

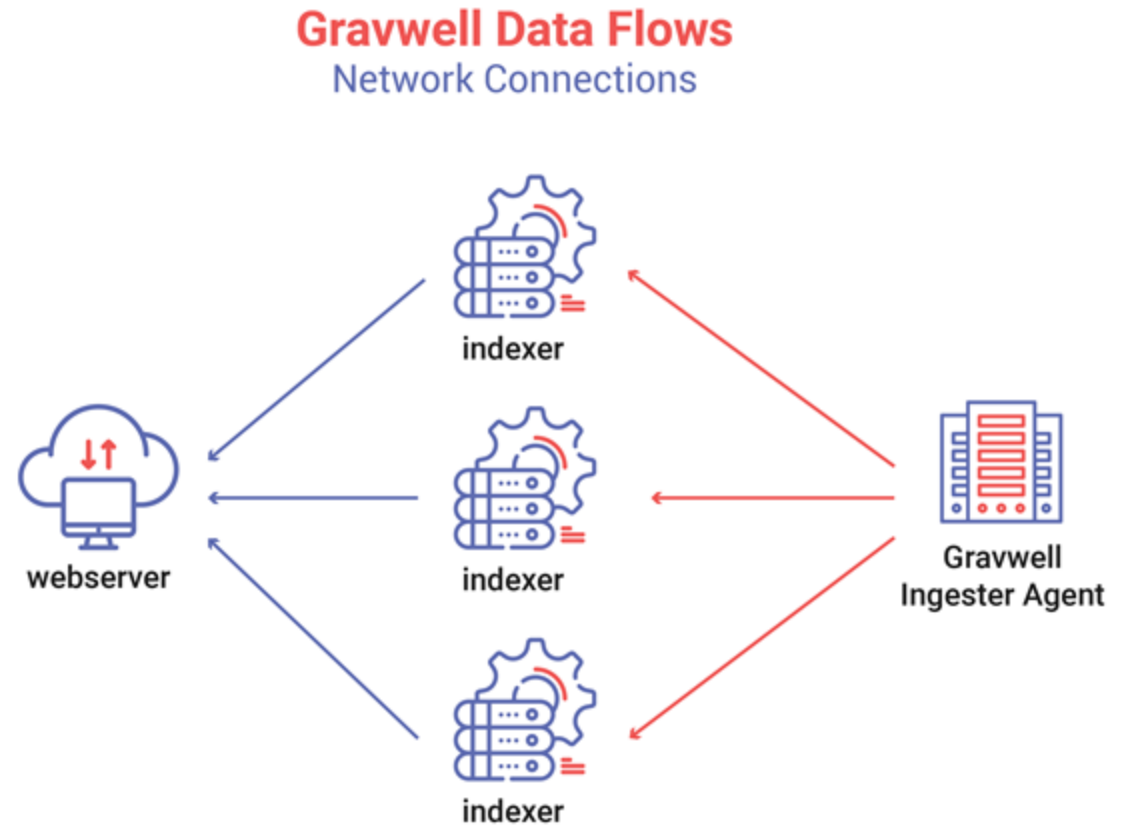


Gravwell

Team 29
GRID-SIEM



Architecture



Architecture Cont.

- Gravwell appears to be a data ingestion tool or data analytics platform and could be used to in addition to Security Onion to better handle the data coming in.
- Gravwell allows data to be explored by using a central ingester which have [options that are premade](#) by Gravwell, as well as an open-source API created by Gravwell, available on [Github](#).
- This ingester will divide up the data and send it to indexers which divide it up into storage wells that are segmented based on type, for example syslogs vs linux logs.
- The indexers can then send data to a webserver to be viewed or searched.

Hardware Configuration

Webserver

- Focusing point for all searches
 - interface
- Doesn't need much actual storage
 - Speed > size
- Benefits from high amounts of ram
 - 16GB recommended
- Built to run everything concurrently
 - More cpu cores = better
 - 4 cores recommended

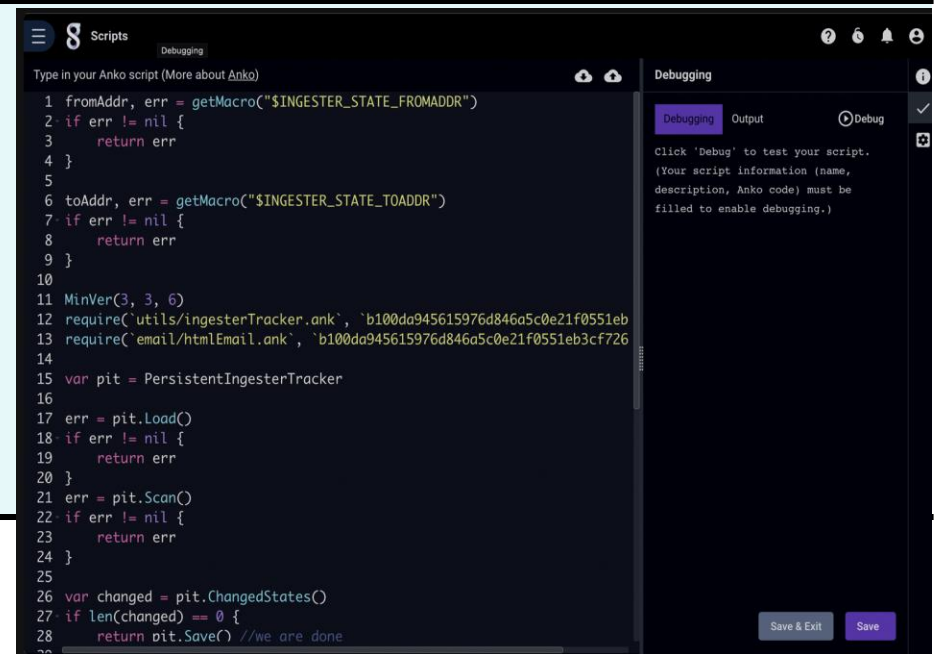
Indexer

- Storing, retrieving, and processing data
- Decent amount of very high speed storage
 - High speed NVME SSD recommended
- Lots of low latency storage
 - 32GB Recommended
- 8 CPU Cores recommended

Automation Pt.1

How To:

- Gravwell Search agents: automated search scripts that can be scheduled to detect malicious behavior. Or any other specific action.
- The Search Agent is installed automatically by the Gravwell Debian package.
- In case a scheduled script is missed, Gravwell has a feature called Backfill scheduling which can perform the script after an update is done. So information from that time period is not lost.
- Script debugging options are available.
- The Anko and Eval modules help analysts create scripts with greater flexibility.

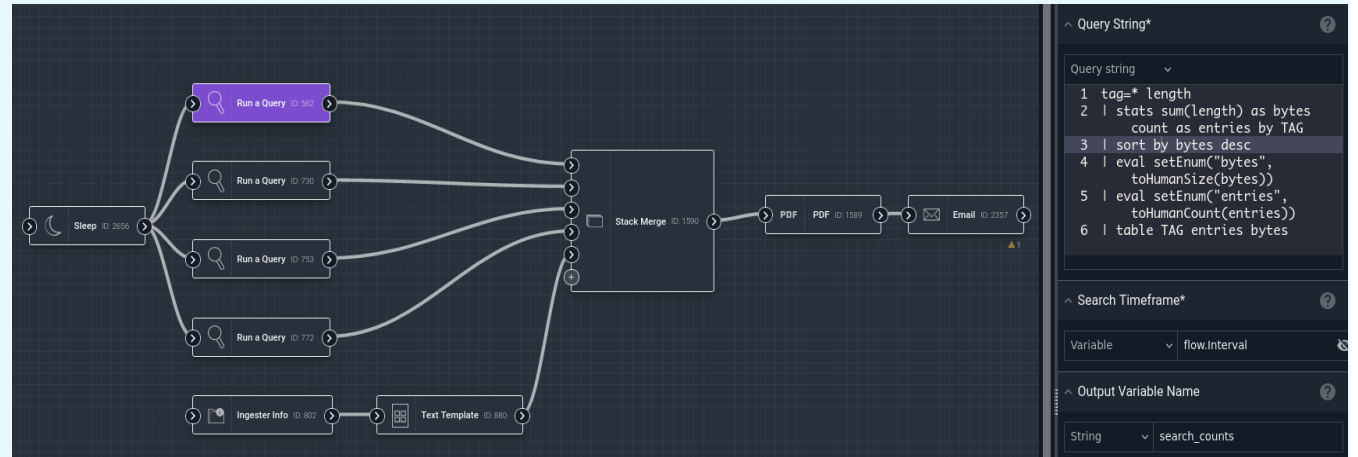


The screenshot shows the Anko script editor interface. The main area displays a script with the following code:

```
1 fromAddr, err = getMacro("$INGESTER_STATE_FROMADDR")
2 if err != nil {
3     return err
4 }
5
6 toAddr, err = getMacro("$INGESTER_STATE_TOADDR")
7 if err != nil {
8     return err
9 }
10
11 MinVer(3, 3, 6)
12 require('utils/ingesterTracker.ank', 'b100da945615976d846a5c0e21f0551eb')
13 require('email/htmlEmail.ank', 'b100da945615976d846a5c0e21f0551eb3cf726')
14
15 var pit = PersistentIngesterTracker
16
17 err = pit.Load()
18 if err != nil {
19     return err
20 }
21 err = pit.Scan()
22 if err != nil {
23     return err
24 }
25
26 var changed = pit.ChangedStates()
27 if len(changed) == 0 {
28     return pit.Save() //we are done
```

The right-hand side of the interface shows a 'Debugging' panel with a 'Debug' button and instructions: 'Click "Debug" to test your script. (Your script information (name, description, Anko code) must be filled to enable debugging.)' At the bottom right, there are 'Save & Exit' and 'Save' buttons.

Automation Pt. 2



Flows/Playbooks:

- Flows provide an easy way to automate work in Gravwell. That would otherwise require manual analyst action.
- Actions such as sending emails, generating files, alert processing, send Slack messages and more tasks could be integrated in to a flow.
- The basic unit of a flow is the NODE. A node can contain one or more payload input sockets(as seen above) but they are typically confined to one task-socket at the most to reduce complexity and debugging time.
- A chain of nodes, or a Flow is always executed one at a time, in order.
- Flows are created using Gravwell's flow editor, Gravwell documentation also provides generic template flow patterns for the most commonly used flow patterns and has a set of node lists if you want to customize your own.

Collaboration with existing SIEM

- Query any environment (Azure, AWS, etc.)
- Can search multiple sources at once
- Also works for separate instances and environments for different teams (business units)
- Combine the output of each query and combine to common output
- Will allow the auto-extractor to pull out desired fields and process as JSON without needing to alter the schema
- Gravewell can automatically extract and process specific data fields as JSON without needing to change the data structure, ensuring that no data is lost due to a rigid schema.
- Lots of log parsing options.

Helpful Links

<https://www.gravwell.io/gravwell-for-siem#:~:text=We%20understand%20that%20teams%20have,log%20at%20a%20lower%20cost.>

https://docs.gravwell.io/scripting/schedule_dsearch.html

<https://docs.gravwell.io/flows/flows.html>

<https://docs.gravwell.io/architecture/architecture.html>