy of the scanned email
  - Stores email in archive file for later collection
  - Scans email content for keyword to callback MOSDEF shell to encoded IP

Further breach - I

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

Further breach - II

- Exploited Email chatter between user and 3rd party
- Used mail attachment to infect internal Desktop (PINK)
- 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 content not on the hard drive. USB drive!

Breaching the Air-Gap - I

- USB drive goes between segmented development network and the Internet network
- Error logs from 3rd party product are emailed to the support group
- Logs carried from segmented network to the Internet network
- USBDumper comes to mind!

Breaching the Air-Gap – II

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

Scenario Conclusions

- AntiVirus gateways are a serious security risk
  - Complex parser on crucial hosts!
- USB drives can be high value targets
- Relationship mapping is required in professional attack toolkits
  - More than just X knows Y – needs technical information about email content as well. Does X talk to Y about Z? Do they send PDFs about Q?

IMMUNITY

Agenda

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

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

Current Botnet C&C technology

- IRC
  - Easy to tear down, take over
- HTTP to single server
  - Share IRC's cons
- Fast-Flux of DNS Servers
  - Easy to block the domain requests
- Storm P2P protocols
  - Reliable but not covert
  - Does not pass through strict proxies

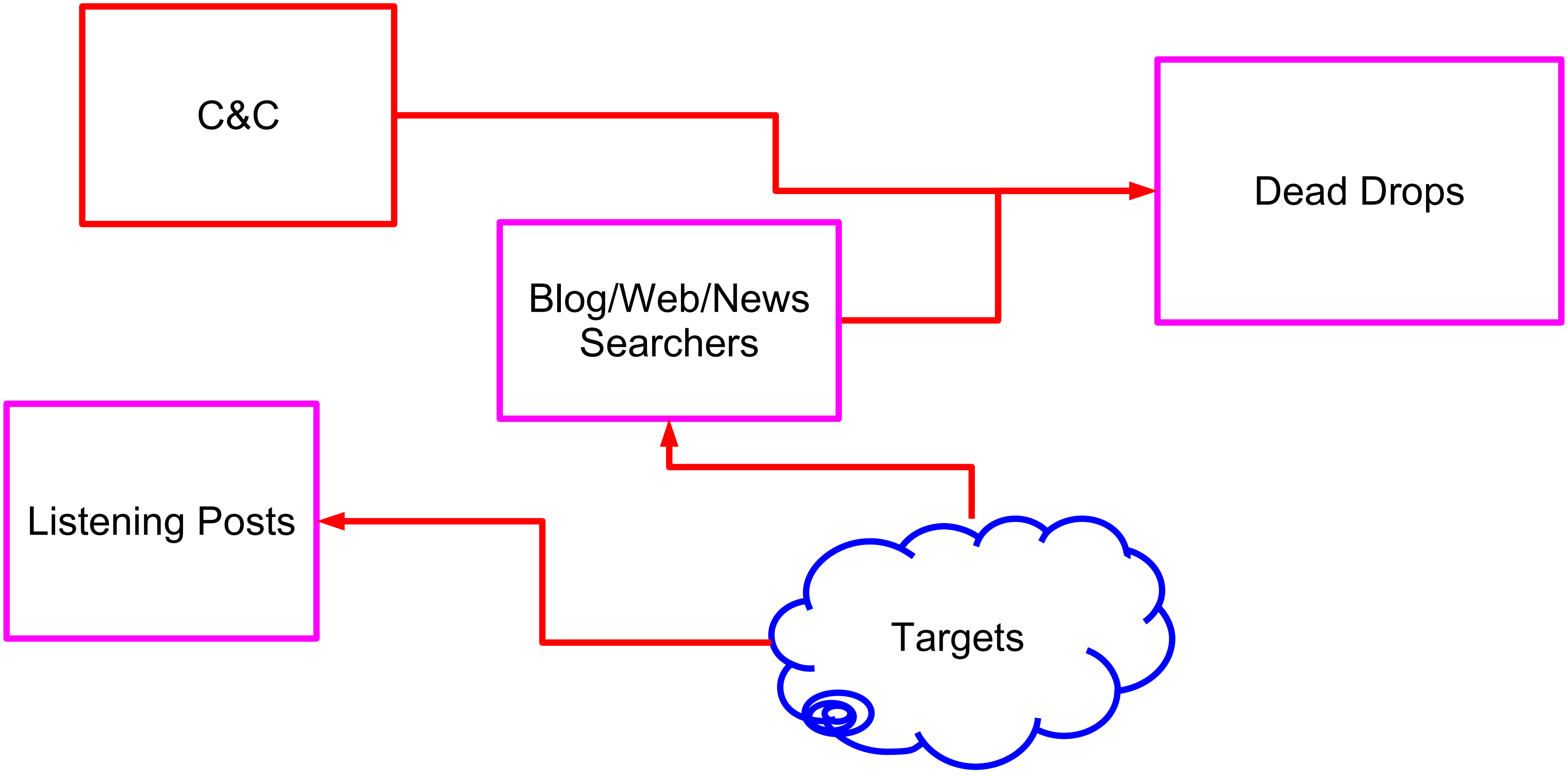
New C & C

- Need a new Command &Control technology
  - Scalable
  - Covert
  - Portable

Agenda

- ~~A Real to Life Scenario~~
- ~~Problems of scale when hacking~~
- Immunity's PINK Framework
- Trojaning hard targets
  - Immunity Debugger Parasitic Infection

PINK C&C Framework



IMMUNITY

Blog Search

- Blog searching is currently the best parasitic host protocol for PINK
  - Almost instantaneous responses
  - Easy to find hosts for our blogs
  - Lots of signal to hide in
  - RSS feeds
- Other search operations can be implemented as well

PINK Dead Drops

<Cover Text>

<TRIGGER>

<base 64><RC4 Encrypted/RSA Signed Commands></base64>

<END TRIGGER>

<More Cover Text>

PINK Dead Drops

- Signed and Encrypted payloads prevent replay attacks with removal kits
- Triggers need to be signed with time-based key as well. PINK verifies signature before command execution
- Trigger strings of random words makes it hard for search engines to filter our requests

PINK Tech - I

- Installs itself as a Shell Extension
- Does not require Admin privs due to current user-only registry key injection
- Persistent across reboots
- In DLL format within Explorer.exe
- Takes itself out of PEB loaded modules list
- Invisible in user mode

PINK Tech - II

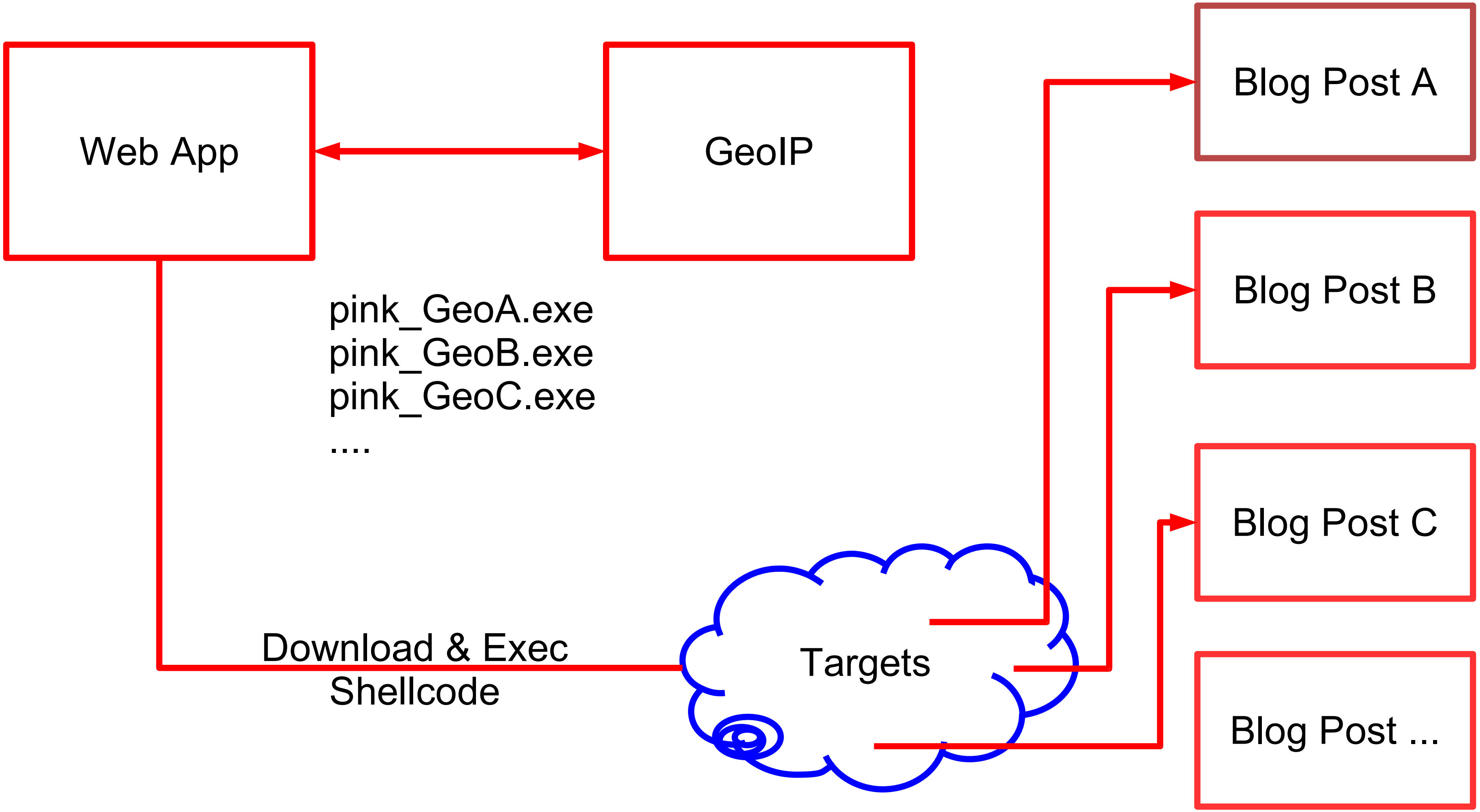
- No known AV product checks for malicious Shell Extensions.
- Initial loading of the shell extension requires a shell activity such as; copy, paste, delete, right-click, drag & drop etc. by end user
- Personal firewalls might trigger on Explorer.exe outbound connection. Easy problem to solve, hard to port across the whole market.

PINK Tech - III

- 3 components
  - PINK backdoor dll (shell extension)
  - PINK installer (dll embedded within)
  - Blog content generator TriggerText((RSA_sign(RC4_enc(Commands)));
- PINK installer changes before download to reflect a certain drone subnet
- GeoIP <-> Blog search

IMMUNITY

PINK Subnets



IMMUNITY

Targets & Triggers

- Goal is to divide our targets into manageable sets, Could be;
  - Per Country
  - Per Company
  - Per Domain
  - Per Time-of-exploit
  - etc
- Could than do things like;
  - "All hosts from immunityinc.com domain" please contact listeningpost.my.com using HTTP MOSDEF on port 443

PINK Tech - IV

- Internet searches on configurable timer. Every X hour
- When the timer expires, checks for user mouse, keyboard activity
- If none, sleeps on shorter intervals to check for user activity more often
- If user active, google search, find dead drop block, verify signature, decode
- Run commands, sleep on timer again

Current Pink Commands

- Callback over HTTP/HTTPS MOSDEF to CANVAS
- Callback over TCP MOSDEF to CANVAS
- Download from URL and Exec
- Download from URL and LoadLibrary
- Exec given string
- Upload file(s) to URL (ftp/http/https)
- Key log
- Update self
- Coming: Vbscripting

PINK conclusions

- Currently in Beta-testing state – pushing out to CANVAS shortly
- Parasitic C&C is:
  - Hard 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

PINK exploitation setup

- Client-Side exploit
  - Acrobat PDF reader through IE7
- Shellcode
  - UrlDownloadToCacheFile & WinExec
  - Downloads pink installer into IE cache and runs it
- Pink installer extracts pink.dll into a user directory
- Adds pink.dll as a shell extension
- Clean up

PINK demo

- TBD

Overall Conclusions

- IO proven itself. MTA compromised, Endpoint compromised, Air gap breached
- PINK introduces stealth and persistence on endpoints
- Recent market shift to automated incident response as part of vulnerability analysis faces ongoing challenges as attackers build one-time custom-use trojans and one-time use exploits

Epilogue

- Invest in human capital
  - Build and train teams
- Be on the offense

Agenda

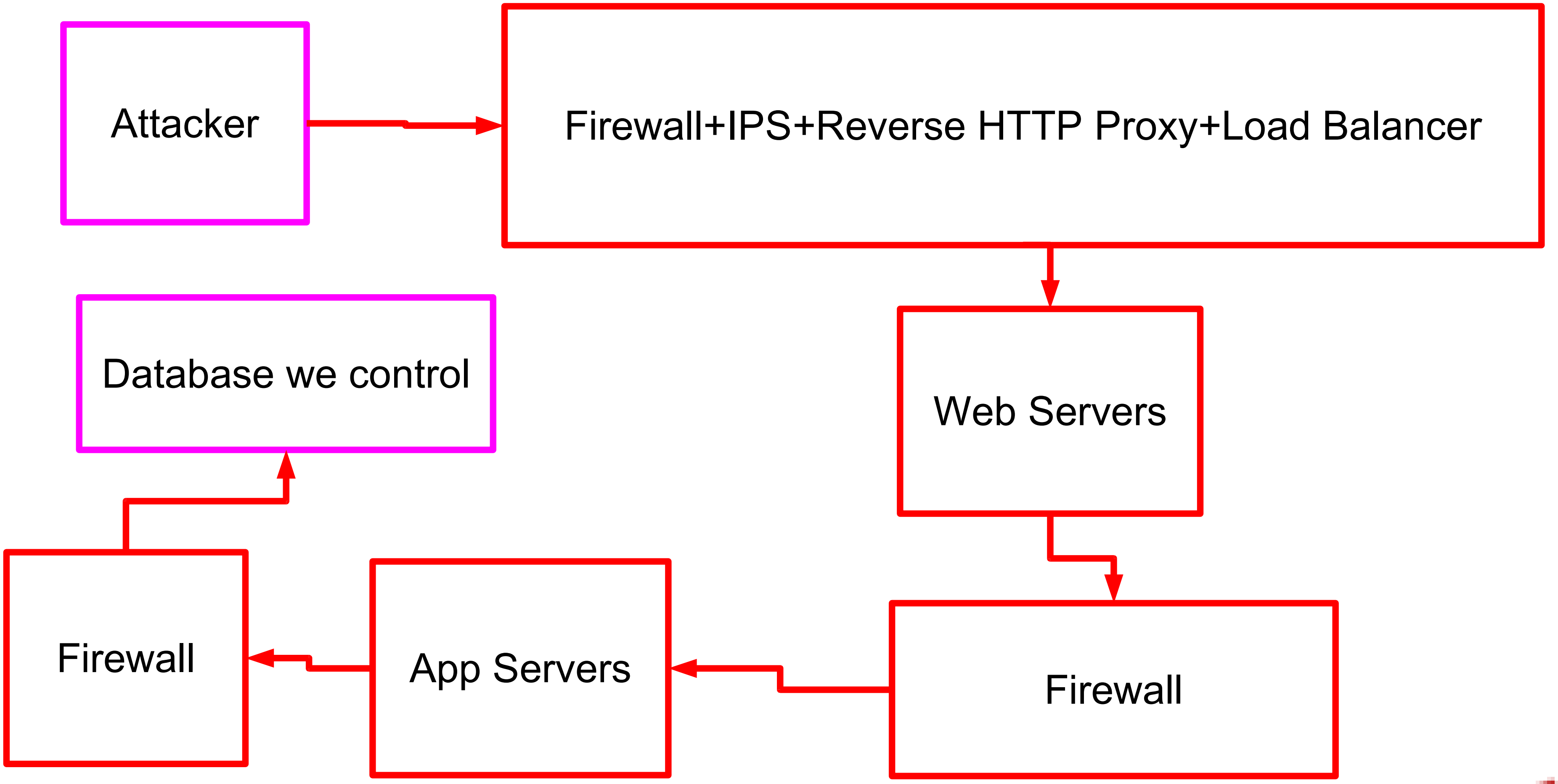
- ~~A Real to Life Scenario~~
- ~~Problems of scale when hacking~~
- ~~Immunity's PINK Framework~~
- Trojaning hard targets
  - Immunity Debugger Parasitic Infection

IMMUNITY

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



Custom automatic backdoors

- Use Immunity Debugger to analyze target .exe/.dll
- Send traffic to it and trace where our triggers are seen
- Create custom backdoor .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
- Trojan in memory or on disk or both

IMMUNITY