KC7 - 101 notes

Created @July 4, 2025 10:33 AM

```
Employees
| take 10

Employees
| where role == "Pharmacist"

// when in doubt take 10
OutboundNetworkEvents
| take 10

// IP address how many time particular has access a system, how many failed login attempts
// Identify suspecious activity, detect patterns,based on abnormormal or high activity

// How many employees work at the company?
Employees
| count
// How many roles? Distinct use to clutter results
// 1. Identify unique IP addresses that have accessed a system
// 2. List all different usernames that have attempted to log in
// 3. Find the variety of processes running on a machine

Employees
|distinct role

Employees
| distinct ip_addr

Employees
| distinct username

// Where command helps to filter results based on specific conditions
Employees
| where role == "Radiologist"

// == operator used to find exact matches in a column. Look up a specific user, IP address or event
Employees
| where name == "Noemi Tep"

// In instances where we dont have an exact value used contains operator instead
// eg we know the security staff but we cant rem the exact title. Hmmm security guard or manager?
Employees
| where role contains "security"

//a phishing campaign, you start investigating emails related to health policies, looking for messages that mention health in the subject li
// along with emails about also pulling other emails. Since contains lets which only particular word that necessary
Email
| where subject has "health"

// !contains filter out certain values of findings. To find value that include a keyword, !contains excludes anything that matches
Email
| where recipient == "noemi_tep@jojoshospital.org"
| where sender !contains "jojoshospital"

// To filter data using multiple conditions at once. The and operator narrows down results by requiring all conditions are true
```

```
// eg. analyzing login attempts to find only failed logins from a specific IP address:
AuthenticationEvents
| where src_ip == "10.10.0.144" and result == "Failed Login"

// While and narrows down results by requiring multiple conditions to are true, or broadens them by allowing multiple possibilities.
// instead of running two separate queries we use or to check for both in one search
AuthenticationEvents
| where src_ip == "10.10.0.144" or src_ip == "10.10.0.86"

// AuthenticationEvents | where src_ip == "10.10.0.144" orsrc_ip == "10.10.0.86"orsrc_ip == "10.10.0.86"or src_ip == "10.10.0.20"  or src_i
// can be simplify using in:
AuthenticationEvents
| where src_ip in ("10.10.0.144", "10.10.0.86", "10.10.0.86", "10.10.0.20", "10.10.0.109") and result == "Failed Login"

// Suspect Nancy Roberts has been accessing unauthorized websites, ip_adr is unknown then used OutboundNetworkEvents
// Step 1. find Nancy ip_adr
Employees
| where name == "Nancy Roberts"

Employees
| where ip_addr == "10.10.0.30"

// Now we use that IP to check her browsing activity.
OutboundNetworkEvents
| where src_ip == "10.10.0.30"
| distinct url

// Finding Nancy Roberts' IP address to check web browsing
Employees
| where name == "Nancy Roberts"
//Nancy's ip address: 10.10.0.30
// Find all IT support employees
Employees
| where role contains "IT support"

// Look for logins from IPs used by IT support
AuthenticationEvents
| where src_ip in ("10.10.0.75", "10.10.0.42", "10.10.0.34", "10.10.0.10", "10.10.0.2")

// Count all failed logins
AuthenticationEvents
| where result == "Failed Login"
| count

//1. We use a `let` statement to create a temporary variable called `mary_ips`.
//2. This variable stores the `IP addresses` of employees whose name contains "Mary" (found in the `Employees` table).
//3. We then use `mary_ips` to filter the `OutboundNetworkEvents` table, checking how many websites those employees visited.

let mary_ips =
Employees
| where name has "Mary"
| distinct ip_addr;
OutboundNetworkEvents
| where src_ip in (mary_ips)

let mary_ips = Employees
| where name has "Mary"
| distinct username;
AuthenticationEvents
```

```
| where username in (mary_ips)
| count
```