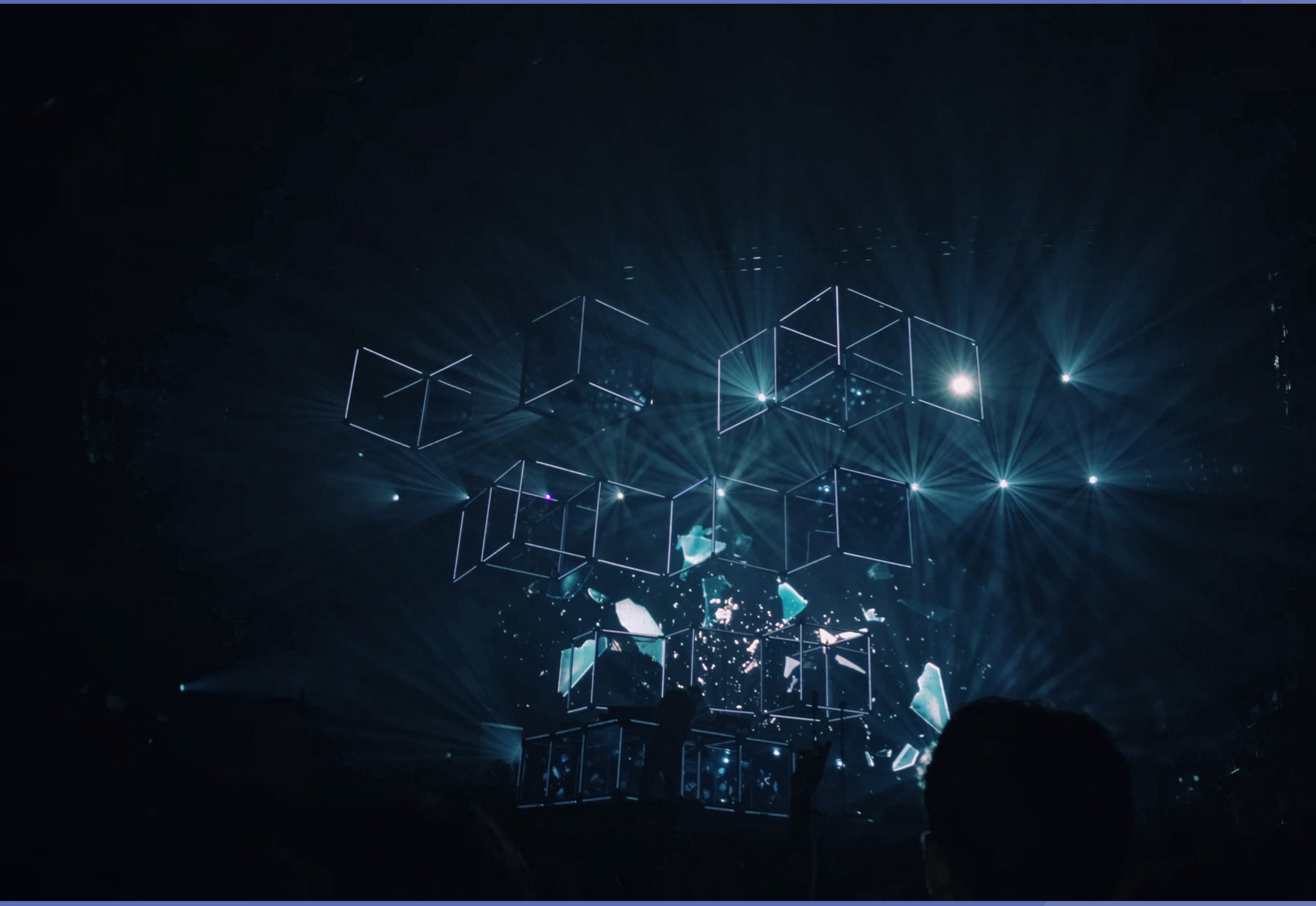
Case Study

# A Splunk Expert Tries Gravwell



We frequently have discussions with prospective customers about the differences between Gravwell and Splunk. A common theme of those questions is "How easy is it for a Splunk expert to get spun up and fully integrated using Gravwell?" In this case study, we'll hear about one of our new integration service partner’s first experiences with Gravwell. The following is a "mock customer" use case put together over the course of a single day as they experimented with setting up a customer in Gravwell.

Gravwell

## Scenario

A customer has the following systems they would like to monitor:

- Prestashop ECommerce web store
- Opnsense firewall logs
- Linux software installs

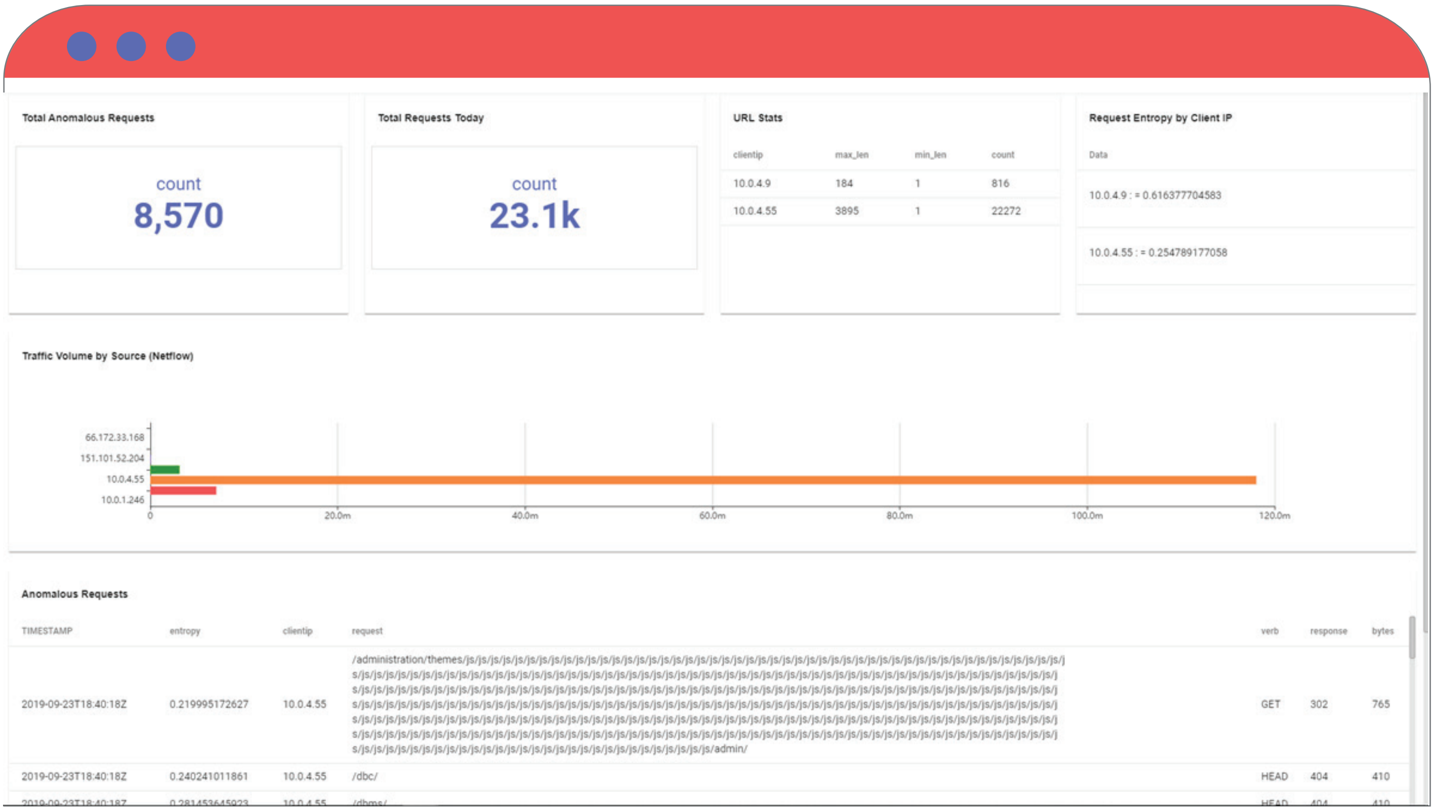
The customer was previously sshing and grepping logs on various machines in a reactive manner. The main issues they encounter include a) detecting web-based attacks against their ecommerce server in a timely manner, b) fully understanding the types of traffic passing through their firewall, and c) problems with developers installing unauthorized software (e.g.,Telnet) onto their Linux hosts.

## Solution

This proof of concept using Gravwell addresses these problems without requiring the installation of any additional security tools via the three dashboards below:

- ### Ecommerce Anomalies

This dashboard specifically looks for anomalous requests hitting the customer's ecommerce server. It relies on Apache web logs and netflow as data sources.  At a glance, a user can identify bots and scanners, which are good targets for Gravwell's automated remediation feature.



Total Anomalous Requests

count
8,570

Total Requests Today

count
23.1k

URL Stats

| clientip | max_len | min_len | count |
|---|---|---|---|
| 10.0.4.9 | 184 | 1 | 816 |
| 10.0.4.55 | 3895 | 1 | 22272 |

Request Entropy by Client IP

| Data |
|---|
| 10.0.4.9 : = 0.616377704583 |
| 10.0.4.55 : = 0.254789177058 |

Traffic Volume by Source (Netflow)

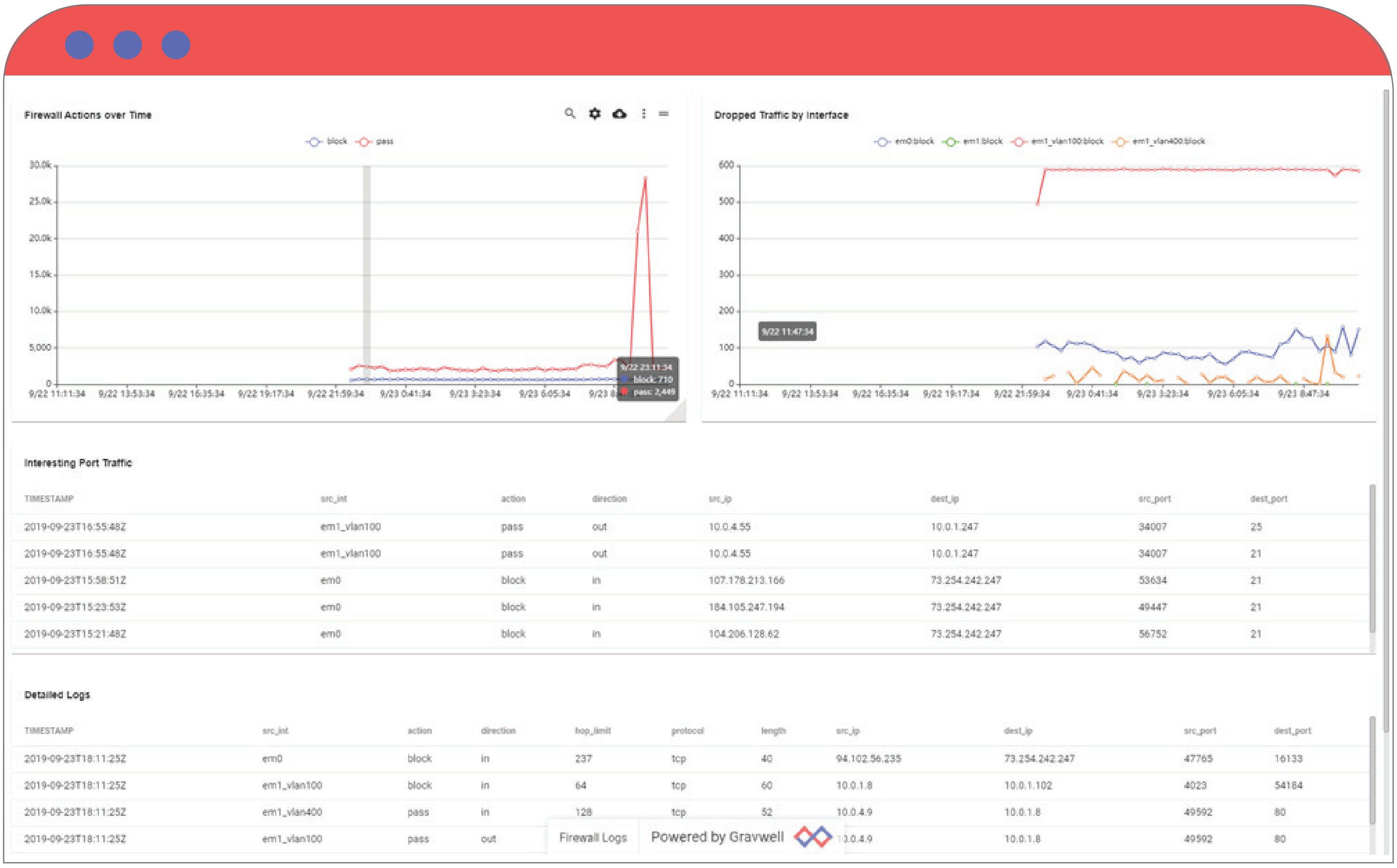
Anomalous Requests

| TIMESTAMP | entropy | clientip | request | verb | response | bytes |
|---|---|---|---|---|---|---|
| 2019-09-23T18:40:18Z | 0.219995172627 | 10.0.4.55 | /administration/themes/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/js/admin/ | GET | 302 | 765 |
| 2019-09-23T18:40:18Z | 0.240241011861 | 10.0.4.55 | /dbc/ | HEAD | 404 | 410 |
| 2019-09-23T18:40:18Z | 0.281453645923 | 10.0.4.55 | /dbms/ | HEAD | 404 | 410 |

## Firewall Activity

This dashboard provides a general overview into the firewall's activities:

- The rate of traffic passes versus drops.
- Which interfaces are dropping the most traffic.
- The types of "interesting" traffic seen crossing or attempting to cross through the firewall.



Interesting Port Traffic

| TIMESTAMP | src_int | action | direction | src_ip | dest_ip | src_port | dest_port |
|---|---|---|---|---|---|---|---|
| 2019-09-23T16:55:48Z | em1_vlan100 | pass | out | 10.0.4.55 | 10.0.1.247 | 34007 | 25 |
| 2019-09-23T16:55:48Z | em1_vlan100 | pass | out | 10.0.4.55 | 10.0.1.247 | 34007 | 21 |
| 2019-09-23T15:58:51Z | em0 | block | in | 107.178.213.166 | 73.254.242.247 | 53634 | 21 |
| 2019-09-23T15:23:53Z | em0 | block | in | 184.105.247.194 | 73.254.242.247 | 49447 | 21 |
| 2019-09-23T15:21:48Z | em0 | block | in | 104.206.128.62 | 73.254.242.247 | 56752 | 21 |

Detailed Logs

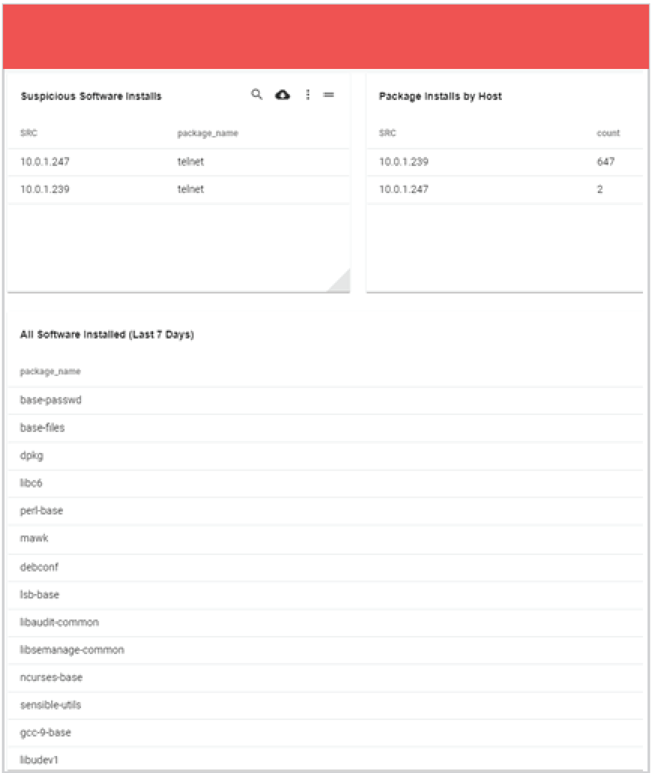
| TIMESTAMP | src_int | action | direction | hop_limit | protocol | length | src_ip | dest_ip | src_port | dest_port |
|---|---|---|---|---|---|---|---|---|---|---|
| 2019-09-23T18:11:25Z | em0 | block | in | 237 | tcp | 40 | 94.102.56.235 | 73.254.242.247 | 47765 | 16133 |
| 2019-09-23T18:11:25Z | em1_vlan100 | block | in | 64 | tcp | 60 | 10.0.1.8 | 10.0.1.102 | 4023 | 54184 |
| 2019-09-23T18:11:25Z | em1_vlan400 | pass | in | 128 | tcp | 52 | 10.0.4.9 | 10.0.1.8 | 49592 | 80 |
| 2019-09-23T18:11:25Z | em1_vlan100 | pass | out |  |  |  | 10.0.4.9 | 10.0.1.8 | 49592 | 80 |

The “Interesting Port Traffic” panel can easily be set to alert the firewall administrator.

## Software Inventory

This dashboard helps address the customer's problem with undesirable software installations on their systems. It also helps provide a detailed view into all packages installed across all systems. There is also the possibility for the customer to rely on a lookup table to identify all undesired software (this demo simply looks at "Telnet").

## Additional Technical Info & Considerations



Suspicious Software Installs

| SRC | package_name |
|---|---|
| 10.0.1.247 | telnet |
| 10.0.1.239 | telnet |

Package Installs by Host

| SRC | count |
|---|---|
| 10.0.1.239 | 647 |
| 10.0.1.247 | 2 |

All Software Installed (Last 7 Days)

| package_name |
|---|
| base-passwd |
| base-files |
| dpkg |
| libc6 |
| perl-base |
| mawk |
| debconf |
| lsb-base |
| libaudit-common |
| libsemanage-common |
| ncurses-base |
| sensible-utils |
| gcc-9-base |
| libudev1 |

File followers were deployed to each server; tagging setup and file following was implemented as appropriate. In addition, the firewall was configured to ship netflow v5 data to the Gravwell instance's IP.

For the demo's purposes, all data was fed into the default well. In a real environment, they would likely use a different well for netflow for better control over age-out and retention policies. They would also want to consider the use of a packet capture instead of netflow for root-cause analysis on a shorter retention -- however for this scenario, they wanted to be able to deploy Gravwell and address the sample problems without requiring the customer to make any network changes.

They created an auto-extractor for the opnsense firewall logs and added a grokfile as a resource (using https://raw.githubusercontent.com/gravwell/resources/master/grok/all.grok). They relied fairly heavily on the grokfile as a quick way to parse out apache logs. With more time, they would explore making these searches more efficient by only looking at the specific fields needed from the apache logs.

## Conclusion

We at Gravwell are excited to be expanding our partner network. We have been happy to see that Splunk experts can easily pick up Gravwell and go from zero to value in a very, very short amount of time. Getting Gravwell spun up is so much easier than the other tools on the market thanks to our tech stack and our amazing engineering team.

If you're not a Splunk expert, we have a wonderful Customer Success team that's ready to get you up and running in no time.

If you're experiencing any of the Splunk pain that so many others are feeling, give Gravwell a spin and see the power of data analytics re-thought for the age when every company is a data company.