MENTAL MODELS FOR EFFECTIVE SEARCHING

CHRIS SANDERS | @CHRISANDERS88

1 PLAYER GAME
2 PLAYER GAME

Top- 06201969
Leveling Up Search

What to search for?

How to search for it?

What to search for?
Asking the Right Question

Attack Timeline

Attack Starts → Event 1 → Event 2 → Event 3 → Event 4 → Attack Ends

The timeline of events leading your current reference point.

The timeline of events following your current reference point.

Context about the nature of entities or relationships
Over 95% of case searches involve refinement.
The amount of refinement decreases as analysts gain experience, but not a lot.
Even highly experienced analysts rely on refinement.
REFINEMENT

**Expansion**: Broadening scope
- Time: 1 day to 1 week
- Criteria: 1 host to 10 hosts

**Reduction**: Narrowing scope
- Time: 12 hours to 10 minutes
- Criteria: All ports to 1 port
Start Broad or Narrow?

**NARROW SEARCHES**

- Useful for answering specific questions.
- May miss interesting things if they’re too specific.
  - “What HTTP transactions led up to the suspicious download on this host?”

**BROAD SEARCHES**

- Useful when you’ve run out of leads or need to explore a timeline, host, or indicator.
  - “What else has this external IP communicated with on my network?”

Specific searches get specific results.
How to Search for it?

Questions manifest in tools as syntax is search tools. The average security practitioner will use ~4 search syntaxes in a 10 year career.

**Common Query Languages**
- SQL – Osquery
- Lucene – Elastic
- MQL – Mandiant
- EQL – Endgame
- SPL – Splunk
- RegEx – Everything

Once you learn one search tool, it’s easier to learn others.
Learning Search Syntax
Learning Search Syntax

- Backend
- Syntax
- Partial Matching
- Features
- Output
Search Back End

Relational DB

Graph DB

Plain Text

Inverted Index

**Cypher (Graph)**
MATCH (p:Person)-[:WORKS_FOR]-(t:Finance)

**SQL (Relational)**
SELECT * FROM users WHERE dept='Finance'

<table>
<thead>
<tr>
<th>Users</th>
<th>Dept_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>1</td>
</tr>
<tr>
<td>Kim</td>
<td>2</td>
</tr>
<tr>
<td>Sally</td>
<td>2</td>
</tr>
<tr>
<td>Steve</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dept_ID</th>
<th>Dept</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Admin</td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
</tr>
<tr>
<td>3</td>
<td>IT</td>
</tr>
</tbody>
</table>
Search Syntax

**LUCENE**: field:value

**SPLUNK**: field=value

**LUCENE**: src_ip:192.168.1.1 OR dst_ip:192.168.1.1

**SPLUNK**: src_ip=192.168.1.1 OR port=80

**SPLUNK**: src_port=80 src_port=53

**LUCENE**: domain(\textit{xyz.com,abc.com,123.com})

**SPLUNK**: username!=\textit{chris} (matches when no field exists)

**SPLUNK**: NOT username=\textit{chris} (field must exist but not have value)
Partial Matching

Matching smaller parts of larger things.

**Wildcards** match one or more characters:
- Single Character: `st?ak` will match steak, st1ak, or stqak
- Multiple Characters: `st*ak` will match steak, st1ak, st2983928392ak

**Prefix search** will match a specific value followed by other data.
- `csan*` will match csanders, csanderson, csan12345

**Suffix search** will match a specific value after other data.
- `*ders` will match csanders, sanders, esanders, 123ders
Features and Output

**FEATURES**

- **Ranges**: Search between values
  - port: 1-1024
- **Exists**: Return results when a field is present
  - `_exists_:username`
- **Regular Expressions**: Complex pattern matching
  - `domain://^[a-zA-Za-z0-0-9]{2,}$`
- **Subsearch**: Search within a search

**OUTPUT**

- Statistical Transforms
- Portable Data Types (JSON, CSV)
## Search Features to Learn

<table>
<thead>
<tr>
<th>Search Backend</th>
<th>Partial Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Storage</td>
<td>Wildcards</td>
</tr>
<tr>
<td>Typed Data</td>
<td>Prefix Search</td>
</tr>
<tr>
<td>Filter vs. Query Context</td>
<td>Suffix Search</td>
</tr>
</tbody>
</table>

### Search Syntax

<table>
<thead>
<tr>
<th>Basic Structure</th>
<th>Regular Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compound Queries</td>
<td>Subsearch / Nested Search</td>
</tr>
<tr>
<td>Boolean Operators</td>
<td>Ranges</td>
</tr>
<tr>
<td>Escaped Characters</td>
<td>Exists Searches</td>
</tr>
<tr>
<td>Comparison Operators</td>
<td></td>
</tr>
</tbody>
</table>

### Output

<table>
<thead>
<tr>
<th>Transformations</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
THE END

Twitter: @chrissanders88
Blog: http://chrissanders.org
Training: http://networkdefense.co