Detecting Kerberoasting activity using Azure Security Center

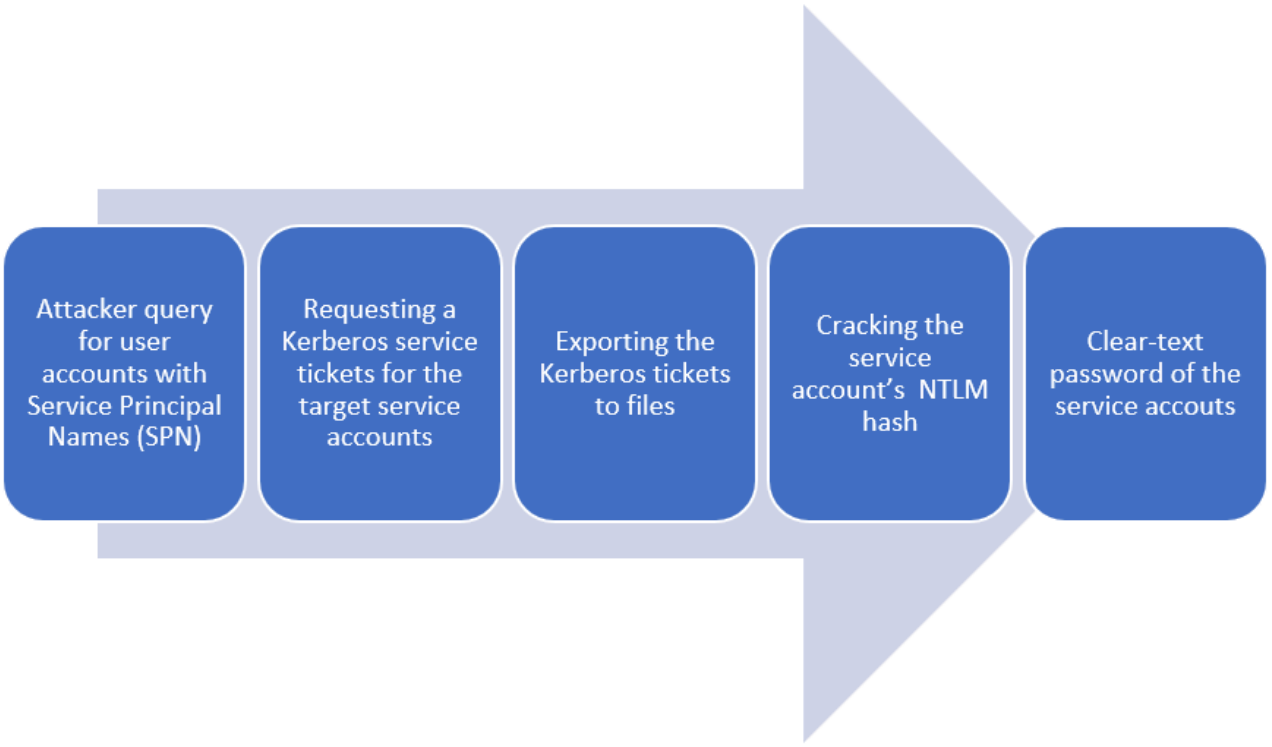
By kexugit

Kerberoasting, a term coined by Tim Medin, is a privilege escalation technique which proves to be very effective in extracting service account credentials in a domain environment. A service account is standard user account that has been configured with the specific task of running a service or scheduled task.

Many organizations are using service accounts with weak passwords that never expired, and usually these accounts enjoy excessive privileges (local administrator or domain administrator). And last but not least, actions taken by service accounts are not sufficiently audited in most environments.

Kerberoasting technique

The Kerberoasting strategy in this example is as follows:



If you're new to Kerberoasting and want to learn more, I recommend any of the following resources:

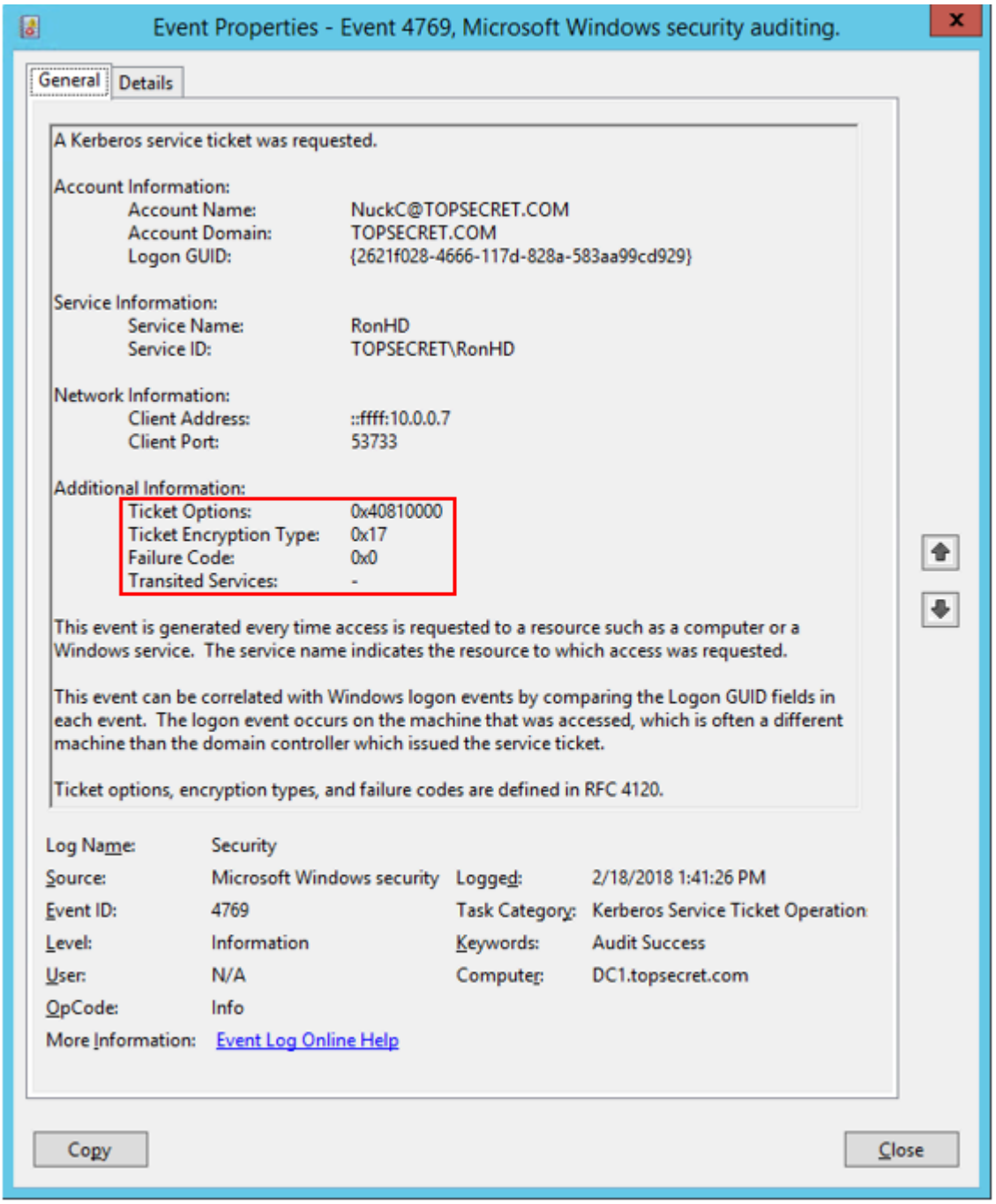
- https://adsecurity.org/?p=3458
- https://www.harmj0y.net/blog/powershell/kerberoasting-without-mimikatz/

Kerberoasting detection

Now, how to detect Kerberoasting activity in your network? We can enable "**Audit Kerberos Service Ticket Operations"** in advanced audit policy and the Domain Controllers will start to log TGS requests.

But it is not enough, detection of Kerberoasting can be challenging because requesting service tickets happens regularly as users are accessing resources in the domain. Sean Metcalf did some research and discovered that Kerberoasting activity has some unique indicators we can leverage:

- Excessive requests to different resources with small time difference (second or two)
- Kerberos TGS service tickets are requested with RC4 encryption (Type 0x17)



By collecting and analyzing security events in Azure Security Center, you can detect attacks like the one above. To enable these detections, you must have:

1. Azure subscription and Azure Security center enabled for the domain controllers
2. Enable collection of security event data in your Log Analytics workspace
3. Define custom alerts in Security Center

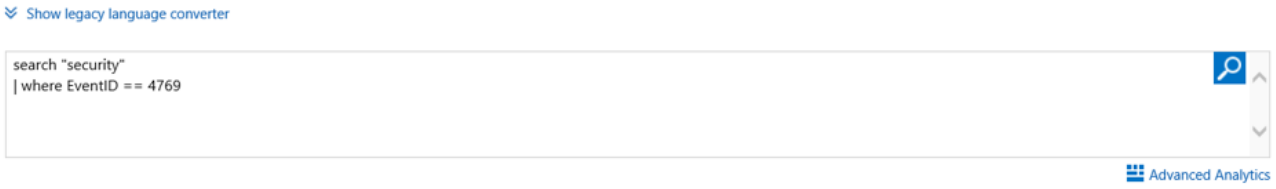
Azure Security Center provides advanced threat protection across hybrid cloud workloads. Among other features such as security assessments and threat intelligence customers can use data collection, search, and analysis (from both cloud and on-premise resources).

Define a detection logic

Log Analytics is the log search feature which allows you to combine and correlate any machine data that was collected from multiple sources within your environment.

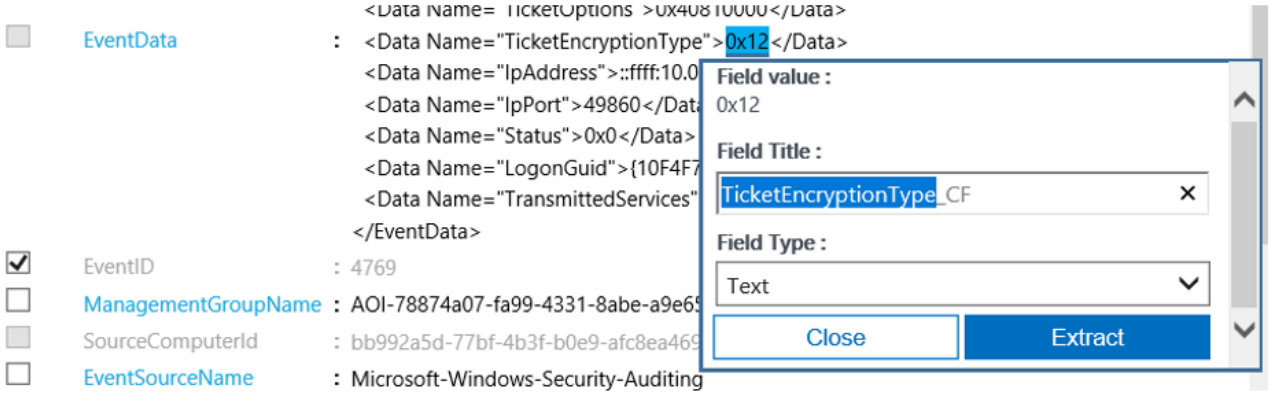
- First step: Execute the following query in Log Analytics:

Search “Security” | Where EventID==4769



- Now, we want to create alerts based on specific criteria in the 4769 events rather than alerting on all events that are collected. This can be achieved by creating custom fields and then defining alert rules based on querying these fields.

- Click on one event below and make sure that you’ve selected the fields you want to filter by. Than highlight the area that you want to use as the example for your data. In my case, I’ve highlighted the value after “**TicketEncryptionType**”. When the value is highlighted name the field with a name that would be easily understood for anyone working with Log Analytics.



When 'Extract' is clicked it will lead you to samples of the results visible if you saved that extraction. If you see some results that should not be there you can individually edit them out or simply ignore them. That should help the extraction algorithm in providing better results. Once you are ok with the results click Save Extraction.

Remember that custom field extraction will be applied only on new events.

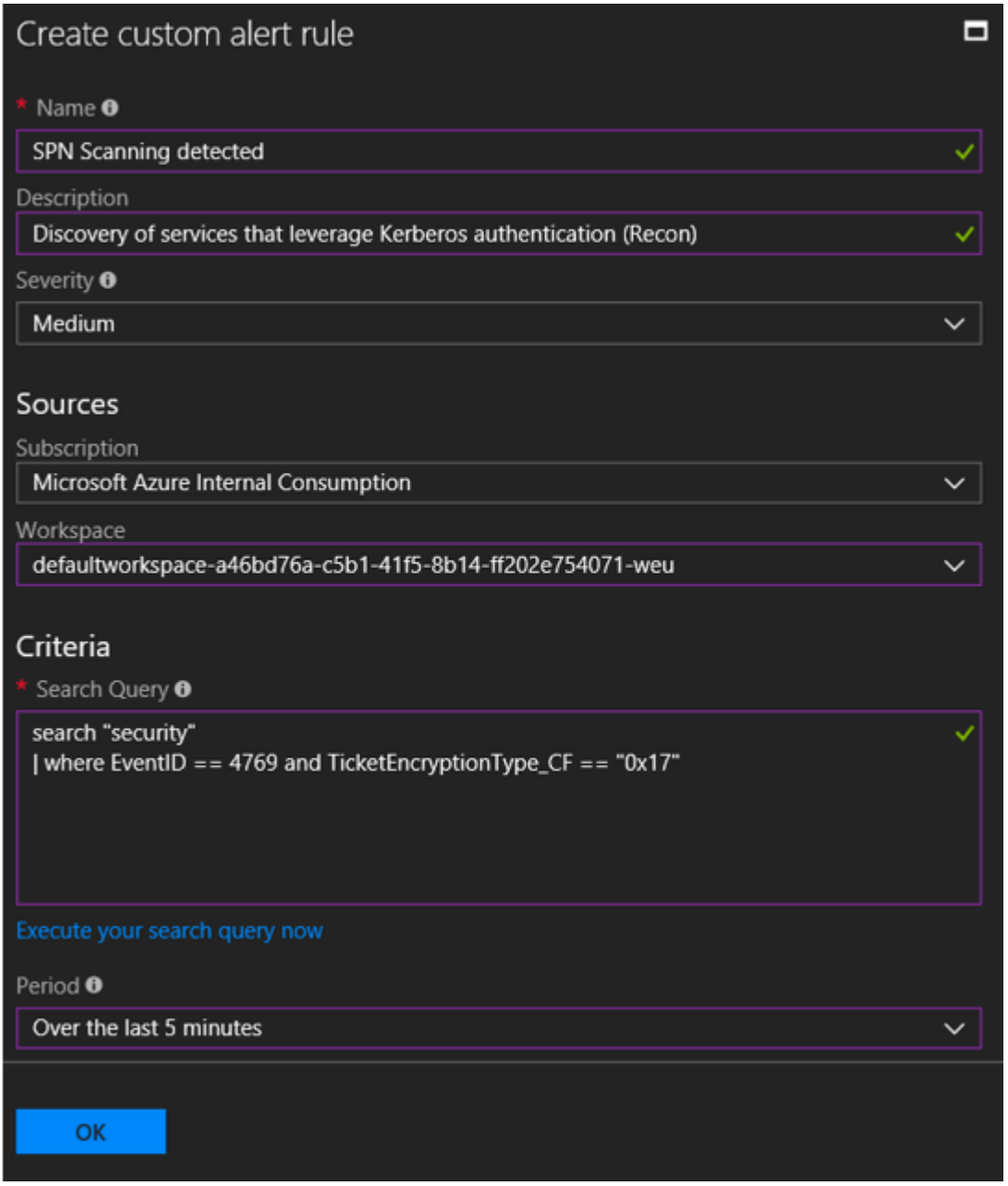
Define custom alerts in Azure Security Center

In the example query above only the highly targeted events are returned and it's very likely that they're malicious. Therefore, we should alert on any events that are being collected and match the specific query.

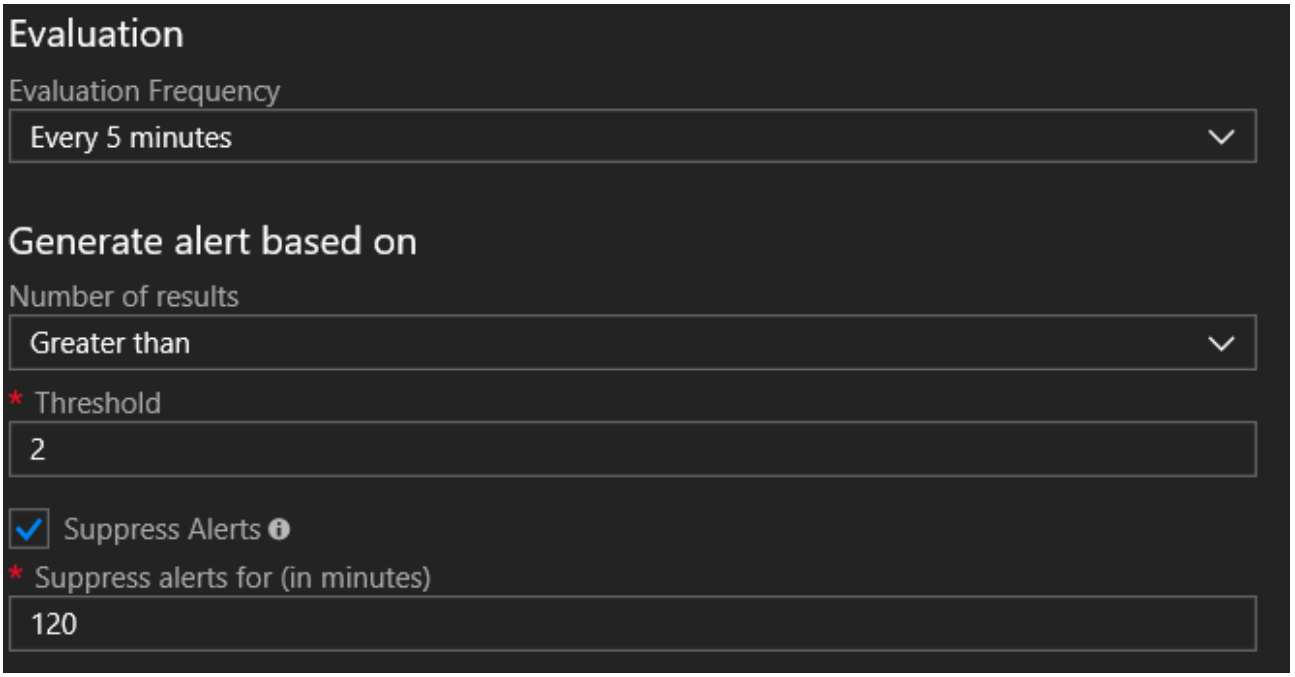
To do so:Open Security Center in the Azure portal, select Customer Alerts, and New Custom Alert Rule, specify the alert details and use the following query

search "security" | where EventID == 4769 and TicketEncryptionType_CF == "0x17"

(Assuming you mapped **TicketEncryptionType** to custom field **TicketEncryptionType_CF**)

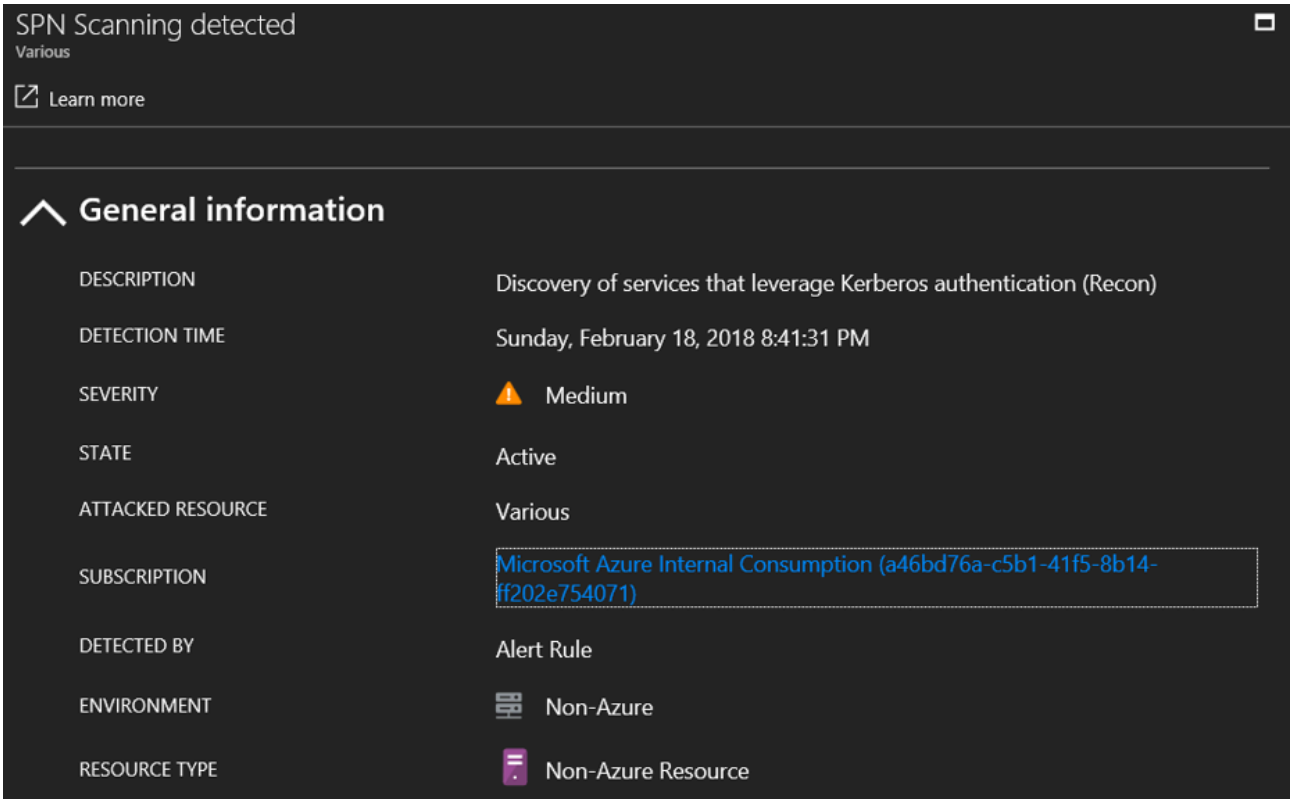


We can configure that only 2-3 consecutive events will triger our alert on the time period we selected:



I wrote the following script to simulate SPN scanning:

As we can see the alert was triggered shortly after the script was executed



Final Thought

The ability to detect advanced attacks is certainly valuable. However, the easiest way to prevent these attacks is to simply use secure practices for handling service accounts:

- Use complex and long passwords for service accounts, and rotate them frequently
- Better option, if feasible, is to use Group Managed Service Accounts - random and complex passwords that can be automatically rotated by Active Directory

---