28th ANNUAL FIRST CONFERENCE

SEOUL

JUNE 12 - 17, 2016

GETTING TO THE SOUL OF INCIDENT RESPONSE

DIY Threat Intelligence with Real-Time Data

Dr. Paul Vixie (vixie@fsi.io)
CEO
Farsight Security, Inc.

28th ANNUAL FIRST CONFERENCE
SEOUL
JUNE 12 - 17, 2016

I. Introduction

"*Today most incident response teams rely on vendor threat feeds to gain additional intelligence about the attacks against their network.*

Yet vendor threat intelligence alone is limited – if the IOCs, signatures, or other feeds don't match what investigators have found in their network the investigation itself can come to an abrupt end."

[part of the abstract for this talk]

The "Magic" Behind Many Security Vendors' Threat Feeds

- Cybercriminals like to minimize their effort, and will **reuse an attack**, if successful, against many other potential victim sites.
- Because attacks are recycled, sharing the attack's attributes can help other victim sites **identify and respond** to these attacks.
- *Required assumption #1:* your goal is probably to do two things:
  - **Block** the malicious behavior (if possible), but at least
  - **Detect** the malicious behavior (in case efforts at blocking fail)
- *Required assumption #2:* statistically, you're unlikely to be one of the first sites hit, so you'll have **time** to learn from the experiences of others and take appropriate measures (but if you are attacked first, that attack at least provides intelligence for everyone else).
- *Required assumption #3:* **false positives/collateral damage** can be kept low through whitelisting and professional feed curation, etc.

"Abracadabra" Doesn't Always Yield A Rabbit

- ***Sometimes the magic of threat feeds simply doesn't work...***
- You might get hit by a **unique attack** meant just for you. You weren't protected from it, and no one else may ever see it.
- Sometimes there may **not be a traditional control point** at which a detected attack can be automatically mitigated (example: classic firewalls may allow all *outbound* connection attempts by default).
- You may **not have visibility** into all network traffic (example: encrypted network traffic such as PGP-encrypted email messages).
- If blocking fails, **detection is a distinctly inferior secondary outcome** ("hey, we did at least *spot* the incoming nuclear missile, even though we couldn't prevent it from blasting our city").
- **Collateral damage/false positives** MAY exist & be problematic.
- Sharing indicators can result in **intelligence being leaked** to the bad guys (disclosure of "sources and methods").

More Empty Hats

- **Attribution** often remains a huge unsolved problem, so the community largely ignores the attribution problem (or employs non-scalable manual efforts in isolated cases, such as the Mandiant China report).
- **Threat feeds are a tactical "solution"** that focuses on observable manifestations (like cough syrup for lung cancer) while what we need is a genuine **strategic solution** that focuses on correcting **root causes** (analogy: discourage smoking and other **causes** of lung cancer, rather than improve oncological treatments or suppress symptoms)
  - Cyber example: sites NOT doing SAV are still tolerated by the community, so spoofed DoS traffic remains a problem
  - Criminal sanctuary networks aren't summarily de-peered
  - Criminals may be non-extraditable from some jurisdictions

II. DIY

"That aesthetic of the Star Wars universe: **the do-it-yourself, hotrod ethic** that George Lucas exported from his childhood, is exactly the same kind of soul behind what we do and build for the show. **It may not look pretty, but it gets the job done.**"

Adam Savage, co-host of *Mythbusters* [emphasis added]

Why Consider A DIY Model? Many Reasons

- The market **doesn't have** what you need/want
- What you want is available, but you **can't afford** to buy it
- You've tried what exists, but it **isn't working well enough**
- There's something available, but what's available is **proprietary and poorly disclosed,** even under NDA (and relying on "witch doctoring" seems to be less-than-standard-of-care treatment)
- **You like layered approaches** to security (and DIY might be able to give you at least part of "another nine's worth" of incremental improvement)
- You like **crafting solutions/controlling your own destiny,** much like F/OSS for operating systems or OpenFlow/SDN networking
- **No one knows your unique environment as well as you do.**
- Also: creative "tinkerers" can potentially drive **innovation** and also potentially drive **ecosystem improvements**

Implicit Assumptions Applicable To DIY Models

- DIY can be a sweet way to save cash, but it isn't going to be totally "free." You WILL need to invest some **"sweat equity,"** instead.
- A DIY approach shouldn't be just totally *ad hoc*, it should have an articulable **theoretical basis/rational foundation**
- The approach employed must be able to be **horizontally replicated** (e.g., be generalizable to at least your friends, if not the whole Internet), and thus **cannot rely on the local existence of a willing expert (or secret heuristics) in order to succeed**
- NOT require a total (and totally impractical!) redesign of your operational environment–-**you need to be able to just "drop it in"**
- A DIY approach CANNOT require that you **"stand at the stove and stir continually"** – you've got other stuff you still have to do.
  - For example, manually adding IPv4 /32's to a local block list (as spam/phishing/malware gets locally noticed and manually reported) doesn't scale

Managing Security Exposures with DNS RPZ

- As someone who has worked with DNS a little, I think **DNS may be a promising substrate for implementing DIY security measures**
- DNS Response Policy Zones (RPZ) allow us to use DNS as a control point: **DNS RPZ can make identified unwanted domains locally return NXDOMAIN** (thereby keeping users from accidentally wandering into online minefields and experiencing traumatic cyber amputations)
- RPZs can be published/shared with other sites, but currently **there are only a relatively small number of large-scale RPZ publishers** (mostly the "usual suspects," see http://dnsrpz.info/).
- **It's wonderful to have those mass market/at scale security options, thank you all, but we need more small RPZ providers (the online equivalent of hobby farmers offering exotic fruit/heirloom vegetables at the local Saturday farmer's market).**

III. DIY Example #1:
Blocking Sources of Unwelcome Behavior By Leveraging Passive DNS and RPZ

Fool me once, shame on you;
fool me twice, shame on me.

Anonymous

Everyone Sees Attacks – But What Do You *Do* About Them?

- Everyone connected to the Internet sees attempted attacks
- Sometimes those attacks are already known to the vendors of the threat feeds you use; other times, they may not be.
- Some of you may automatically submit threat data to your threat intelligence provider, enriching those feeds and improving the protection that everyone enjoys (including yourself)
- But sometimes NOTHING gets done with that attack information. When nothing is done after an attack, a bad guy can pound on you, and **keep pounding on you** from what should now be a well-known-to-be-bad location. Permitting that is dumb.
- Other times there may be a delay between the time threat information gets shared, and the time that threat information gets incorporated into public threat feeds. It would be useful to reduce that window of vulnerability.

Leveraging Passive DNS

- Passive DNS is a well-known approach among threat analysts. Normally a threat analyst will take an initial "clue" (such as a suspicious IP, suspicious domain, or suspicious DNS server) and use passive DNS to find additional related bits of badness.
- This same process can also be leveraged for the development of domain lists to be blocked via a "DNS firewall" implemented with RPZ, complementing and extending IP-based blocking.
- For example, from a recent syslog file on an employee system:
  May 3 11:34:10 [snip] sshd: refused connect from 118.175.5.100
  May 3 11:59:12 [snip] sshd: refused connect from 118.175.5.100
  [etc]
- Those attempts *are* getting automatically blocked, but being a "belt and suspenders" sort of person, what else might we block?
- Let's check passive DNS...

Simple Passive DNS for 118.175.5.100

```
$ dnsdb_query.py -i 118.175.5.100 --after=30d
makarak.com. IN A 118.175.5.100
www.makarak.com. IN A 118.175.5.100
[no other domains seen in the last month]
$ whois makarak.com
[...]
Registrant Name: makarak
Registrant Organization: makarak
Registrant Street: makarak
Registrant City: makarak
Registrant State/Province: Krung Thep Maha Nakhon
Bangkok
Registrant Postal Code: 99999
Registrant Country: TH
Registrant Phone: +999.99999999
[etc]
```

Potential Action Options

- **Do nothing** (After all, the unauthorized ssh access attempts are currently getting blocked, but doing nothing feels... incomplete).
- **Report the obviously incomplete/inaccurate whois via WDPRS** (see https://forms.icann.org/en/resources/compliance/complaints/whois/inaccuracy-form). The problematic whois information may be an innocent clerical error, a domain that's been hijacked, or something less savory. We don't know/can't say. Cleaning up the whois is a nice first step to finding out.
- **Add that domain to a locally maintained RPZ zone.**
  Why? Assume the domain moves to a new IP. If we're **blocking by IP,** once the bad guy moves, he's free to do bad stuff again (at least until he gets relisted). If we **block by domain name**, the bad guy's attempt to avoid blocklisting by moving to a new IP address will accomplish precisely nothing – he'll still be blocked.

"Hold On. What's RPZ?"

- RPZ == DNS Response Policy Zones, see https://dnsrpz.info/ RPZ is supported by current versions of multiple name server software products.

- RPZ allows a local site to intentionally rewrite/override how a domain would normally resolve.

- For instance, if you don't want to allow your local users to accidentally access example.com, you can make your DNS return NXDOMAIN for that domain, redirect to a captive web portal, etc.

- This allows DNS to be used as a "firewall" of sorts, protecting all applications that might otherwise try to access a bad domain.

"But Vixie! I Don't Want to Chase Dotted Quads!"

- Okay. You can still leverage the power of passive DNS and RPZ.
- For instance: take the list of CIDRs on the Spamhaus DROP and EDROP lists (www.spamhaus.org/drop) as input to passive DNS, checking to see what domains are used in those 868 CIDRs...
- Those lists currently expand via passive DNS to 200,680 unique hostnames seen within the past 30 days, or, if we simplify that list by running it against the effective TLD list, we can find 65,459 unique domains (43,742 of those are from the com TLD, FWIW)
- Domain names seen include domain names with:
  -- randomly-generated-appearing components (DGA's?)
  -- domains associated with the online sale of RX drugs
  -- brands heavily targeted for infringement (Nike, Oakley, etc)
  -- brands heavily targeted by phishers (Paypal, etc.)
  -- "antivirus"-related domains

IV. DIY Example #2: "Cheap Public Suffixes" RPZ Zone

Cheap things are not good, good things are not cheap.

Chinese Proverb

Hypothetical: "Cheap Public Suffixes" RPZ

- Miscreants need a continual stream of **new domains** because current ones get blocklisted as soon as they begin to be used.
- Miscreants use free domains (or subdomains), or buy the **cheapest domains** they can find (that aren't widely block listed).
- Typical end users largely (but not exclusively) buy domains in **traditional gTLDs** or a relatively **small set of ccTLDs.**
- Price isn't critical for most users with just a few domains.
- **HYPOTHETICALLY, some cheap public suffixes may be a disproportionate source of unwanted traffic (and, conversely, NOT a material source of legit traffic)**
- A site might thus construct a DIY "threat feed" that blocks traffic from cheap public suffixes via RPZ (prices change relatively slowly, and new public suffixes are uncommon, so maintaining such a zone shouldn't be very painful).

Wait, Wait: What's a Public Suffix Again?

- Quoting https://publicsuffix.org/

  A "public suffix" is one under which Internet users can (or historically could) directly register names. Some examples of public suffixes are .com, .co.uk and pvt.k12.ma.us. The Public Suffix List is a list of all known public suffixes.

- There are just under 8,000 public suffixes at this time. Many of them you will never see, much less see heavily abused. Some public suffixes you may ONLY see in conjunction with abuse.

- If you're running an enterprise network (rather than an ISP), you might decide that there are some public suffixes that you can "live without."

Blocking Entire Public Suffixes: A "Nuclear" Option That Apparently Does Nonetheless Get Used

- **Blocking entire public suffixes is a potentially hugely problematic practice,** and **will** likely cause collateral damage. Thus, this is something that we really hope would normally not be necessary. We'd hope that those responsible for public suffixes would curb the worst abuses associated with their part of the namespace.
- Therefore, **normally at least one dot is required in an RPZ filter rule** (e.g., by default RPZ expects you to be filtering foo.bar, not just a TLD such as *.bar). However, this default **can** be changed.
- We know (from first hand reports) that some (typically enterprise-ish) sites DO currently block access to some entire public suffixes.
- Commercial managed DNS services (such as OpenDNS Umbrella), do offer this – see for example https://support.opendns.com/entries/26514730-Web-Content-Filtering-and-Security.

Which Public Suffixes Are Currently Least Expensive?

- There are sites that track at least part of this: https://tld-list.com/
- If we operationalize "inexpensive" Public Suffixes as those that are available for <= $1/domain, at the time this was prepared, TLDs known to be under that dollar per domain threshold include:

  .xyz, .top, .bid, .science, .loan, .racing, .win, .faith, .review, .trade, .date, .webcam, .party, .download, .accountant, .cricket, .pw, .press, .website, .site, .tech, .space, .online, .club, and .in
- That list would also includes .info, .com, and .us (at least right now), but we should *probably* exclude those legacy TLDs due to collateral damage considerations.
- There are other TLDs in that list that also appear to be dealing with the abuse issues they face, such as .site and .in, and which therefore might also be candidates for exclusion.
- What you do/don't block is up to you: **your network, your rules.**

"What If All The Listed Suffixes Just Raised Their Price To $1.01 or $2 or [fill in the numbere here]?"

- *Answer #1:* This would be good: criminal costs just increased.
- *Answer #2:* If necessary, the listing threshold could obviously be floated up, particularly if there were indications that pricing was being set to "game" a protective zone of this sort.
- *Answer #3: E*ventually we'd expect that most suffixes would increase in price until eventually they'd be on par with normal/non-sale dot com domain pricing (this is a decision for the entity controlling each public suffix).
- *Overall Answer:* RPZ can be used as a way for sites to deal with a particular **category** of domains (such as the current lower tail of the public suffix cost distribution), regardless of what exact "cut point" might happen to be.

"What About All Those Already-Registered Domains in Cheap Public Suffixes?"

- Traditional per-domain-based blocklisting can deal with legacy already-registered domain inventory.
- Most cheap domains are only registered for a year, and, at renewal, new pricing would typically apply.
- The crucial point for this hypothetical model is denying cyber-criminals a cheap and reliable supply of newly-created domains.
- Aside: this is the same problem Farsight already directly attacks with our Newly Observed Domain (NOD) RPZs, but this puts pressure on a different dimension of the problem.

V. DIY Example #3:
Bayesian Registrar Scoring

"He that walketh with the wise, shall be wise:
a friend of fools shall become like to them."

Proverbs 13:20,
Douay-Rheims 1899 American Edition

Another Hypothetical Example: Bayesian Filtering of Bad Guy-Preferred Registrars

- Each domain has an associated registrar. Some registrars are favorites of the Fortune 500. A second category of registrar might specialize in handling high volume domainer registrations. Other registrars specialize in providing domains for cyber criminals.
- Let's assume that there are some registrars loved by the bad guys and little used by legitimate domain registrants.
- Now imagine a publicly available DNS zone that maps domain names to registrars (much as the University of Oregon's Routeviews Project offers DNS zones mapping IP addresses to ASNs).
- The registrar data needed for such a zone is currently available from domain name registry Whois (no need to do recursion to the registrar's Whois data).

Example of Domain Name Registry Whois

Domain Name: FARSIGHTSECURITY.COM

Registrar: GANDI SAS

Sponsoring Registrar IANA ID: 81

Whois Server: whois.gandi.net

Referral URL: http://www.gandi.net

Name Server: NS5.DNSMADEEASY.COM

Name Server: NS6.DNSMADEEASY.COM

Name Server: NS7.DNSMADEEASY.COM

Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited

Updated Date: 14-dec-2015

Creation Date: 24-jan-2013

[etc]

How We Might Use f(domain)→registrar ?

- That function could hypothetically be used in email to **map spamvertised URL domains to the registrar used,** and then let Bayes classifiers do their thing with that additional token.

- E.G., like this, but for registrars rather than ASNs

  https://spamassassin.apache.org/full/3.2.x/doc/Mail_SpamAssassin_Plugin_ASN.txt

- Anyone interested in this sort of zone?

VI. DIY Example #4: Default Deny for DNS?

"I always have issues with trust."

Vin Diesel, American Popular Actor

"Risk Management"

- We used to all be philosophical purists: we'd do the right thing, for the right reason, because it was the right thing to do.
- Now "everyone" (well, a lot of people) have become pragmatists.
  - They do what seems to help right now (what's that about long term?)
  - We do what "pencils out," cost/benefit wise
  - We may only do what compliance requirements say we must do.
- This often means giving up historically-enjoyed "luxuries:"
  - Trust-by-default, Convenience, Privacy, Being A Good Network Neighbor
  - Etc.
- Example: because it is so hard to tell friends from enemies, assume everyone is hostile unless proven otherwise
- Network/system version of this: "default deny" policies

System/Network Examples of "Default Deny" Today

- $ umask 077
- Email addresses are not shared by default (try to find a publicly available email directory for an institution other than a university!)
- Social media pages are increasingly private by default (e.g., mashable.com/2014/05/22/facebook-private-default-setting/)
- Apps/executables are all untrusted by default, except for those that have been heavily scrutinized and whitelisted.
- All ports are blocked inbound at the border firewall, except for specifically allowed exceptions.
- This is all generally accepted as an example of people being "network savvy" or "streetwise online."
- **The big exception? DNS. DNS is the last "hippie protocol." DNS remains idealistic/"free love"/"default permit." (Of course, that means DNS also tends to work pretty well by default)**

FWIW, "DNS Deny By Default" Would Not Mean Just Blocking End User Access to Arbitrary Resolvers...

- Forcing users to use a specified recursive resolver (normally their ISP's recursive resolver or their company's recursive resolver) has become pretty common since DNS Changer and similar threats. See for example "Messaging Anti-Abuse Working Group (MAAWG) Overview of DNS Security - Port 53 Protection," https://www.m3aawg.org/sites/default/files/maawg_dns_port_53 v1.0_2010-06.pdf
- That document's full of great recommendations, but it doesn't go as far as calling for a full "Deny by Default" model for DNS.
- Today we're actually talking about forcing use of a specified recursive resolver **AND controlling the resolution (domain by domain) that does (or doesn't) take place on that resolver, changing from default permit (resolve anything) to default deny (only resolve the domains that are locally necessary).**

A Conceptual Model For "Default Deny" via RPZ

- Conceptually, rather than a default permit ("resolve everything by default, except for the following bad things we'll edit out") model, a default deny approach might redirect users to a web "portal" where they could request permission to access a new, never-before-requested domain. Having requested and received permission for that domain, the domain would then resolve, and continue to resolve unless/until revoked by the site.
- As part of adding a requested domain, a site might automatically check the domain characteristics, or review its reputation at sites such as WOT.
- Permission could even be granted semi-automatically (ask for permission, maybe complete a simple Captcha, then you're GTG).
- Permitted domains can also be reviewed in real time by a site's security team, or audited retrospectively (including reviewing who requested what domains).

VII. Conclusion

"It's a great country: you can say whatever you like so long as it is strictly true—nobody will ever take you seriously."

Edward Abbey, *Desert Solitaire*

Key Takeaways

- Do-it-yourself can make sense as a strategy for leveraging threat intelligence without having to rely on traditional vendor threat feeds.
- Passive DNS and DNS Response Policy Zones can be powerful tools in your DIY threat intelligence toolbox, complementing and supplementing other tools you may already be using.
- We've considered multiple examples of how this might be done:
  1) Leveraging Passive DNS with RPZ
  2) A "Cheap Public Suffixes" RPZ
  3) Bayesian Registrar Scoring
  4) Moving to "Default Deny" for DNS
- We hope you experiment a little with these approaches, and share what works for you.
- Thank you! Are there any questions?