Security

- Tips
  - Microsoft
- Microsoft Sentinel
  - Ingestion Lag
  - Rule Sample
  - Union Alert
  - Splunk to Sentinel Logic
  - Microsoft Security
- Qradar
  - Ignore when another rule matches
  - Qradar expiring whitelist
  - Qradar global whitelist
  - Reference Maps
- ArcSight
- Splunk
  - Splunk Use Case tracker
  - Splunk total run time
  - Splunk Drill Down Events
  - Splunk Results Token
- LogRhythm
  - LogRhythm Or/And Previous
- Mitre Framework
- Cybersecurity Strategy

    - Cybersecurity Mesh Architecture
    - NTLM Brute Force

- Databricks
    - Databricks Training

- Work
    - Jira Story Prompts

Tips

Tips

Microsoft

[Azure AD Authentication and authorization error codes]

MFA Number matching

```
a new security feature called number matching will be enabled which will replace the current Microsoft Authenticator Approval method (push notification). This enhancement is being implemented to protect users against multi-factor authentication (MFA) fatigue attacks (also known as MFA spamming).

Why does this matter?
MFA fatigue attacks rely on a user’s ability to approve a simple SMS or push notification that doesn’t require the user to have context of the session being authenticated. According to Microsoft, the use of simple approvals such as “click to approve” or “enter your PIN to approve” has resulted in a corresponding rise in MFA attacks; this security upgrade will address this identified vulnerability.

What will number matching entail?
During the MFA process, users will be prompted to enter a randomly generated number from the login screen to verify the session/application being authenticated. This change will also show the application that is requesting MFA along with the location from where it is accessed.
```

Microsoft Sentinel

Microsoft Sentinel

Ingestion Lag

```
| where ingestionTime > 5m
```

Setting:

- Run query every 5m
- Lookup data from the last 24 hours
- Stop running query after alert disabled

Microsoft Sentinel

Rule Sample

Log Analytics:

> Logs

Rules

> Analytics

```
OfficeActivity
| where ingestion_time() > ago(5m)
| where tolower(OfficeWorkload) matches regex "onedrive|sharepoint" and tolower(Operation)
matches regex "filesyncdownload|filedownload" and UserId != "app@sharepoint"
| summarize initial_time = min(TimeGenerated), end_time =
max(datetime_add("Second",1,TimeGenerated)), operation=make_set(Operation,5),
workload=make_set(OfficeWorkload,5), file_name=make_set(SourceFileName,5),
file_count=dcount(OfficeId) by bin(TimeGenerated,5m),UserId, ClientIP
| project-rename src_ip=ClientIP, src_user=UserId
| where file_count>500
```

Microsoft Sentinel

Union Alert

```
// The query_now parameter represents the time (in UTC) at which the scheduled analytics rule ran to produce this alert.
set query_now = datetime(2022-03-23T02:57:37.9729472Z);
let Alert1 =
SecurityAlert
| where AlertName == "Unfamiliar sign-in properties"
| extend UserPrincipalName = tostring(parse_json(ExtendedProperties).["User Account"])
| extend Alert1Time = TimeGenerated
| extend Alert1 = AlertName
| extend Alert1Severity = AlertSeverity
;
let Alert2 =
SecurityAlert
| where AlertName == "Atypical travel"
| extend UserPrincipalName = tostring(parse_json(ExtendedProperties).["User Account"])
| extend Alert2Time = TimeGenerated
| extend Alert2 = AlertName
| extend Alert2Severity = AlertSeverity
| extend CurrentLocation =
strcat(tostring(parse_json(tostring(parse_json(Entities)[1].Location)).CountryCode), "|",
tostring(parse_json(tostring(parse_json(Entities)[1].Location)).State), "|",
tostring(parse_json(tostring(parse_json(Entities)[1].Location)).City))
| extend PreviousLocation =
strcat(tostring(parse_json(tostring(parse_json(Entities)[2].Location)).CountryCode), "|",
tostring(parse_json(tostring(parse_json(Entities)[2].Location)).State), "|",
tostring(parse_json(tostring(parse_json(Entities)[2].Location)).City))
| extend CurrentIPAddress = tostring(parse_json(Entities)[1].Address)
| extend PreviousIPAddress = tostring(parse_json(Entities)[2].Address)
;
Alert1
| join kind=inner Alert2 on UserPrincipalName
| where abs(datetime_diff('minute', Alert1Time, Alert2Time)) <=10
| extend TimeDelta = Alert1Time - Alert2Time
| project UserPrincipalName, Alert1, Alert1Time, Alert1Severity, Alert2, Alert2Time,
```

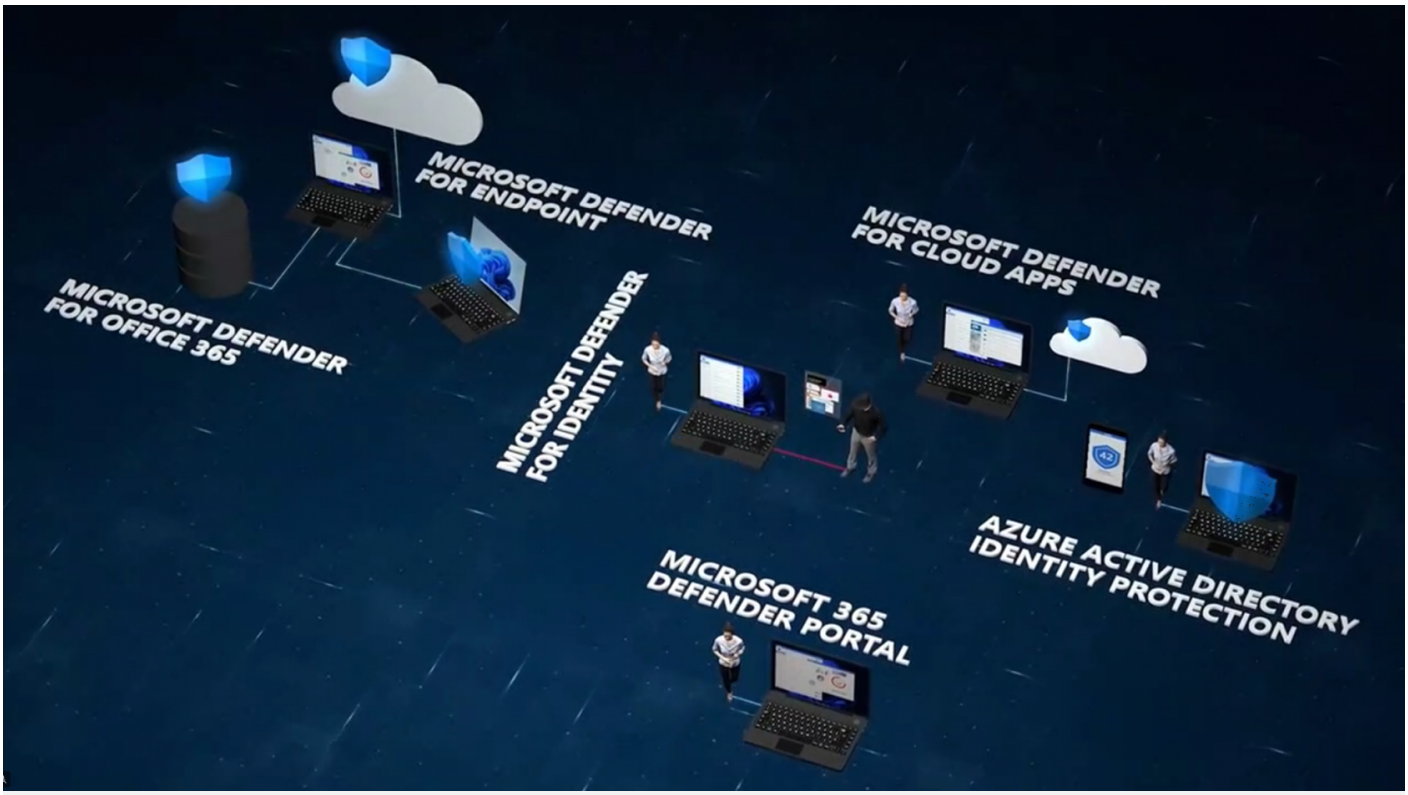
```
Alert2Severity, TimeDelta, CurrentLocation, PreviousLocation, CurrentIPAddress,
PreviousIPAddress
| extend AccountCustomEntity = UserPrincipalName
| extend IPCustomEntity = CurrentIPAddress
```

Splunk to Sentinel Logic

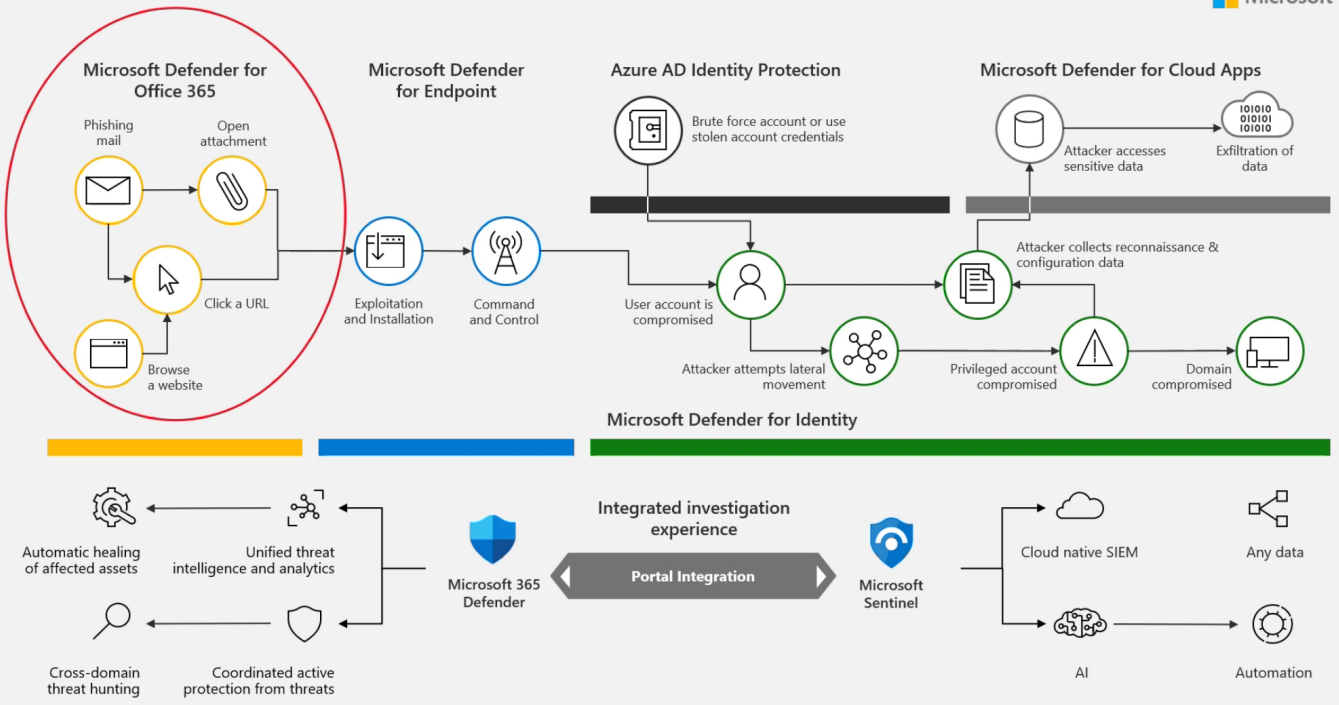
bin time doc

> | summarize initial_time = min(TimeGenerated), end_time = max(datetime_add("Second",1,TimeGenerated)) by bin(TimeGenerated,15m), src_user

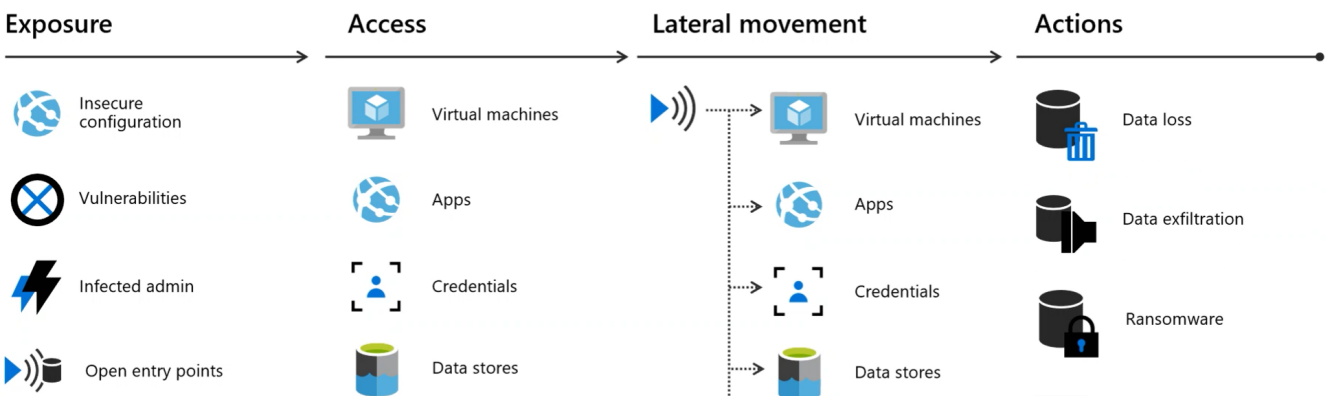
Microsoft Security



Defend Across Attack Chains



The cloud kill chain model



Qradar

Qradar

Ignore when another rule matches

When a building block or rule matches it's specific fields, do not fire matched events

```
and NOT when a subset of at least this number of these rules, in order, from the
same|different source IP to the same destination IP, over this many seconds
```

Qradar

Qradar expiring whitelist

Use AQL filter query

```
username LIKE 'testUser' and LONG(DATEFORMAT(starttime, 'yyyyMMdd')) < 20190429
```

Qradar global whitelist

Use Routing Rules with **forwarding > bypass correlation**

For IP ranges, use Network Hierarchy.

Reference Maps

[http://www.siem.su/docs/ibm/Technical_remarks/Reference_Data_Collections_Technical_Note.pdf]

ArcSight

Splunk

Splunk

Splunk Use Case tracker

```
| rest splunk_server=local count=0 /services/saved/searches
| where disabled=0
| rename action.correlationsearch.label as csearch_label, alert.suppress.period as Throttling, alert.suppress.fields as "Grouped By", action.notable.param.rule_title as "Notable Title", action.notable.param.drilldown_search as "Drilldown Search", updated as "Last Updated", actions.email.to as "Email Recipients"
| search "Notable Title"="*TUC*"
| convert num(Throttling)
| eval "Throttling (Days)"=Throttling/86400
| table csearch_label, "Notable Title", description, search, "Drilldown Search", "Throttling (Days)", "Grouped By", cron_schedule, dispatch.earliest_time, dispatch.latest_time, author, actions, "Email Recipients", "Last Updated"
```

Splunk

Splunk total run time

```
(index=_audit host=* action=search sourcetype=audittrail search_id!="rsa_*")
| eval user = if(user="n/a", null(), user)
| eval search_id=replace(search_id, "'(.*)'", "\1")
| eval search=if(isnull(savedsearch_name) OR savedsearch_name=="", search, savedsearch_name)
| stats min(_time) as _time, values(user) as user, max(total_run_time) as total_run_time,
first(search) as search, first(apiStartTime) as apiStartTime, first(apiEndTime) as apiEndTime
by search_id
```

Splunk

Splunk Drill Down Events

```
earliest=$initial_time$ latest=$end_time$ index=$index$ EventCode=4624 NOT Logon_Type IN ("5")
host=$orig_host$
```

> orig_action_name
> orig_host
> orig_rid
> orig_sid

Splunk Results Token

fieldsummary

LogRhythm

LogRhythm Or/And Previous

OR PREVIOUS works like an OR statement in parenthesis.

a AND b OR PREVIOUS c

would look like

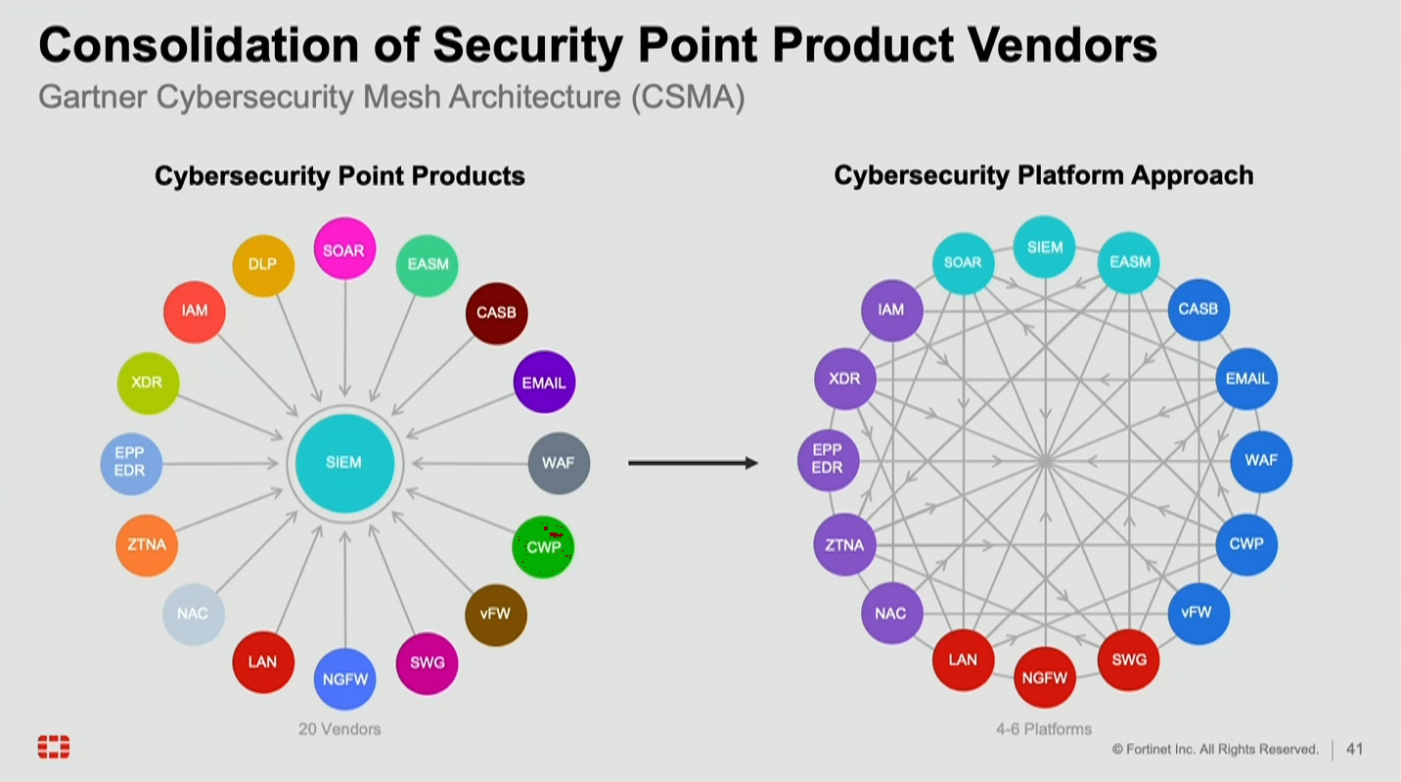
a && (b || c)

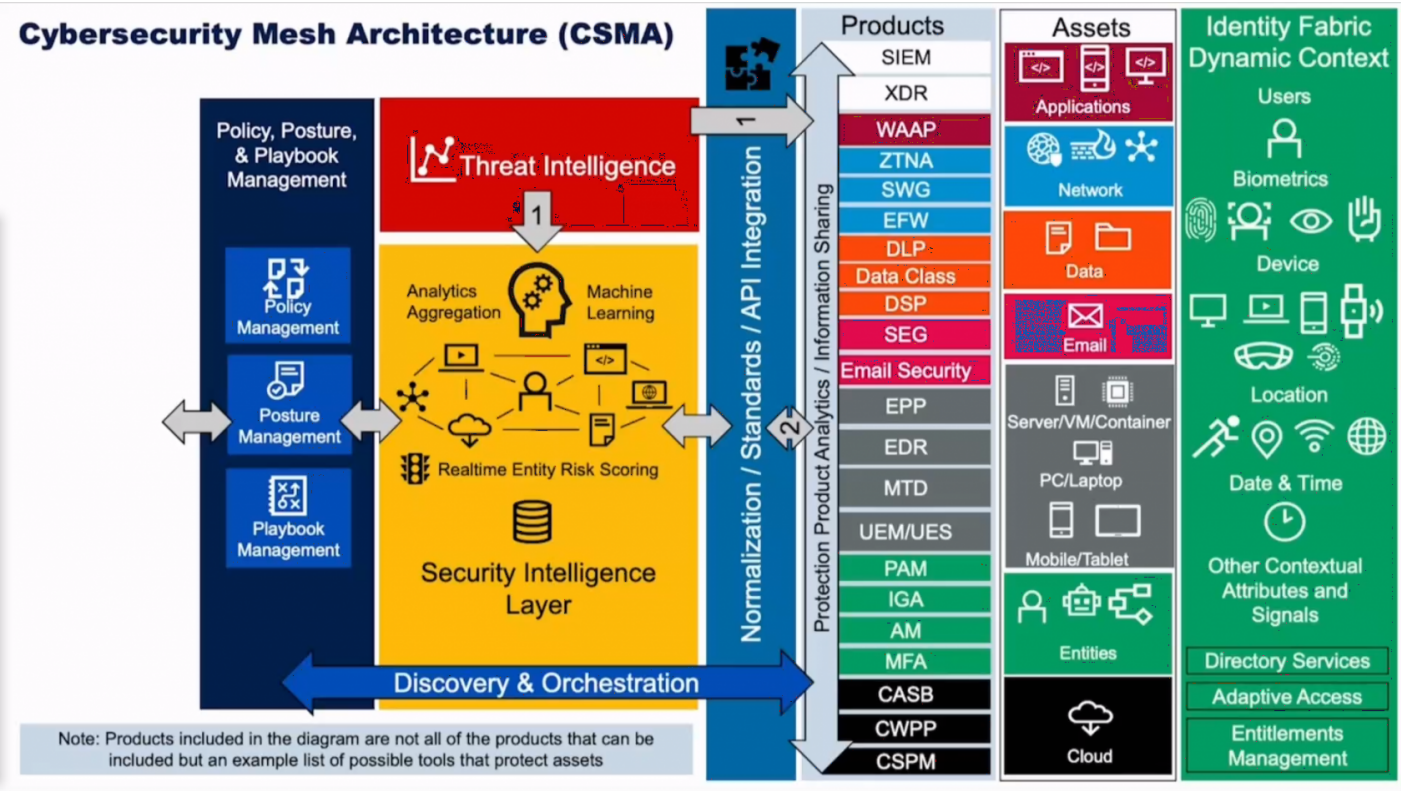
Mitre Framework

Cybersecurity Strategy

Security Topics and Theories.

Cybersecurity Mesh Architecture





Cybersecurity Strategy

NTLM Brute Force

https://www.varonis.com/blog/investigate-ntlm-brute-force

> More specifically, you will need to use Event ID 8004 in Event Viewer to identify the actual device that is on the receiving end of these NTLM brute force attack attempts. Locating the victim device will be the first step in the remediation process.

> 8004 events are typically not enabled by default and may require configuration changes in specific Domain Controller group policies to enable logging.

Databricks

Databricks Training

Partner Databricks training - https://www.databricks.com/learn/training/login

Work

Work

Jira Story Prompts

You are a jira scrum leader, you are to help write out story descriptions together. Below is an explanation of the stories format.

User story template and examples
User stories are often expressed in a simple sentence, structured as follows:

“As a [persona], I [want to], [so that].”

Breaking this down:

"As a [persona]": Who are we building this for? We’re not just after a job title, we’re after the persona of the person. Max. Our team should have a shared understanding of who Max is. We’ve hopefully interviewed plenty of Max’s. We understand how that person works, how they think and what they feel. We have empathy for Max.
“Wants to”: Here we’re describing their intent — not the features they use. What is it they’re actually trying to achieve? This statement should be implementation free — if you’re describing any part of the UI and not what the user goal is you're missing the point.
“So that”: how does their immediate desire to do something this fit into their bigger picture? What’s the overall benefit they’re trying to achieve? What is the big problem that needs solving?

For example, user stories might look like:

As Max, I want to invite my friends, so we can enjoy this service together.
As Sascha, I want to organize my work, so I can feel more in control.
As a manager, I want to be able to understand my colleagues progress, so I can better report our sucess and failures.

This structure is not required, but it is helpful for defining done. When that persona can capture their desired value, then the story is complete. We encourage teams to define their own structure, and then to stick to it.

Ask what I'm doing in my next story, reformat it to fit the user stories style and it should be written in context of someone non-technical reading it.