HACKERSTORM

Thinking outside the cage

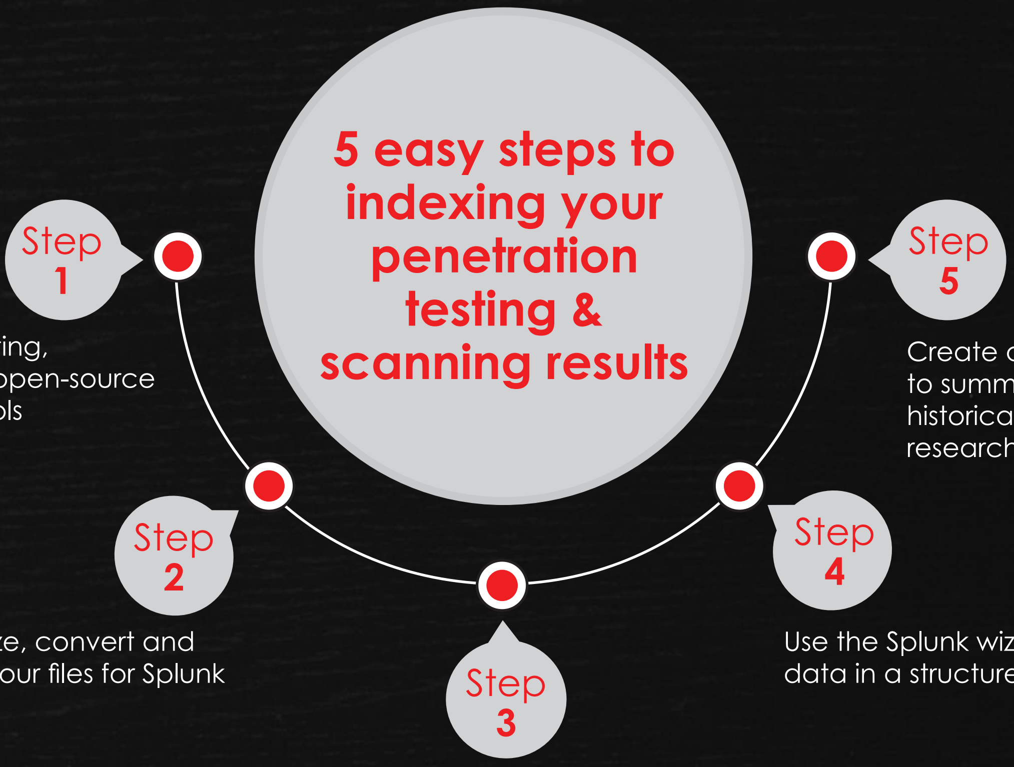
Fast Historical Threat Research

Indexing penetration testing and vulnerability scan results with Splunk for fast retrieval of past weaknesses during cyber response incidents, audits and compliance checks

January 2018

Overview

This guide describes how to use vulnerability assessment tools and penetration test reports with Splunk to enable analysts to review the history of a host, network or service within mintues to speed up incident response, enable faster audit and improve compliance reporting for cyber security threats and risks. With a centralized logging system like Splunk we will describe how easy it is to achieve and provide example dashboards for searching and reviewing the results from scanning and testing.



5 easy steps to indexing your penetration testing & scanning results

Step 1

Select your testing, scanning and open-source intelligence tools

Step 2

Centralize, convert and format your files for Splunk indexing

Step 3

Upload manually or automate uploads to Splunk

Step 4

Use the Splunk wizard to extract data in a structured way

Step 5

Create dashboards to summarize host historical data for fast research

What this guide covers

- The challenge
- The benefits
- How the data gets into Splunk
- Step by step guide importing data
- Ideas for dashboards

The Challenge

1. **We are still too slow in both detecting and containing suspected, suspicious or actual breaches and;**
2. **We are not making the best use of tools available to make our lives easier.**

According to the IBM Global Cost of Data Breach Study 2017, it takes on average 66 days for businesses to contain a breach.

Time to contain a data breach 66 DAYS

These delays impact incident analysis and the ability to audit and report on compliance. Its essentially the same data being used for different purposes so all assurance activities are impacted. Can we really trust any process that takes 66 days to answer simple questions like;

- What has changed?
- When did it change?
- Did it have known weaknesses or problems?
- How long were weaknesses exposed?

If it takes 66 Days to obtain answers for questions like these, its almost certainly going to be unreliable and will have gaps so what we want is to speed up the process by capturing as much data we can on observed or identified weaknesses and get them logged centrally, preferably in an automated way and make the whole process much more reliable and trustworthy to interrogate..

IBM: "The faster the data breach can be identified and contained, the lower the costs."

The Benefits

What are the benefits of centralizing and indexing outputs from penetration testing and VA scanning?

This is by no means exhaustive, but should hopefully give you an idea.

1. **Instant view.** An instant view of a hosts entire history in a single view of with all relevant data. Depending on what tools you have and use, you can see if anything has changed and when. It should take less than 15 minutes to understand the hosts history.

2. **Cyber response.** Fragmented security responsibilities slow incident investigations. Teams or people hoarding data that could have helped another identify potential issues or pro-actively prevent an incident can become a thing of the past.

3. **Culture.** By asking all security teams to centralize or make their information available to be indexed, we can begin to introduce a culture of knowledge sharing. Executives need to decide, whats more important, the data or internal office politics?. A lack of knowledge sharing inside organizations compounds the lack of knowledge sharing in industry, governments and suppliers.

4. **Better Assurance.** Without a centralized view of the entire history of a host, network or service means its difficult to provide any meaningful assurance to asset owners. What good is an anti-virus report on its own when security has many layers?.

"15 minutes or less to understand an assets entire history"

How the data gets into Splunk

Below is a flow diagram which shows the use of tools like nmap, openvas, nikto, sqlmap and penetration test report findings as the source data which is checked by Splunk, indexed and presented to the analyst via dashboards (or search tool).

Once you decide on which data to index in Splunk, you need to save the files to a folder on a server that Splunk can reach. Ideally you would convert the files into csv and make the findings look like a typical log file entry (covered later). Splunk can then automatically check for new csv files and index the data in them (or you can choose to manually upload the csv files using drag and drop tool in Splunk). Once indexed, the data can be used for searching and historical analysis, but if you schedule your scan jobs, you can monitor the results and alert too and become aware of issues as they are discovered, something traditional SIEM solutions would struggle to do with such data sources.

There is no right or wrong way in terms of choices of tools or data sources. Automation is preferred as we are lazy, but if its easier for you, files can be manually uploaded to Splunk very easily.

Hacking tools & methods to Splunk Sequence

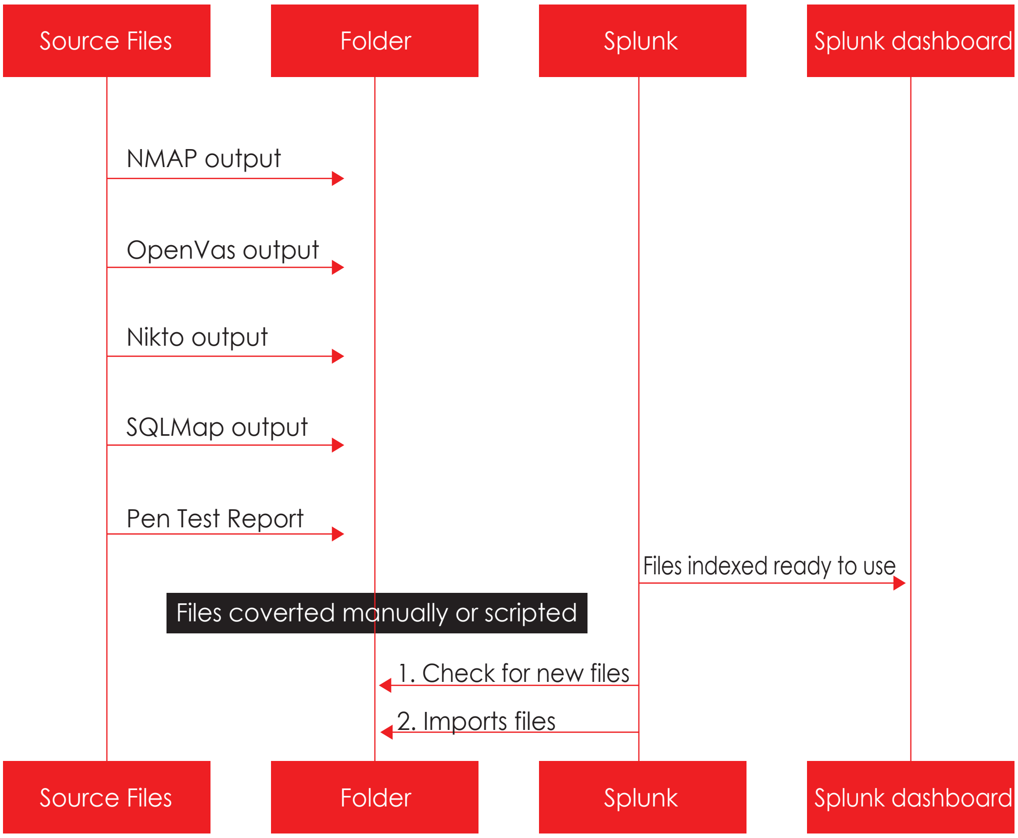


Diagram showing flow from testing & scanning output into Splunk

Step by Step Guide to Importing Data

Step 1. Tools and data source selection

The choice of tools and data sources is entirely up to you, we have chosen some tools readily available on Kali (the hacking distribution from offensive security). What you use should be tailored to your business, it should map to your identified threats, risks and business impact assessments.

You never want to rely on a single product or tool and it makes sense to use a combination of commercial and open-source tools in addition to penetration testing. For the purposes of this guide we have chosen for our example business to compliment commercial tools with:

- NMAP IP and port scanner,
- Nikto Web-server scanner
- SQLMap for SQL injections and;
- Penetration test report findings.

Step 2. Centralize, convert and format outputs

Once you decide to use a particular tool or source data, you need to give it a home that the Splunk servers can reach. Luckily, Splunk is very good at accessing folders on servers, the first thing to do is create a dedicated folder for each tool or data set. In our example, we would create a folder for each of our selections and give them simple descriptive names e.g.

- nmap
- nikto
- sqlmap
- pentest

Next, the most important thing of all, make sure all outputs get saved or copied there. If tools are being used as part of a scheduled job to conduct scans, it should be trivial copying or saving the outputs to there respective folders automatically.

Convert the files into a csv. Within each csv file, you ideally want to construct the same column names, ordering of the columns is not an issue, Splunk wont care, but having the same column names will make it easier to find data once indexed across multiple data sources. For each finding in the original output you want to ensure there is a line detailing the IP address, host-name, date, and findings plus any extras you want. You can create these files manually, but if you want to automate, we would suggest a simple python script to read the outputs and for each finding write a new line to a csv file.

We created the following columns for our example source files;

For NMAP: DST_IP, HOSTNAME, DST_PORT, PROTOCOL, FINDING, SERVICE, DATE

For NIKTO: DST_IP, HOSTNAME, DST_PORT, URL, DATE, FINDING

For SQL Map: DST_IP, HOSTNAME, DST_PORT, URL, DATE, FINDING

For Penetration test reports: DST_IP, HOSTNAME, DST_PORT, DATE, FINDING,SEVERITY

(Tip: ask the pen-testers to provide findings in a csv format to save time).

The reason for **dst_ip** and **dst_port** (dst is destination) is that the same results can be reported with suspicious activity or logs from IDS, IPS, Firewalls and other applications or appliances as this is how they will normally record the events. But don't worry if you forget, Splunk lets you create an alias very easily later to map them where different.

Example NMAP csv file;

```
dst_ip,hostname,dst_port,protocol,finding,service,date
10.1.0.132, some.server,21,tcp,open,ftp,06/07/2017 17:00
10.1.0.132, some.server,23,tcp,closed,telnet,06/07/2017 17:00
10.1.0.132, some.server,25,tcp,open,smtp,06/07/2017 17:00
10.1.0.132, some.server,53,tcp,closed,domain,06/07/2017 17:00
10.1.0.132, some.server,80,tcp,open,http,06/07/2017 17:00
10.1.0.132, some.server,110,tcp,open,pop3,06/07/2017 17:00
10.1.0.132, some.server,113,tcp,closed,ident,06/07/2017 17:00
```

This should be repeated for each type of source file you intend to have Splunk index.

File names. We need a naming convention and we want it to be used in a way that can allow as a wild-cards in searches, reports and dashboards within Splunk later on, we would suggest for vulnerability assessment tools something like **vascan-[tool-name]-[date].csv**. This will allow a search wild-card search like 'vascan*' dst_ip=10.0.0.1 which will show results from all VA scans indexed, or if you know what you want, you can be specific e.g. vascan-nikto for web-server vulnerabilities only, this is especially useful for dashboards which we cover later.

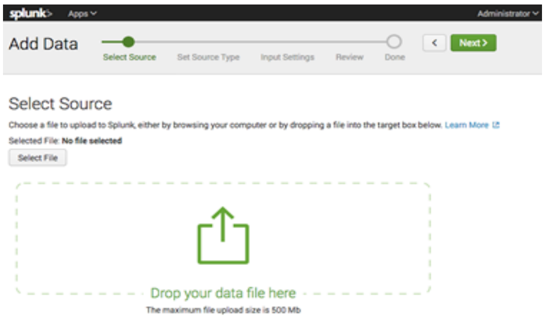
Last step, move or hide the original output file, again you can script this or do it manually, Splunk if configured to automatically check files in folders will try to index all files its sees in the folder (unless you are manually uploading of course).

Step 3. Uploading files into Splunk

The first time you do this, we suggest manually uploading, then configure Splunk to automatically check if needed after. The manual process has wizard to make life easy.

Upload your data files

1. Drag and drop your csv file.



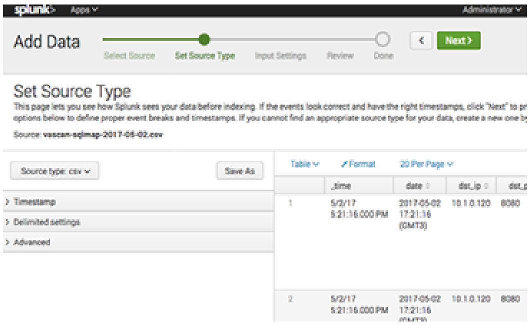
Select Source

Choose a file to upload to Splunk, either by browsing your computer or by dropping a file into the target box below. Learn More

Selected File: **No file selected**

The maximum file upload size is 500 Mb

2. Ensure csv type is selected.



Set Source Type

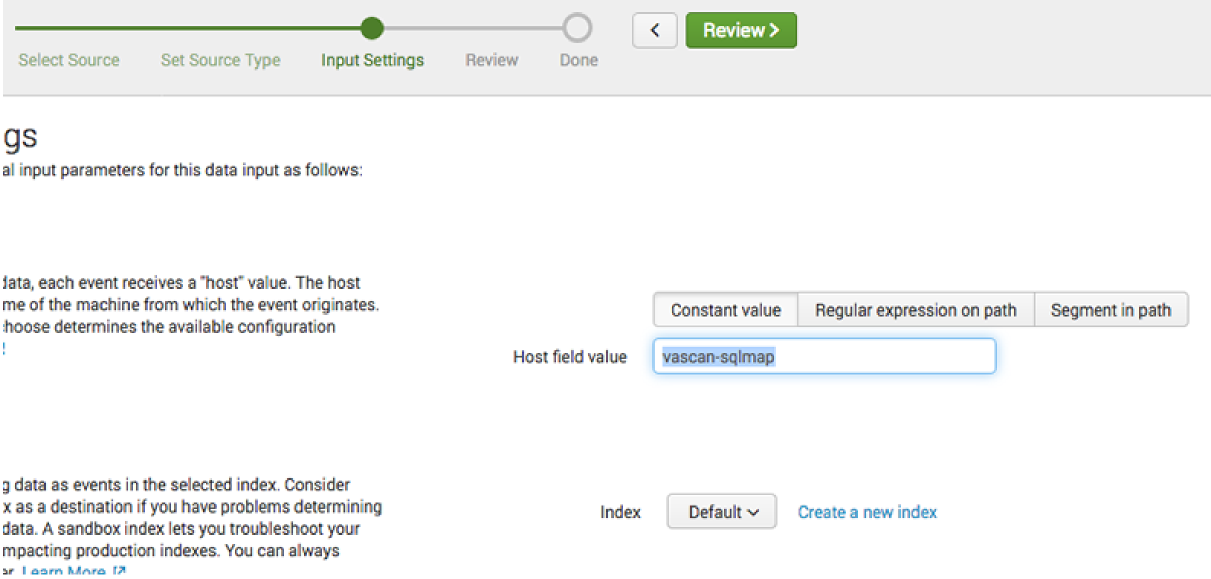
This page lets you see how Splunk sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to pr
options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by

Source: **vascan-sqlmap-2017-05-02.csv**

Source type: csv

	_time	date	dst_ip	dst_p
1	5/2/17 5:21:16.000 PM	2017-05-02 17:21:16 (GMT3)	10.1.0.120	8080
2	5/2/17 5:21:16.000 PM	2017-05-02 17:21:16 (GMT3)	10.1.0.120	8080

3. Specify host, use naming convention 'vascan-[tool]' this way you always know where the data came from and allow analysts to look up specific types of results, below we specify vascan-sqlmap. Complete review and save.



gs

al input parameters for this data input as follows:

fata, each event receives a "host" value. The host me of the machine from which the event originates. hoose determines the available configuration

Constant value | Regular expression on path | Segment in path

Host field value: vascan-sqlmap

g data as events in the selected index. Consider x as a destination if you have problems determining data. A sandbox index lets you troubleshoot your mpacting production indexes. You can always ar Learn More

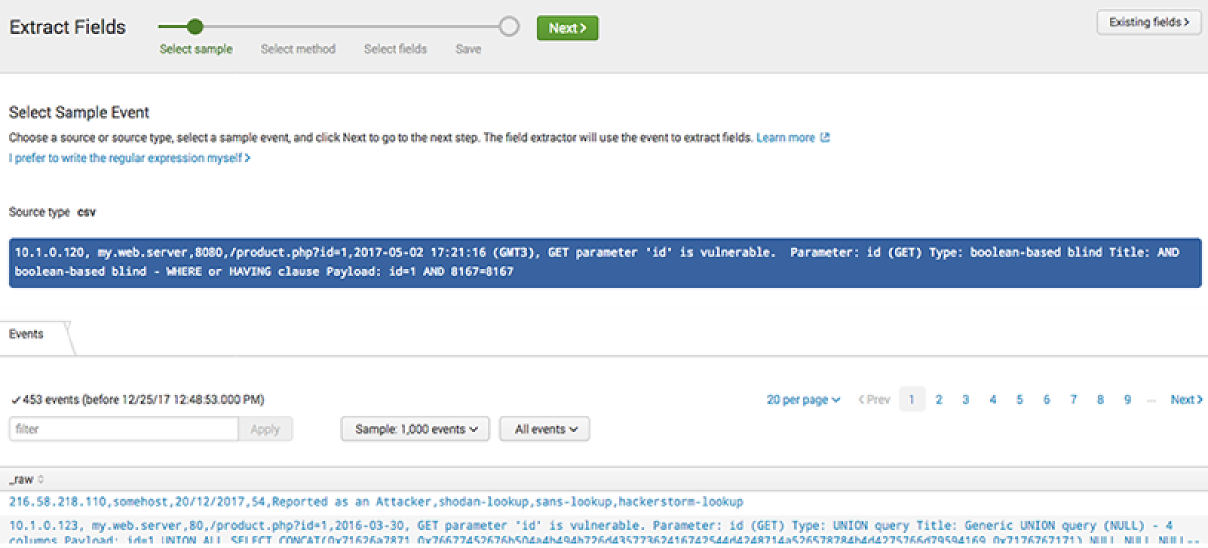
Index: Default

Create a new index

Step 4. Extract Fields using the Splunk wizard.

You need to do this for each type of data source you add, luckily, only the once then you simply add files as you need or want.

1. Select 'Extract Fields' after completing the upload earlier, then select a line from the file just uploaded, use the filter if needed to find a line from your newly uploaded file.



Select Sample Event

Choose a source or source type, select a sample event, and click Next to go to the next step. The field extractor will use the event to extract fields. Learn more

I prefer to write the regular expression myself >

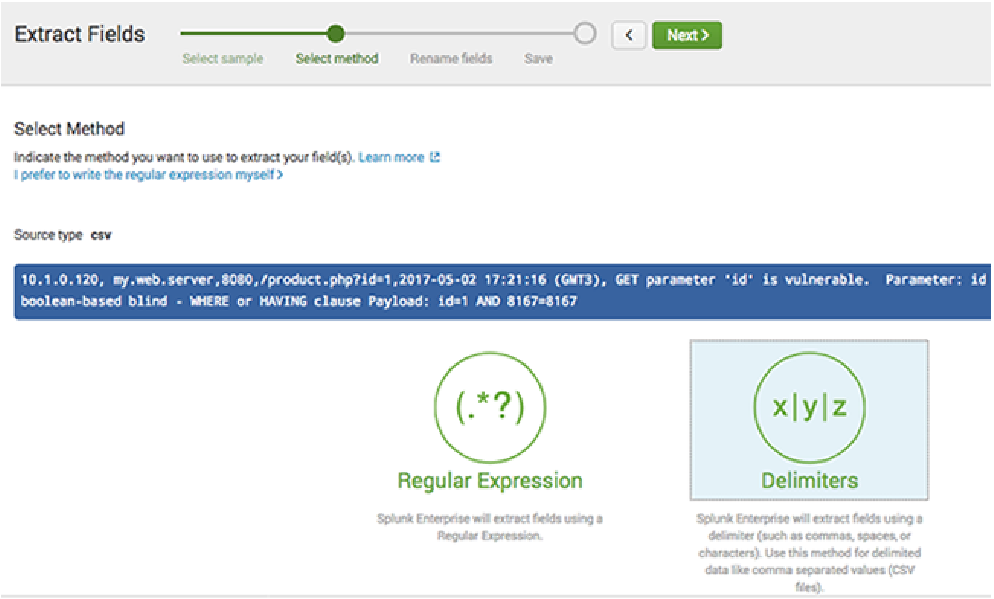
Source type csv

```
10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 (GMT3), GET parameter 'id' is vulnerable.  Parameter: id (GET) Type: boolean-based blind Title: AND
boolean-based blind - WHERE or HAVING clause Payload: id=1 AND 8167=8167
```

✓453 events (before 12/25/17 12:48:53.000 PM)

```
216.58.218.110,somehost,20/12/2017,54,Reported as an Attacker,shodan-lookup,sans-lookup,hackerstorm-lookup
10.1.0.123, my.web.server,80,/product.php?id=1,2016-03-30, GET parameter 'id' is vulnerable. Parameter: id (GET) Type: UNION query Title: Generic UNION query (NULL) - 4
columns Payload: id=1 UNION ALL SELECT CONCAT(0x71626a7871,0x76677452676b504a4b494b726d43577362416742544d4248714a526578784b4d4275766d79594169,0x7176767171),NULL,NULL,NULL--
```

2. Tell Splunk to use 'Delimiters', advanced users can use unstructured data files with regular expressions.



Select Method

Indicate the method you want to use to extract your field(s). Learn more

I prefer to write the regular expression myself >

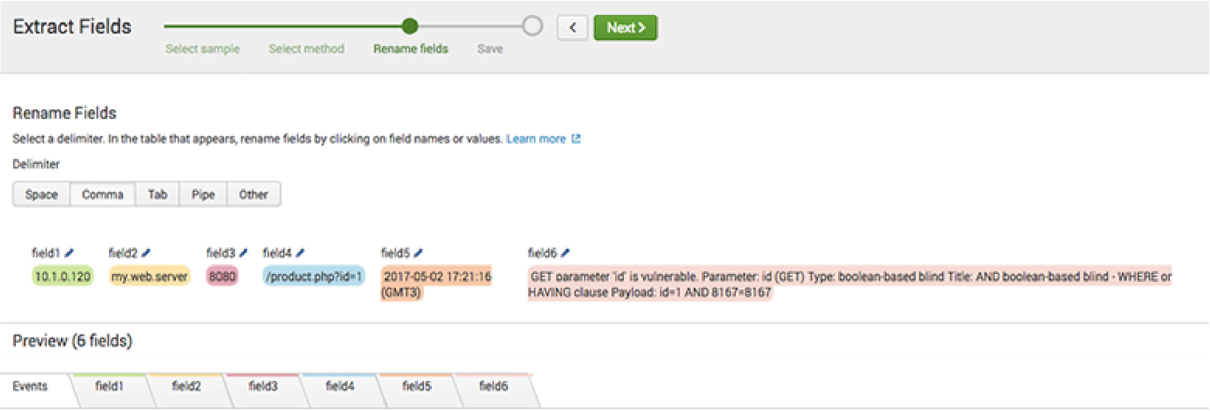
Source type csv

```
10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 (GMT3), GET parameter 'id' is vulnerable.  Parameter: id
boolean-based blind - WHERE or HAVING clause Payload: id=1 AND 8167=8167
```

Splunk Enterprise will extract fields using a Regular Expression.

Splunk Enterprise will extract fields using a delimiter (such as commas, spaces, or characters). Use this method for delimited data like comma separated values (CSV files).

3. Choose comma as it's a csv file separated by commas, Splunk shows you the fields.



Rename Fields

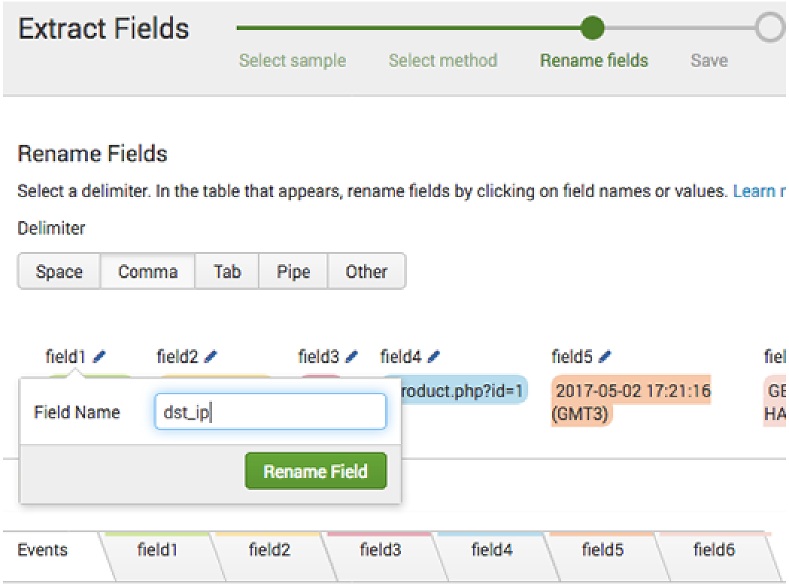
Select a delimiter. In the table that appears, rename fields by clicking on field names or values. Learn more

Delimiter

field1	field2	field3	field4	field5	field6
10.1.0.120	my.web.server	8080	/product.php?id=1	2017-05-02 17:21:16 (GMT3)	GET parameter 'id' is vulnerable. Parameter: id (GET) Type: boolean-based blind Title: AND boolean-based blind - WHERE or HAVING clause Payload: id=1 AND 8167=8167

Preview (6 fields)

4. Rename (or edit) the field names to match you csv column names.



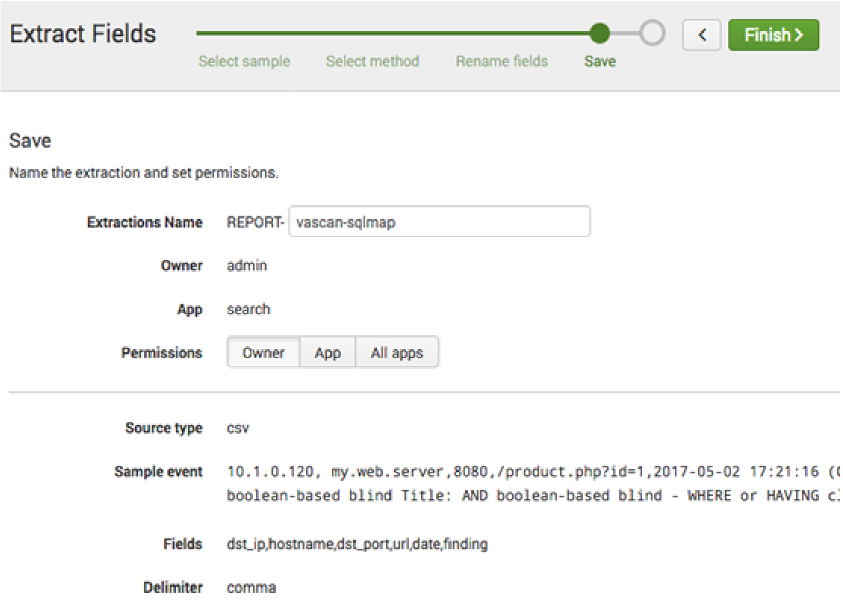
Rename Fields

Select a delimiter. In the table that appears, rename fields by clicking on field names or values. Learn r

Delimiter

Field Name: dst_ip

5. Give the field extract a report name, suggest you stick to naming convention here so for an SQLMap data-source we are calling it similar to the file and host name earlier, vascan-sqlmap. Now your done.



Save

Name the extraction and set permissions.

Extractions Name: REPORT- vascan-sqlmap

Owner: admin

App: search

Permissions: Owner | App | All apps

Source type: csv

Sample event:

```
10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 ((
boolean-based blind Title: AND boolean-based blind - WHERE or HAVING c]
```

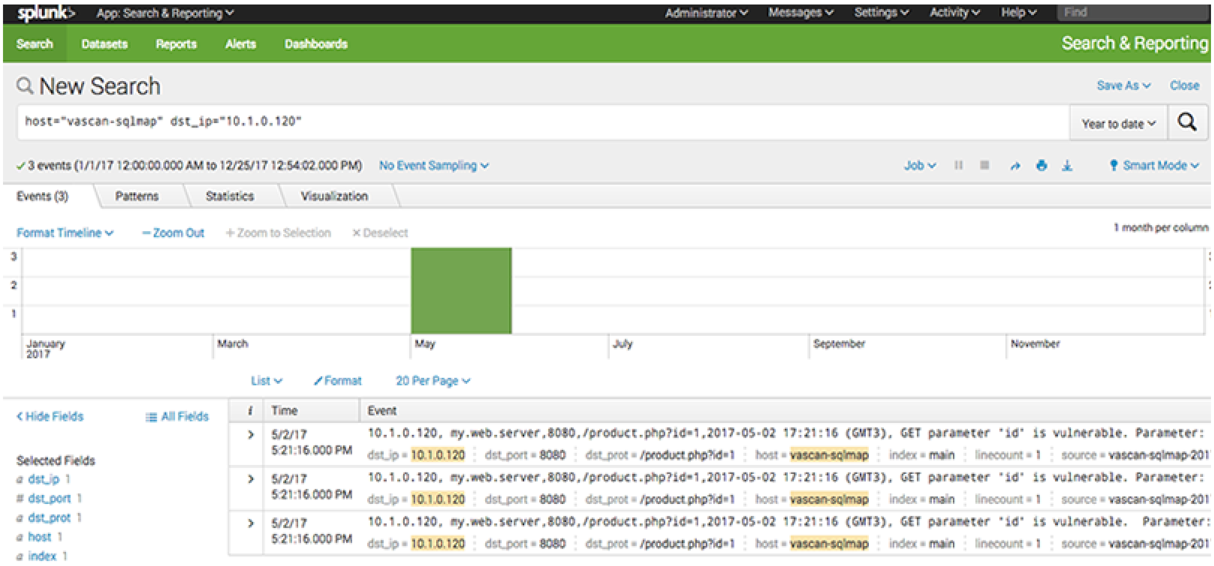
Fields: dst_ip,hostname,dst_port,url,date,finding

Delimiter: comma

Automating Files & Directories Monitoring

Splunk can easily monitor files and directories for you including allowing you to do one-time input and many other clever tricks. The following link will show you how http://docs.Splunk.com/Documentation/Splunk/6.6.2/Data/Monitorfilesanddirectories

Now that the data is indexed and fully researchable, the below image shows we are asking Splunk to show us any SQL findings (host=vascan-sqlmap) for IP 10.1.0.120, and in an instant, we have 3 findings back in May 2017.



```
host="vascan-sqlmap" dst_ip="10.1.0.120"
```

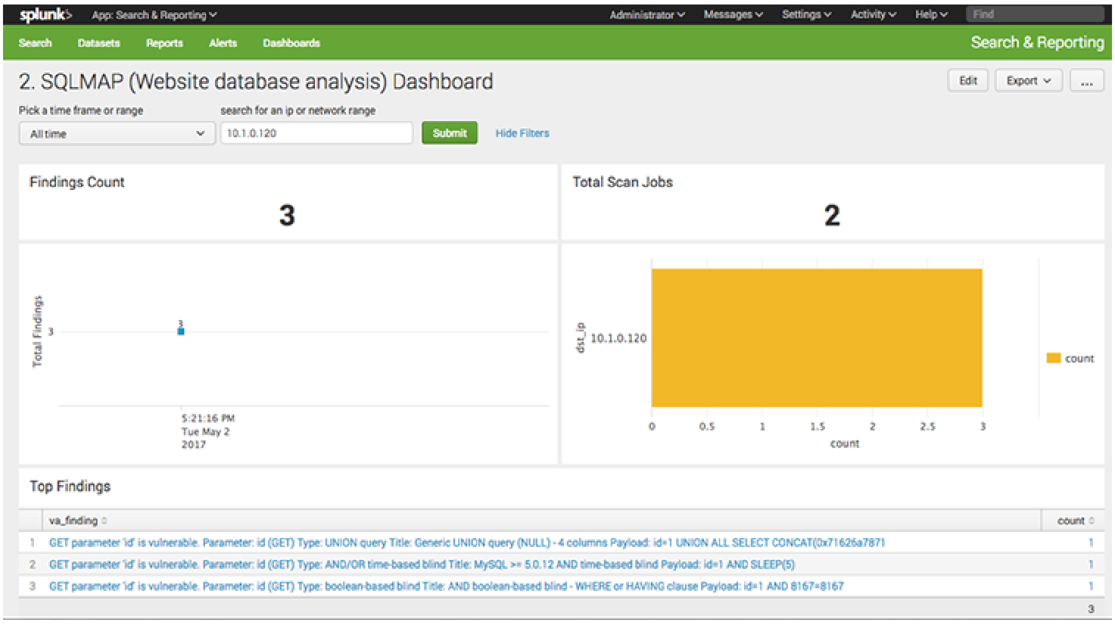
✓ 3 events (1/1/17 12:00:00.000 AM to 12/25/17 12:54:02.000 PM) No Event Sampling

```
i  Time                Event
>  5/2/17              10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 (GMT3), GET parameter 'id' is vulnerable. Parameter:
   5:21:16.000 PM      dst_ip = 10.1.0.120  dst_port = 8080  dst_prot = /product.php?id=1  host = vascan-sqlmap  index = main  linecount = 1  source = vascan-sqlmap-201
>  5/2/17              10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 (GMT3), GET parameter 'id' is vulnerable. Parameter:
   5:21:16.000 PM      dst_ip = 10.1.0.120  dst_port = 8080  dst_prot = /product.php?id=1  host = vascan-sqlmap  index = main  linecount = 1  source = vascan-sqlmap-201
>  5/2/17              10.1.0.120, my.web.server,8080,/product.php?id=1,2017-05-02 17:21:16 (GMT3), GET parameter 'id' is vulnerable.  Parameter:
   5:21:16.000 PM      dst_ip = 10.1.0.120  dst_port = 8080  dst_prot = /product.php?id=1  host = vascan-sqlmap  index = main  linecount = 1  source = vascan-sqlmap-201
```

Step 5. Creating Dashboards

Splunk as ever has made it exceedingly easy to create dashboards. Once you have created one, you can clone them using the built in wizard, or you can copy and paste the source file code, all within the same window in Splunk. Heres a simple dashboard for our SQLMap data source. We added a search filter for IP address where you can type in the IP, if you leave it blank, it will show you results for all IP. Splunk is so versatile, you can specify networks, and if you know your external network ranges from your internal, you can use that to give a view of external and internal threats. The world is your oyster.

We also added a filter for time frame, click the time range and a whole world of time options are available to you to make your searches even quicker.



2. SQLMAP (Website database analysis) Dashboard

Pick a time frame or range: All time

search for an ip or network range: 10.1.0.120

Findings Count: 3

Total Scan Jobs: 2

Top Findings

	va_finding	count
1	GET parameter 'id' is vulnerable. Parameter: id (GET) Type: UNION query Title: Generic UNION query (NULL) - 4 columns Payload: id=1 UNION ALL SELECT CONCAT(0x71626a7871	1
2	GET parameter 'id' is vulnerable. Parameter: id (GET) Type: AND/OR time-based blind Title: MySQL >= 5.0.12 AND time-based blind Payload: id=1 AND SLEEP(5)	1
3	GET parameter 'id' is vulnerable. Parameter: id (GET) Type: boolean-based blind Title: AND boolean-based blind - WHERE or HAVING clause Payload: id=1 AND 8167=8167	1
		3

Heres the UI Source code, you can cut and paste this (assuming you have used our suggested naming conventions e.g. for host vascan-sqlmap and for IP addresses extract field 'dst_ip'

```
<form>
<label>SQLMAP (Website database analysis) Dashboard</label>
<fieldsetautoRun="true" submitButton="true" searchWhenChanged="true">
<input type="time">
<label>Pick a time frame or range</label>
<default>
<earliest>-7d@h</earliest>
<latest>now</latest>
</default>
</input>
<input type="text" token="dst_ip_tok">
<label>search for an ip or network range</label>
<default>*</default>
</input>
</fieldset>
<row>
<panel>
<title>Findings Count</title>
<single>
<search>
<query>host="vascan-sqlmap" dst_ip="$dst_ip_tok$" | stats count</query>
<earliest>$earliest$</earliest>
<latest>$latest$</latest>
</search>
<option name="drilldown">all</option>
<option name="height">50</option>
</single>
```

```
</panel>
<panel>
<title>Total Scan Jobs</title>
<single>
<search>
<query>host=”vascan-sqlmap”  dst_ip=”*” | stats count by host,source | stats count</query>
<earliest>$earliest$</earliest>
<latest>$latest$</latest>
</search>
<option name=”drilldown”>all</option>
<option name=”height”>50</option>
</single>
</panel>
</row>
<row>
<panel>
<chart>
<search>
<query>host=”vascan-sqlmap” dst_ip=”$dst_ip_tok$” | timechart count(_raw)</query>
<earliest>$earliest$</earliest>
<latest>$latest$</latest>
</search>
<option name=”charting.axisTitleX.visibility”>collapsed</option>
<option name=”charting.axisTitleY.text”>Total Findings</option>
<option name=”charting.chart”>column</option>
<option name=”charting.chart.overlayFields”>count(_raw)</option>
<option name=”charting.chart.showDataLabels”>all</option>
<option name=”charting.drilldown”>all</option>
<option name=”charting.legend.placement”>none</option>
</chart>
</panel>
<panel>
<chart>
<search>
<query>host=”vascan-sqlmap” dst_ip=”$dst_ip_tok$” | top dst_ip</query>
<earliest>$earliest$</earliest>
<latest>$latest$</latest>
</search>
<option name=”charting.chart”>bar</option>
<option name=”charting.drilldown”>all</option>
</chart>
</panel>
</row>
<row>
<panel>
<title>Top Findings</title>
<table>
<search>
```

```
<query>host=”vascan-sqlmap” dst_ip=”$dst_ip_tok$” | top limit=50 va_findingshowperc=f</
query>
<earliest>$earliest$</earliest>
<latest>$latest$</latest>
</search>
<option name=”count”>100</option>
<option name=”dataOverlayMode”>none</option>
<option name=”drilldown”>cell</option>
<option name=”percentagesRow”>false</option>
<option name=”rowNumbers”>true</option>
<option name=”totalsRow”>true</option>
<option name=”wrap”>true</option>
</table>
</panel>
</row>
</form>
```

For an overview on Splunk dashboards including creating them, using forms and filters, visit the Splunk website for lots of good advice.

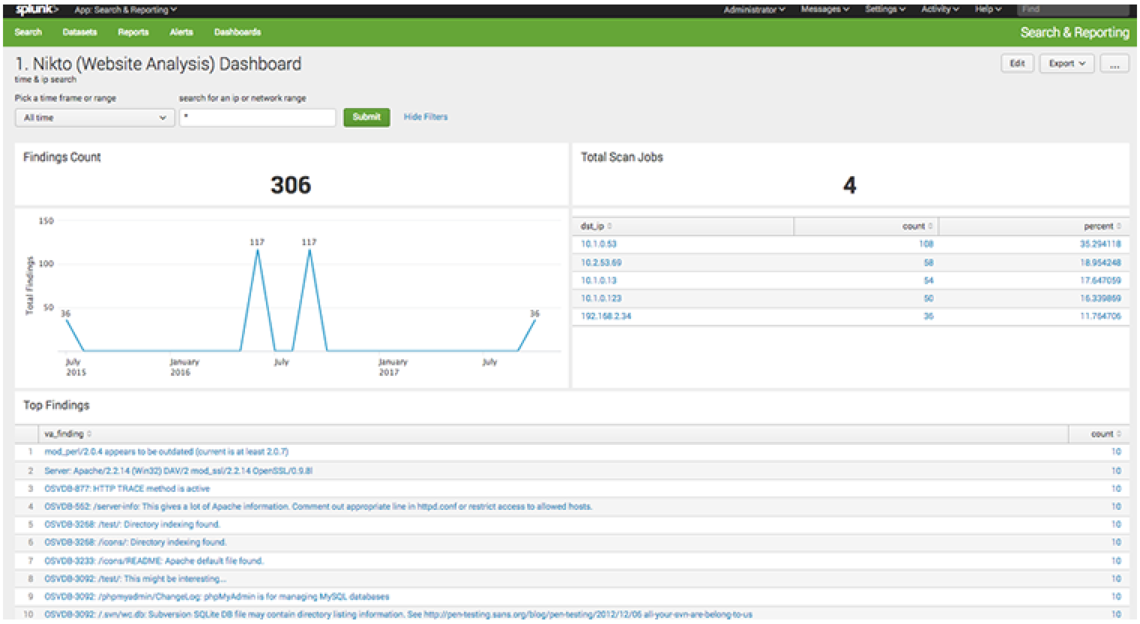
http://docs.Splunk.com/Documentation/Splunk/latest/Viz/Overviewofdashboards

Ideas for Dashboards

Here are some more dashboards we created, the source UI files are all available on our website at hackerstorm.co.uk.

Nikto website Scanner

This dashboard has total findings, how many times scanned (scan jobs), a time-line for the findings and a detailed list of the findings with a ‘time’ and IP address filter to enable searches for the entire history for all devices or one in particular you care about.



1. Nikto (Website Analysis) Dashboard

time & ip search

Pick a time frame or range: All time

search for an ip or network range: *

Findings Count

306

Total Scan Jobs

4

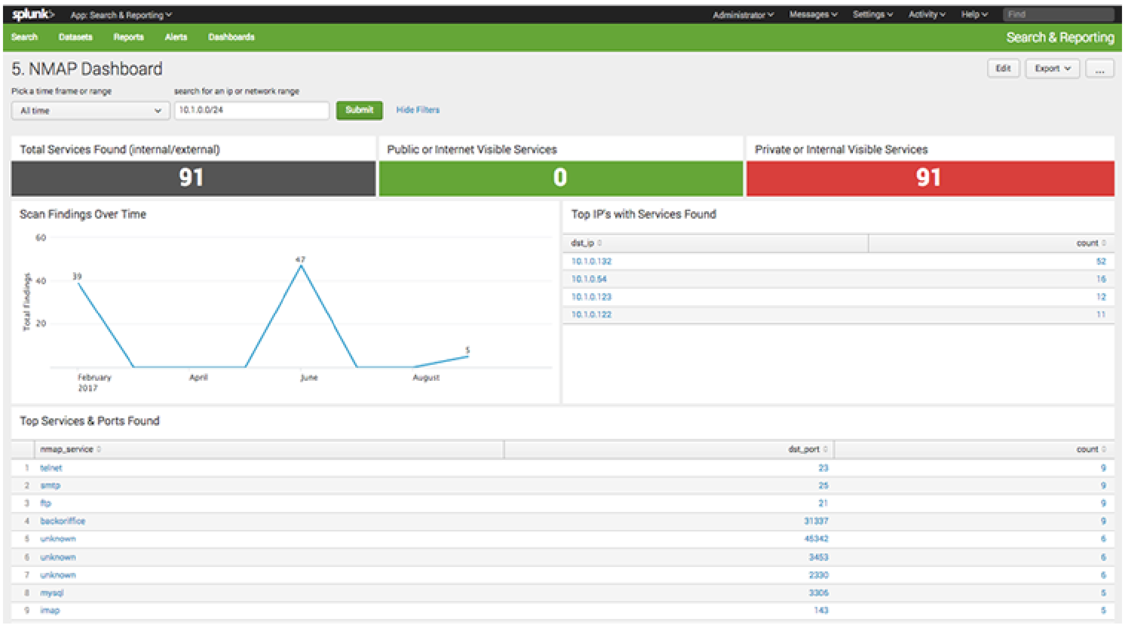
dst_ip	count	percent
10.1.0.53	108	35.294118
10.2.53.69	58	18.954248
10.1.0.13	54	17.647059
10.1.0.123	50	16.339869
192.168.2.34	36	11.764706

Top Findings

	va_finding	count
1	mod_perl/2.0.4 appears to be outdated (current is at least 2.0.7)	10
2	Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l	10
3	OSVDB-877: HTTP TRACE method is active	10
4	OSVDB-552: /server-info: This gives a lot of Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.	10
5	OSVDB-3268: /test/: Directory indexing found.	10
6	OSVDB-3268: /icons/: Directory indexing found.	10
7	OSVDB-3233: /icons/README: Apache default file found.	10
8	OSVDB-3092: /test/: This might be interesting...	10
9	OSVDB-3092: /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases	10
10	OSVDB-3092: /.svn/wc.db: Subversion SQLite DB file may contain directory listing information. See http://pen-testing.sans.org/blog/pen-testing/2012/12/06 all-your-svn-are-belong-to-us	10

NMAP Scanner

Here we have a dashboard similar to Nikto but with colored widgets to indicate public and private visible services. Splunk makes it easy to copy panels (widgets) of other dashboards.



5. NMAP Dashboard

Total Services Found (internal/external)	Public or Internet Visible Services	Private or Internal Visible Services
91	0	91

Scan Findings Over Time

Top IP's with Services Found

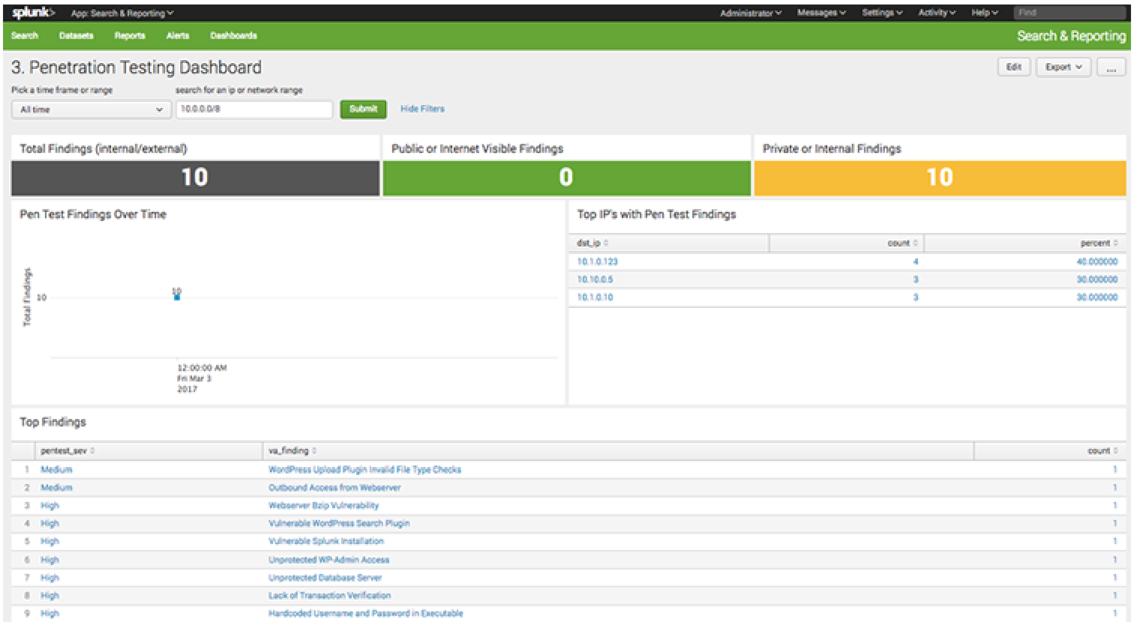
dst_ip	count
10.1.0.132	52
10.1.0.54	16
10.1.0.123	12
10.1.0.122	11

Top Services & Ports Found

	nmap_service	dst_port	count
1	telnet	23	9
2	smtp	25	9
3	ftp	21	9
4	backoriffice	31337	9
5	unknown	45342	6
6	unknown	3453	6
7	unknown	2330	6
8	mysql	3306	5
9	imap	143	5

Penetration Test Results

This dashboard is a clone of the NMAP view, knowing how good or bad something was at a particular point in time in an instant cant be underestimated. The alternative is asking someone for a report and reading through many pages of results, potentially many reports over many years. Not an easy task.



3. Penetration Testing Dashboard

Total Findings (internal/external)	Public or Internet Visible Findings	Private or Internal Findings
10	0	10

Pen Test Findings Over Time

Top IP's with Pen Test Findings

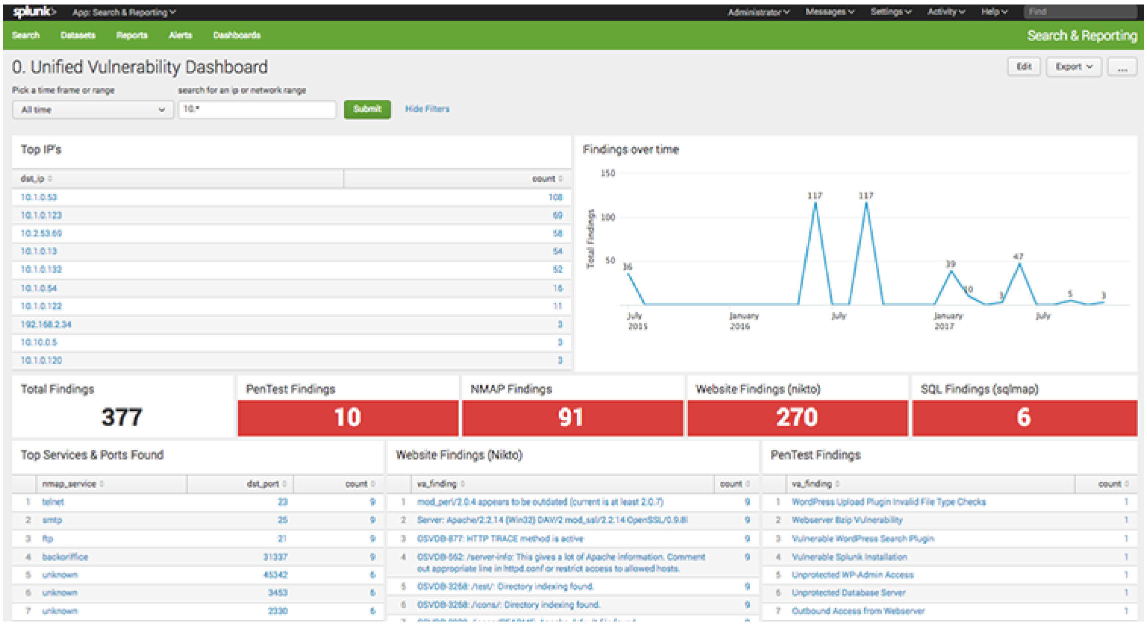
dst_ip	count	percent
10.1.0.123	4	40.000000
10.10.0.5	3	30.000000
10.1.0.10	3	30.000000

Top Findings

	pentest_sev	va_finding	count
1	Medium	WordPress Upload Plugin Invalid File Type Checks	1
2	Medium	Outbound Access from Webserver	1
3	High	Webserver Bzip Vulnerability	1
4	High	Vulnerable WordPress Search Plugin	1
5	High	Vulnerable Splunk Installation	1
6	High	Unprotected WP-Admin Access	1
7	High	Unprotected Database Server	1
8	High	Lack of Transaction Verification	1
9	High	Hardcoded Username and Password in Executable	1

Unified Dashboard combining all data sources

Here we have combined all four data sources into one dashboard. You can view results for entire networks, subnet, specific hosts, public or private networks or hosts all on one page for any time frame you need. This is gold for an analyst trying to understand what went wrong, when, how and what sort of condition something was in at any given point in time.



0. Unified Vulnerability Dashboard

Pick a time frame or range: All time

search for an ip or network range: 10.*

Top IP's

dst_ip	count
10.1.0.53	108
10.1.0.123	69
10.2.53.69	58
10.1.0.13	54
10.1.0.132	52
10.1.0.54	16
10.1.0.122	11
192.168.2.34	3
10.10.0.5	3
10.1.0.120	3

Total Findings	PenTest Findings	NMAP Findings	Website Findings (nikto)	SQL Findings (sqlmap)
377	10	91	270	6

Top Services & Ports Found

	nmap_service	dst_port	count
1	telnet	23	9
2	smtp	25	9
3	ftp	21	9
4	backorifice	31337	9
5	unknown	45342	6
6	unknown	3453	6
7	unknown	2330	6

Website Findings (Nikto)

	va_finding	count
1	mod_perl/2.0.4 appears to be outdated (current is at least 2.0.7)	9
2	Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l	9
3	OSVDB-877: HTTP TRACE method is active	9
4	OSVDB-562: /server-info: This gives a lot of Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.	9
5	OSVDB-3268: /test/: Directory indexing found.	9
6	OSVDB-3268: /icons/: Directory indexing found.	9

PenTest Findings

	va_finding	count
1	WordPress Upload Plugin Invalid File Type Checks	1
2	Webserver Bzip Vulnerability	1
3	Vulnerable WordPress Search Plugin	1
4	Vulnerable Splunk Installation	1
5	Unprotected WP-Admin Access	1
6	Unprotected Database Server	1
7	Outbound Access from Webserver	1

You can also clone the dashboard and change it to only show public facing servers for example or create 'service' specific dashboards, tailor it to your organization setup, needs and risks.

More dashboard Ideas

Risk based dashboards. Create dashboards for risk and compliance managers showing events mapped back to corporate risks. If the corporate risk register has a risk for e.g. PII Data loss , a dashboard can be created to summarize all relevant events for hosts storing or processing PII data.

Combine security events with vulnerability and penetration test findings. Add events from networks, access control, configuration compliance, anti-malware, patch management compliance, integrity checking, SIEM etc. Why look at what problems existed historically when you can combine event based behaviors alongside the weaknesses for a truly complete view.

Open-source Intelligence. One idea that you may also be very interested in is using Open-source intelligence. You can save the results, convert them to csv and begin capturing data about yourself from places like Shodan, SANs Storm Centre as well as other API enabled reputation checkers. If you automate it, you can get a regular view over time and have historical reputation data for any of your publicly visible hosts, something that would otherwise be a time consuming task, here we have a way to be informed in an instant and generate reports in seconds.

References

Splunk Guides *http://docs.Splunk.com/Documentation*

HackerStorm Dashboards UI source codes *http://www.hackerstorm.co.uk/home*

Kali (Offensive Security) *https://www.kali.org*

NMAP *https://nmap.org*

Nikto *https://cirt.net/Nikto2*

SQLMap *http://sqlmap.org/*

IBM *http://info.resilientsystems.com/hubfs/IBM_Resilient_Branded_Content/White_Papers/2017_Global_CODB_Report_Final.pdf*

NCSC *https://www.ncsc.gov.uk/scheme/cyber-incidents*

NIST *https://www.nist.gov/sites/default/files/documents/cyberframework/cybersecurity-framework-021214.pdf*

OWASP *https://www.owasp.org/index.php/Application_Threat_Modeling*

HACKERSTORM

Thinking outside the cage

Threat, Hacking & Vulnerability Research
News, Alerts and Advisories
Reputation checks and reports

Copyright © 2018 hackerstorm.co.uk