Weapon System Training - 2

LAB 3: CREATING AN INVESTIGATION DASHBOARD

Maj Michael Lester and Capt Jon Bynum
32 WPS/DOA | NELLIS AFB, NEVADA

Weapon System Training – 2: Creating an Investigation Dashboard

CONTENTS

Symbols Table

Symbol	Name	Meaning
✅	**Note**	Detailed information that is required to fully understanding the concept or to be able to execute a procedure but is not necessarily related to a key learning objective.
💡	**Learning Point**	Information related to key learning objectives.
⚠	**Warning**	Important information related to safety and security.
✋	**Raise Hand**	Raise your hand for instructor assistance. This is often used at critical points to validate your understanding of the material.

LAB 3: CREATING AN INVESTIGATION DASHBOARD

OVERVIEW

Summary: The purpose of this lab is to create an investigation dashboard that can be used to gather context about what events took place in and around the time of a given suspicious event. The purpose of this dashboard is **NOT** to be a starting location to find malicious activity, but rather a next step to validate whether a suspicious event or an alert is related to malicious activity.

Outcomes: By the end of the lab, you will be able to perform the following:

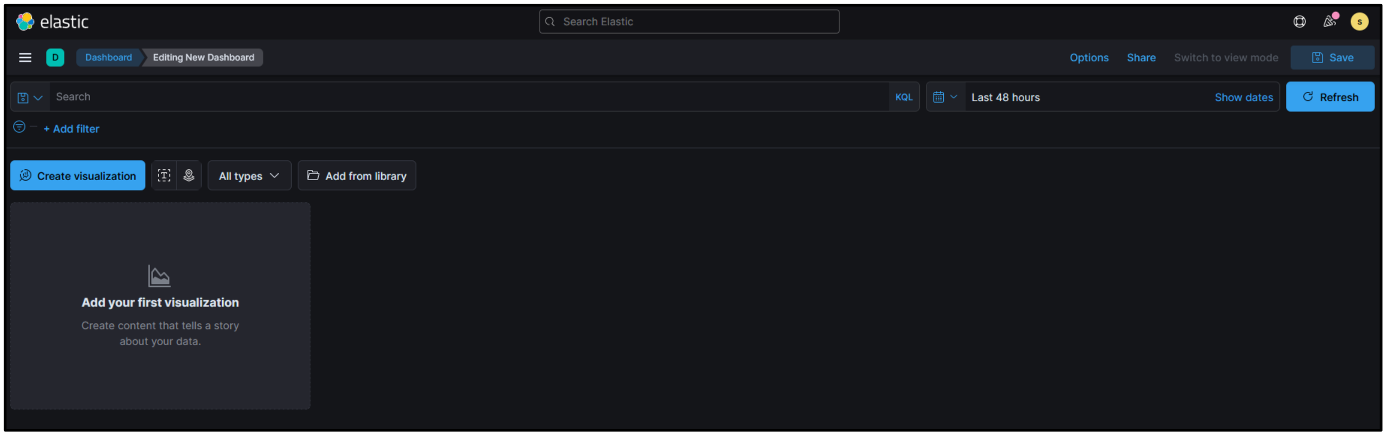
- Build summary statistic visualizations.
- Build visualizations that show events over time.
- Build a visualization that shows data from multiple types of events.
- Create links that open secondary dashboards and pass parameters.

PROCEDURES

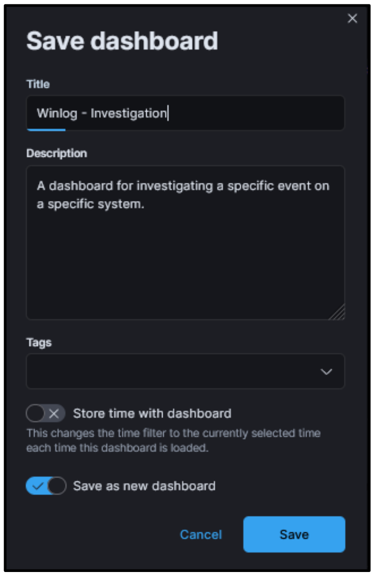
STEP 1. CREATE AN INVESTIGATION DASHBOARD

We're going to start off by creating an Investigation dashboard, then we'll add more and more visualizations to it in order to investigate a specific event.

1. Start by navigating to **Main Menu → Analytics → Dashboard**.
2. Click on the bright blue "Create dashboard" button.



3. Start by saving the Dashboard so that we can find it and return to it later. Click on the "Save" button in the top, right-hand corner of the screen. Name the new dashboard "Winlog – Investigation". If multiple students are using the same range, add your last name to the dashboard name to make it unique. Give it a short description, then click the "Save" button at the bottom of the screen.


Save dashboard

Title: Winlog - Investigation

Description: A dashboard for investigating a specific event on a specific system.

Tags

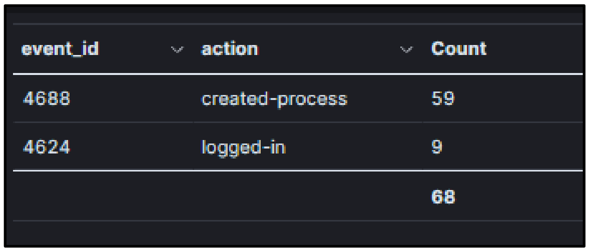
○ Store time with dashboard
This changes the time filter to the currently selected time each time this dashboard is loaded.

● Save as new dashboard

STEP 2. CREATE VISUALIZATION OF EVENT IDS

The first visualization that we are going to add is a summary of event types that are taking place, given the time frame and filters applied to the current view. This gives us a high level idea of what events and how many took place (e.g. process creation, scheduled task creation, service creation, user logon, PowerShell command execution, etc.). It also gives us a visual panel where we can apply quick filters by clicking on specific values. For example, if we want to only see process creation events, we could click on the 4688 event which would add a quick filter to the entire dashboard.

1. From the “Winlog – Investigation” dashboard, click on the “All types” drop down and select **Aggregation based -> Data table**.
2. Select the “so-beats*” index pattern.
3. Under “Buckets” in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   3.1. **Aggregation:** Terms
   3.2. **Field:** winlog.event_id
   3.3. **Order by:** Metric: Count
   3.4. **Order:** Descending
   3.5. **Size:** 1000
   3.6. **Group other values in separate bucket:** Checked
   3.7. **Custom label:** event_id
4. Under “Buckets” in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   4.1. **Aggregation:** Terms
   4.2. **Field:** winlog.event_action
   4.3. **Order by:** Metric: Count
   4.4. **Order:** Descending
   4.5. **Size:** 1000
   4.6. **Group other values in separate bucket:** Checked
   4.7. **Custom label:** action
5. Click on the blue “Update” button in the bottom, right-hand corner of the screen and verify that the visualization looks like the image below:



event_id	action	Count
4688	created-process	59
4624	logged-in	9
		68

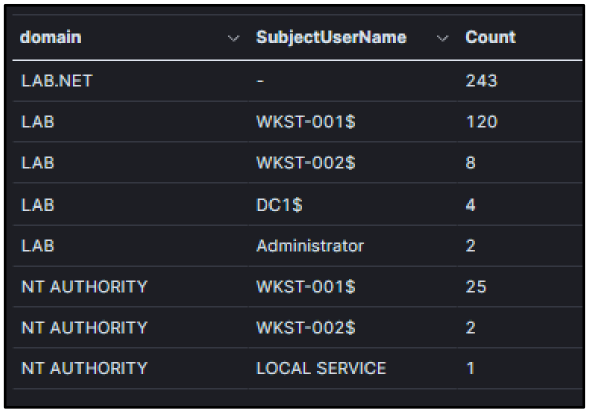
6. Click on the blue “Save to library” button in the top, right-hand corner of the screen.
   6.1. **Title:** Winlog – Event Id Summary
   6.2. **Add to Dashboard after saving:** Checked
7. Click on the blue “Save and return” button.

✅**NOTE: We purposefully did not create a filter to only show Event Id 4688 events like we did in previous visualizations, because this visualization needs to pull from multiple event types.**

STEP 3. CREATE VISUALIZATION OF SUBJECTUSERNAME

The next visualization is going to be a summary of usernames that generated events. This is a useful visualization so that we can see what users were active on the system in and around the time of a suspicious event. It also gives us another panel to create quick filters from. For example, we may not care about events created by NT AUTHORITY\SYSTEM if the suspicious event came from a standard user. We can easily filter out the background noise by removing events generated by SYSTEM. We also want to know what domain the user account is tied to or if it is a local account. We'll add a column for that information as well.

1. From the "Winlog – Investigation" dashboard, click on the "All types" drop down and select **Aggregation based -> Data table**.
2. Select the "so-beats*" index pattern.
3. Under "Buckets" in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   3.1. **Aggregation:** Terms
   3.2. **Field:** user.domain.keyword
   3.3. **Order by:** Metric: Count
   3.4. **Order:** Descending
   3.5. **Size:** 9
   3.6. **Group other values in separate bucket:** Checked
   3.7. **Custom label:** domain
4. Under "Buckets" in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   4.1. **Aggregation:** Terms
   4.2. **Field:** winlog.event_data.SubjectUserName
   4.3. **Order by:** Metric: Count
   4.4. **Order:** Descending
   4.5. **Size:** 9
   4.6. **Group other values in separate bucket:** Checked
   4.7. **Custom label:** domain
5. Click on the blue "Update" button in the bottom, right-hand corner of the screen and verify that the visualization looks like the image below:



domain	SubjectUserName	Count
LAB.NET	-	243
LAB	WKST-001$	120
LAB	WKST-002$	8
LAB	DC1$	4
LAB	Administrator	2
NT AUTHORITY	WKST-001$	25
NT AUTHORITY	WKST-002$	2
NT AUTHORITY	LOCAL SERVICE	1

✅**NOTE: Some events don't have a username associated with them because most of the Windows event logs are not standardized. This is important to keep in mind when filtering events. When you apply a filter for SubjectUserName, you will eliminate events that don't have a username even though those events may have been generated as a result of actions by that user.**

6. Click on the blue "Save to library" button in the top, right-hand corner of the screen.
   6.1. **Title:** Winlog – SubjectUserName Summary
   6.2. **Add to Dashboard after saving:** Checked
7. Click on the blue "Save and return" button.

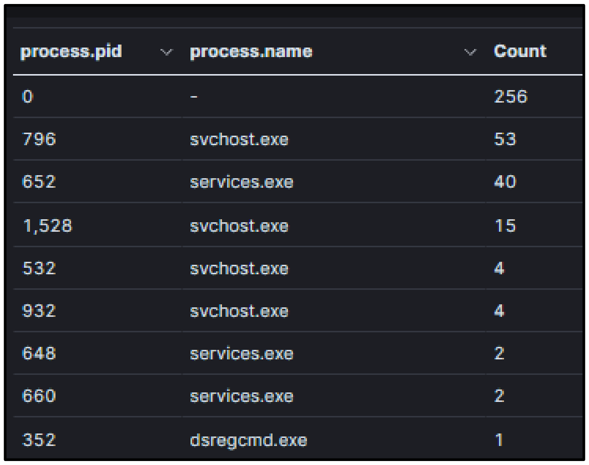
STEP 4. CREATE VISUALIZATION OF PROCESSES

Next, we want a summary of what processes created events. This is helpful to get a quick idea of what was happening on the system, but it is also helpful for creating some quick filters. Additionally, not all Windows events are associated with a process name or path, many are associated with a PID. It may be interesting to filter down the dashboard to only show events that were generated by a particular PID.

1. From the "Winlog – Investigation" dashboard, click on the "All types" drop down and select **Aggregation based -> Data table**.
2. Select the "so-beats*" index pattern.
3. Under "Buckets" in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   3.1. **Aggregation:** Terms
   3.2. **Field:** process.pid
   3.3. **Order by:** Metric: Count
   3.4. **Order:** Descending
   3.5. **Size:** 1000
   3.6. **Group other values in separate bucket:** Checked
   3.7. **Custom label:** process.pid
4. Under "Buckets" in the panel on the right-hand side of the screen, click **Add -> Split rows**. Configure bucket with the following options:
   4.1. **Aggregation:** Terms

4.2. **Field:** process.name
4.3. **Order by:** Metric: Count
4.4. **Order:** Descending
4.5. **Size:** 1000
4.6. **Group other values in separate bucket:** Checked
4.7. **Custom label:** process.name

5. Click on the blue “Update” button in the bottom, right-hand corner of the screen and verify that the visualization looks like the image below:



process.pid	process.name	Count
0	-	256
796	svchost.exe	53
652	services.exe	40
1,528	svchost.exe	15
532	svchost.exe	4
932	svchost.exe	4
648	services.exe	2
660	services.exe	2
352	dsregcmd.exe	1

✅**NOTE: Just like before, not all events have a PID. If you filter by PID or process name, you will lose context of some events.**

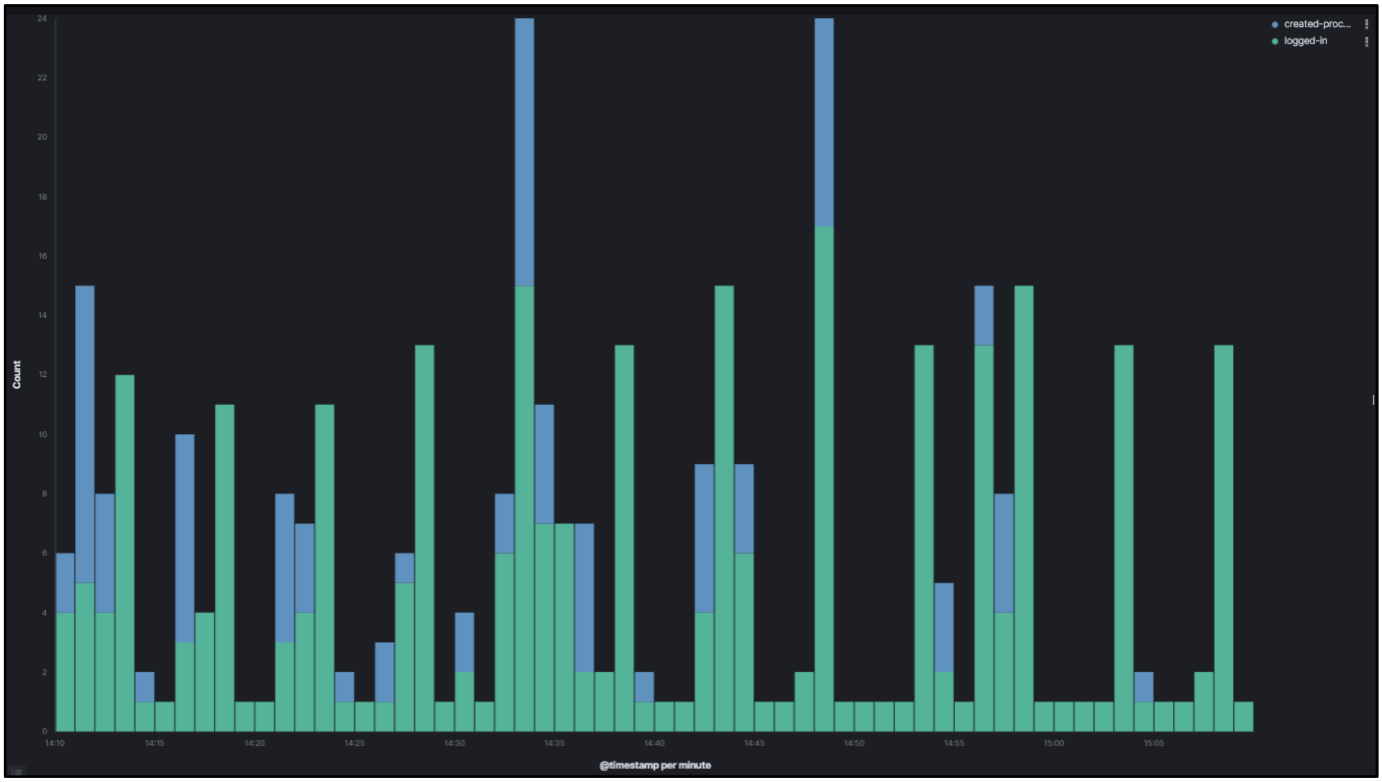
6. Click on the blue “Save to library” button in the top, right-hand corner of the screen.
   6.1. **Title:** Winlog – PID Summary
   6.2. **Add to Dashboard after saving:** Checked
7. Click on the blue “Save and return” button.

STEP 5. CREATE VISUALIZATION OF EVENT TYPES OVER TIME

The next visualization is a timeline chart with a stacked bar graph that will show us the volume of events by type over time. We can use this to show spikes in the event timeline such as when a threat actor runs a host enumeration script or brute force logon attempts.

1. From the “Winlog – Investigation” dashboard, click on the “All types” drop down and select **Aggregation based -> Vertical bar**.
2. Select the “so-beats*” index pattern.
3. Under “Buckets” in the panel on the right-hand side of the screen, click **Add -> X-axis**. Configure bucket with the following options:
   3.1. **Aggregation:** Date Histogram
   3.2. **Field:** @timestamp
   3.3. **Minimum interval:** Minute

4. Under “Buckets” in the panel on the right-hand side of the screen, click **Add -> Split series**. Configure bucket with the following options:
   4.1. **Aggregation:** Terms
   4.2. **Field:** event.action.keyword
   4.3. **Order by:** Metric: Count
   4.4. **Order:** Descending
   4.5. **Size:** 10
   4.6. **Group other values in separate bucket:** Checked
5. Click on the blue “Update” button in the bottom, right-hand corner of the screen and verify that the visualization looks like the image below:

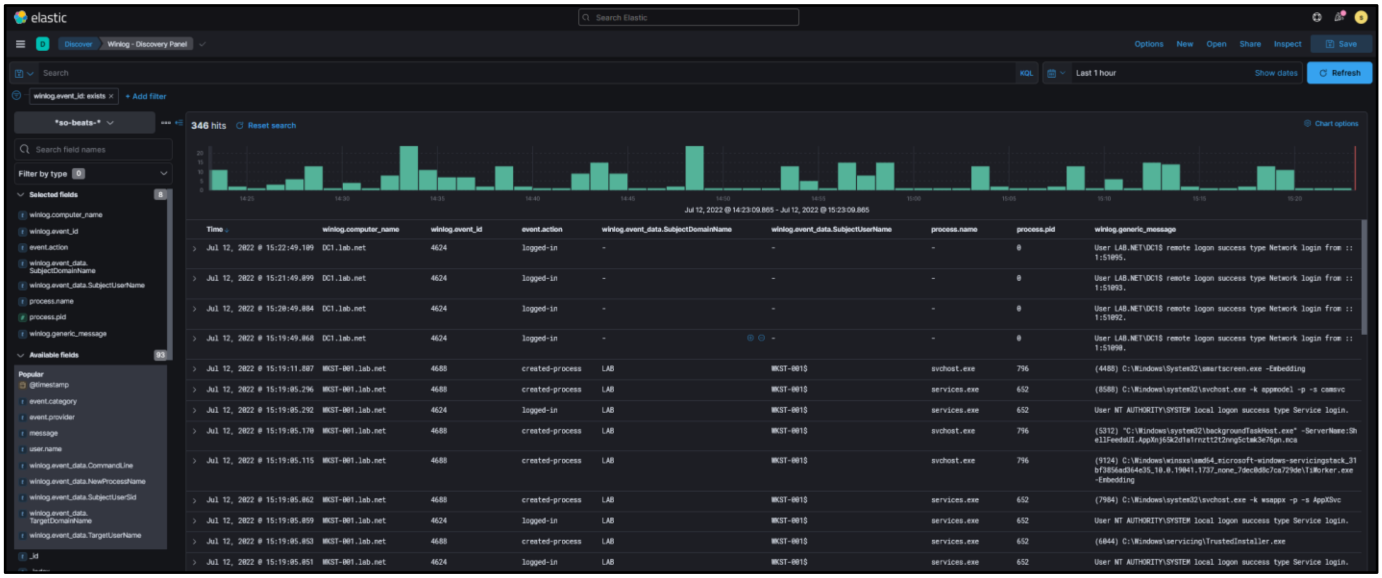


6. Click on the blue “Save to library” button in the top, right-hand corner of the screen.
   6.1. **Title:** Winlog - Bar Chart of Event Types over Time
   6.2. **Add to Dashboard after saving:** Checked
7. Click on the blue “Save and return” button.

STEP 6. CREATE VISUALIZATION OF EVENT DETAILS

The next visualization we want is a detailed table of events that you can easily expand to view all of the fields of that particular event. We want this visualization to take up the entire width of the screen in order to show us much information in as dense a format as possible. We want to be able to visualize a 1 to n list of all events with a summary column that explains what happened in each event (winlog.generic_message). This will help us to quickly see what was happening throughout time on this system.

1. Start by expanding the three bars in the top, left-hand corner of the screen and navigating to **Analytics -> Discover**.
2. First, we are going to configure a filter so that this search only shows Windows event log events.
   2.1. Click on the blue "+ Add filter" button in the top, left-hand corner of the screen.
   2.2. **Field:** winlog.event_id.keyword
   2.3. **Operator:** exists
3. Next, use the "Search field names" box in the left-hand side of the screen to find the following events and add them to the table by clicking on the + symbol that appears when mousing over each field:
   3.1. winlog.computer_name
   3.2. winlog.event_id
   3.3. event.action
   3.4. winlog.event_data.SubjectDomainName
   3.5. winlog.event_data.SubjectUserName
   3.6. process.name
   3.7. process.pid
   3.8. winlog.generic_message
4. You should see something similar to the following:

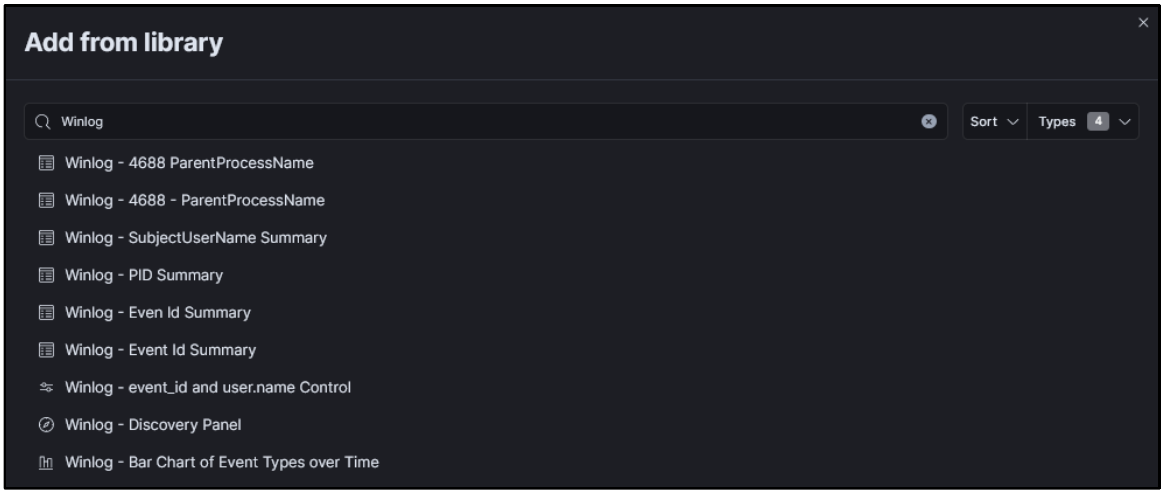


Time	winlog.computer_name	winlog.event_id	event.action	winlog.event_data.SubjectDomainName	winlog.event_data.SubjectUserName	process.name	process.pid	winlog.generic_message
Jul 12, 2022 @ 15:22:49.109	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from ::1:51095.
Jul 12, 2022 @ 15:21:49.099	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from ::1:51093.
Jul 12, 2022 @ 15:20:49.084	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from ::1:51092.
Jul 12, 2022 @ 15:19:49.068	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from ::1:51090.
Jul 12, 2022 @ 15:19:11.807	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	796	(4488) C:\Windows\System32\smartscreen.exe -Embedding
Jul 12, 2022 @ 15:19:05.296	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(8588) C:\Windows\system32\svchost.exe -k appmodel -p -s camsvc
Jul 12, 2022 @ 15:19:05.292	WKST-001.lab.net	4624	logged-in	LAB	WKST-001$	services.exe	652	User NT AUTHORITY\SYSTEM local logon success type Service login.
Jul 12, 2022 @ 15:19:05.170	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	796	(5312) "C:\Windows\system32\backgroundTaskHost.exe" -ServerName:ShellFeedsUI.AppXnj65k2d1a1rnztt2t2nng5ctmk3e76pn.mca
Jul 12, 2022 @ 15:19:05.115	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	796	(9124) C:\Windows\winsxs\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.1737_none_7dec0d8c7ca729de\TiWorker.exe -Embedding
Jul 12, 2022 @ 15:19:05.062	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(7984) C:\Windows\system32\svchost.exe -k wsappx -p -s AppXSvc
Jul 12, 2022 @ 15:19:05.059	WKST-001.lab.net	4624	logged-in	LAB	WKST-001$	services.exe	652	User NT AUTHORITY\SYSTEM local logon success type Service login.
Jul 12, 2022 @ 15:19:05.053	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(6044) C:\Windows\servicing\TrustedInstaller.exe
Jul 12, 2022 @ 15:19:05.051	WKST-001.lab.net	4624	logged-in	LAB	WKST-001$	services.exe	652	User NT AUTHORITY\SYSTEM local logon success type Service login.

5. We want to shorten the column titles for some of the fields in order to make more room for the "generic_message" field at the end. Click on the following fields under the "Selected fields" list on the left-hand side of the Discovery page, then click on the "pencil" icon in the top, right-hand side of the popup:
   5.1. **Winlog.event_data.SubjectDomainName:**
      5.1.1. **Set custom label:** Checked
      5.1.2. **Custom label:** Domain
   5.2. **Winlog.event_data.SubjectUserName:**

5.2.1. **Set custom label:** Checked
5.2.2. **Custom label:** Username

6. Click on the blue “Save” button in the top, right-hand corner of the screen.
   6.1. **Title:** Winlog – Discovery Panel
7. Click on the blue “Save” button.
8. Navigate back to the “Winlog – Investigation” dashboard.
9. Click on the “add from library” button.
10. Type “Winlog” into the search box and then click on “Winlog – Discovery Panel”.

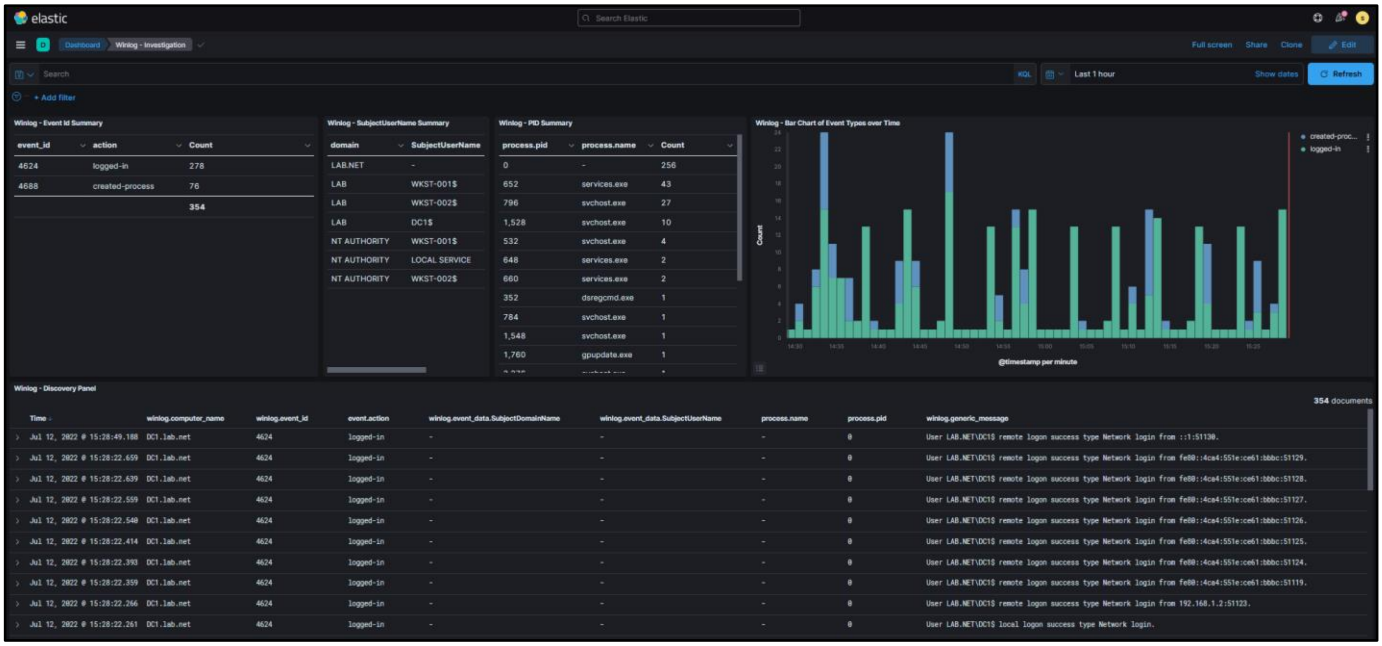


Add from library

- Winlog - 4688 ParentProcessName
- Winlog - 4688 - ParentProcessName
- Winlog - SubjectUserName Summary
- Winlog - PID Summary
- Winlog - Even Id Summary
- Winlog - Event Id Summary
- Winlog - event_id and user.name Control
- Winlog - Discovery Panel
- Winlog - Bar Chart of Event Types over Time

STEP 7. ARRANGE THE DASHBOARD

Next, arrange the visualizations we just created until the dashboard looks like the one below.



Winlog - Event Id Summary

event_id	action	Count
4624	logged-in	278
4688	created-process	76
		354

Winlog - SubjectUserName Summary

domain	SubjectUserName
LAB.NET	-
LAB	WKST-001$
LAB	WKST-002$
LAB	DC1$
NT AUTHORITY	WKST-001$
NT AUTHORITY	LOCAL SERVICE
NT AUTHORITY	WKST-002$

Winlog - PID Summary

process.pid	process.name	Count
0	-	256
652	services.exe	43
796	svchost.exe	27
1,528	svchost.exe	10
532	svchost.exe	4
648	services.exe	2
660	services.exe	2
352	dsregcmd.exe	1
784	svchost.exe	1
1,548	svchost.exe	1
1,760	gpupdate.exe	1

Winlog - Discovery Panel

354 documents

Time	winlog.computer_name	winlog.event_id	event.action	winlog.event_data.SubjectDomainName	winlog.event_data.SubjectUserName	process.name	process.pid	winlog.generic_message
Jul 12, 2022 @ 15:28:49.188	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from ::1:51130.
Jul 12, 2022 @ 15:28:22.659	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51129.
Jul 12, 2022 @ 15:28:22.639	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51128.
Jul 12, 2022 @ 15:28:22.559	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51127.
Jul 12, 2022 @ 15:28:22.540	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51126.
Jul 12, 2022 @ 15:28:22.414	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51125.
Jul 12, 2022 @ 15:28:22.393	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51124.
Jul 12, 2022 @ 15:28:22.359	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from fe80::4ca4:551e:ce61:bbbc:51119.
Jul 12, 2022 @ 15:28:22.266	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ remote logon success type Network login from 192.168.1.2:51123.
Jul 12, 2022 @ 15:28:22.261	DC1.lab.net	4624	logged-in	-	-	-	0	User LAB.NET\DC1$ local logon success type Network login.

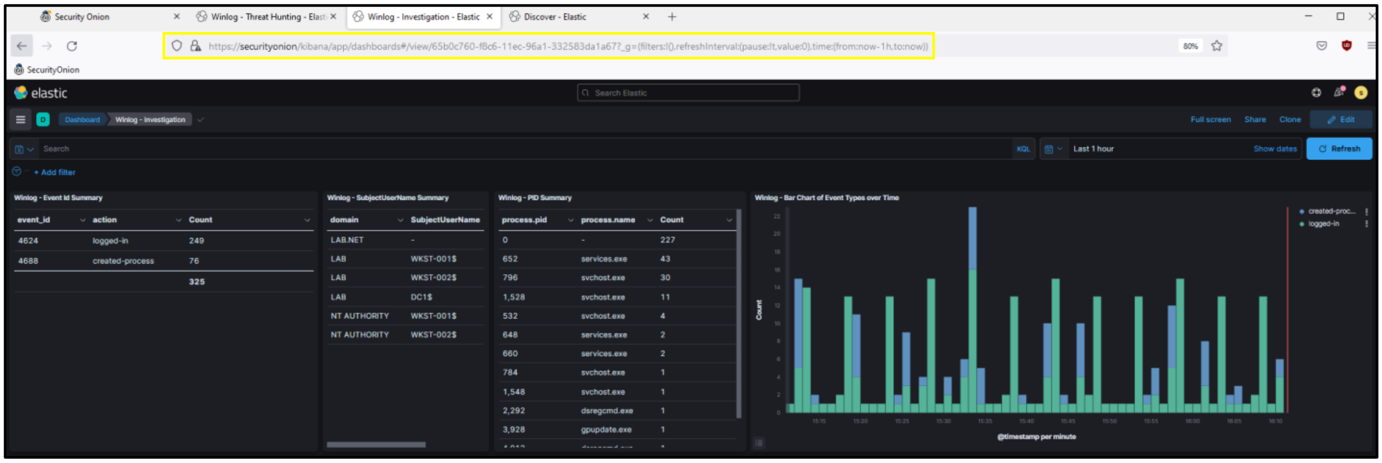
💡**LEARNING POINT: I prefer to setup my dashboards with high level summary statistics, visualizations, filters, and analytics at the top of the dashboard with a table of detailed events towards the bottom. As I apply more filters and queries at the top, the changes cascade down towards the details at the bottom.**

STEP 8. CREATE A URL DRILL DOWN IN THE THREAT HUNTING DASHBOARD

Now that we have successfully created our threat hunting dashboard, we are going to create a URL Drill Down from the "Winlog – Threat Hunting" dashboard that will link to the "Winlog – Investigation" dashboard. Essentially, we want to click on an event in the "Winlog – Threat Hunting" dashboard and navigate to the "Winlog – Investigation" dashboard with filters automatically applied that only show events that occurred in and around the time of the source event on the system the source event originated.

✅**NOTE: URL Drill Down events require a license; however, we can get around that temporarily by using a 30-day trial license, which can be enabled from the License Management page located at Management -> Index Management -> Stack -> License Management.**

1. First, we need to get the base URL to the "Winlog – Investigation" dashboard we just created. Start by navigating to that dashboard.



https://securityonion/kibana/app/dashboards#/view/65b0c760-f8c6-11ec-96a1-332583da1a67?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-1h,to:now))

Winlog - Event Id Summary

event_id	action	Count
4624	logged-in	249
4688	created-process	76
		325

Winlog - SubjectUserName Summary

domain	SubjectUserName
LAB.NET	-
LAB	WKST-001$
LAB	WKST-002$
LAB	DC1$
NT AUTHORITY	WKST-001$
NT AUTHORITY	WKST-002$

Winlog - PID Summary

process.pid	process.name	Count
0	-	227
652	services.exe	43
796	svchost.exe	30
1,528	svchost.exe	11
532	svchost.exe	4
648	services.exe	2
660	services.exe	2
784	svchost.exe	1
1,548	svchost.exe	1
2,292	dsregcmd.exe	1
3,928	gpupdate.exe	1

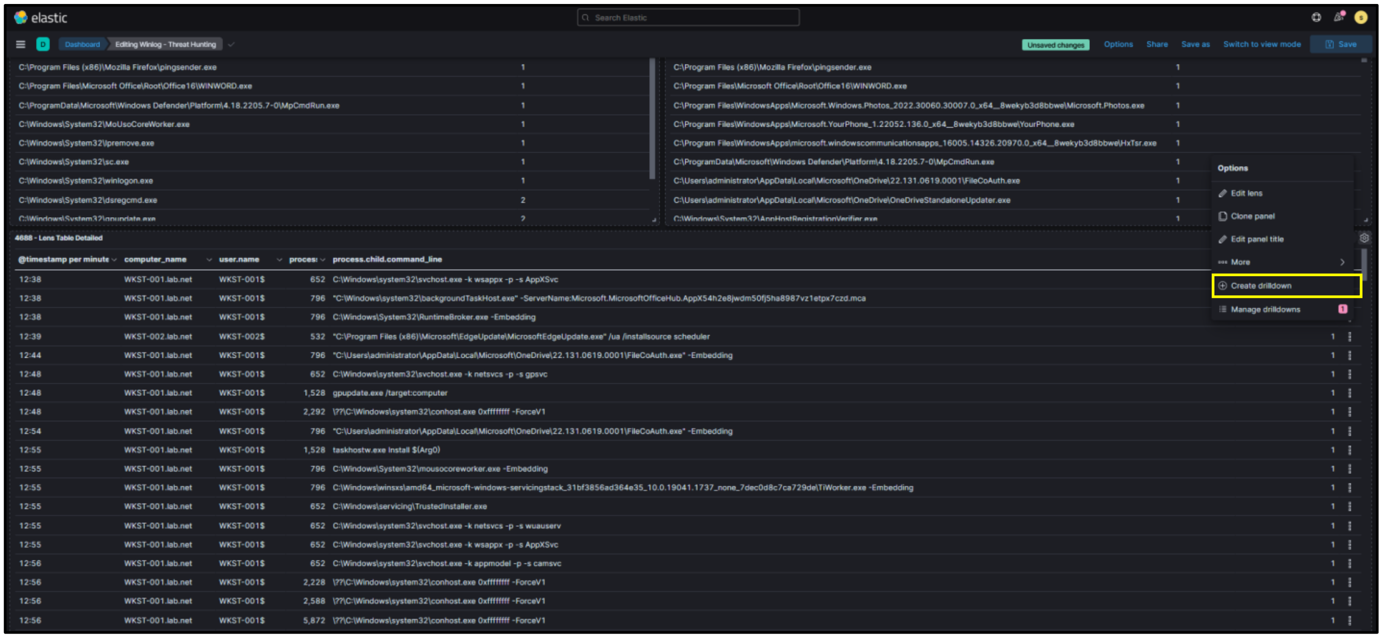
Winlog - Bar Chart of Event Types over Time

2. In the URL, copy everything left of the start of the URL parameters (e.g. "?_g="). It should look something like the link below.

```
https://securityonion/kibana/app/dashboards#/view/65b0c760-f8c6-11ec-96a1-332583da1a67
```

✅**NOTE: Your domain/IP component of the URL as well as the GUID that identifies your dashboard will probably be different than the values shown above.**

3. Next, navigate back to the "Winlog – Threat Hunting" dashboard you created in Lab 2.
4. Ensure you are in "edit" mode.
5. Click on the "gear" icon in the top, right-hand corner of the "Winlog – 4688 Lens Table Detailed" visualization at the bottom of the dashboard.



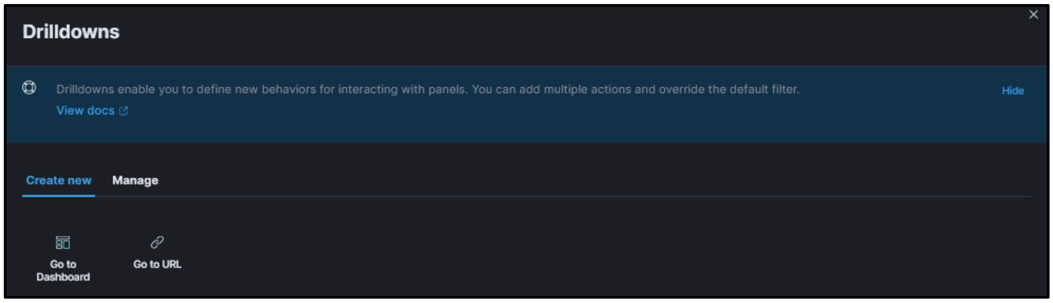
C:\Program Files (x86)\Mozilla Firefox\pingsender.exe	1
C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe	1
C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2205.7-0\MpCmdRun.exe	1
C:\Windows\System32\MoUsoCoreWorker.exe	1
C:\Windows\System32\lpremove.exe	1
C:\Windows\System32\sc.exe	1
C:\Windows\System32\winlogon.exe	1
C:\Windows\System32\dsregcmd.exe	2
C:\Windows\System32\gpupdate.exe	2

C:\Program Files (x86)\Mozilla Firefox\pingsender.exe	1
C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe	1
C:\Program Files\WindowsApps\Microsoft.Windows.Photos_2022.30060.30007.0_x64__8wekyb3d8bbwe\Microsoft.Photos.exe	1
C:\Program Files\WindowsApps\Microsoft.YourPhone_1.22052.136.0_x64__8wekyb3d8bbwe\YourPhone.exe	1
C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_16005.14326.20970.0_x64__8wekyb3d8bbwe\HxTsr.exe	1
C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2205.7-0\MpCmdRun.exe	1
C:\Users\administrator\AppData\Local\Microsoft\OneDrive\22.131.0619.0001\FileCoAuth.exe	1
C:\Users\administrator\AppData\Local\Microsoft\OneDrive\OneDriveStandaloneUpdater.exe	1
C:\Windows\System32\AppHostRegistrationVerifier.exe	1

4688 - Lens Table Detailed

@timestamp per minute	computer_name	user.name	proces:	process.child.command_line	
12:38	WKST-001.lab.net	WKST-001$	652	C:\Windows\system32\svchost.exe -k wsappx -p -s AppXSvc	
12:38	WKST-001.lab.net	WKST-001$	796	"C:\Windows\system32\backgroundTaskHost.exe" -ServerName:Microsoft.MicrosoftOfficeHub.AppX54h2e8jwdm50fj5ha8987vz1etpx7czd.mca	
12:38	WKST-001.lab.net	WKST-001$	796	C:\Windows\System32\RuntimeBroker.exe -Embedding	
12:39	WKST-002.lab.net	WKST-002$	532	"C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe" /ua /installsource scheduler	1
12:44	WKST-001.lab.net	WKST-001$	796	"C:\Users\administrator\AppData\Local\Microsoft\OneDrive\22.131.0619.0001\FileCoAuth.exe" -Embedding	1
12:48	WKST-001.lab.net	WKST-001$	652	C:\Windows\system32\svchost.exe -k netsvcs -p -s gpsvc	1
12:48	WKST-001.lab.net	WKST-001$	1,528	gpupdate.exe /target:computer	1
12:48	WKST-001.lab.net	WKST-001$	2,292	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	1
12:54	WKST-001.lab.net	WKST-001$	796	"C:\Users\administrator\AppData\Local\Microsoft\OneDrive\22.131.0619.0001\FileCoAuth.exe" -Embedding	1
12:55	WKST-001.lab.net	WKST-001$	1,528	taskhostw.exe Install $(Arg0)	1
12:55	WKST-001.lab.net	WKST-001$	796	C:\Windows\System32\mousocoreworker.exe -Embedding	1
12:55	WKST-001.lab.net	WKST-001$	796	C:\Windows\winsxs\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.1737_none_7dec0d8c7ca729de\TiWorker.exe -Embedding	1
12:55	WKST-001.lab.net	WKST-001$	652	C:\Windows\servicing\TrustedInstaller.exe	1
12:55	WKST-001.lab.net	WKST-001$	652	C:\Windows\system32\svchost.exe -k netsvcs -p -s wuauserv	1
12:55	WKST-001.lab.net	WKST-001$	652	C:\Windows\system32\svchost.exe -k wsappx -p -s AppXSvc	1
12:56	WKST-001.lab.net	WKST-001$	652	C:\Windows\system32\svchost.exe -k appmodel -p -s camsvc	1
12:56	WKST-001.lab.net	WKST-001$	2,228	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	1
12:56	WKST-001.lab.net	WKST-001$	2,588	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	1
12:56	WKST-001.lab.net	WKST-001$	5,872	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	1

6. Click on the "+ Create drilldown" button.



Drilldowns enable you to define new behaviors for interacting with panels. You can add multiple actions and override the default filter.
View docs

7. Click on "Go to URL".
8. Configure the following options:
   - 8.1. **Name:** Investigate
   - 8.2. **Trigger:** Table row click
   - 8.3. **Open in new window:** Checked
   - 8.4. **Encode URL:** Checked
   - 8.5. **Enter URL:**

```
{{kibanaUrl}}/app/dashboards#/view/65b0c760-f8c6-11ec-96a1-
332583da1a67?_g=(filters:!((query:(match_phrase:({{event.keys.[1]}}:{{event.values.[1]}})))),refreshInterval:
(pause:!t,value:0),time:(from:'{{date event.values.[0]}}',to:'{{date event.values.[0]}}||%2B10m'))
```

💡**LEARNING POINT: Make sure to replace the GUID shown above with the GUID specific to your dashboard.**

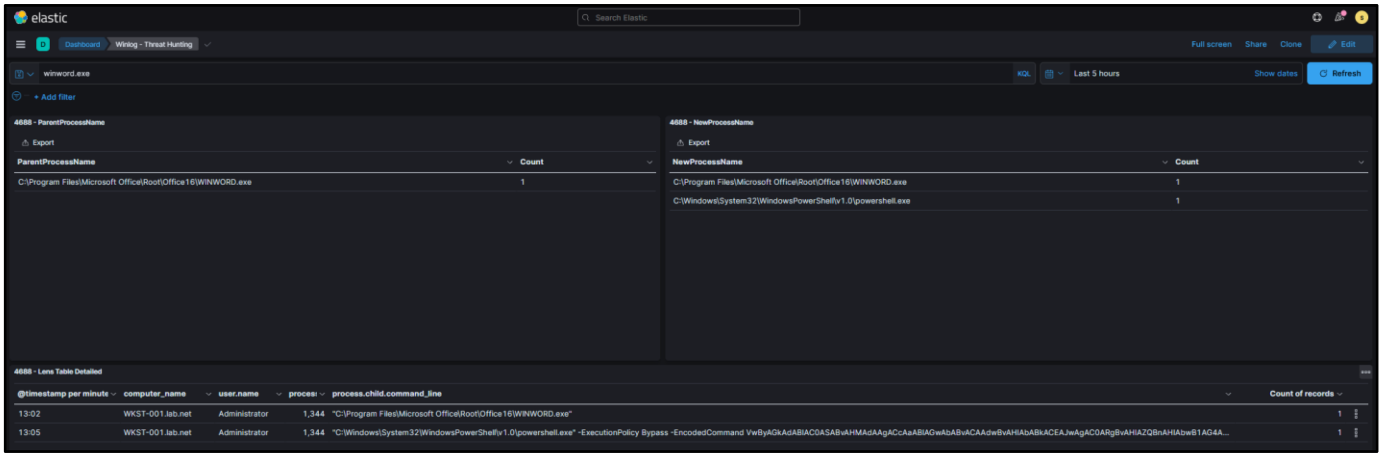
The URL Drill Down string shown above passes a filter to the next dashboard that will only show events where winlog.computer_name is equal to the value of the field this Drill Down event was executed from. It also includes a time filter from the timestamp of the source event to the timestamp of the source event plus

10 minutes. Refer to Appendix A in the back of this lab guide for a detailed description of how this URL is formatted.

9. Click on the blue “Save” button.
10. Go back to the dashboard and click on the blue “Refresh” button in the top, right-hand corner of the screen.
11. Enter the following search into the search bar:

```
winword.exe
```

You should see output similar to the screenshot below:



4688 - ParentProcessName

Export

ParentProcessName	Count
C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe	1

4688 - NewProcessName

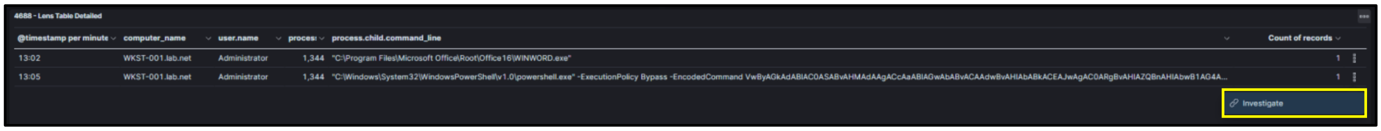
Export

NewProcessName	Count
C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe	1
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	1

4688 - Lens Table Detailed

@timestamp per minute	computer_name	user.name	proces:	process.child.command_line	Count of records
13:02	WKST-001.lab.net	Administrator	1,344	"C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe"	1
13:05	WKST-001.lab.net	Administrator	1,344	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy Bypass -EncodedCommand VwByAGkAdABlAC0ASABvAHMAdAAgACcAaABlAGwAbABvACAAdwBvAHIAbABkACEAJwAgAC0ARgBvAHIAZQBnAHIAbwB1AG4A...	1

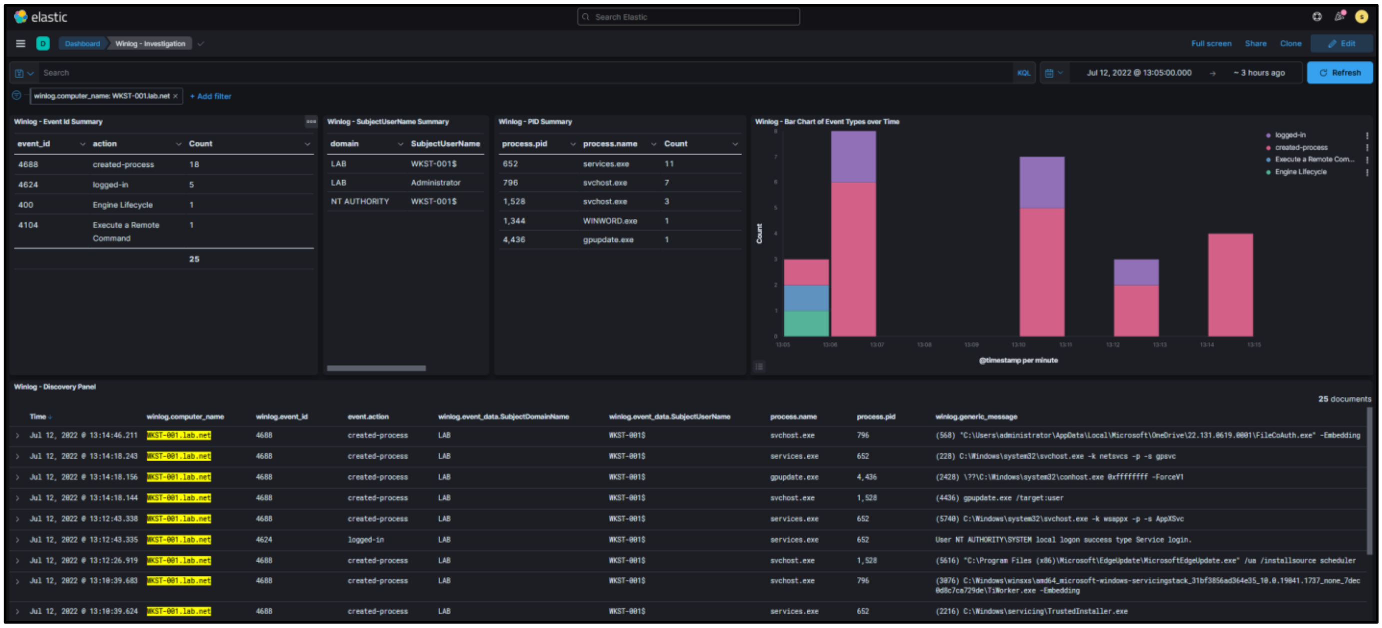
12. Click on the three vertical square dots on the far, right-hand side of one of the “4688 – Lens Table Detailed” panel rows and then click “Investigate”. This will trigger our URL Drill Down and take us to the “Winlog – Investigation” dashboard and configure the appropriate filters based on the source event.



4688 - Lens Table Detailed

@timestamp per minute	computer_name	user.name	proces:	process.child.command_line	Count of records
13:02	WKST-001.lab.net	Administrator	1,344	"C:\Program Files\Microsoft Office\Root\Office16\WINWORD.exe"	1
13:05	WKST-001.lab.net	Administrator	1,344	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy Bypass -EncodedCommand VwByAGkAdABlAC0ASABvAHMAdAAgACcAaABlAGwAbABvACAAdwBvAHIAbABkACEAJwAgAC0ARgBvAHIAZQBnAHIAbwB1AG4A...	1

13. Review the “Winlog – Investigation” dashboard to see if you can tell what actions the attacker took after gaining initial execution through spear phishing.



winlog.computer_name: WKST-001.lab.net

Winlog - Event Id Summary

event_id	action	Count
4688	created-process	18
4624	logged-in	5
400	Engine Lifecycle	1
4104	Execute a Remote Command	1
		25

Winlog - SubjectUserName Summary

domain	SubjectUserName
LAB	WKST-001$
LAB	Administrator
NT AUTHORITY	WKST-001$

Winlog - PID Summary

process.pid	process.name	Count
652	services.exe	11
796	svchost.exe	7
1,528	svchost.exe	3
1,344	WINWORD.exe	1
4,436	gpupdate.exe	1

Winlog - Discovery Panel

25 documents

Time	winlog.computer_name	winlog.event_id	event.action	winlog.event_data.SubjectDomainName	winlog.event_data.SubjectUserName	process.name	process.pid	winlog.generic_message
Jul 12, 2022 @ 13:14:46.211	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	796	(568) "C:\Users\administrator\AppData\Local\Microsoft\OneDrive\22.131.0619.0001\FileCoAuth.exe" -Embedding
Jul 12, 2022 @ 13:14:18.243	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(228) C:\Windows\system32\svchost.exe -k netsvcs -p -s gpsvc
Jul 12, 2022 @ 13:14:18.156	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	gpupdate.exe	4,436	(2428) \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
Jul 12, 2022 @ 13:14:18.144	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	1,528	(4436) gpupdate.exe /target:user
Jul 12, 2022 @ 13:12:43.338	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(5740) C:\Windows\system32\svchost.exe -k wsappx -p -s AppXSvc
Jul 12, 2022 @ 13:12:43.335	WKST-001.lab.net	4624	logged-in	LAB	WKST-001$	services.exe	652	User NT AUTHORITY\SYSTEM local logon success type Service login.
Jul 12, 2022 @ 13:12:26.919	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	1,528	(5616) "C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe" /ua /installsource scheduler
Jul 12, 2022 @ 13:10:39.683	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	svchost.exe	796	(3076) C:\Windows\winsxs\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.1737_none_7dec0d8c7ca729de\TiWorker.exe -Embedding
Jul 12, 2022 @ 13:10:39.624	WKST-001.lab.net	4688	created-process	LAB	WKST-001$	services.exe	652	(2216) C:\Windows\servicing\TrustedInstaller.exe

✅**NOTE: There is a pre-configured filter in the top, left-hand corner of the screen that only shows events that originated from the same system as the source event we clicked on in the "Winlog – Threat Hunting" dashboard.**

✅**NOTE: The timespan in the top, right-hand corner of the screen goes from just prior to the source event to +10 minutes after the source event. This gives us a very narrow time window of data to review and will hopefully provide some context about what events happened on the system in and around the time of the suspicious event.**

SUMMARY

In this lab, we created a tailored investigation dashboard that can be used to determine if a specific event is related to malicious activity. We created several summary visualizations to get an overview of what was happening on the system. Finally, we created a URL Drill Down that will automatically populate the investigation dashboard with filters and time ranges based on a suspicious source event.

In the next lab, we will add some advanced filters and queries that can be applied to our dashboards to help us identify malicious activity. We will then follow the process that we just created where we threat hunt from the "Winlog – Threat Hunting" dashboard, and we investigate by executing a URL Drill Down to the "Winlog – Investigation" dashboard.

APPENDIX

APPENDIX A: URL DRILLDOWN TEMPLATES

URL TEMPLATING LANGUAGE REFERENCE LINKS

Kibana has a link for the URL Templating Language that can be used as a reference for building drilldowns.

https://www.elastic.co/guide/en/kibana/current/url_templating-language.html

There is also a separate link for drilldown documentation.

https://www.elastic.co/guide/en/kibana/7.17/drilldowns.html

PASSING A FILTER AND TIME RANGE

Description:	This URL Drilldown template will pass the computer name of the current event row and the timestamp with a +10 minute range afterwards. This template only works with Table row click event on Lens Table visualization. **Kibana URL Variable:** `{{kibanaUrl}}` **Resource Path to the Dashboard:** `/app/dashboards#/view/65b0c760-f8c6-11ec-96a1-332583da1a67` **Query Filter passed to the Dashboard:** `(query:(match_phrase:({{event.keys.[1]}}:{{event.values.[1]}})))` **Time Range passed to Dashboard:** `time:(from:'{{date event.values.[0]}}',to:'{{date event.values.[0]}}||%2B10m')` 💡**LEARNING POINT: Some characters will need to be manually URL encoded in order for Kibana to properly interpret any operations you're trying to perform with variables inside of the URL string. For example, in the time range above the statement** `'{{date event.values.[0]}}||+10m'` **is not properly interpreted by Kibana unless the '+' character is URL encoded to %2B resulting in** `'{{date event.values.[0]}}||%2B10m'`**.** 💡**LEARNING POINT: The 'date' command shown above converts a timestamp field, which is internally a long integer representing the number of milliseconds since epoch (January 1st, 1970 at 00:00:00 UTC). The Kibana URL parser for the time range filter doesn't accept timestamps formatted as long integers, so they must be converted to something human readable. The date command performs this conversion from a long integer to a string that can be interpreted by the Kibana dashboard URL handler.** 💡**LEARNING POINT: You can perform some *very limited* operations within the Kibana URL templates. Here we use this ability to add 10 minutes to our time range filter that is passed to the target dashboard. The syntax for this is <timestamp>||+10m. This allows us to narrow down the range of results that will be displayed to only events that occurred in and around the event we want to investigate as those events should be the most relevant to understanding the context of the event we observed in the current dashboard.**
Query:	`{{kibanaUrl}}/app/dashboards#/view/65b0c760-f8c6-11ec-96a1-332583da1a67?_g=(filters:!((query:(match_phrase:({{event.keys.[1]}}:{{event.values.[1]}})))),refreshInterval:(pause:!t,value:0),time:(from:'{{date event.values.[0]}}',to:'{{date event.values.[0]}}||%2B10m'))`

GENERIC DRILLDOWN TO PASS A COLUMN FIELD AS A FILTER

Description:	This URL Drilldown template will pass the user selected field name and value to the target dashboard as a filter. A specific column of the data table does not need to be identified. This template only works on Single click events, but supports both Lens and Aggregation based visualizations. **Kibana URL Variable:** `{{kibanaUrl}}` **Resource Path to the Dashboard:** `/app/dashboards#/view/86fc5e40-ee60-11ec-823e-238becd7f8b3` **Query Filter passed to the Dashboard:** `(query:(match_phrase:({{event.key}}:{{event.value}})))` 💡**LEARNING POINT: For Single click events, the name of the field and the value of the field the user is clicking on can be referenced with the variables `{{event.key}}` and `{{event.value}}` respectively.**
Query:	`{{kibanaUrl}}/app/dashboards#/view/86fc5e40-ee60-11ec-823e-238becd7f8b3?_g=(filters:!((query:(match_phrase:({{event.key}}:{{event.value}})))))`