KustoWorks
SmartSecurity

EXCEPTIONAL DETECTIONS IN YOUR MICROSOFT SECURITY STACK

10 Sep 2024

PresenterInfo | project About

Gianni Castaldi

Supporting customers and their security products

Improving the usage of existing security products

Implementing security new products

Background as an IT Administrator

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Introduction to Detection Engineering

Brief definition

Importance in modern cybersecurity

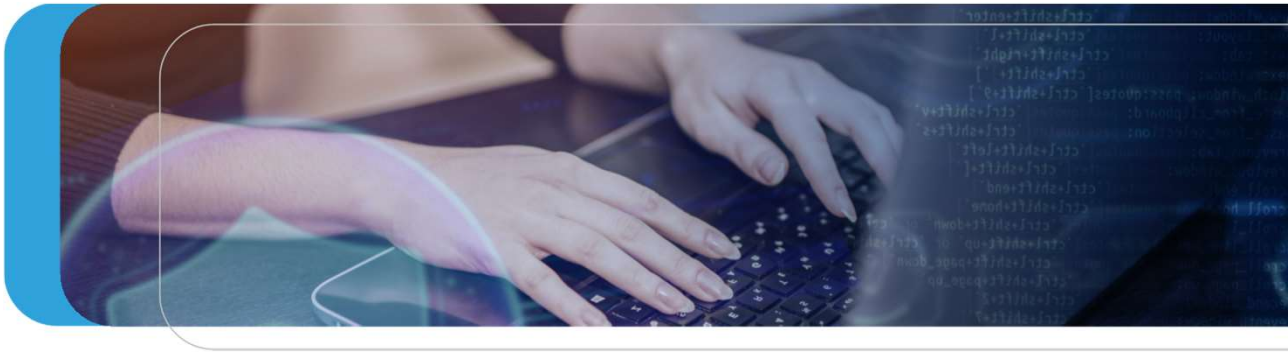
Overview of roles and responsibilities

ROLES IN DETECTION ENGINEERING

- Detection Engineer
- Threat Hunter
- Incident Responder
- SOC Analyst

TODAYS USE CASE

During an incident response we found out that a user clicked on a phishing link in an E-mail. Clicked on the URL and entered their credentials. After entering their credentials, the adversary got access to the user's account.



Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

FUNDAMENTALS OF DETECTION ENGINEERING

- Tools
- Detection engineering process
- Signal Detection Theory

TOOLS

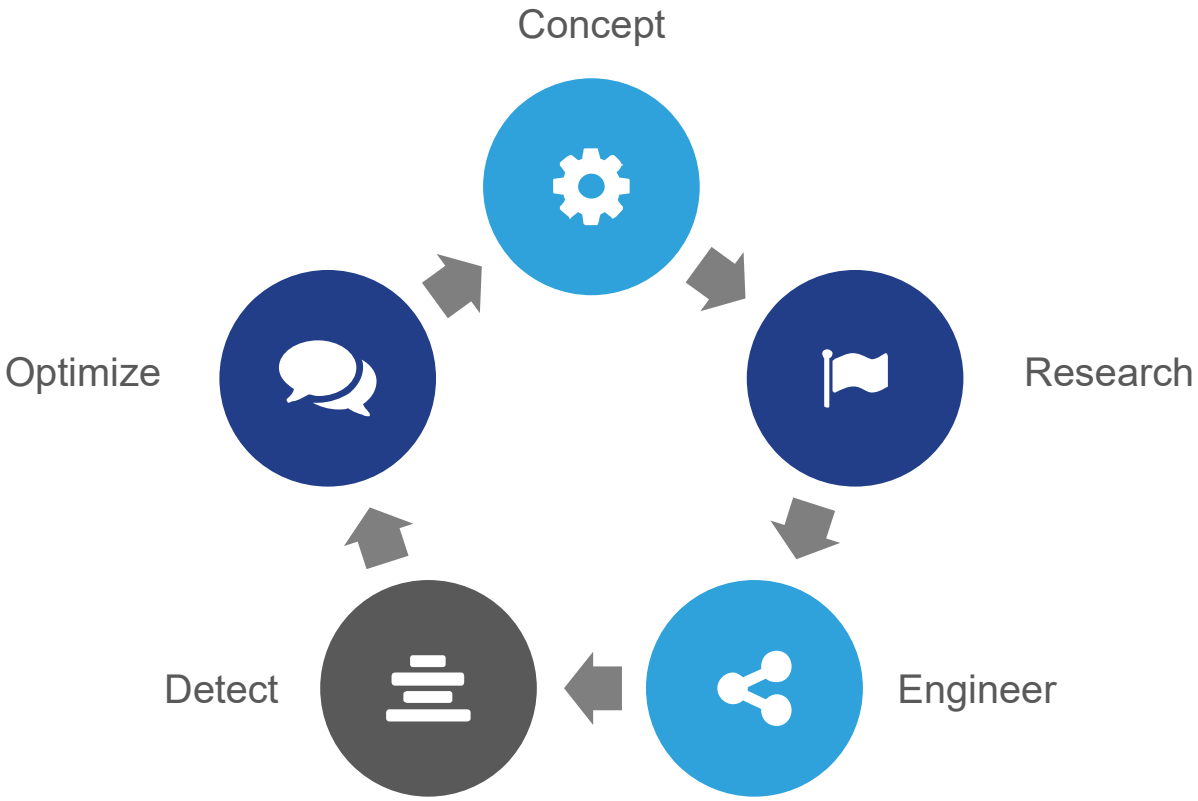
Excel

OneNote

Azure DevOps

Jira

PROCESS



CONCEPT

Item	Description
Title	Name of the detection
Goal	Description of what we want to detect
Scope	What entities should be targeted by the detection
Detection source	What is the source of this detection
Data Source	What is the source of the data
Target Engine	Which engine is the detection made for

SOURCES

Business rules

Incidents

Malware

Offensive Tools & research

Red Team

Threat hunting

Threat Intelligence

CONCEPT

Item	Description
Title	Successful Login Post-URL Click Detection
Goal	Identify successful logins that occur shortly after a user clicks a URL in a phishing email, indicating potential account compromise
Scope	All users
Detection source	Incidents
Data Source	Microsoft Defender XDR
Target Engine	Microsoft Defender XDR

RESEARCH

Item	Description
Strategy Abstract	High level description of what we want to detect
Technical Context	Detailed description for responder to understand the detection
MITRE ATT&CK Tactic identifier	MITRE ATT&CK Tactic ie. Execution
MITRE ATT&CK Technique identifier	MITRE ATT&CK Technique ie.T1059.001
Feasibility	Yes/No
Severity	High Medium Low Informational
Artifacts	What are the items the detection should trigger on

RESEARCH

Item	Description
Strategy Abstract	Detect instances where a user successfully logs in shortly after clicking on a phishing URL from an email, indicating potential account compromise.
Technical Context	This detection identifies correlations between URL click events and subsequent login activities. It uses base64 encoding to match shortened user identifiers within URLs, then cross-references with login attempts within a 15-minute window.
MITRE ATT&CK Tactic identifier	Credential Access
MITRE ATT&CK Technique identifier	T1078 - Valid Accounts
Feasibility	Yes
Severity	High
Artifacts	UrlClickEvents: UrlChain | AccountUpn

ENGINEER

Item	Description
Blind Spots and assumptions	Known gaps and assumptions
False Positives	Known false positives
Validation	How to generate a true positive
Version	What is the current version
Response	3-5 steps to investigate this
Additional Resources	References
Comments	Changelog
Suppressions	Which suppressions are made
Detection Logic	Logic

ENGINEER

Item	Description
Blind Spots and assumptions	When the E-mail is sent to a distribution list with a different domain it will not be detected
False Positives	Claude.ai registrations
Validation	Simulate a phishing scenario where a test user clicks a malicious URL and logs in within the specified time frame
Version	1.0
Response	Verify the login location and timing Check the associated email content for phishing indicators Confirm user activity with the impacted individual Isolate the account if suspicious activity is verified Review logs for further signs of compromise
Additional Resources	
Comments	Initial logic, awaiting feedback and tuning
Suppressions	
Detection Logic	KQL

DEPLOY

Item	Description
Deployable Object	[{}]

```
{
  "displayName": "Successful Login Post-URL Click Detection",
  "schedule": {
    "nextRunDateTime": "2024-09-07T17:39:41.6533333Z",
    "period": "1H"
  },
  "isEnabled": true,
  "detectionAction": {
    "alertTemplate": {
      "severity": "high",
      "mitreTechniques": [],
      "title": "Successful Login Post-URL Click Detection",
      "recommendedActions": "Verify the login location and timing\nCheck the associated email content for phishing indicators\nConfirm user activity with the impacted individual\nIsolate the account if suspicious activity is verified\nReview
      "category": "CredentialAccess",
      "description": "Identify successful logins that occur shortly after a user clicks a URL in a phishing email, indicating potential account compromise",
      "impactedAssets": [
        {
          "@odata.type": "#microsoft.graph.security.impactedMailboxAsset",
          "identifier": "accountUpn"
        },
        {
          "@odata.type": "#microsoft.graph.security.impactedUserAsset",
          "identifier": "accountUpn"
        }
      ]
    },
    "organizationalScope": null,
    "responseActions": []
  },
  "lastRunDetails": {
    "lastRunDateTime": "2024-09-07T16:39:41.6533333Z",
    "errorCode": null,
    "status": "completed",
    "failureReason": null
  },
  "createdBy": "gianni@kustoking.com",
  "lastModifiedBy": "gianni@kustoking.com",
  "@odata.context": "https://graph.microsoft.com/beta/$metadata#security/rules/detectionRules/$entity",
  "queryCondition": {
    "lastModifiedDateTime": null,
    "queryText": "// Define the length for the encoded UPN substring\r\nlet EncodedUpnLength = 20;\r\n// Extract and transform data from UrlClickEvents\r\nUrlClickEvents\r\n| extend AccountUpnLower    = tolower(AccountUpn)\r\n| extend Encodec
  },
  "lastModifiedDateTime": "2024-09-07T16:39:40.9814832Z",
  "detectorId": "5edb2469-2a8c-460c-b5c1-2468efcf49f7",
  "id": "17016",
  "createdDateTime": "2024-09-07T16:39:40.9814832Z"
}
```

OPTIMIZATION

Item	Description
Change Log	What changes are made
Notables	What are the findings of the reviews

OPTIMIZATION

Item	Description
Change Log	Added filtering
Notables	Low false positive rate after adding filtering

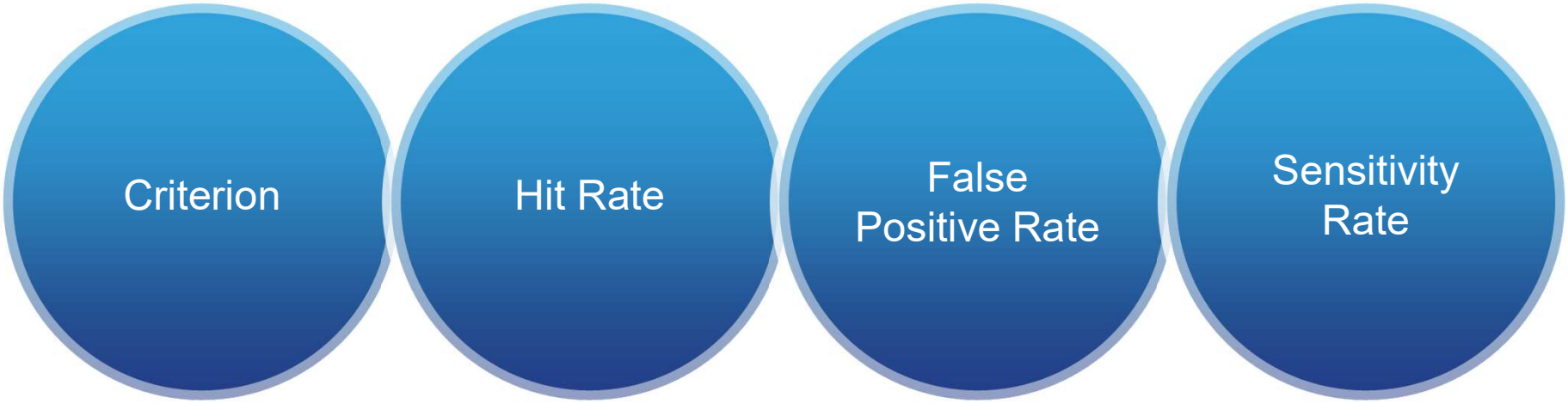
Item	Description
Title	Name of the detection
Goal	Description of what we want to detect
Scope	What entities should be targeted by the detection
Detection source	What is the source of this detection
Data Source	What is the source of the data
Target Engine	Which engine is the detection made for
Strategy Abstract	High level description of what we want to detect
Technical Context	Detailed description for responder to understand the detection
MITRE ATT&CK Tactic identifier	MITRE ATT&CK Tactic ie. Execution
MITRE ATT&CK Technique identifier	MITRE ATT&CK Technique ie.T1059.001
Feasibility	Yes/No
Severity	High Medium Low Informational
Artifacts	What are the items the detection should trigger on
Blind Spots and assumptions	Known gaps and assumptions
False Positives	Known false positives
Validation	How to generate a true positive
Version	What is the current version
Response	3-5 steps to investigate this
Additional Resources	References
Comments	Changelog
Suppressions	Which suppressions are made
Detection Logic	Logic
Deployable Object	[{}]
Change Log	What changes are made
Notables	What are the findings of the reviews

Signal Detection Theory

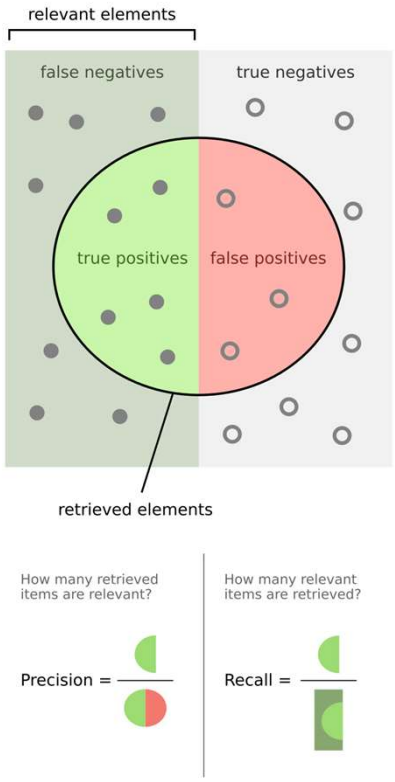
OVERVIEW

	Response present	Response absent
Signal present	True positive	False negative
Signal absent	False positive	True negative

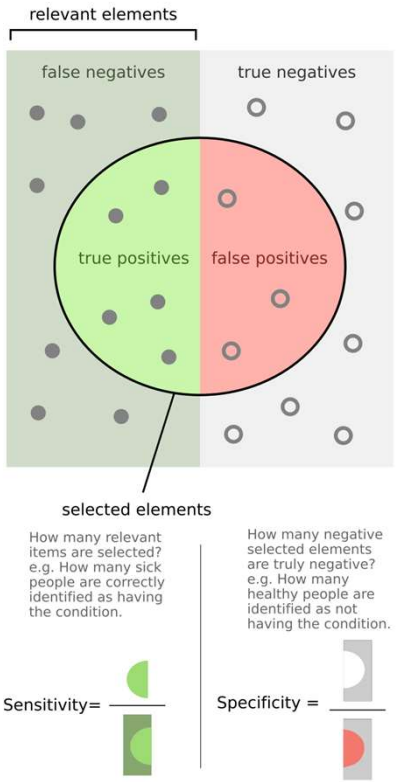
IMPORTANT VARIABLES

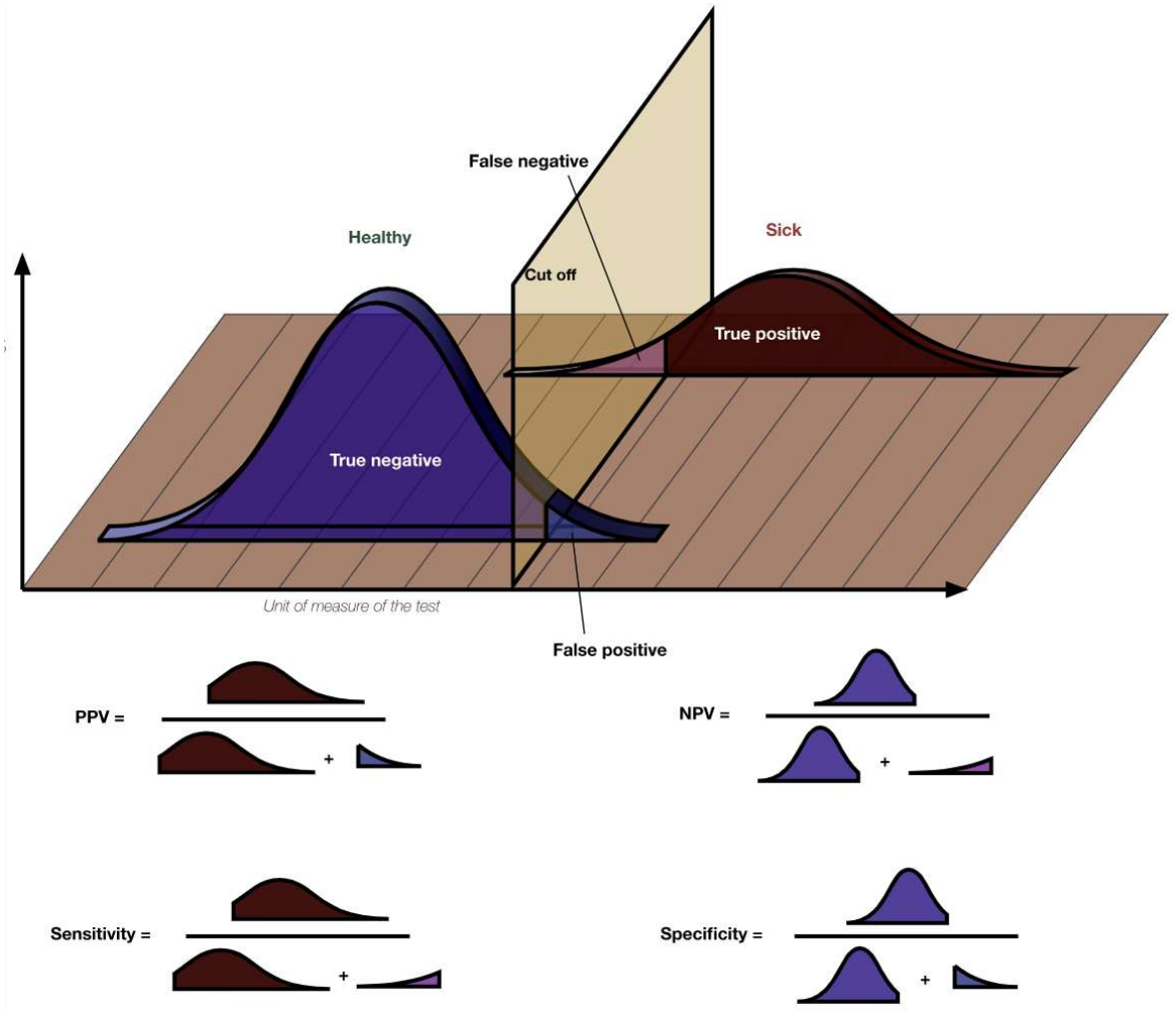


Precision Recall



Sensivity Specificity





DO THE MATH

Hit Rate:

Correctly identified threats = 85

Threats that were missed = 15

False Positive Rate:

Safe activities incorrectly flagged = 20

Correctly identified safe activities =80

THE GOOD

KustoWorks

```python
def calculate_rates(true_positives, false_positives, true_negatives, false_negatives):
    # Calculate Hit Rate
    hit_rate = true_positives / (true_positives + false_negatives)

    # Calculate False Positive Rate
    false_positive_rate = false_positives / (false_positives + true_negatives)

    return hit_rate, false_positive_rate

# Example inputs
true_positives = 85  # Number of correctly identified threats
false_positives = 20 # Number of safe activities incorrectly flagged as threats
true_negatives = 80  # Number of correctly identified safe activities
false_negatives = 15 # Number of threats that were missed

# Calculate rates
hit_rate, false_positive_rate = calculate_rates(true_positives, false_positives, true_negatives, false_negatives)

print(f"Hit Rate: {hit_rate:.2f}")
print(f"False Positive Rate: {false_positive_rate:.2f}")
```

```
PS C:\Users\GianniCastaldi> & "C:/Program Files/Python311/python.exe"
iveRate.py"
Hit Rate: 0.85
False Positive Rate: 0.20
```

THE GOOD

```python
from scipy.stats import norm

# Hit Rate
hit_rate = 0.85 # Hit rate of 85%
z_hit = norm.ppf(hit_rate)  # Outputs approximately 1.04

# False Positive Rate
false_positive_rate = 0.20 # False positive rate of 20%
z_false_positive = norm.ppf(false_positive_rate)  # Outputs approximately -0.84

# Calculate Sensitivity (d')
d_prime = z_hit - z_false_positive  # Sensitivity calculation

# Function to classify Sensitivity (d')
def classify_sensitivity(d_prime):
    if d_prime < 0.5:
        return "Poor Sensitivity"
    elif 0.5 <= d_prime < 1.0:
        return "Fair Sensitivity"
    elif 1.0 <= d_prime < 1.5:
        return "Moderate Sensitivity"
    elif 1.5 <= d_prime < 2.0:
        return "Good Sensitivity"
    else:
        return "Excellent Sensitivity"

# Classification of d'
sensitivity_classification = classify_sensitivity(d_prime)

print("Z-score for Hit Rate:", z_hit)
print("Z-score for False Alarm Rate:", z_false_positive)
print("Sensitivity (d'):", d_prime)
print("Sensitivity Classification:", sensitivity_classification)
```

```
PS C:\Users\GianniCastaldi> & "C:/Program Files/Python311/python.exe"
Z-score for Hit Rate: 1.0364333894937898
Z-score for False Alarm Rate: -0.8416212335729142
Sensitivity (d'): 1.8780546230667041
Sensitivity Classification: Good Sensitivity
```

THE GOOD

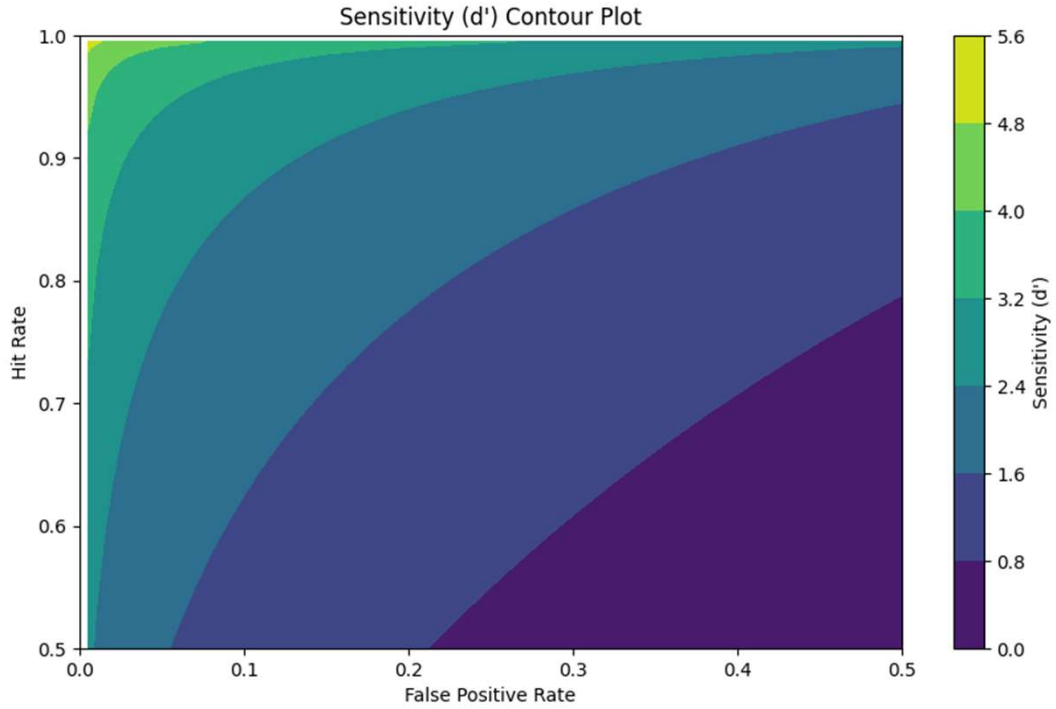
Z-Hit: =NORM.S.INV(B1)

Z-FP: =NORM.S.INV(B2)

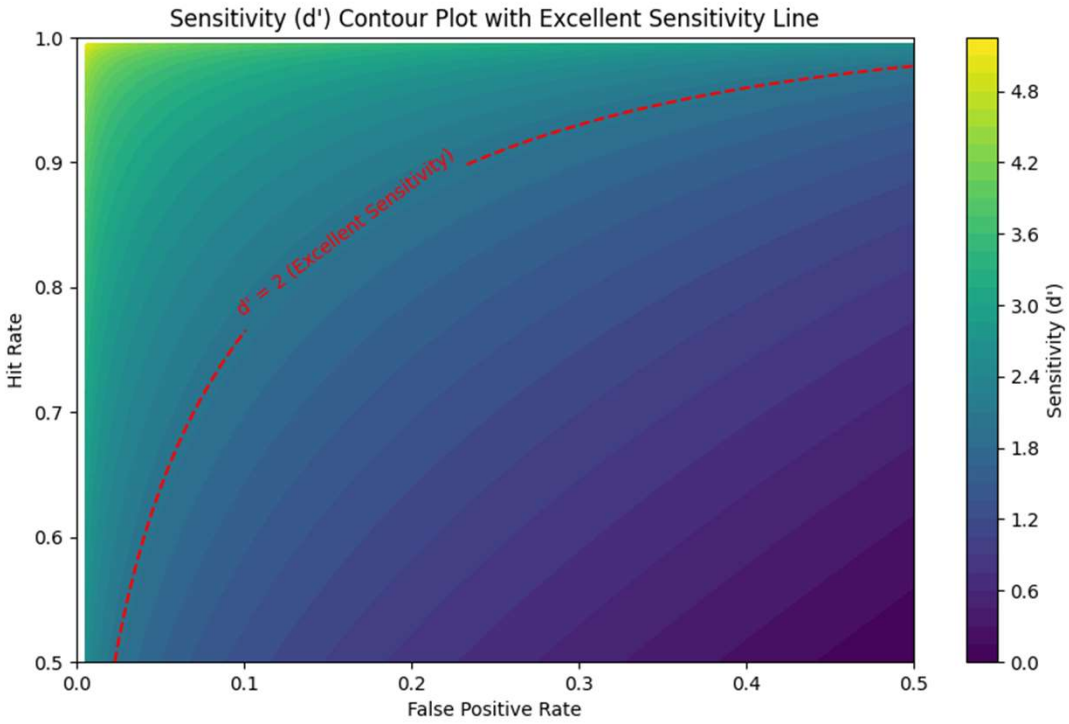
d’: =B4-B5

=IF(B7 < 0,5; "Poor Sensitivity";
  IF(B7 < 1; "Fair Sensitivity";
    IF(B7 < 1,5; "Moderate Sensitivity";
      IF(B7 < 2; "Good Sensitivity";
        "Excellent Sensitivity"))))

THE GOOD



THE GOOD



THE BAD

Run | Time range : Custom | Limit : 1000 | KQL mode

```
SecurityIncident
| summarize
    Total           = count(),
    Empty           = countif(isempty(Classification)),
    BenignPositive  = countif(Classification == "BenignPositive"),
    Undetermined    = countif(Classification == "Undetermined"),
    TruePositive    = countif(Classification == "TruePositive"),
    FalsePositive   = countif(Classification == "FalsePositive") by Title, bin(TimeGenerated,30d), Severity
| extend PrecisionPercentage = round(
    (TruePositive*1.00 / (TruePositive + FalsePositive + BenignPositive))
    ,2
    )*100
| extend Precision = case(
    PrecisionPercentage >= 80, "High",
    PrecisionPercentage between (50 .. 80), "Medium",
    "Low")
| project-away Empty, BenignPositive, Undetermined, TruePositive, FalsePositive
```

Results | Chart | Add bookmark

Title	TimeGenerated [UTC]	Severity	Total	PrecisionPercentage	Precision
'AskToolbar' unwanted software was detected during a scheduled scan on one endpoint	12/17/2023, 12:00:00.000 AM	Informational	5	0	Low
'CandyOpen' unwanted software was prevented on one endpoint	11/17/2023, 12:00:00.000 AM	Informational	3	100	High
'CustomCertEnterpriseBlock' malware was prevented on one endpoint	6/14/2024, 12:00:00.000 AM	Informational	11	0	Low
'EICAR_Test_File' malware was detected on one endpoint	12/17/2023, 12:00:00.000 AM	Informational	3	100	High
'EICAR_Test_File' malware was prevented on one endpoint	3/16/2024, 12:00:00.000 AM	Informational	3	100	High
'Edrblok' hacktool was detected on one endpoint	8/13/2024, 12:00:00.000 AM	Low	1	NaN	Low
'Edrblok' hacktool was prevented on one endpoint	8/13/2024, 12:00:00.000 AM	Low	3	100	High
'ICBundler' unwanted software was prevented on one endpoint	11/17/2023, 12:00:00.000 AM	Informational	4	100	High
'InstallCore' unwanted software was prevented on one endpoint	12/17/2023, 12:00:00.000 AM	Informational	4	100	High
'MalUri' malware was prevented on one endpoint	2/15/2024, 12:00:00.000 AM	Informational	4	100	High

THE UGLY

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

KUSTO QUERY LANGUAGE (KQL)

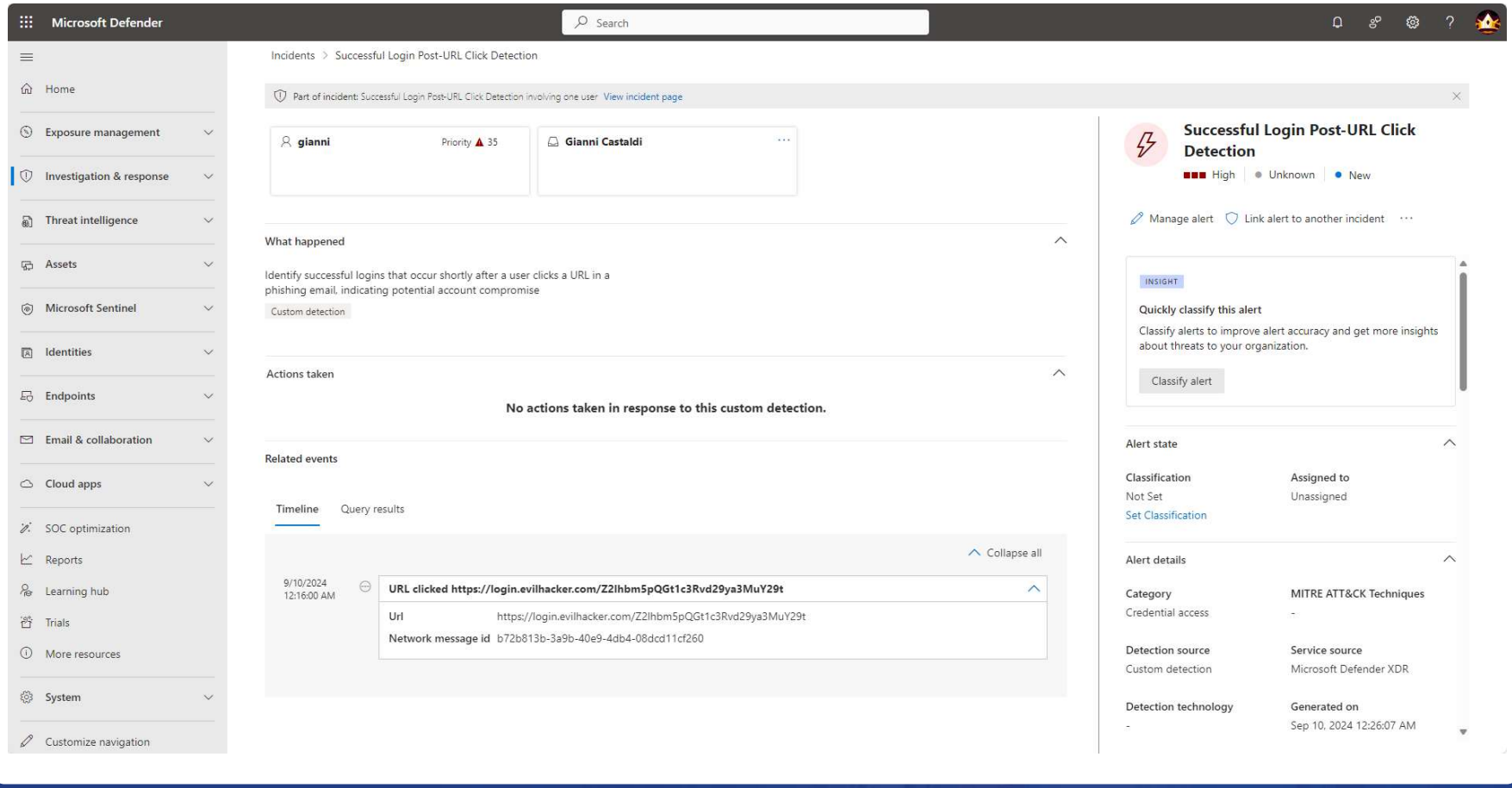
Read Only
Query Language

Next
PowerShell

Used in a lot
of Microsoft Products

Query

```
// Define the length for the encoded UPN substring
let EncodedUpnLength = 20;
// Extract and transform data from UrlClickEvents
UrlClickEvents
| extend AccountUpnLower    = tolower(AccountUpn)
| extend EncodedUpn         = base64_encode_tostring(AccountUpnLower)
| extend ShortUpn           = substring(EncodedUpn, 0, EncodedUpnLength)
| extend AccountDomain      = tostring(split(AccountUpnLower, "@")[-1])
| extend EncodedDomain      = base64_encode_tostring(AccountDomain)
| extend DomainLength       = strlen(EncodedDomain) - 8
| extend ShortDomain        = substring(EncodedDomain, 4, DomainLength)
// Filter events where the URL chain contains the short UPN or domain
| where UrlChain contains ShortUpn or
    UrlChain contains ShortDomain
// Join with IdentityLogonEvents on the lowercased UPN
| join kind=inner (
    IdentityLogonEvents
    | where LogonType in ("Login:login", "Login:reprocess", "SAS:BeginAuth")
    | extend AccountUpnLower = tolower(AccountUpn)
) on AccountUpnLower
// Calculate the time difference between events
| extend Delta = datetime_diff('minute', Timestamp, Timestamp1)
// Filter events where the time difference is between 0 and 15 minutes
| where Delta between (0 .. 15)
// Summarize to get the earliest event by NetworkMessageId
| summarize arg_min(Delta, *) by NetworkMessageId
// Join with EmailEvents on NetworkMessageId
| join EmailEvents on NetworkMessageId
```



Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Agenda

Introduction to Detection Engineering

Fundamentals of Detection Engineering

Kusto Query Language (KQL)

Practical Applications and Best Practices

Best Practices and Common Pitfalls

Review detections every 6 months
Document your detections

Not having a classification framework
Not measuring detections

KustoWorks

SUMMARY

CREDO
Formalize your detection engineering

Signal Detection Theory
Make your detections measurable

KQL
Optimize the performance

REFERENCES

https://detect.fyi/a-research-driven-process-applied-to-threat-detection-engineering-inputs-1b7e6fe0412b

https://digitalcommons.usf.edu/etd/6728/

https://www.wikipedia.org/wiki/Detection_theory

https://www.kqlsearch.com

https:/www.kqlcafe.com

KustoWorks

KustoCon

Learn | Share | Practice

November 8th, 2024

giannicastaldi

castello_johnny

gianni@kustoworks.com