What's in a (user|host) name?

That which we call a rose, by any other name, would still prick your fingers.
Logs 101: the basics

Logs - observable effects of an operation
Field - discrete components of a log
Parsing - identifying fields
Enrichment - adding information based on parsed values
Logstash - the one true log processor; log geeks want it, log software wants to be it
Translate - a logstash plugin that lets us do dictionary-style lookups based on field type/values
Logs 102: formats

Syslog - the devil, or at least his anarchist cousin; unstructured, arbitrary, ubiquitous

XML - the format used by Windows, highly structured

JSON - the only appropriate logging format

YAML - represents objects by a keyed value (multidimensional array++)
Logs 103: syslog and fields

20/11 10:30 10.10.15.18 password accepted for ceabhain

<date> <time_utc> <host_ip> <login_type> <login_status> for <username>

- date: “20/11”
- time_utc: “10:30”
- host_ip: “10.10.15.18”
- login_type: “password”
- login_status: “accepted”
- username: “ceabhain”
Why do I *really* like JSON?

```json
{
    "date": "20/11",
    "time_utc": "10:30",
    "host_ip": "10.10.15.18",
    "login_type": "password",
    "login_status": "accepted",
    "username": "ceabhain"
}
```
ceabhain, we hardly knew ye...
LDAP, Active Directory

- first/last name
- last password change
- certificate info
- department/division
- phone number
- building/office number
- photograph
- date created
- preferred/nicknames
- computer name (they’re in AD, right?!)
- asset tag/information
- group memberships - “memberOf”
- email address
- hardware vendor
- primary user
- pretty much any other attribute you want to store

ldapsearch, getad-user, getad-computer, python + ldap3, powershell, pick your language
Build a User’s Lookup Table (this is yaml)

ceabhain:
  dn: CN=Just A. Charlatan,OU=osg_users,DC=osg,DC=local
displayname: Just A. Charlatan
memberof:
  - CN=InfoSec,DC=osg,DC=local

conaire:
  dn: CN=Conaire MacLeoid,OU=osg_users,DC=osg,DC=local
displayname: Conaire MacLeoid
memberof:
  - CN=Mergers and Acquisitions,DC=osg,DC=local

https://github.com/kevinwilcox/python-elk/blob/master/crawl_ad_ldap/get_ad_users.py
Sample logstash "translate" Config

```ruby
filter {
  if [username] {
    translate {
      field => "username"
      destination => "user_info"
      dictionary_path => "ldap_users.yml"
    }
  }
}
```
What Would That Login Log Look Like?

{  
  "date": "20/11",
  "time_utc": "10:30",
  "host_ip": "10.10.15.18",
  "login_type": "password",
  "login_status": "accepted",
  "username": "ceabhain",
  "user_info": {
    "dn": "CN=Just A. Charlatan,OU=osg_users,DC=osg,DC=local",
    "displayname": "Just A. Charlatan",
    "memberof": "CN=InfoSec,DC=osg,DC=local"
  }
}
TELL ME MORE, TELL ME MORE!

(aka, what about the computer?)
Suppose an Inventory

10.10.14.1:
  hostname: infosec-paw-1
  mac_address: 00:12:34:56:78:90

10.10.15.18:
  hostname: mergers-01
  mac_address: 00:00:15:36:19:86

10.10.15.36:
  hostname: clancy-01
  mac_address: 00:00:10:05:09:70
  system_type: FreeBSD bastion host (headless)
With a Little Help From the User Script

infosec-paw-01:
  dn: CN=infosec-paw-01,OU=osg_paws,DC=osg,DC=local
  assigned_to: ceabhain
  location: 36.216, -81.682

mergers-01:
  dn: CN=mergers-01,OU=Mergers and Acquisitions,DC=osg,DC=local
  assigned_to: conaire
  location: 56.869687, -5.438271
  ups_connected: true
Two Logstash Translate Filters...

```ruby
filter {
  if [host_ip] {
    translate {
      field => "host_ip"
      destination => "inv_info"
      dictionary_path => "inv_computers.yml"
    }
  }
}

filter {
  if [inv_info][hostname] {
    translate {
      field => "[inv_info][hostname]"
      destination => "ldap_computer_info"
      dictionary_path => "ldap_computers.yml"
    }
  }
}
```
And one filter for the GPS info...

```plaintext
filter {
    if [ldap_computer_info][location] {
        mutate {
            add_field => {
                "g_maps" => "https://www.google.com/maps/@%{[ldap_computer_info][location]},15z?hl=en"
            }
        }
    }
}
```
The enriched log!

```json
{
  "date": "20/11",
  "time_utc": "10:30",
  "host_ip": "10.10.15.18",
  "login_type": "password",
  "login_status": "accepted",
  "username": "ceabhain",
  "user_info": {
    "dn": "CN=Just A. Charlatan,OU=osg_users,DC=osg,DC=local",
    "displayname": "Just A. Charlatan",
    "memberof": "CN=InfoSec,DC=osg,DC=local"
  },
  "inv_info": {
    "hostname": "mergers-01",
    "mac_address": "00:00:15:36:19:86"
  },
  "ldap_computer_info": {
    "dn": "CN=mergers-01,OU=Mergers and Acquisitions,DC=osg,DC=local",
    "assigned_to": "conaire",
    "location": "56.869687, -5.438271",
    "ups_connected": "true"
  },
  "g_maps": "https://www.google.com/maps/@56.869687,-5.438271,15z?hl=en"
}
```
With just one log entry, automated enrichment from existing systems and *one click* from an analyst...
The story “in words”

From the log: on 20th November at 10.30 AM UTC, “ceabhain”‘s username and password were used to login to a computer.

From enrichment: “ceabhain” is an InfoSec account and accessed a system in Mergers & Acquisitions that is powered via UPS. The system “ceabhain” logged into is assigned to Conaire MacLeod, has a IP ending in 15.18, in the village of Gleann Fhionnain, on the shores of Loch Seile.

And it is immortal (at least until the UPS dies).
Real-life scenario

<table>
<thead>
<tr>
<th>ou:0</th>
<th>Unique count of username.keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>OU=English</td>
<td>2</td>
</tr>
<tr>
<td>OU=HPC</td>
<td>2</td>
</tr>
<tr>
<td>OU=Office of Transfer Services</td>
<td>2</td>
</tr>
<tr>
<td>OU=Art</td>
<td>1</td>
</tr>
<tr>
<td>OU=College of Business</td>
<td>1</td>
</tr>
<tr>
<td>OU=General Education</td>
<td>1</td>
</tr>
<tr>
<td>OU=Health and Exercise Science (HES)</td>
<td>1</td>
</tr>
<tr>
<td>OU=Institutional Research &amp; Planning</td>
<td>1</td>
</tr>
<tr>
<td>OU/Library-Instruction</td>
<td>1</td>
</tr>
<tr>
<td>OU=MKT</td>
<td>1</td>
</tr>
<tr>
<td>OU=Marketing</td>
<td>1</td>
</tr>
<tr>
<td>OU=SOC</td>
<td>1</td>
</tr>
<tr>
<td>OU=Special Projects</td>
<td>1</td>
</tr>
<tr>
<td>OU=Sustainable Techni &amp; Built Envirn</td>
<td>1</td>
</tr>
<tr>
<td>OU=Technology Support Services</td>
<td>1</td>
</tr>
<tr>
<td>OU=University Communications</td>
<td>1</td>
</tr>
<tr>
<td>Missing</td>
<td>1</td>
</tr>
</tbody>
</table>
## Real-life 2

### Metrics
- **Metric**: Unique count of `username.keyword`

### Buckets
- **OU**: AVC for Enrollment Management - 2
- **OU**: College of Arts and Sciences - 2
- **OU**: College of Business - 2
- **OU**: College of Fine and Applied Arts - 2
- **OU**: GS - 2
- **OU**: University Affairs - 1
- **OU**: Athletics Office - 1
- **OU**: College of Health Sciences - 1
- **OU**: GC - 1
- **OU**: Institutional Research & Planning - 1
- **OU**: Library - 1
- **OU**: Provost & Exec VC - Academic Affairs - 1
- **OU**: University College - 1
- **OU**: University Communications - 1
- **Missing**: 1
The Takeaways...

We have multiple sources of information about various names:

- Active Directory
- LDAP
- DNS
- Inventory Databases

By enriching logs from those data sources:

- "business logic" built into SIEM "flow", increasing value
- analysts get a "big picture" view earlier in the process
- analysts can spend their time more effectively
- group-based reporting is quicker and easier
- mismatches and anomalies start to become more obvious
- seemingly "common" events can become actionable indicators