Inject: Web Shell

| Date |  |
| --- | --- |
| Name: |  |
| Team Number: |  |

# Summary

We have reason to believe that a shady threat group known as ‘Red Team’ has either compromised our systems or is planning. Specifically, we believe they have targeted our business critical web application. We need you to determine if the web server has indeed been compromised. Unfortunately, threat actors generally don’t announce when they’re going to attack. You need to monitor and be prepared.

# Hints

We’re looking for threat actor activity that doesn’t want to be found. You know your network. Keep in mind how traffic flows in and out of your network. How does traffic enter? Through what device? Where is that traffic forwarded to? You can inspect the payloads of the traffic. The scoreboard and your team are all connecting to the web server. What does that known-good traffic look like? Answer these and you can find anomalies. Happy hunting!

**There is a user guide for NetWitness at the end of this document.**

# Part 1 Instructions

You have been provided with the tools needed to investigate this threat. Other than the built in linux utilities available via command line on the web server, you have access to a security information and event management (SIEM) and network detection and response (NDR) solution called NetWitness. This solution's job is to collect and analyze event logs and packet information. Your job is to carve that data and identify anomalies. Good and bad data is mixed together. We need to separate it so that we can respond appropriately.

Your goal is to identify the following (feel free to add screenshots):

* attacker’s source IP address
* insecure file/utility name
  + What are the credentials?
* what HTTP header shows the credentials?
* name of the user input field allowing remote command execution (RCE)

| Attacker Source IP | <ip> |
| --- | --- |
| Filename | <filename> |
| Credentials | <username>:<password> |
| HTTP Header Name | <header name> |
| Input Field Name | <field name> |

# 

# 

# Part 2 Instructions

Severity Guidelines

| Critical | Multiple systems compromised, severe systems impact, data loss, loss of productivity |
| --- | --- |
| High | Multiple systems impacted, high risk of data theft/loss, potential systems outage |
| Medium | Potential risk identified, initial stages of attack, potential for spread |
| Low | Identification of enabler of compromise, low risk activity/behavior |

| Incident Severity | <value> |
| --- | --- |

| Notes: (reasoning for determined severity, feel free to delete this text) |
| --- |

*Check all that apply*

| Compromised System |  | Physical Break-in |  |
| --- | --- | --- | --- |
| Compromised User |  | Social Engineering |  |
| Network Attack |  | Law Enforcement Request |  |
| Malware |  | Policy Violation |  |
| Reconnaissance |  | Misconfiguration |  |
| Lost Equipment/Theft |  | Other (please note below) |  |

Create and insert a timeline of events in the space below (you may make the timeline using whatever tool you like). A timeline needs to include an identifiable artifact (such as IP address) and show events in chronological order. You may use the table below to organize your event artifacts to be used in the timeline visual. Change the size of the table (rows/columns) as you see fit.

Think of the following questions when forming a timeline:

* What did I identify?
* What happened before this event (activity associated with the artifact)?
* What happened after?
* How did the attack start?
* Was data stolen?

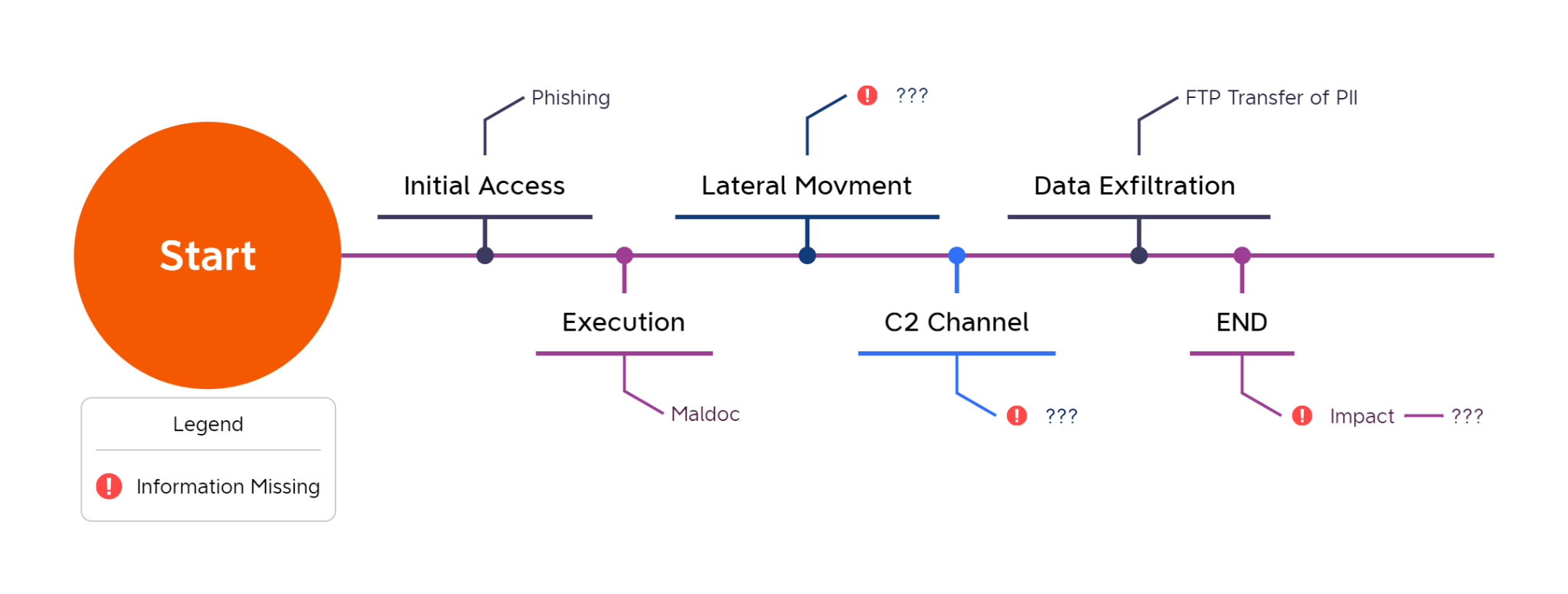
| Event # | IP Source | IP Destination | Notes (short one-liner description of event) |
| --- | --- | --- | --- |
| Event 1 |  |  |  |
| Event 2 |  |  |  |
| Event 3 |  |  |  |
| Event 4 |  |  |  |

## 

## EXAMPLE TIMELINE

\*This is an example only and the details don’t relate to the challenge/inject. It is meant to be an example of what forming a timeline may look like.\*

A timeline represents the attack chain associated with an incident. It must include (to the best of your knowledge) the start, middle, and end of the attack.



## 

## Timeline

<insert your timeline here>

## Indicators of Compromise (IOCs)

IOCs are used to help identify the spread of activity from the same threat actor.

| Threat Actor Sources (malicious IPs, emails, domains etc., ) |  |
| --- | --- |
| Impacted Systems (IPs) |  |

Based on the information above, what steps would you recommend for triage/mitigation of the threat?

# 

# NetWitness Basics

## Accessing the UI

The web UI for NetWitness is hosted at <https://172.18.121.2>.

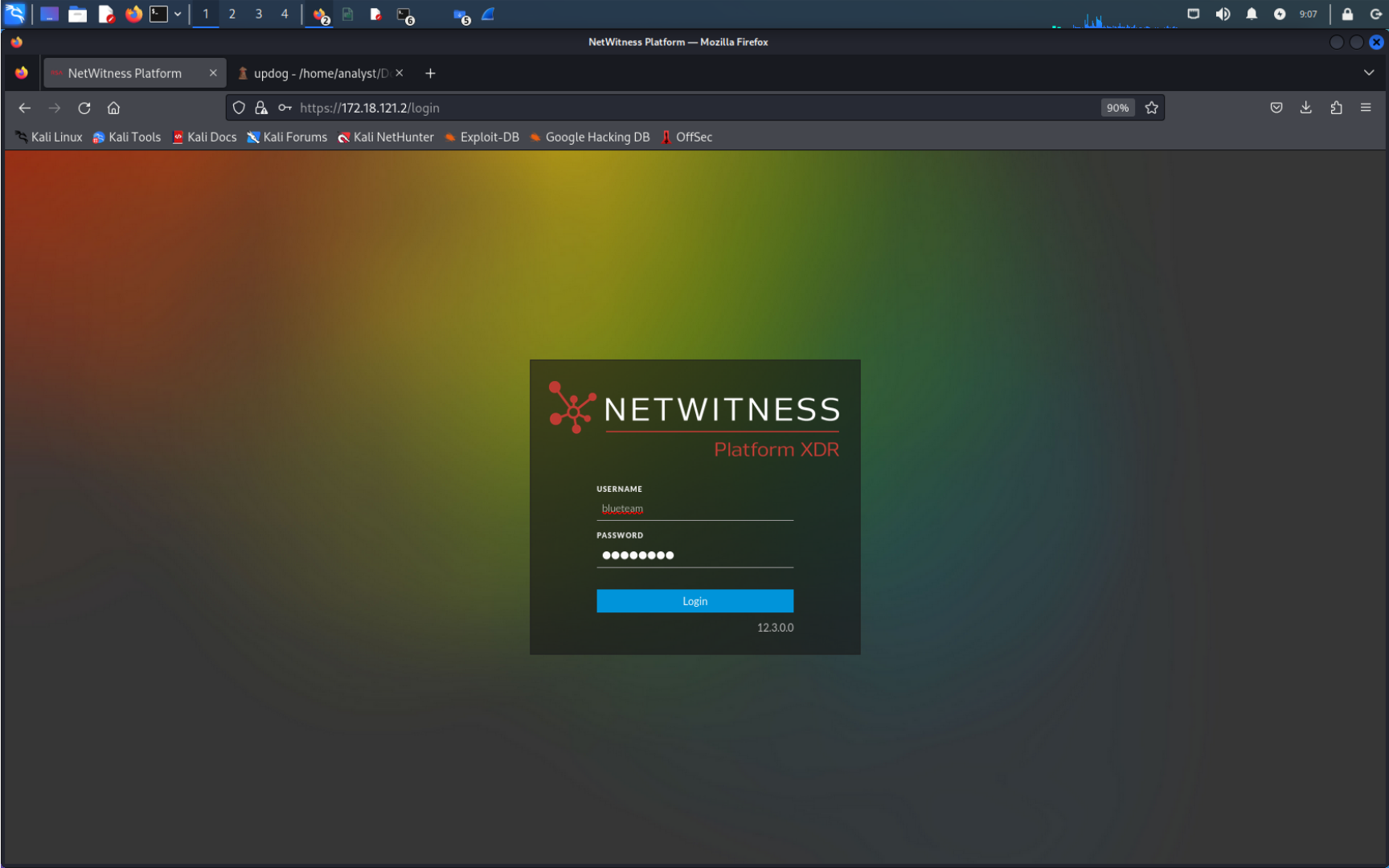
You can access this via the following methods:

* SOCK5 proxy to jumphost
* Using your external Kali machines
* Using your internal Kali machines (once you have configured your internal networking)

## Logging In

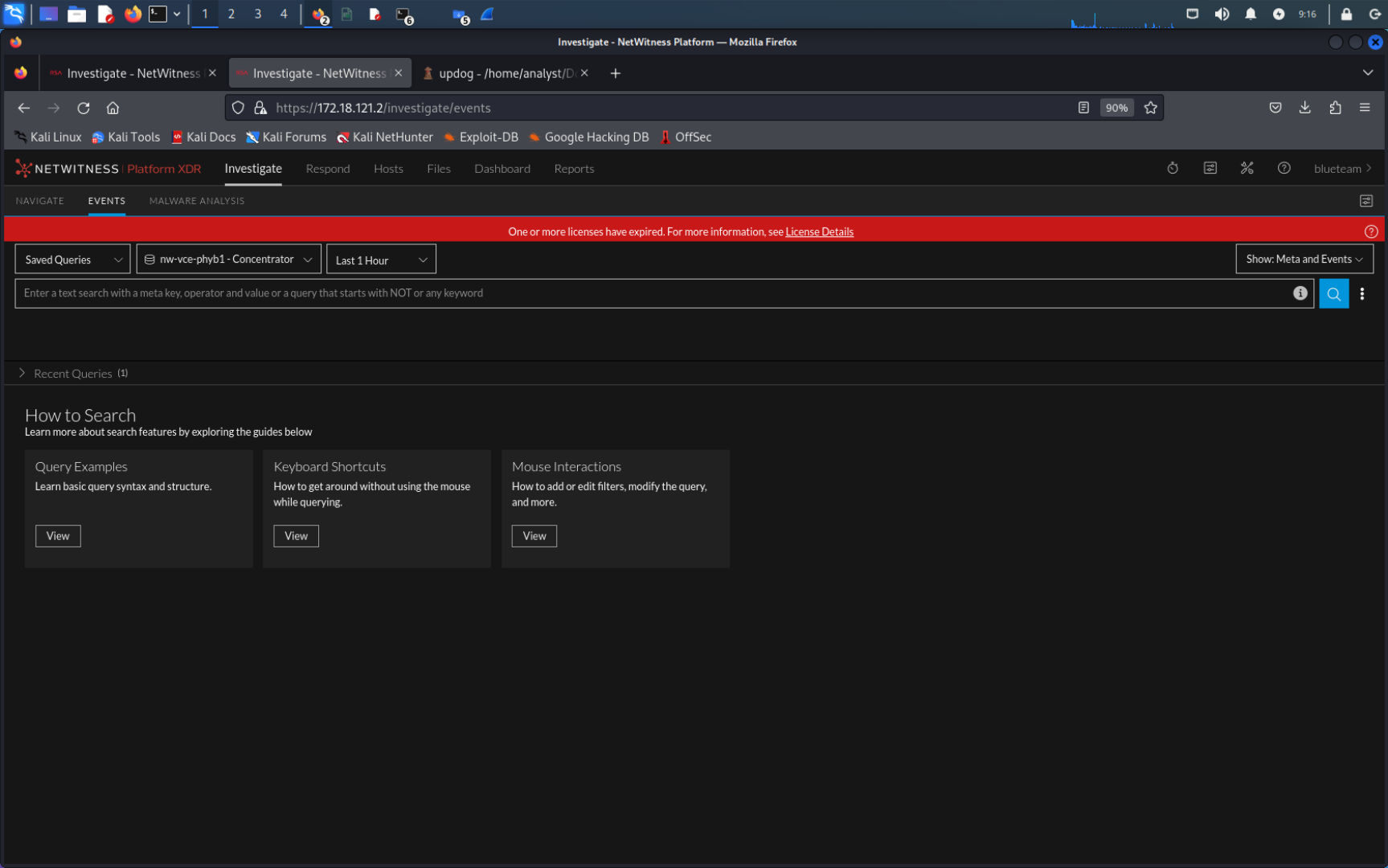
Once you reach the UI, you’ll be greeted with a login prompt. The access credentials are:

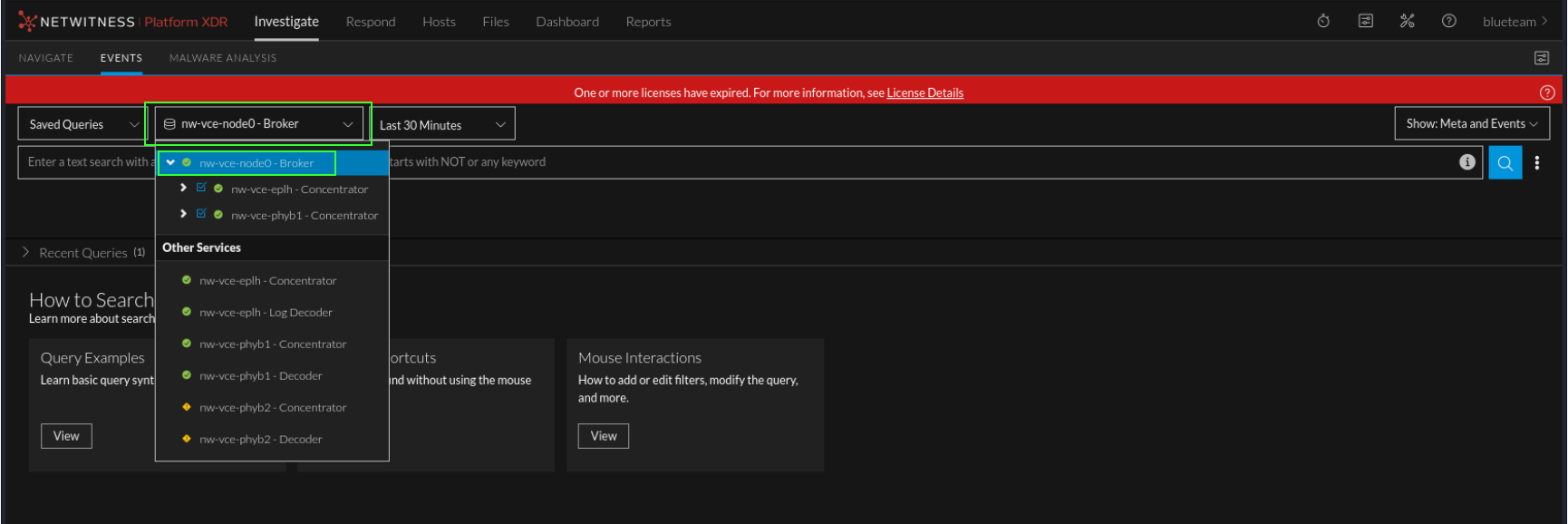
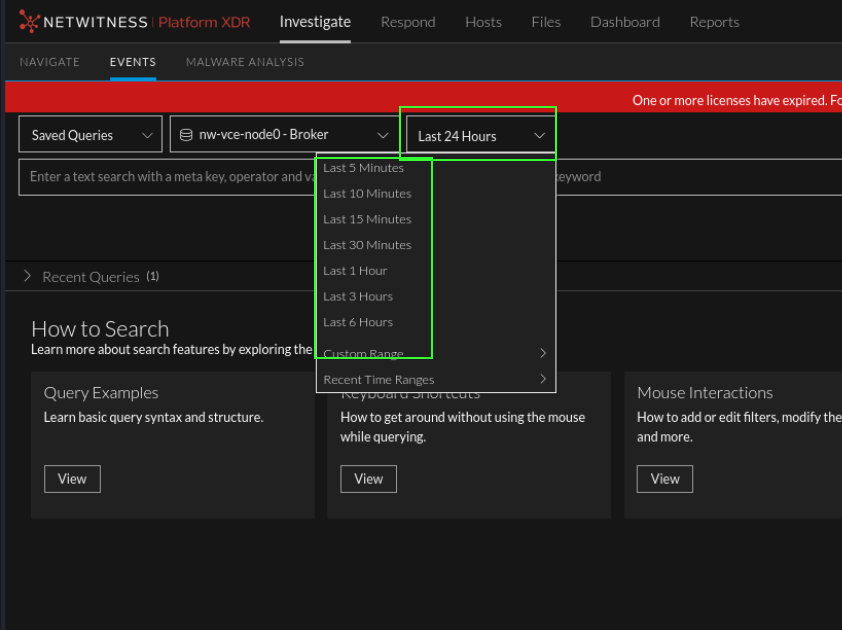
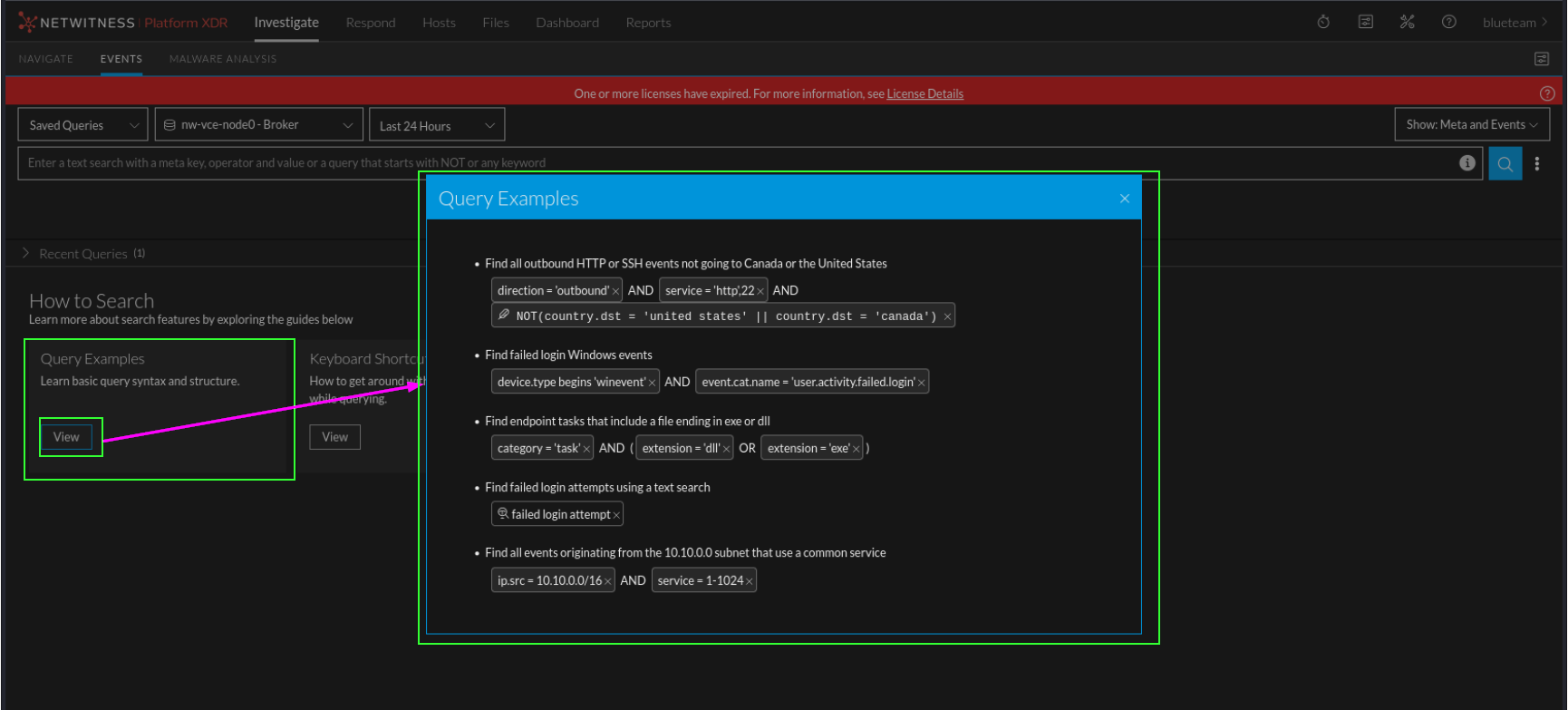
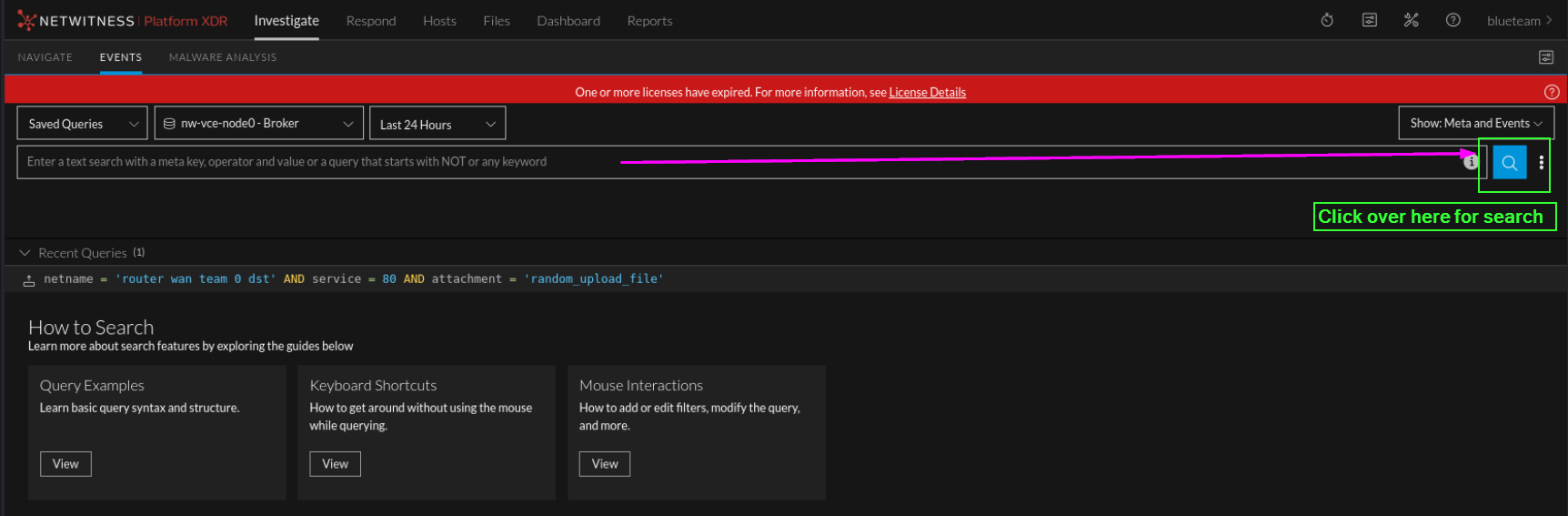
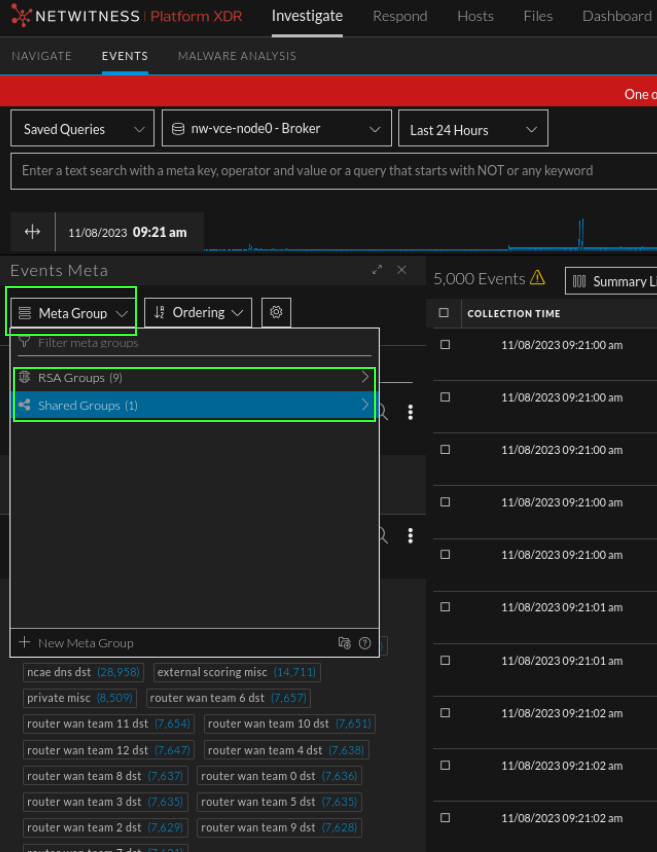
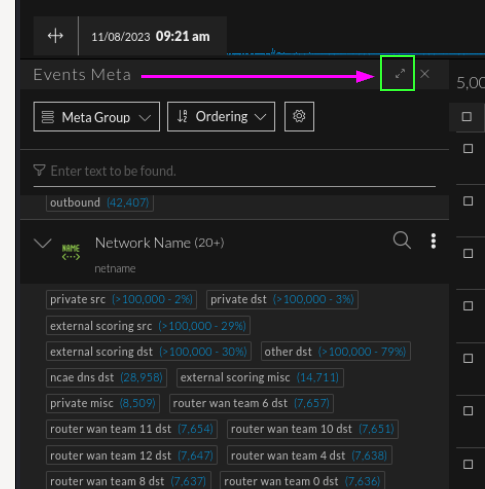
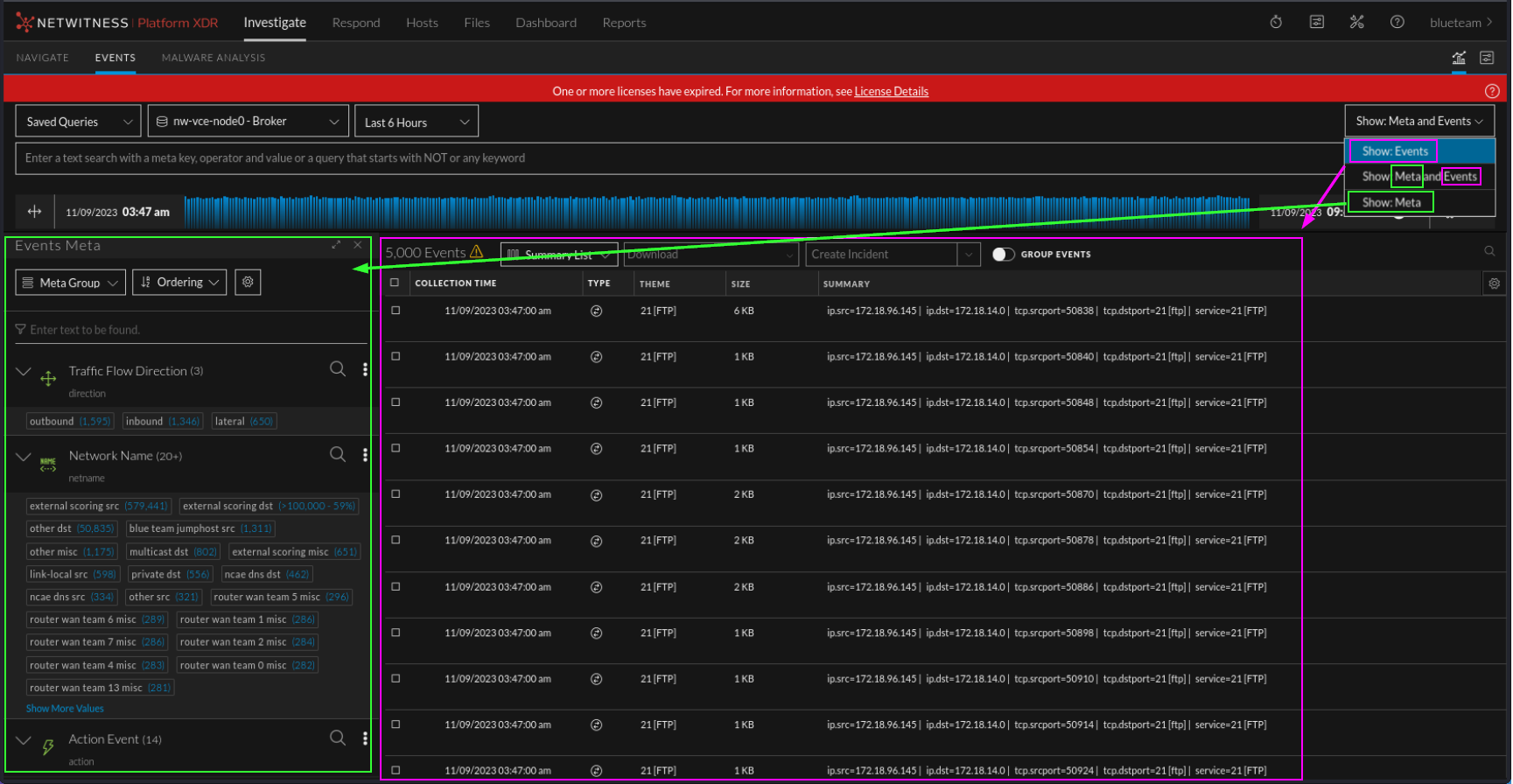
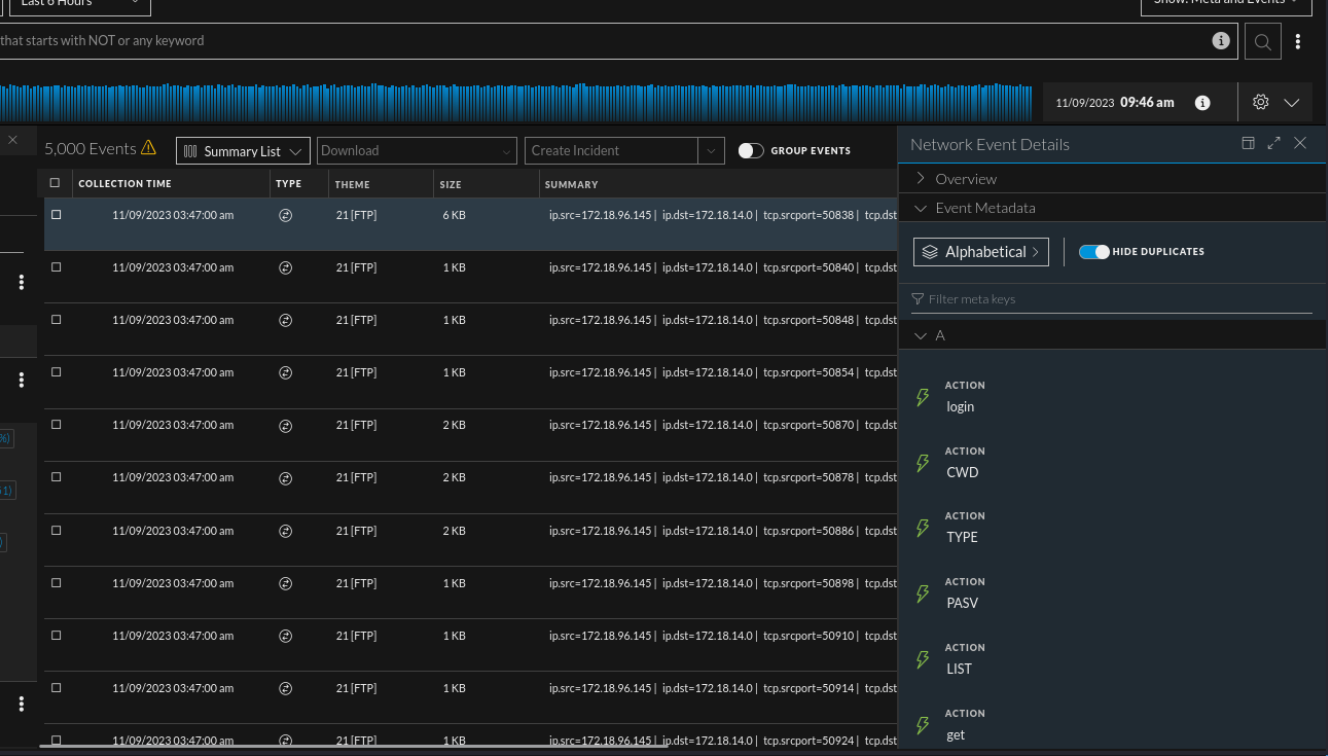
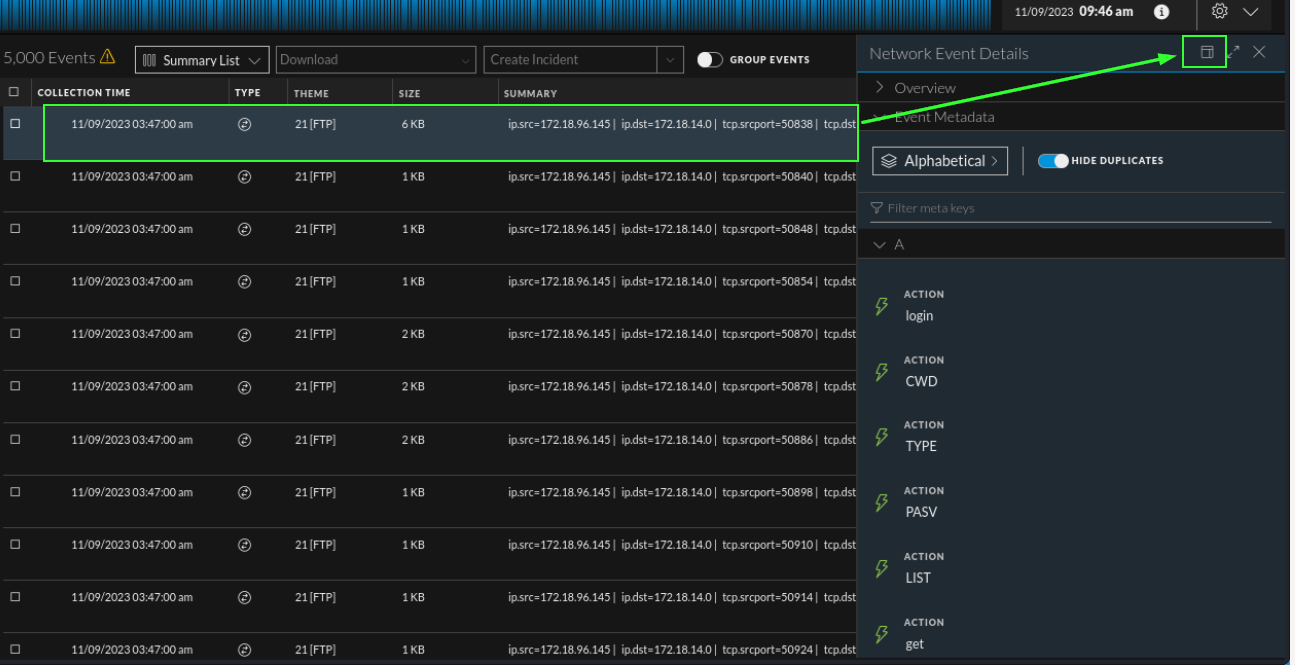
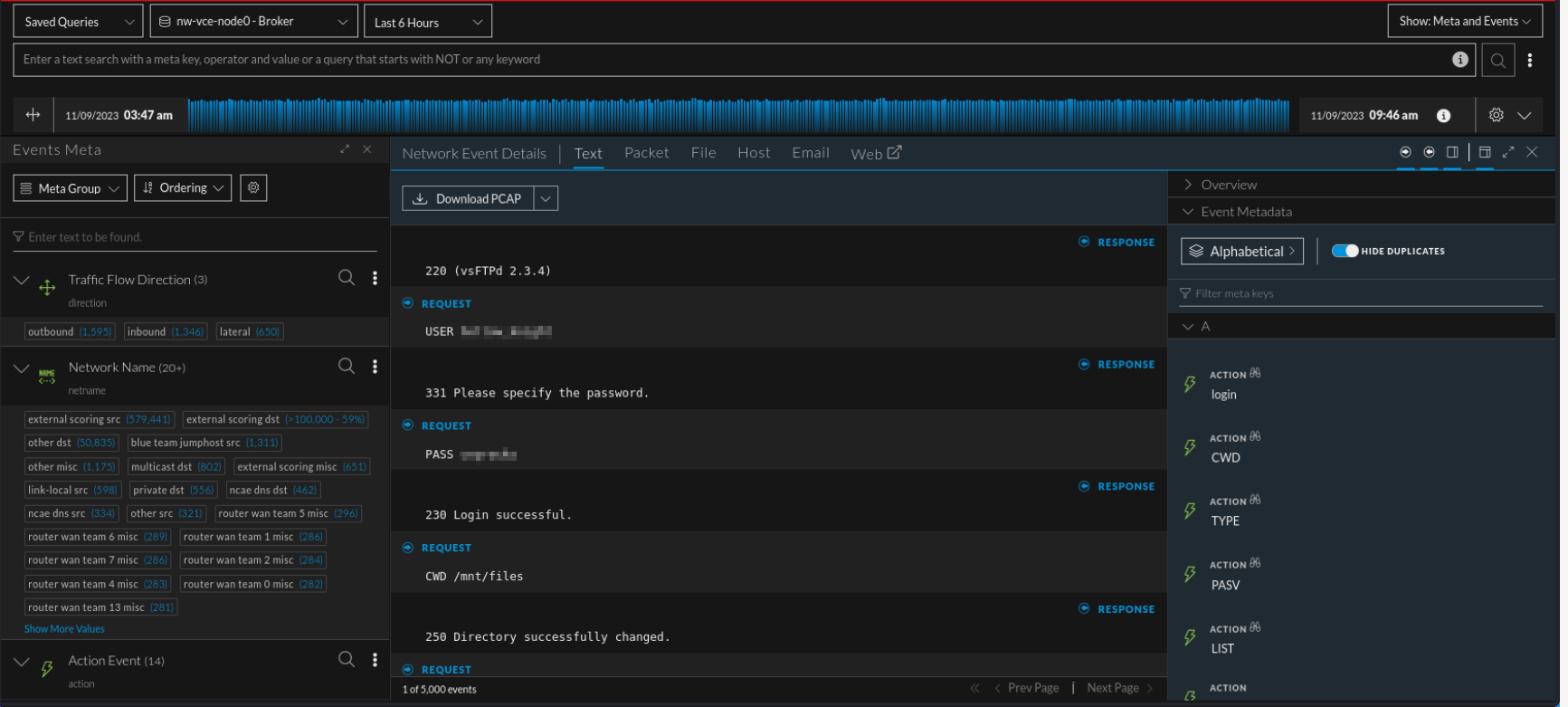
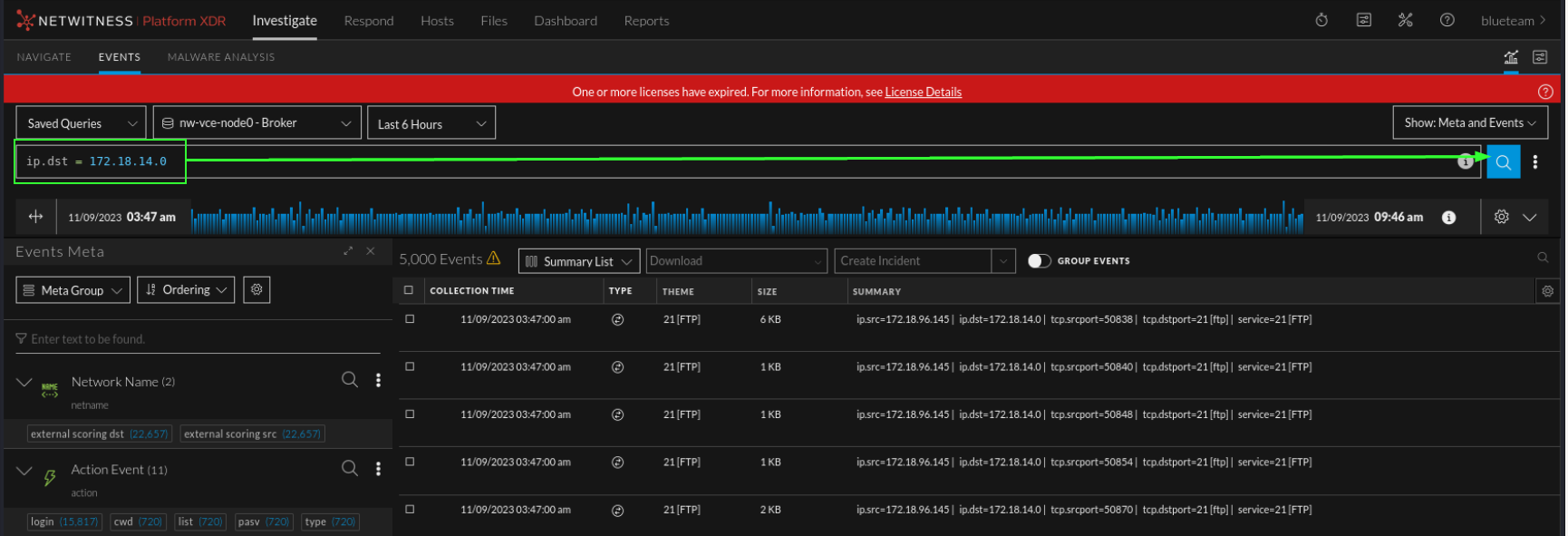
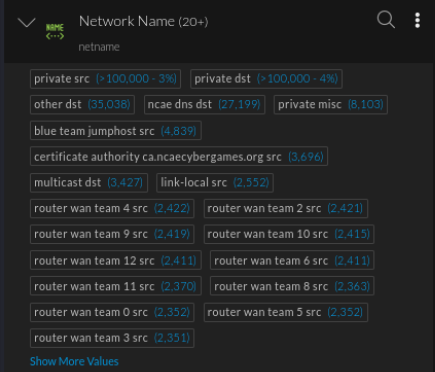
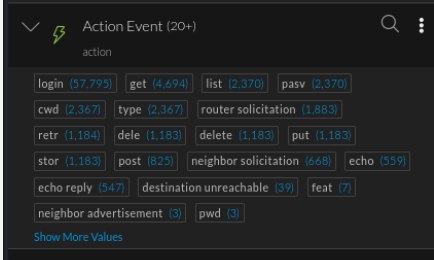
* Username: blueteam
* Password: blueteam



## Getting Started

Once you login, you’ll land at the Investigation module. This is where you will get started with data carving. NetWitness is meta data oriented platform. This means that the tool parses data that turns into meta values and places those values within categories called meta keys. You carve the data by looking for combinations of meta values and meta keys. Unlike other packet capture tools, NetWitness uses the concept of sessions. If you have used wireshark in your classes, this would be similar to the ‘Follow Stream’ function. In NetWitness, a single session can contain multiple packets.



1. Ensure that the ‘broker’ is selected as your data source (see below)
   1. 
2. Select a time range that makes sense for what you’re trying to do. For example, if you’re looking for red team activity on competition day, you probably only need to go back a few hours at most. This will lower your dataset so that you have less data carving to do.
   1. 
3. Review the example queries if you’d like. Much like wireshark, you can just type out what you want to look for via the query bar.
   1. 
4. Click the blue search icon to the right of the screen. Optionally, if you did the hunting challenges in 2022 that used NetWitness, you can get started with a query and then click search.
   1. 
5. Once you click search, you end in the data carving and reconstruction view. There are a few options in this view to play with.
   1. Meta Groups - meta groups organize the data the you carve. You can choose the order and keys/categories that are displayed. There are 9 built in groups and 1 custom group. Use whatever you want. The custom group is what I created when looking for the answers to this challenge.
      1. 
   2. Important buttons
      1. The little ←→ icon on the Events meta panel can clicked to expand the view to full screen. This is useful when getting started to reduce your data set.
         1. 
   3. Meta vs. Events View
      1. The drop down in the top right hand corner of the screen that says ‘Show: Meta and Events’, can be changed. The panel to the left of the screen is the meta carving screen. You click on meta to add to the query (left click to append to query, right click for options to add to query). The Meta panel is highlighted in green below. The ‘Events’ panel is in the center of the screen. The table lists out events that can be selected to view their specific meta keys/values and optionally reconstruct the data.
6. 
7. Event reconstruction will be crucial to solving this challenge. When you select an event in the center table, the meta for that session is available to review. That may look like the following where when selecting an event in the table, a panel opens on the right hand side showing the related meta.
   1. 
   2. If you select this squarish icon at the top of the meta panel, you can reconstruct the event.
   3. 
   4. When you select that button, you’ll be able to see the raw packets via text or packet form as well as any files that were extracted.
   5. 
   6. HINT: Nothing is stopping you from looking at your other services, like FTP. Might help you troubleshoot or find more shenanigans.
8. Search/Query Bar - the search bar allows you to manually type out queries if you know what you’re looking for. For example, if you wanted to look for FTP traffic going to your FTP server, you could use a query such as: ip.dst = 172.18.14.0 (review your topology for addressing specific to your team). This looks like the following.
   1. 
   2. Once you have clicked ‘search’, both the meta panel to the left and the events table in the center return data that match the search criteria (in this case, only traffic destined for team 14’s FTP server). Alternatively, you could scroll down through the left hand panel, find the ‘Destination IP’ key/category and simply click on the value. It’ll automatically append the search bar.
9. Useful keys to get started (not a complete list, play around):
   1. Direction - outbound, lateral, inbound … this key can help you carve your data based on flow of traffic
   2. Netname - provides friendly names for network subnets
      1. 
   3. Action - contains values such as GET,POST, login, etc.. Focussed on high level action event
      1. 
   4. Service - tags based on identified protocol. Parses the industry standard port for the identified protocol. For example, if HTTP traffic is identified, then the service key would show 80 as a value. (ex. service=80, service=80,21, service != 80)
   5. Tcp.dstport - parsers the port used in the packets/session regardless of protocol.
   6. Ip.src - search for traffic originating from a specific IPv4 address or subnet. Supports CIDR notation (ex. ip.src=172.18.0.0/16, ip.src=172.18.14.0)
   7. Ip.dst - search for traffic going to a specific IPv4 address or subnet, supports CIDR notations (ex. ip.dst=172.18.0.0/16, ip.dst = 172.18.14.0)
   8. Referer - shows values from the referer header
   9. Query - will show the path following the filename and/or domain.
      1. 
   10. Ethernet Keys
       1. Eth.src - source mac address
       2. Eth.dst - destination mac address
       3. HINT: might be useful for reacting to red team?
   11. Client - contains user-agent strings from the user-agent header. Very useful for finding non-browser activity
   12. Error - tag all non status code ‘200’ values. Useful for troubleshooting web related services (ex. ‘error exists’ would return only non 200 status code related traffic, ‘error !exists’ would return only status code 200 web traffic )
   13. Filename - contains extracted filenames
   14. Attachment - alternative to filename, for certain protocols you may see file names parse into this key
10. Query Operators
    1. Not - negates a query. Ex. NOT(netname contains ‘scoring’) . This would return data where the netname key does not contain values with the string ‘scoring’.
    2. = - equal sign checks if a key is the same as a specific value. Ex. ip.dst = 172.18.14.0 , direction = ‘inbound’
    3. != - the opposite of the = sign, not equal to a value. Ex. ip.dst != 172.18.14.0, direction !- ‘inbound’
    4. AND, && - both can be used. && is the same as the word AND. Chain criteria together. If you wanted to look for inbound web server traffic you could use: direction = ‘inbound’ AND ip.dst = 172.18.0.14 AND service = 80
    5. OR, || - The word OR or double pipes || can be used. Ex. ip.src = 172.18.14.0 OR ip.src = 172.18.0.14.
       1. Optionally, if you’re using an OR statement with the same key you could do the following: ip.src = 172.168.14.0,172.168.0.14 . Comma delimiting the values acts as an OR function.
    6. Contains - checks if the values within a key contain a specific string. Ex. domain contains ‘google’
    7. Example of all the operators in one query, you probably won’t need to do this. Example only.

