Ch 4: Monitoring and Detecting Security Breaches

Updated 10-24-17
Monitoring

• Four useful types of data
  – Log data
  – Network flow data
  – Packet data
  – Application level metadata
Log data
Types of Log Data

• Format errors in queries
• Lame delegations
  – Referral from a parent zone to an invalid name server for the child zone
• Queries for nonexistent domains
BIND's Logging in named.conf

```conf
logging {
    [ channel channel_name {
        ( file path name
            [ versions ( number | unlimited ) ]
            [ size size spec ]
            | syslog syslog_facility
            | stderr
            | null );
        [ severity (critical | error | warning | notice | info | debug [ level ] | dynamic ); ]
        [ print-category yes or no; ]
        [ print-severity yes or no; ]
        [ print-time yes or no; ]
    } ]
    [ category category_name {
        channel_name ; [ channel_name ; ... ]
    } ]
... 
};
```
Clauses

• Channel
  – Defines output medium, such as files, syslog, stderr, or null to eliminate output

• Versions
  – Max. number of files that can be used
  – Files are rolled when "size" is reached

• Severity
  – "critical" logs only critical events
  – "info" stores much more

• Print
  – print-time, print-severity, print-category
  – Controls what is printed (link Ch 4a)
Categories

• queries
  – Logs client IP & port, question name, type and class of query
  – Useful to record which hosts are querying for what domains
  – + indicates recursive query
  – S indicates signed query
  – E indicates Extended DNS (EDNS)

Example:
12-Sep- 15:45:49:053 queries: info: client
192.168.0.100#1876: query: www.example.com IN A +SE
Categories

• security
  – Requests that were denied
  – Rejected by access control lists (ACLS) that define which hosts are allowed to send queries, zone transfers, etc.
  – ACLs are set using these **options** statements
    • allow-query
    • allow-recursion
    • allow-transfer
Categories

• update-security
  – Denied requests to update DNS zone data dynamically, because of ACLs or policies
  – ACLs and policies defined with
    • allow-update
    • allow-update-forwarding
    • update-policy
  – BIND tool "nsupdate" generates dynamic updates

20-Sep 21:21:11.499 update-security: info: client 127.0.0.1#42445: update 'ppdev.net/IN' denied
Categories

• dnssec
  – Only works if DNS server supports DNSSEC and is configured to perform record validation
  – DNSSEC statements
    • dnssec-enable
    • dnssec-validation
DNSSEC Example

• Line prefix omitted in figure below
  – Date dnssec: debug 3:

```plaintext
validating @0xb904ace8: nist.gov A: starting
validating @0xb904ace8: nist.gov A: attempting positive response validation
validating @0xb904c510: nist.gov DNSKEY: starting
validating @0xb904c510: nist.gov DNSKEY: attempting positive response validation
validating @0xb904c510: nist.gov DNSKEY: verify rdataset (keyid=41227): success
validating @0xb904c510: nist.gov DNSKEY: signed by trusted key; marking as secure
validator @0xb904c510: dns_validator_destroy
validating @0xb904ace8: nist.gov A: in fetch_callback_validator
validating @0xb904ace8: nist.gov A: keyset with trust 7
validating @0xb904ace8: nist.gov A: resuming validate
validating @0xb904ace8: nist.gov A: verify rdataset (keyid=63462): success
validating @0xb904ace8: nist.gov A: marking as secure
validator @0xb904ace8: dns_validator_destroy
```
Categories

- xfer-in
- xfer-out
  - Report zone transfers

```
25-Sep 14:12:12.179 xfer-out: info: client 127.0.0.1#38077: transfer of 'ppdev.net/IN': AXFR started
25-Sep 14:12:12.186 xfer-out: info: client 127.0.0.1#38077: transfer of 'ppdev.net/IN': AXFR ended
```
Packet Data
SPAN Port

• Capture packets with *tcpdump* or *Wireshark*
• From a SPAN port on a router or switch
  – Provides a copy of every packet
• Or use an optical or electronic splitter
  – Or a hub
• Data sent to a server that captures and stores all the packets
• Usually uses *libpcap* or *WinPcap* with standard *pcap* format
Network Flow Data
Flow Data

• Summarized record of a network traffic session
• Packets with common characteristics
  – Source and destination IP, Port, and Protocol
• Each flow typically goes in only one direction
• NetFlow
  – Originally developed by Cisco
  – Standardized by IETF as IP Flow Information Export (IPFIX)
Packet Grouping

• TCP sessions
  – Export flow as soon as session ends with FIN or RST

• UDP traffic
  – Must guess when flow ends
  – Activity timer expiration exports after a period of time, even if flow is still in progress
  – Inactivity times generates a flow record when there is inactivity for a period of time
Flow Records

- Don't contain a complete summary of a session between two hosts
- Very long sessions, or sessions with periods of inactivity, may appear in multiple flow records

Router1, 10.173.163.76, 10.246.128.147, 171, 8313, 1255112063, 1255233063, 64126, 41450, 26, 6

where the fields correspond to: Router name, Source IP address, Destination IP address, number of packets transferred, number of bytes transferred, UTC start time of flow in seconds since 1/1/1970, UTC end time, source port, destination port, cumulative TCP flags (in decimal representation), and protocol number.
Application-Level Metadata
Metadata

- Flow records provide very little information
- Packet data are overwhelming, containing too much data
  - Also raise privacy concerns
- Application layer metadata
  - Keeps some packet fields from application and other layers

Domain, A_record, first_time, last_time, number_of_responses
www.example.com, 10.20.30.40, Jan 1 2009, June 30 2010, 15288
Detection
Cache Poisoning Attack Detection

• Brute force attempts to guess Transaction ID and Source Port
  • Of a query from a recursive DNS server to an authoritative server
• First, attacker makes a request for a record that is not cached
  • Then blasts server with spoofed responses with many Transaction ID and Source Port values
Flow Records

- Keep flows with source or destination port 53 (TCP or UDP) and source or destination IP of the DNS server

Table 5: Example of a sequence of flow records indicating a possible cache poisoning attack.

<table>
<thead>
<tr>
<th>Sip</th>
<th>Dip</th>
<th>Sport</th>
<th>Dport</th>
<th>Stime</th>
<th>Etime</th>
<th>Pkts</th>
<th>Bytes</th>
<th>Proto</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.5.100</td>
<td>10.10.1.1</td>
<td>1024</td>
<td>53</td>
<td>0.000</td>
<td>0.000</td>
<td>1</td>
<td>70</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.001</td>
<td>0.001</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.002</td>
<td>0.002</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.003</td>
<td>0.003</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.004</td>
<td>0.004</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.005</td>
<td>0.005</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
<tr>
<td>192.168.0.50</td>
<td>10.10.1.1</td>
<td>53</td>
<td>1024</td>
<td>0.006</td>
<td>0.006</td>
<td>1</td>
<td>90</td>
<td>17</td>
</tr>
</tbody>
</table>
Limitations of Flow Records

- No Layer 7 data
  - Such as the DNS request
- Cannot pinpoint the domains being targeted
- Or the addresses being injected
Selecting Relevant Data

- DNS requests are irrelevant
- Poisoning is performed by replies
- Data needed
  - Source & destination IP
  - Domain name in the question section
  - Answer, authority, and additional sections
  - Transaction ID
  - Timestamp
  - Only include authoritative replies (AA set)
Transient Domains

- Resolve to a small number of IP addresses
- Change over hours or days
- IP addresses are not owned by the same autonomous system (AS)
- Typically they are botnet controllers, malware downloads, or file drop sites
- Could be an innocent software bug, or a security research site
Identifying Transient Domains

• Collect DNS traffic with
  – Small TTLs
  – Collect at peering links to other AS networks

• Record
  – Domain that was queried
  – Answer given
  – Timestamp
  – Exclude client IP address for privacy
Round-Robin DNS

• If there's more than one A record
  – The order changes for each request
    • Link Ch 4b
• This is the default for most DNS servers
• Demo:
  – `dig a microsoft.com`
  – Repeat a few times
<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Class</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>microsoft.com.</td>
<td>2488</td>
<td>IN</td>
<td>104.40.211.35</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>2488</td>
<td>IN</td>
<td>104.43.195.251</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>2488</td>
<td>IN</td>
<td>191.239.213.197</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>2488</td>
<td>IN</td>
<td>23.96.52.53</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>2488</td>
<td>IN</td>
<td>23.100.122.175</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Class</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>microsoft.com.</td>
<td>1348</td>
<td>IN</td>
<td>191.239.213.197</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1348</td>
<td>IN</td>
<td>104.40.211.35</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1348</td>
<td>IN</td>
<td>23.100.122.175</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1348</td>
<td>IN</td>
<td>23.96.52.53</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1348</td>
<td>IN</td>
<td>104.43.195.251</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Class</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>microsoft.com.</td>
<td>1346</td>
<td>IN</td>
<td>104.43.195.251</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1346</td>
<td>IN</td>
<td>191.239.213.197</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1346</td>
<td>IN</td>
<td>23.96.52.53</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1346</td>
<td>IN</td>
<td>104.40.211.35</td>
</tr>
<tr>
<td>microsoft.com.</td>
<td>1346</td>
<td>IN</td>
<td>23.100.122.175</td>
</tr>
</tbody>
</table>
Fast Fluxing Domains

• TTLs set to a few seconds
• IP changes rapidly

• Purposes
  • Evade detection
  • Resilience: maintain control of a botnet despite attempts to block malicious traffic
Example from Conficker

Answer at time 0

- www.refaourma.info. 60 IN A 65.54.40.75
- www.refaourma.info. 60 IN A 65.118.223.203
- www.refaourma.info. 60 IN A 65.130.228.46

Answer 28 sec. later

- www.refaourma.info. 32 IN A 65.130.228.46
- www.refaourma.info. 32 IN A 65.54.40.75
- www.refaourma.info. 32 IN A 65.118.223.203
Example from Conficker

Answer at 56 sec.
• www.refaourma.info. 4 IN A 65.118.223.203
• www.refaourma.info. 4 IN A 65.130.228.46
• www.refaourma.info. 4 IN A 65.54.40.75

Answer at 83 sec.
• www.refaourma.info. 32 IN A 209.17.184.203
• www.refaourma.info. 32 IN A 209.228.250.75
• www.refaourma.info. 32 IN A 209.229.142.35

• When cache expires, IP addresses are all new
## Detecting Fast-Flux Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>IP</th>
<th>Last_activity</th>
<th>IP_registration</th>
<th>Country_code</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.refaourma.info">www.refaourma.info</a></td>
<td>10.5.172.203</td>
<td>10-25@16:01:03</td>
<td>10.5.160.0</td>
<td>19</td>
</tr>
<tr>
<td><a href="http://www.refaourma.info">www.refaourma.info</a></td>
<td>20.203.221.67</td>
<td>10-25@15:01:05</td>
<td>20.203.221.64</td>
<td>27</td>
</tr>
<tr>
<td><a href="http://www.refaourma.info">www.refaourma.info</a></td>
<td>30.135.46.203</td>
<td>10-25@15:00:59</td>
<td>30.135.0.0</td>
<td>16</td>
</tr>
<tr>
<td><a href="http://www.refaourma.info">www.refaourma.info</a></td>
<td>40.54.159.203</td>
<td>10-25@14:05:12</td>
<td>40.52.0.0</td>
<td>14</td>
</tr>
</tbody>
</table>
Phantom Domains

- Register a domain
- Use it for only a few hours or days
- Defends malware against *sinkhолing*
  - Resolving to an address that offers no service
- Works best with domain registrars who offer a free trial period
Detecting Phantom Domains

- Find domains that have been active recently
- Find current addresses
- Find domains with no matching historical IP addresses
- Find records with very different IP addresses for the same domain
Wannacry Ransomware

- Caused hospitals across England to divert emergency patients in May 2017
- Used NSA-developed attacks leaked by "Shadow Brokers" (Russians)
- Microsoft released a patch but hospital systems didn't install it in time
  - Link Ch 1y
have to be propagated using another method). I was quickly able to get a sample of the malware with the help of Kafeine, a good friend and fellow researcher. Upon running the sample in my analysis environment I instantly noticed it queried an unregistered domain, which I promptly registered.

Darien Huss
@darienhuss

#WannaCry propagation payload contains previously unregistered domain, execution fails now that domain has been sinkholed

10:29 AM - May 12, 2017

65 1,458 2,186

• Link Ch 1z1
Conficker Worm Domains

- Algorithm made 50,000 new domains per day
- Registrars tried to block them all
  - Links Ch 1u, 1v

<table>
<thead>
<tr>
<th>Variant, Date, Index, Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, 02/12/2009, 0, puxqy.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 1, elvyodjjiyao.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 2, ltxbshpv.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 3, ykzjaluthux.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 4, lpiishmjljy.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 5, arpsyp.com</td>
</tr>
<tr>
<td>A, 02/12/2009, 6, txkjinjucnths.org</td>
</tr>
<tr>
<td>A, 02/12/2009, 7, vhslzulwn.org</td>
</tr>
<tr>
<td>A, 02/12/2009, 8, jcqavkkhg.net</td>
</tr>
<tr>
<td>A, 02/12/2009, 9, dmszsyfp.info</td>
</tr>
<tr>
<td>B, 02/12/2009, 0, tvxwoajflewad.info</td>
</tr>
<tr>
<td>B, 02/12/2009, 1, blojvbcbrwx.biz</td>
</tr>
<tr>
<td>B, 02/12/2009, 2, wimmugmq.biz</td>
</tr>
<tr>
<td>B, 02/12/2009, 3, fwnvylja.org</td>
</tr>
<tr>
<td>B, 02/12/2009, 4, umgrzaybbf.ws</td>
</tr>
<tr>
<td>B, 02/12/2009, 5, btgoyr.cc</td>
</tr>
<tr>
<td>B, 02/12/2009, 6, zboycplmkhc.cc</td>
</tr>
<tr>
<td>B, 02/12/2009, 7, qszqzhbn.biz</td>
</tr>
<tr>
<td>B, 02/12/2009, 8, xqdvims.cn</td>
</tr>
<tr>
<td>B, 02/12/2009, 9, wgrrrrr.biz</td>
</tr>
</tbody>
</table>
Corrupted Local DNS Server Settings (DNS Changer)

• Redirect victims to evil DNS server
• Most resolutions are correct
• Some lead to fake websites
  – Such as banking sites, antivirus sites, etc.
Detecting DNS Changers

- Recursive DNS requests to suspicious remote addresses
  - Not in ISP's address range
  - Not a known public DNS server
  - Are in an IP address blacklist
  - Associated with transient, fast-flux, phantom, sinkholed or blacklisted domain
  - Located more than 1000 miles away
  - Have no forward DNS domains
Tunneling

• Firewalls allow port 53 through
• Malware can phone home via port 53
• Covert channels via DNS traffic
  – Even embedded in fields of legitimate-looking DNS packets, such as DNSSEC keys or signatures
Detecting Tunneling

- Large UDP Request packets (>300 bytes)
DoS Attacks

• Attacks against the DNS server
  – TCP or UDP flood
  – SYN flood
  – Spoofed source addresses or botnets
DoS Attack Detection

• Watch for these to be different from baseline
  – Incoming bits/sec and outgoing bits/sec
    • Imbalance indicates an attack
  – DNS requests/sec (TCP and UDP)
  – TCP SYN/sec
  – Incoming TCP and UDP packets/sec
  – ICMP incoming and outgoing packets/sec and bits/sec