Elasticsearch, Logstash, and Kibana (ELK)

Dwight Beaver
Sean Hutchison
January 2015
This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon®, CERT®, CERT Coordination Center® and FloCon® are registered marks of Carnegie Mellon University.

DM-0001883
Who are we and what do we do?

I NEED YOU TO ASSEMBLE A HUGE AMOUNT OF TOTALLY INCOMPREHENSIBLE DATA.

MAKE IT BORING SO NO ONE LOOKS AT IT TOO CLOSELY. I'M AIMING FOR QUANTITY OVER QUALITY.

I HAVE A BAD FEELING ABOUT THIS.

NO ONE WOULD PAY YOU TO FEEL GOOD.
What’s our problem?

- Small team
- Lots of users (all untrusted)
- Lots of systems
- Lots of logs
- Luckily, no “sensitive” information BUT
#!/usr/bin/operations

import small_team

tasks = detection(security_team=False)

while max(cat_video.watch()):
    if tasks.detect() == True:
        react(tasks)
    else:
        sleep(5)
Why Elasticsearch

• Easy to deploy (minimum configuration)
• Scales vertically and horizontally
• Easy to use API
• Modules for most programming/scripting languages
• Actively developed with good online documentation
• It’s free
How Elasticsearch Works in 25 seconds

Shards
- Single instance of Lucene on a node
- Can be primary or replica

Index
- Mapping of shards to nodes
- Like a database within a relational database

Nodes
- Keeps a copy of the index
- Maintain primary and replica shards
Hardware and Infrastructure

- Blades
- Network attached storage – NFS
- Aggregate TAP, SPAN off switches (physical and virtual)
- Virtualization (VMware)
- Puppet
Nodes

8 x Nodes – virtualized

- 4x Cores
- 16 GB ram
- 500 GB data partition (NFS->NAS)

Deployed/Configured using Puppet modules.
https://forge.puppetlabs.com/
Software

- logstash
  - (Data Collection)

- redis
  - (Queuing)

- python
  - (Glue/Integration)

- elasticsearch
  - (Storage, index, search)

- kibana
  - (Visualization)
Data Sources

- Windows Event Logs
- Syslog
- Bro (session data/dpi)
- SiLK (flow)
- SNMP
- PCAP (stored on disk, index information in ES)
Can I see a diagram with boxes and arrows?
Things we can do

- Batch analysis (retrospective)
- Correlation between data sets
- Make pretty graphs for displaying on TVs – Kibana
- Alerting – Python/R
Where we want to do

Puppet / Applications containers (ie, Docker)
Our environment is defined in software.
Can we use this to automate auditing?
Batch/Retrospective Analysis

• Say we saw some interesting traffic coming from one of our servers – we want to know which processes were run around that time on that host…

• Set a simple filter in Kibana like…

• Kibana queries ES and returns…
Batch/Retrospective Analysis
Batch/Retrospective Analysis

• You can also use ES Python API to perform queries – http://elasticsearch-py.rtfd.org/
• Lots of query and filter options; JSON syntax; more flexibility and control
• Good for…
  • Running queries on-demand over any period of time
  • Checking on important events that are too cumbersome to alert on
  • Daily review of logs
  • Investigation
Batch/Retrospective Analysis

- Example query bodies

```json
fs_objaxs_body = {
    "_source": ["@timestamp","SubjectUserName","subjectDomainName","SubjectLogonId","ObjectName","ObjectType","host","ProcessName","message"],
    "query":{
        "filtered": {
            "query": { "bool": { "must": [ {
                "match": { "eventlog_id": 4663 }},
                { "match": { "eventlog_category": { "query": "File System", "operator": "and" }}}}]
            }},
            "filter":{ "range": { "@timestamp": { "from": "now-1d" }}}}]
}

reg_objaxs_body = {
    "_source": ["@timestamp","SubjectUserName","host","ProcessName","message"],
    "query":{
        "filtered": {
            "query": { "bool": { "must": [ {"match": { "eventlog_id": 4657 }}
            ]},
            "filter":{ "range": { "@timestamp": { "from": "now-1d" }}}}]
}
```

- And get...
Batch/Retrospective Analysis

$ python2.7 OBJAXS.py -a 30m

[+] Returned 25 hits on file system...
[+] Suspect access to audited file system by USER/SERVICE accounts:

********** 2014-12-16T16:14:28.000Z UTC **********
Host: LogonID: 0x1cfd2383 touched FILE: C:\windows\System32\winevt\Logs\Security.evtx
Process: C:\windows\System32\svchost.exe
Accesses: ReadData (or ListDirectory)

[+] Access to audited file system by COMPUTER accounts (shows processes used):

[+] Returned 1 hits on registry...
[+] Suspect modification of registry by USER/SERVICE accounts:

********** 2014-12-16T16:21:36.000Z UTC **********
On host A registry value was modified.
Subject:
Security ID: S-1-5-21-2723307174-1429147120-1202244634-1703
Account Name:
Account Domain: DTE
Logon ID: 0x7b7b32

Object:
Object Name: \REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
Object Value Name: Userinit
Handle ID: 0xfc
Operation Type: Existing registry value modified

Process Information:
Process ID: 0xb9c
Process Name: C:\windows\regedit.exe

Change Information:
Old Value Type: REG_SZ
Old Value: C:\windows\system32\userinit.exe,
New Value Type: REG_SZ
New Value: C:\windows\system32\userinit.exe, C:\windows\system32\evil.exe

[+] Modifications to registry by COMPUTER accounts (shows processes used):
Correlation of Data Sets & Visualization

• Correlate events within and between data sets to gain context
• Visualizing data with Kibana facets…
  • Makes aspects of data more readily apparent
  • Aids perspective and understanding of data
  • Looks cool
• Typically…
  • Attach one or more Queries to individual facets
  • Drill down on specific data using Filters (whole page)
  • Plethora of info with just one or two filters
Correlation of Data Sets & Visualization
Correlation of Data Sets & Visualization
Alerting – Windows Event Monitoring

- Want to know about certain events as they occur
  - Administrator login
  - Local/SAM account login attempts
  - User account creation/re-enabling
  - Creation/Addition to Groups
  - Scheduled Task creation
  - Log cleared
- Uses ES Python API and CRON
  - Queries ES 15 times per hour
  - Every 4 minutes -> “from”: “now-4m”
Alerting – Windows Event Monitoring

- Example Alerts received via email notifications

Security Group Management Events:


User Account Management Events:

- 2014-12-08T14:31:37: created user account [redacted] in DTE domain.

Local Account Logon Events:

- 2014-12-09T13:45:15: Host [redacted] attempted to locally validate credentials for user [redacted].
  Error Code: 0x0 Successful validation

Logs Cleared:

- 2014-11-25T12:26:48: The Application log file was cleared on [host] by [user].
Alerting – Windows Event Monitoring

- Example Alerts received via email notifications

The following events have recently occurred...

Local Account Logon Events:

2014-11-17T10:32:10  Host  attempted to locally validate credentials for user X
  Error Code: 0xc0000064  User name does not exist

2014-11-17T10:32:10  Host  attempted to locally validate credentials for user APPXMA7Z
  Error Code: 0xc0000064  User name does not exist

2014-11-17T10:33:06  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:06  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:06  Host  attempted to locally validate credentials for user GUEST
  Error Code: 0xc00000072  Account is currently disabled

2014-11-17T10:33:06  Host  attempted to locally validate credentials for user GUEST
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:29  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:29  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc000000072  Account is currently disabled

2014-11-17T10:33:29  Host  attempted to locally validate credentials for user GUEST
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:29  Host  attempted to locally validate credentials for user GUEST
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc000000064  User name does not exist

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password

2014-11-17T10:33:34  Host  attempted to locally validate credentials for user ADMINISTRATOR
  Error Code: 0xc0000006a  Correct user name, wrong password
Alerting – Windows Event Monitoring

- Example Alerts received via email notifications

Administrator Logon Events:

2014-12-11T08:23:49  Administrator account used to logon to host
Process: -
Target Domain: 
Logon ID: 0x416eb3f
Logon Type: 3

2014-12-11T08:23:52  Administrator account used to logon to host
Process: C:\Windows\System32\winlogon.exe
Target Domain: 
Logon ID: 0x4174903
Logon Type: 10

2014-12-11T08:23:52  Administrator account used to logon to host
Process: C:\Windows\System32\winlogon.exe
Target Domain: 
Logon ID: 0x4174bba
Logon Type: 10
Alerting – Windows Event Monitoring

• Example Alerts received via email notifications

Scheduled Tasks:

2014-11-25T12:07:08 LogonId: 0x803f59 created new task named: \tester3
Host:
Description: This is another test
Principals:
  <Principal id="Author">
  <RunLevel>HighestAvailable</RunLevel>
  <UserId/>
  <LogonType>InteractiveToken</LogonType>
  </Principal>

RunOnlyIfNetworkAvailable: true
Hidden: false
RunOnlyIfIdle: false
Actions: context="Author">
  <Exec>
    <Command>C:\Windows\System32\cmd.exe</Command>
  </Exec>
Alerting – Irregular Login Activity

- Want to keep an eye on privileged account use
- Want to know…
  - When users login to hosts they never or rarely ever login to
  - When users login from atypical source IPs
  - When user logins violate certain thresholds based on previous behavior
- Uses ES Python API, CRON, R, and sqlite3 DB
  - Delivers daily login stats
  - Updates weekly and expires old weeks
  - Checks against DB with 4 weeks of aggregated data
Alerting – Irregular Login Activity

24 hour stats

• Processed by R and delivers Daily Login Stats email with plots

RSweekly.txt

• Contains all login activity for the week

lin_stats_weekly.py

• Processes RSweekly.txt with R, expires old weeks / entries, deduplicate

DB
Alerting – Irregular Login Activity

- Example Daily Login Stats with plots
- Email Message

Basic User Login stats for the last 24 hours...

Logins before 6AM and after 8PM:

2014-12-12T05:46:29   RDP   user 10.67.16.165   host
Alerting – Irregular Login Activity

- Example Daily Login Stats with plots
Alerting – Irregular Login Activity

- Example Daily Login Stats with plots
Alerting – Irregular Login Activity

- Example Daily Login Stats with plots
Alerting – Irregular Login Activity

• Example Alerts received via email notification

2014-12-12T10:55:33  SSH user [Redacted] NEVER seen before or has not logged in for TWO or more weeks
Source IP: 10.61.16.146  Host: [Redacted]

[U2S] User NEVER logged in from this Source IP before or last login was TWO or more weeks ago

Source IP: 10.61.16.113

2014-12-15T14:34:07  SSH user [Redacted] logged into host [Redacted] from Source IP 10.61.16.113
[U2H] Last login to this host was NEVER or TWO or more weeks ago

Host: [Redacted]
Questions and Discussion