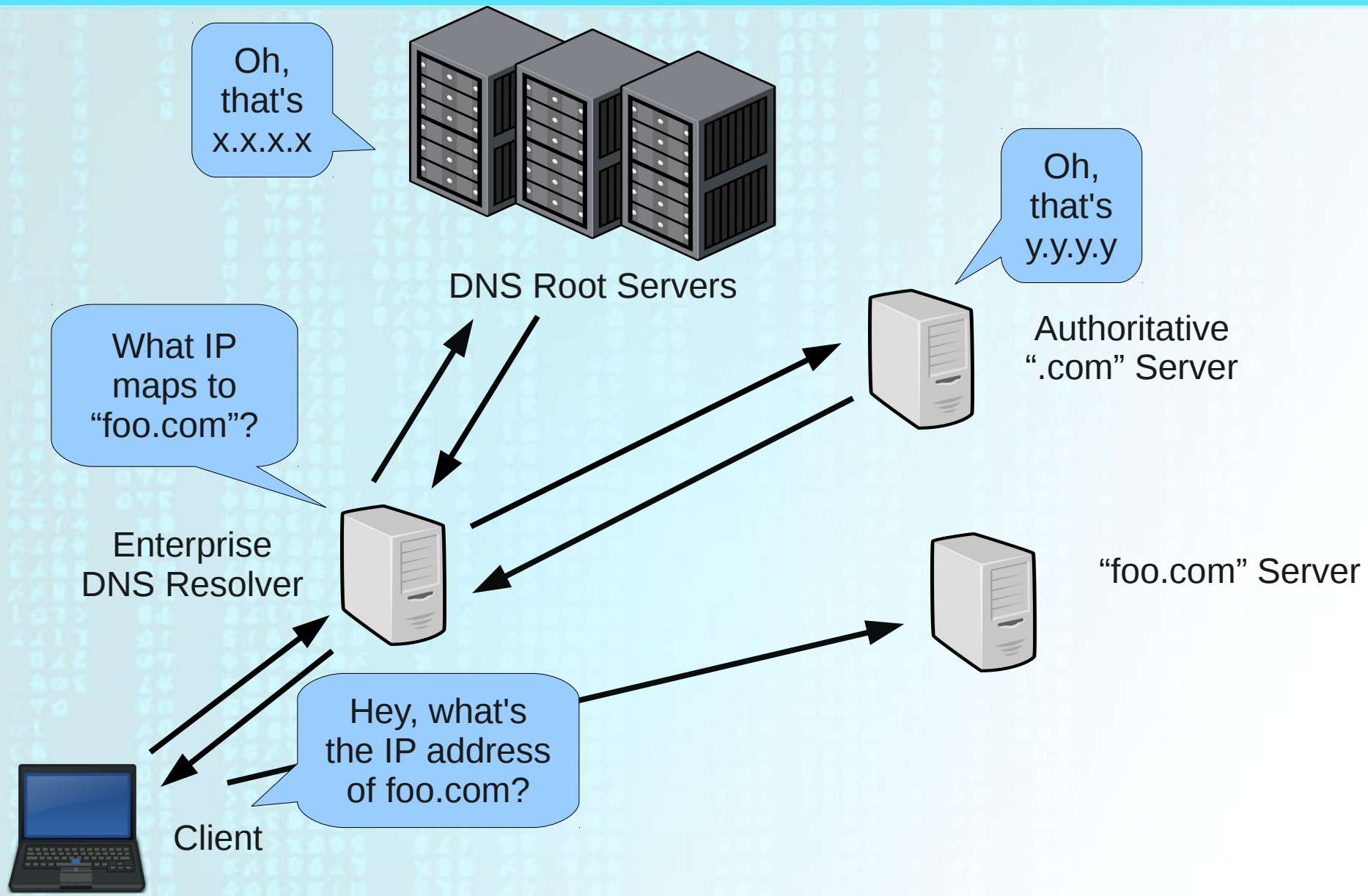
Hacking Your DNS Infrastructure for Security and Counter Intelligence

Scott Janik, CISSP, GREM
Cyber Security Engineer

A Gross Oversimplification of How Traditional DNS Works



Intrusion Kill Chain

1. Reconnaissance	Research, identify and select targets.
2. Weaponization	Stage the payload. (RAT + exploit)
3. Delivery	Transmit the weapon to the target.
4. Exploitation	Trigger intruder's code.
5. Installation	Install a RAT, maintain persistence.
6. C2	Beacon to a Command and Control (C2) server.
7. Actions on Objectives	Achieve original objectives.

E. M. Hutchins, M. J. Cloppert, and R. M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, March 2011. URL `http://papers.rohanamin.com/wp-content/uploads/papers.rohanamin.com/2011/08/iciw2011.pdf`

Courses of Action Matrix

Phase	Detect	Deny	Disrupt	Degrade	Deceive
Reconnaissance	Web analytics	Firewall ACL			
Weaponization	NIDS	NIPS			
Delivery	Vigilant user	Proxy filter	In-line AV	Queing	**DNS redirect**
Exploitation	HIDS	Patch	DEP		
Installation	HIDS	"chroot" jail	AV		
C2	NIDS	Firewall ACL	NIPS	Tarpit	**DNS redirect**
Actions on Objectives	Audit log			Quality of Service	**DNS redirect**

E. M. Hutchins, M. J. Cloppert, and R. M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, March 2011. URL `http://papers.rohanamin.com/wp-content/uploads/papers.rohanamin.com/2011/08/iciw2011.pdf`

DNS Is Commonly Used… for Evil

- **Malware** utilizes DNS:
  - Trojan.Win32.Alureon
  - Trojan-Downloader.Win32.FraudLoad
  - Trojan.Win32.FakeAlert
  - Blackhole Exploit Kit
  - many others...
- **Botnets** utilize DNS
  - DNSChanger
  - Kraken
  - Aurora
  - Fast-flux networks and many others...
- **APT** actors utilize DNS

DNS Blacklist Defense

“Allowing everything except badguy.com.”

Blacklisting is a commonly used method for preventing communication with undesirable domains.

- Spamhaus
- MalwareDomainList
- Bleeding Snort DNS Blackhole
- and many others

DNS Blacklist Defense

“Allowing everything except badguy.com.”

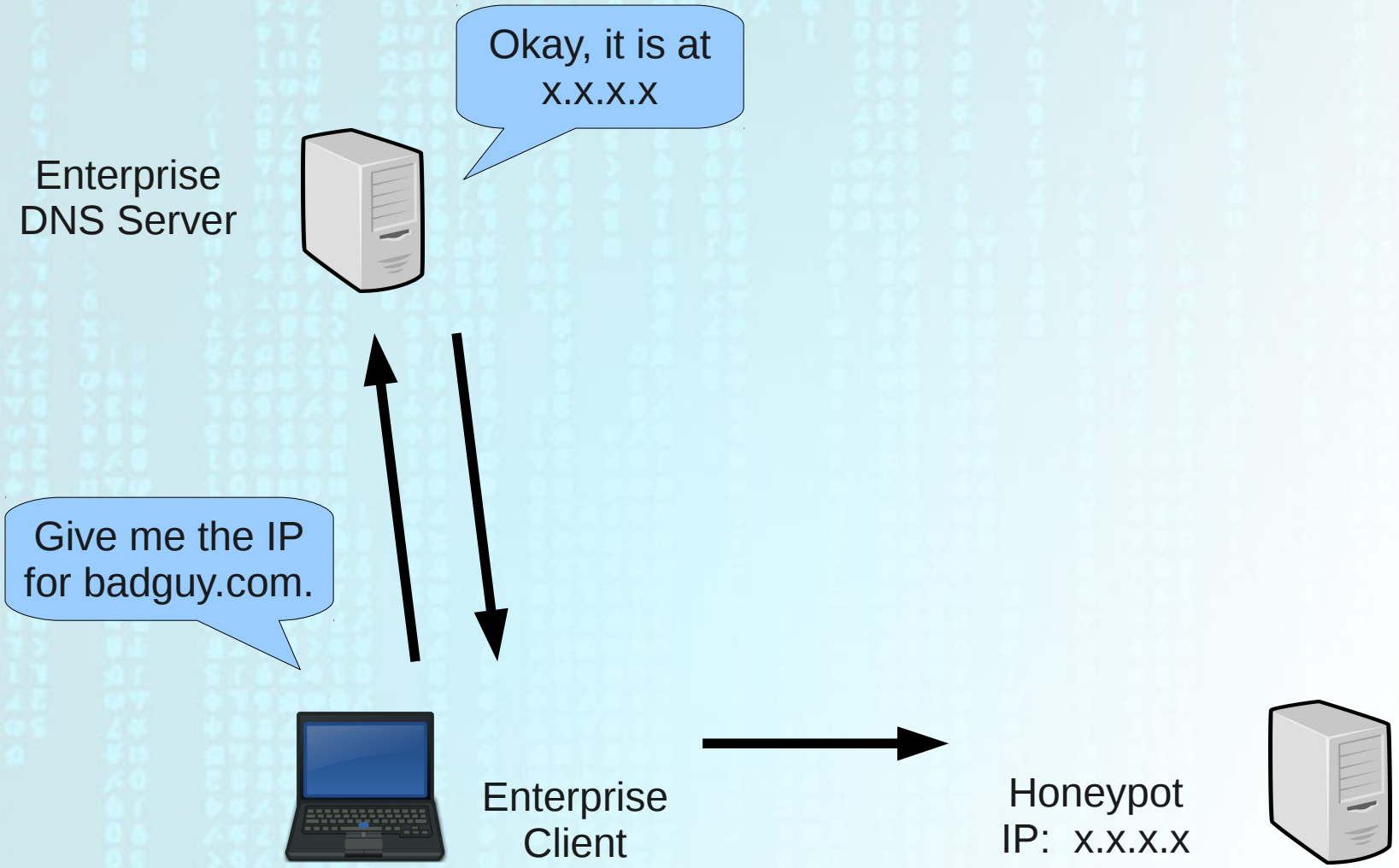
Pros:

- Response can be very customized to the threat.
- Compared to whitelisting, it's profoundly more easy to administer.

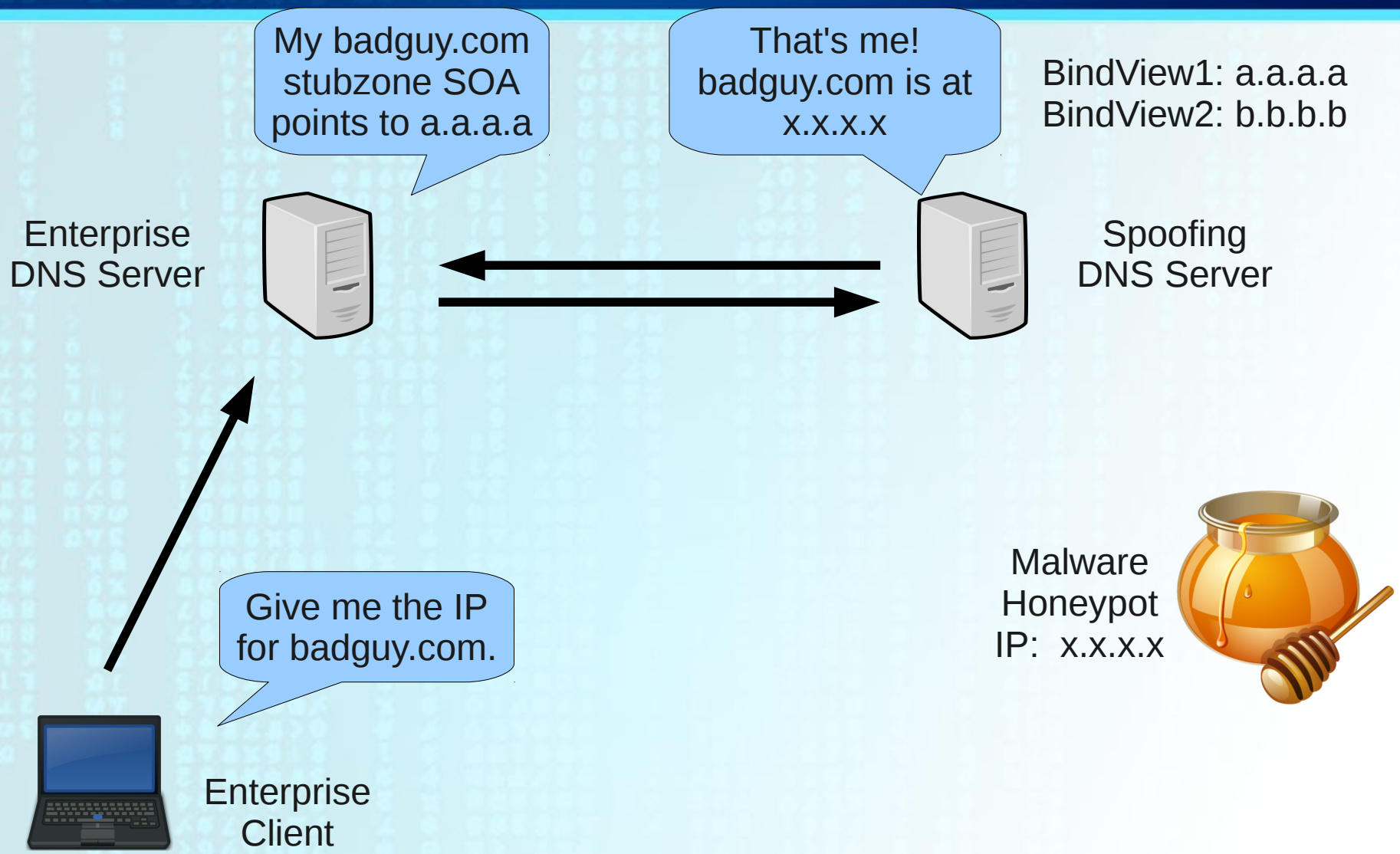
Cons:

- Overly broad (i.e., You don't know what you're missing.)
- Reactive
- Easily bypassed on subsequent attempts

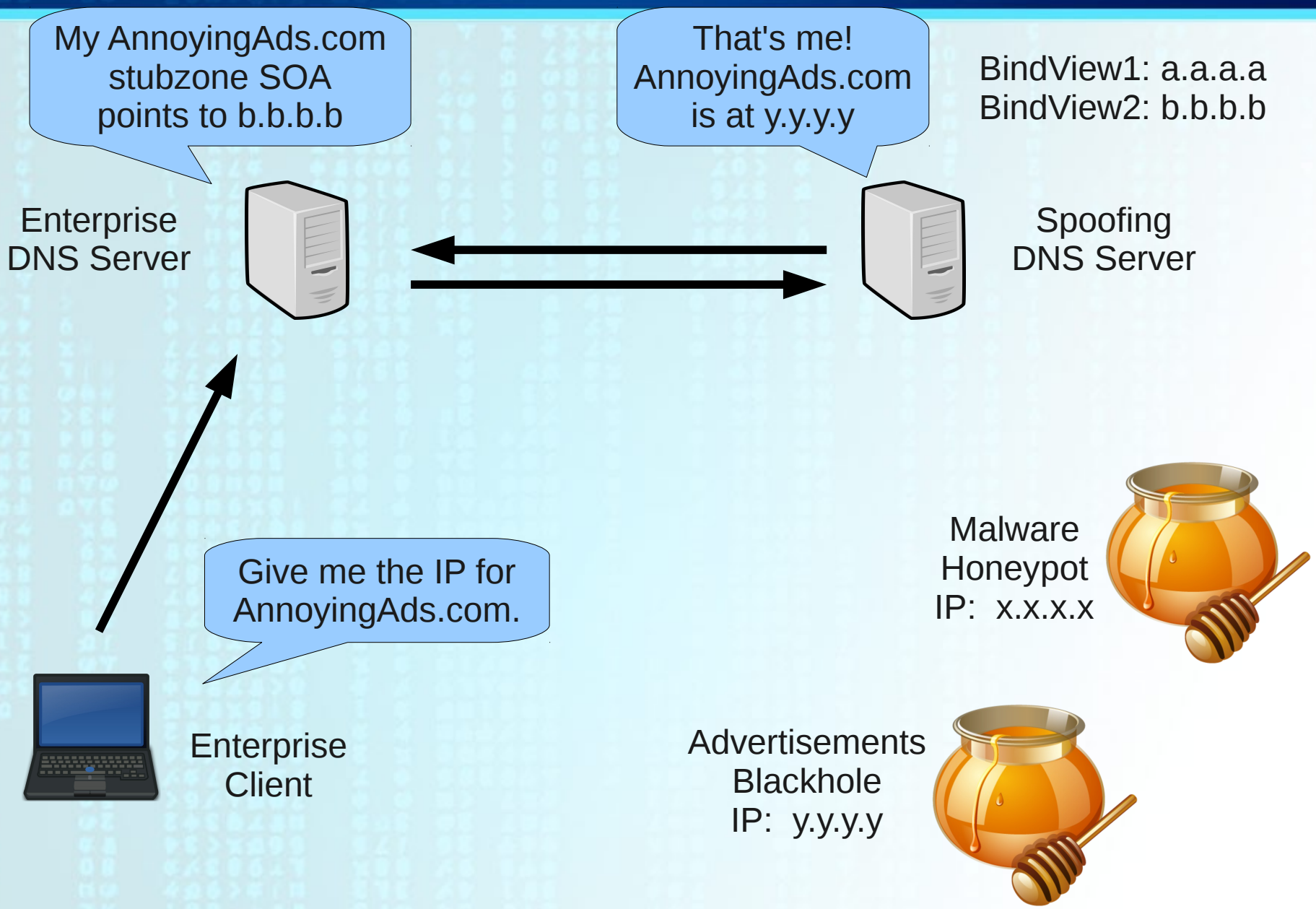
Traditional DNS Blacklisting



DNS Blacklisting



DNS Blacklisting



Demo

Blacklisting in BIND 9 using BIND views

DNS Whitelist Defense

“Blocking everything except goodguy.com.”

Pros:

- Proactive
- Generally more secure than blacklisting
- Can protect against unknown threats

Cons:

- Prevents liberal access to the largest source of information ever available to humankind
- Can be a nightmare to implement and manage


Your users
OMG! I *LOVE*
whitelist-only!

Why Not Both?

- Well, how do we split it up? There must be a sizable chunk of Internet DNS infrastructure that can be easily segregated into a whitelist-only group.
- The threat must be significant enough to justify the cost of whitelisting.
- We would need to reduce administrative effort (read “cost”) as much as possible.

Dynamic DNS

- A service used to maintain a static DNS name for a host with an IP that is not permanent.
- Useful for home users who want to remotely connect to their home network.
- A cheap alternative for organizations that don’t want to pay for managed or self-maintained DNS.

Dynamic DNS

- Very useful for attackers:
  - Bypasses firewall blocks
  - Slow response time from registrars
  - Cost-effective (free)
  - Anonymous
  - Fast

  - Most of the advanced intrusion sets I've responded to use dynamic DNS, and it is used at multiple stages in the kill chain (attack, command & control, exfiltration).

Whitelist-Only DDNS

By requiring a legitimately registered domain, we gain some security:

- Registrars (usually) require more stringent authentication.
- Some responsibility falls on the actual owner of the domain.
- We'd rather have the attackers invest (at least some) financial resource.

Taking Control of DDNS

1. Steal authority inside your organization for select DDNS providers, those that offer free and anonymous registration (like DynDNS, No IP, afraid.org, etc.) and blacklist them.

DynDNS - *.dynamicnameservices.com
(**NOT** *.dyndns.com)

Whitelisting Considerations

- Requires a method for automatically updating zone records (perhaps based on the records' TTLs).
- Automatic whitelisting subdomains of whitelisted domains.
- Maintaining Reverse DNS zones for whitelisted domains (for mail servers).
- Flushing the zones' cache on your organizations caching name servers.

Demo

2.Whitelist legitimate DDNS customer domains.

Legitimate DDNS Site	Auth Nameserver
cityofbrunswick-ga.gov	ns16.zoneedit.com
ci.manassas.va.us	ns13.zoneedit.com
cityofalcoa-tn.gov	ns1.no-ip.com
ci.gardena.ca.us	ns1.no-ip.com

DDNS Blacklist Effectiveness

- ~20% of witnessed APT C2 domains are free & anonymous DDNS registered domains.

- ~50% of witnessed APT attacks use DDNS. (The sets that we see reuse DDNS domains frequently.)

What Could Possibly Go Wrong?

- Too many requests to keep up with in large organizations
- Privacy in requesting a whitelist entry (sites related to specific gender or medical issues, family/personal resources online, etc)
- What if it all “breaks”?
- DNS setups in the real world are not always “correct” (e.g. lame responses) and may require manually administering a zone.

Cool, What's Next?

- Automate improvements:
  - Cron some daily pulls (simple text) from ZeusTracker, Malware Domain Blocklist, etc.
  - Set up user-submission web page on “honeypot” for easy whitelisting.
- Glean intelligence:
  - Set up basic services on the honeypot to receive and alert on incoming connections.
  - Further script listeners on the honeypot to instigate malware infections.

Dynamically Loadable Zones

- Allows users to store zone data in a variety of databases
- Many benefits to off-box zone administration:
  - Saves space in RAM
  - Reduces zone parsing at startup
  - No need to reload zones after changes
  - Reduces reliance on scripting (e.g., rndc). Possibility of running transaction scripts to retrieve RR resolution in near real-time?

Stichting NLnet, Netherlands. BIND DLZ, 2002. URL `http://bind-dlz.sourceforge.net/`