Have I Seen You Before?

Using Splunk to Find Previously Unobserved HTTP and Email Traffic On Your Network

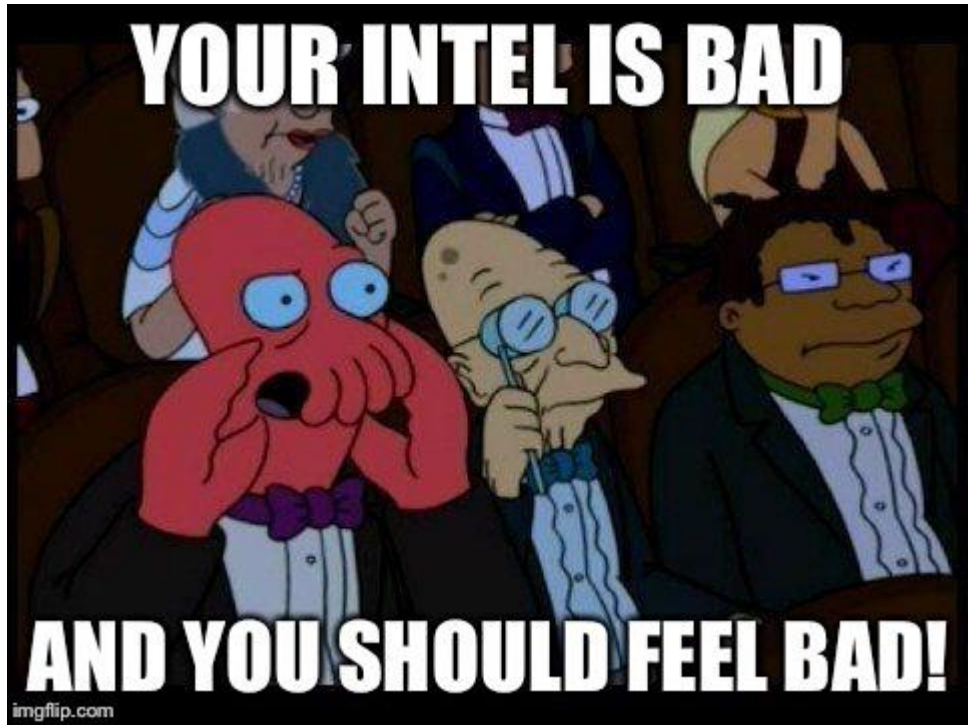
John Lankau

PUNCH Cyber Security Analytics

Who Is This Man And Why Is He Talking To Me?

- John Lankau
  - Professional Keyboard Masher
  - John[at]punchcyber.com
  - Can Google Any Question You Ask Him

Hunting Without Intel

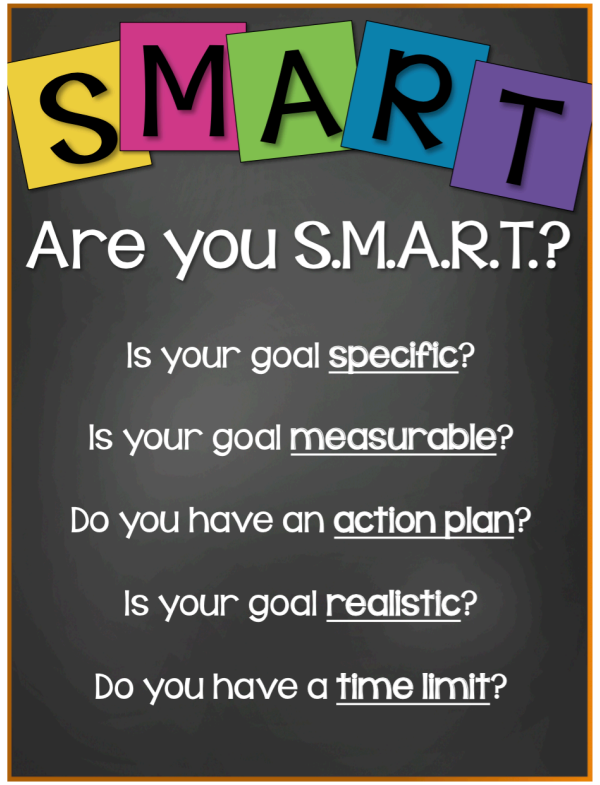


What Are We Doing Here?

- Local Adaptation of the Concept of “Newly Observed Domains”
  - *A Nod to N.O.D.*
- ”New” Domains are Higher Risk
  - Most malicious infrastructure sites are active for less than 24 hours; so they will often be “new”
- Let’s Filter to Find Traffic that is New on Our Network
  - Treat **ALL** the stuff we’ve seen before as a whitelist
- **Assumption**:
  - If we’re not already pwned; malicious things will be new at some point
  - *New stuff isn’t always bad, but bad stuff is usually new*
  - Not the ONLY form of detection; but compliments other methods nicely

S.M.A.R.T. Goals

- Query Goal:
  - Show Me All of The Things I haven't Seen Before
- How to Do?
  - Create a list of all "not-new"
    - Treat this as a whitelist
    - Show me what's left
- What Data Sets Shall We Look At?
  - Email Senders
  - DNS queries



To Do:

- Create the initial whitelists
- Schedule job to update the whitelists daily
- Create a query to find “new” stuff by excluding list matches
- Schedule those queries
- Begin winning.

Problem – Weekends!

- ”Whitelist” gets updated every morning
- Viewing the query on Monday will be using whitelist from Sunday’s traffic
  - Stuff that happened on Saturday & Sunday will already be whitelisted!
  - *You had to be there, man*
- Search results are unique to a “point in time”
  - Relative to the day the query is being run
  - The results will be lost once the Lookup Table is updated

Solution – Schedule A Summary Index!

- Schedule the search to run throughout the day before whitelist is updated
- Record all matches in a summary index
- Records the search results at the ***point in time*** before they are whitelisted
- We maintain a record of the **first** time we first saw something
  - Great for detection
  - Bonus points for forensic value!
- Doesn't count against your precious, precious Splunk license
- **Scheduled to run query every 5 minutes with a 15 minute delay**
  - This gives Splunk time to ingest and index the logs

Ingredients

- Splunk
  - *Featuring the URL Toolkit Splunk App!*
- Bro IDS
  - DNS
  - HTTP
- Email Logs
  - Email Parser (i.e. StoQ)


Lets Get Splunky!

Anatomy of a Splunk Search

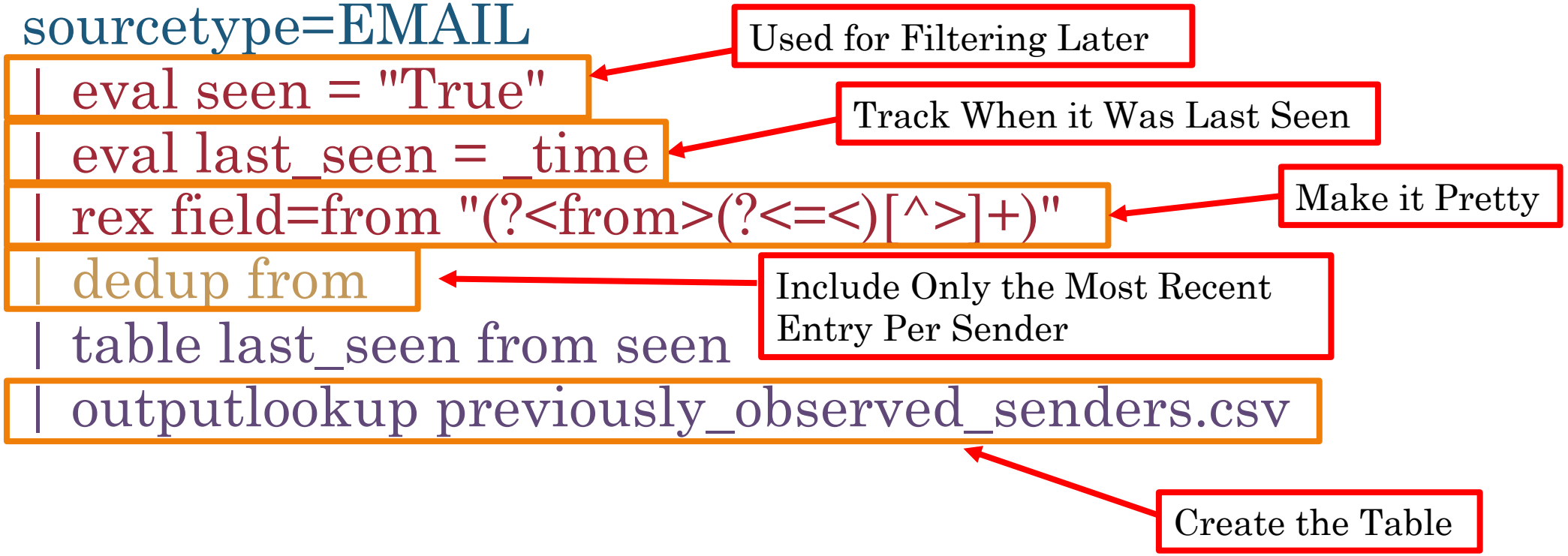
- Search & Filter | Munge | Report | Clean-Up

- *From here: http://www.slideshare.net/Splunk/splunklive-data-models-101*

Create the Initial Lookup Tables

- Email Senders
  - Query to create a lookup table (Run this once over the last 270 days)

```
sourcetype=EMAIL
| eval seen = "True"
| eval last_seen = _time
| rex field=from "(?<from>(?<=<)[^>]+)"
| dedup from
| table last_seen from seen
| outputlookup previously_observed_senders.csv
```



Create the Initial Lookup Tables

- DNS Queries
  - Query to crate a lookup table (Run this once over the last 60 days)

```
index=DNS
| eval seen = "True"
| eval last_seen = _time
| eval list = ”Mozilla" | `ut_parse(query,list)`
| dedup ut_domain
| fields ut_domain seen
| table last_seen ut_domain seen
| outputlookup previously_observed_domains.csv
```

Syntax for URL Toolbox Parsing

Output of URL Toolbox

Schedule Jobs to Update the Tables

- Recurring schedule to update list daily (Run for a 24 hour timeframe each morning)

```
sourcetype=EMAIL
| eval seen = "True"
| eval last_seen = _time
| rex field=from "(?<from>(?<=<)[^>]+)”
| dedup from | table last_seen from seen
| inputlookup previously_observed_senders.csv  append=true
| where last_seen >= now( )-(60*60*24*270)
| dedup from
| outputlookup previously_observed_senders.csv
```

Update & Clean-Up the Lookup Table

Schedule Jobs to Update the Tables

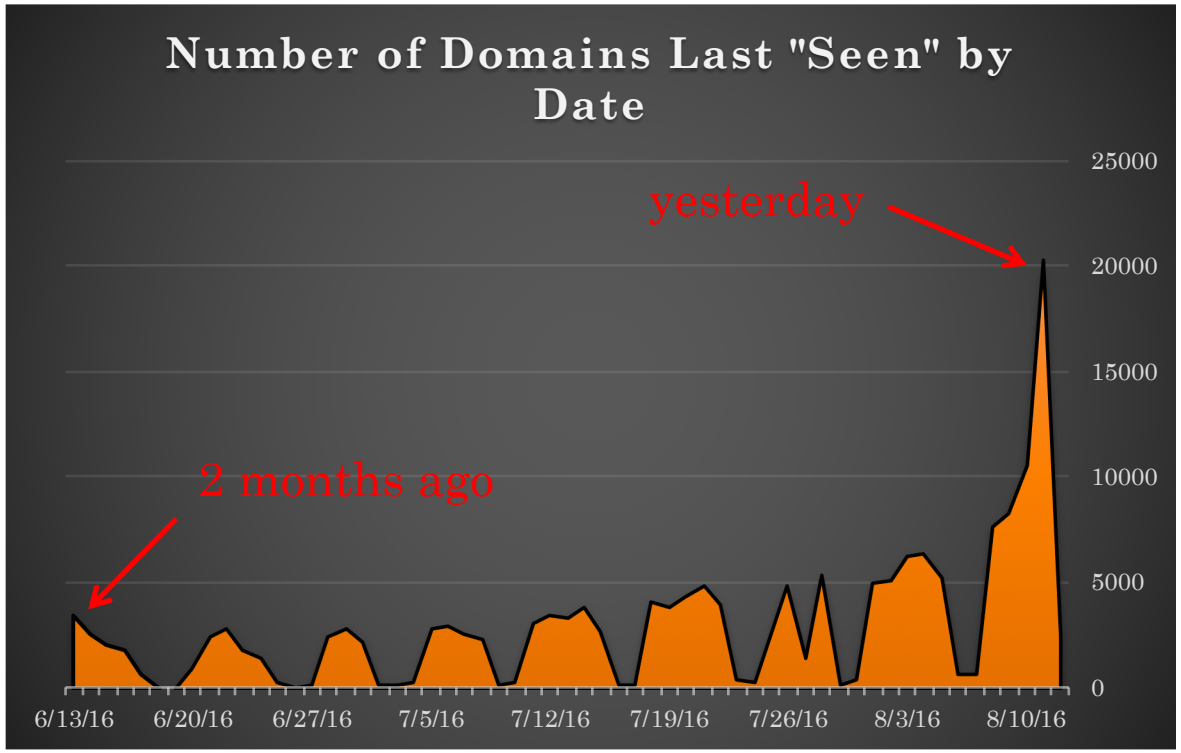
- Recurring schedule to update list daily (Run for a 24 hour timeframe each morning)

```
index=DNS
 | eval last_seen = _time
 | eval seen = "True"
 | eval list = ”Mozilla" | `ut_parse(query,list)`
 | dedup ut_domain
 | table last_seen ut_domain seen
 | inputlookup previously_observed_domains append=true
 | where last_seen >= now( ) - (60*60*24*60)
 | dedup ut_domain
 | outputlookup previously_observed_domains.csv
```

60 days of DNS

Mapping the “Last Seen” Date

- Last **4** business days – **25%** populated
- Last **12** business days- **50%** populated
- Last **23** business days – **75%** populated
- Last **41** business days – **100%** populated
- **After ~23 days, levels off to 1-2% of list added per day**
- **Longer List = Diminishing Returns**



What Does This Mean?

- Increasing the time range feeding the list will make it bigger
  - ***But not necessarily better***
  - ***Find the time range/size range that works for your organization***
- We had issues above 10MB Lookup Tables
  - Reduced time range from 90 days to 60 days; 365 to 270 days
    - No noticeable impact on number of results
  - ~2 weeks would be 50% the 60 day "whitelist"
  - ~ 1 month would be 75% of the 60 day "whitelist"
- Or Change the Splunk Lookup Table size limits*
  - *Default configuration – after 10MB, Splunk indexes lookup tables differently*
  - limits.conf, under the [lookup] stanza, change max_memtable_bytes to a larger number

**Source: https://answers.splunk.com/answers/8228/lookup-table-limits.html*

Hello New(man) - HTTP

- Traffic to New Domains; Filtering Out Noisy MIME Types:

```
index=HTTP_Outbound NOT (dest_ip="NOISY_IP_ADDRESSES")
 | eval list = "Mozilla" | `ut_parse(domain,list)`
 | lookup previously_observed_domains ut_domain as ut_domain
OUTPUTNEW seen as ignore
 | search NOT ignore="True"
 | search NOT (resp_mime_types=image* OR
resp_mime_types=text* OR
resp_mime_types=video* OR
resp_mime_types=audio* OR
resp_mime_types=*font* OR
resp_mime_types=*ocsp-response* OR
resp_mime_types=*xml* OR
resp_mime_types="-")
 | table _time ut_domain src_ip dest_ip resp_mime_types status_code referrer
```

IPs don't have DNS records

Ignore everything in the list

Filter out Noisy MIME Types

Hello New(man) - Email

- All Email from New Senders with Attachments:

```
index=EMAIL attachments=* NOT (from=*@DAYJOB*)
 | rex field=from "(?<from>(?<=<)[^>]+)"
 | rex field=to "(?<to>(?<=<)[^>]+)" max_match=0
 | lookup previously_observed_senders from as from
OUTPUTNEW seen as ignore_sender
 | search NOT (ignore_sender="True")
 | eval attachment_count = mvcount(attachments)
 | dedup from, to, subject, attachments
 | table _time from to subject attachments x-mailer attachment_count
```

Filter Your Org's Senders

Make Recipients List Pretty

Count the Number of Attachments per Email

Let's Get Fancy!
Combine Both Email and DNS Lists
And See What Happens!

Combined Query Powers

- Emails Containing Links to new URLS from New Senders

```
index=EMAIL urls=*
 | rex field=from "(?<from>(?<=<)[^>]+)"
 | rex field=to "(?<to>(?<=<)[^>]+)" max_match=0
 | lookup previously_observed_senders from as from OUTPUTNEW seen as ignore_sender
 | search NOT (ignore_sender="True")
 | eval list = "Mozilla" | `ut_parse(urls,list)`
 | eval ut_domain_dedup=mvdedup(ut_domain)
 | lookup previously_observed_domains ut_domain as ut_domain OUTPUTNEW seen as ignore_url
 | search NOT ignore_url="True"
 | eval url_count = mvcount(urls)
 | table _time from to subject ut_domain_dedup urls url_count
```

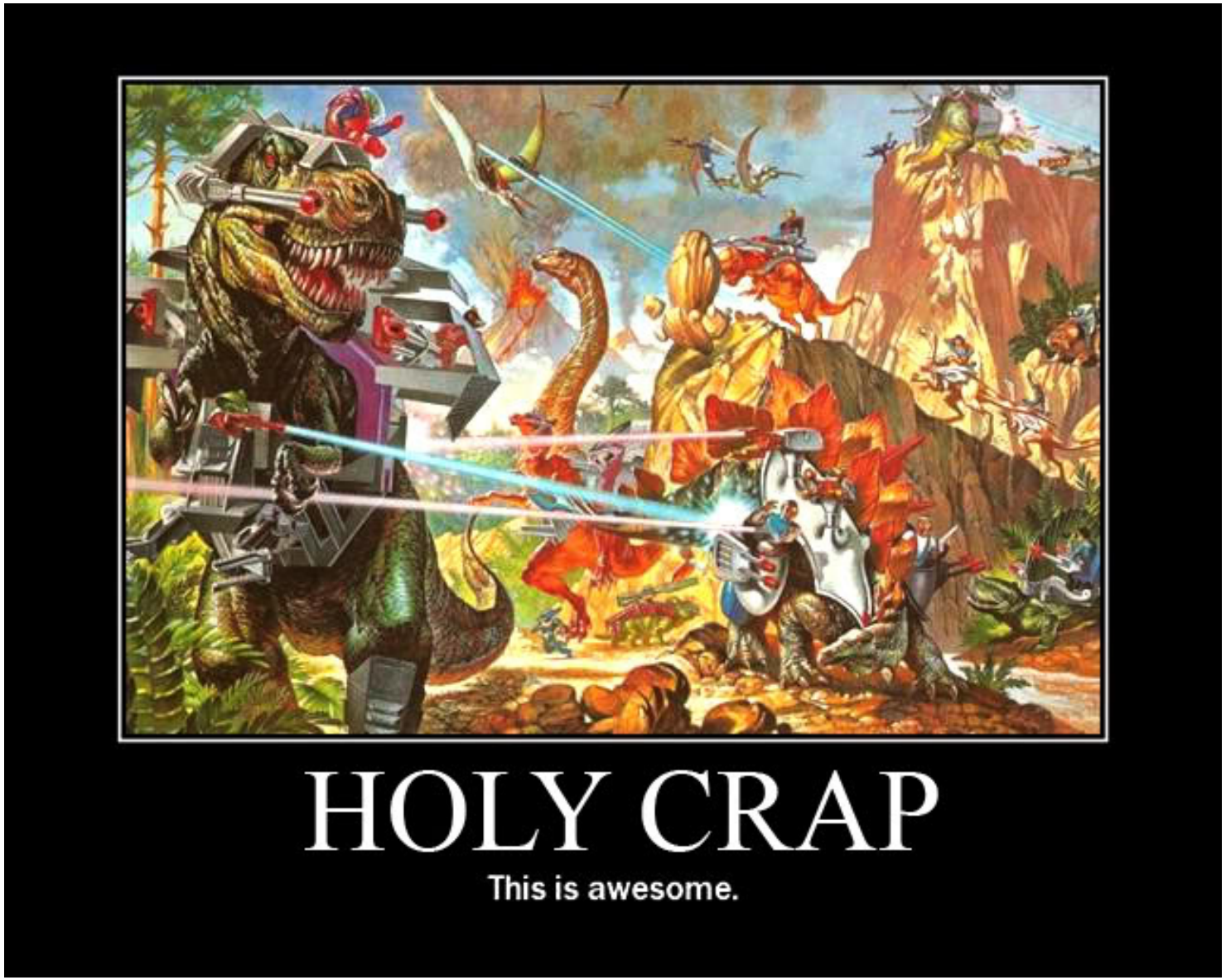
Filter Out Known Senders

Filter Out Known Domains

Detects Stranger Danger!

- Emails from New Senders Containing New URLs
  - ***Behavior – A sender I have never seen before has sent me a link to a domain I have never seen before***
    - *Sounds….hinky*

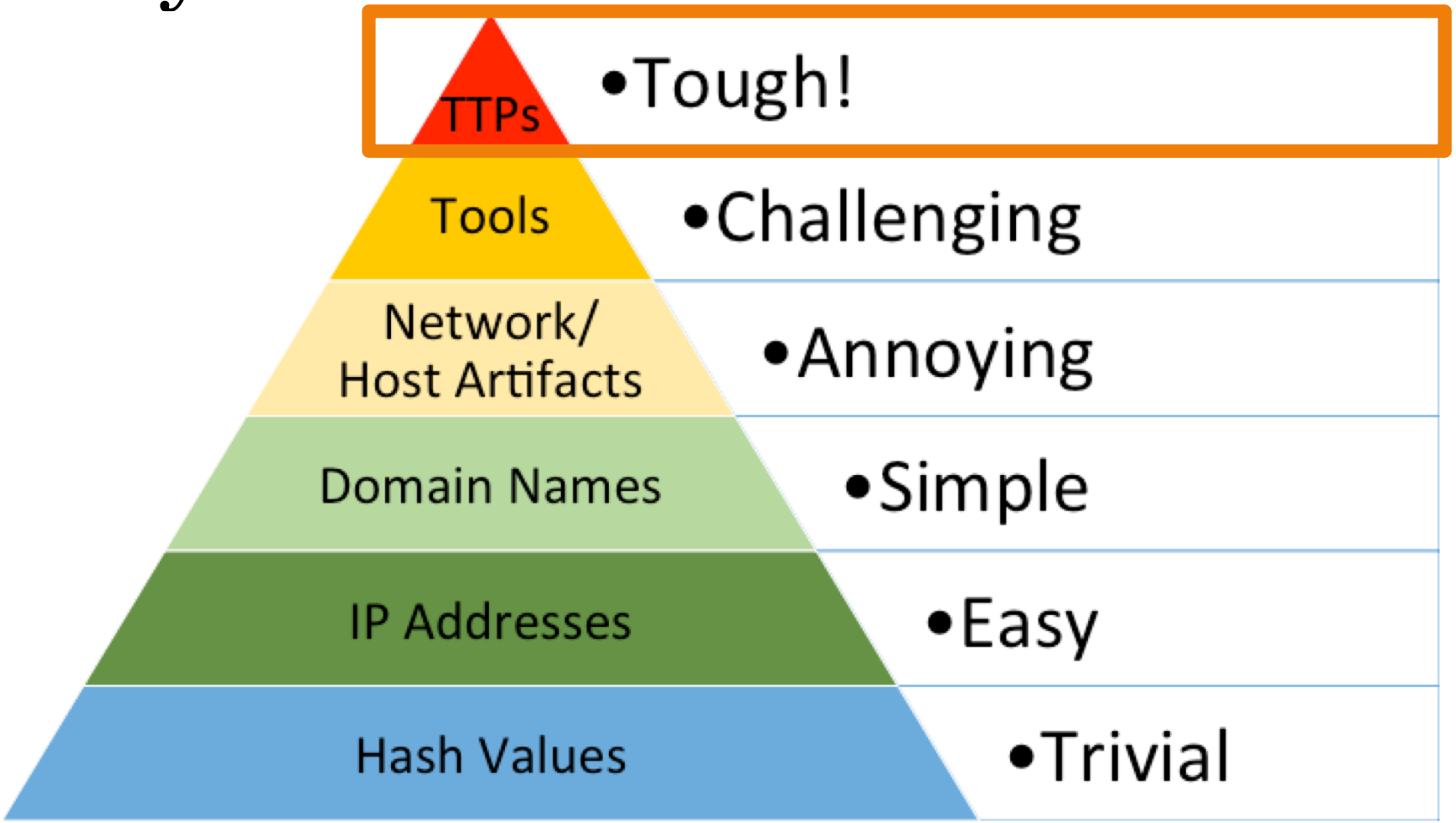
- High percentage of bad things we should probably be aware of
  - Spam; Viruses; Phishing; Suspicious Stuff
  - Can do before or after Spam/AV filtering based on approach
  - *Great place to start hunting*


HOLY CRAP
This is awesome.

So, What *Else* Is New?

- Emails with New Domains that contain a link to .php
- Emails from New Senders that contain a link shortened service
- HTTP Traffic to a New Domain with POST/PUT/HEAD Traffic
- More Data Sources:
  - X509 Certificates!
  - X-Mailers!
  - User-Agents!
  - VPN Login IP Addresses!
  - Recipient Email Addresses Lists where sender=@DAYJOB.gov

Pyramid of Pain



Source: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Summary

- Looking at “new” stuff on your network can result in some interesting datasets
- Self-tuning whitelist – set it and forget it
- Can give insight into strange traffic that requires further investigation
- Can be used to add additional filtering to otherwise noisy and unmanageable detection
  - .php links
  - POST requests
- Detection is based on adversary TTPs that are difficult to change

Questions?

John[at]punchcyber.com