

# Between a Log and a Hard Place

(mis)Adventures in Azure Log Monitoring

DMITRIY BERYOZA



Slides can be found at <https://github.com/0xd13a/presentations>



# About me

*Dmitriy Beryoza*

@0xd13a on Discord/Twitter

<https://www.linkedin.com/in/beryozad>



- Senior Security Researcher at *Vectra AI*
- Prior to that - Pentester/Secure Software Development Advocate with X-Force Ethical Hacking Team at *IBM Security*
- 25+ years in software design and development
- Presented at BSides SF/LV/Ottawa, HackFest, OWASP meetups, and others
- Ph.D. in Computer Science, CEH, OSCP, CISSP, CCSP
- *Interests*: reverse engineering, secure software development, CTFs

# Why This Talk?

- I do security research and incident investigations in the cloud
  - Concentrate on malicious behavior in Azure ecosystem, specifically Azure AD / Microsoft 365
- Found a number of issues in Azure logs that complicate defense (and may help attackers)
- Not an Azure-bashing talk
  - Some of the same issues exist for other cloud providers
- I'm hoping to:
  - Raise awareness for defenders and red-teamers
  - Raise awareness for Azure so that improvements could be made
- *Opinions are my own, not of my employer*



# Agenda

- Azure/AAD/M365 refresher
- Logs in Azure
- Event monitoring on prem/in the cloud
- Problems observed in cloud logs
- Desired behavior



# Azure, AAD, M365

- Azure is one of the Big Three cloud providers
  - About 23% of market share
  - IaaS, PaaS, SaaS; 200+ services
- Azure is massive so we will refer to just 2 parts of it
- Azure AD (aka AAD, now Entra ID, different from Active Directory)
  - Directory and identity management service, provides *authentication* and *authorization*
  - Provides modern auth, MFA, AD syncing, federation, device management, etc.
  - 425M daily active users (Jan 2021)
- Microsoft 365 (aka M365, Office 365)
  - Leader of SaaS office suite market (~48% in Feb 2022)
  - Cloud-based Office + Teams, Exchange Online, Skype, SharePoint, OneDrive, ...
  - 300M+ monthly active users, 2M+ organizations, >60% users - small businesses



# AAD and M365 in Azure Ecosystem



Azure



# Event Monitoring in Azure

- Event monitoring is based on logs generated by services
- *Microsoft Sentinel* - native SIEM solution, ingests logs from variety of sources
  - Has a variety of built-in detections and allows you to define your own
- *Azure Log Analytics* - log analysis platform
  - Good for ad-hoc investigations for security events
  - Implements *Kusto* – powerful log query and analysis language
- Logs can be analyzed in LA or streamed into an external SIEM
- 3<sup>rd</sup> party solutions can help you monitor and interpret the logs
  - But logs are only generated by Azure

# Available Logs

- Azure AD
  - *Sign-ins* (multiple flavors) – human and non-human account sign-in records
  - *Directory Audits* – record of AAD changes
  - Based on Graph API
  - Exist as a base version (1.0) and a Beta
    - Different schemas, Azure actively uses the beta, but it's unclear if they want customers to migrate to it yet
- M365
  - Management Activity API
  - Separate logs for different types of events:
    - Audit.AzureActiveDirectory, Audit.Exchange, Audit.SharePoint, Audit.General, DLP.All
  - Will be migrated to Graph API eventually?



# Event Logging: On-prem and Cloud

- In on-prem setup we have flexibility in event monitoring
- You could collect logs:
  - On the edge
  - On individual machines
  - Put network taps anywhere on your network
- Logging and monitoring solutions (hardware and software) are replaceable
  - If you didn't like one vendor you could go with another
- In the cloud you don't have a choice
  - Provider fully controls the logs
  - If the logs are high quality – great!
- If you see issues – you are at the mercy of provider's willingness to fix them
  - You can raise them with the vendor
  - Factor them into your data analysis queries

# Log Quality Expectations in the Cloud

- Easy to consume API
- Timely logging of events
- Clear and complete documentation
- Log record completeness (no missing or broken values)
- The following data consistently available:
  - Operation and all corresponding parameters
  - Identity and session information
  - IP
  - Geolocation
  - Device information
  - Threat and reputation indicators
- Correlation between logs
- We want to know who did what and when (and what else they did)
- Unfortunately many of these expectations are not always fulfilled in Azure logs

# Difficulties in Log Consumption

- Schema in some of the logs is fluid
  - New parameter in the operation adds a new column
  - Difficult to consume in external SIEM
    - Have to know which fields are important, and predict new ones appearing
- Sometimes even hard for Log Analytics to handle
  - Have seen errors “limit of maximum 500 columns was reached”
- A better way would have been to have complex fields (JSON) to handle variability
- If you pull on the APIs to consume the events be mindful of delays (discussed later)
  - You may get a snapshot in time and miss late events
  - Or request overlapping timeframes and have to do de-duplication



# Log Flow Issues

- There are occasional log outages in Azure
  - I remember several over the past few years
  - One lasted several weeks and was noticed by accident
- Nothing you can do about it
  - Maybe invest in a solution that tracks flow and makes sure the volume is as expected
- Logs are enabled/disabled in-band – single point of failure
  - Pwned admin can turn off all logging and you will be blind
  - Stealthier attacks are also possible:
    - Turn logs off selectively
    - Turn off audit logging in Exchange for specific users
  - Needs to be a better protected privileged operation

# Log Flow Issues: Impact

- **Defenders:** As with many other things, limit the number of people that have the rights to adjust logging; monitor log flow and operations such as `Set-AdminAuditLogConfig` and `Set-Mailbox -AuditEnabled`
- **Attackers:** Turning off the logs is one of the first things you could do when you getting access, turning them off selectively gives more stealth.
- **Azure:** Because of significance of disabling logs more protections are needed for these operations

# Event Delays

- SLA for log ingestion is up to 15 minutes
  - In reality some of the events may arrive later (sometimes *much* later) and out of order
  - In Graph API-based logs I see delays up to 40 minutes
  - In M365 – sometime more than 24 hours
- Delay hurts defenders - extends possible response time
  - Add together time for initial ingestion, time to import into SIEM and analyst response times - and you are late by design
- Advanced attackers with automated exploitation will be ahead of the game
  - Example: Gamaredon attackers steal data within 30 min of breach

Operation_s	max_delay
> AlertTriggered	1.14:32:23.6086490
> Get-ProtectionAlert	1.06:49:34.6184966
> QuarantineViewMessageHeader	1.06:23:04.8198977
> Get-QuarantineMessageHeader	1.06:18:47.5866602
> AtpDetection	1.01:00:56.4671624
> TeamsSessionStarted	17:06:44.6838551



# Event Delays: Impact

- **Defenders:** You may need to run some stats to see if the events you are monitoring are arriving within the timeframes you assume they will
- **Attackers:** Study the log times of the events your actions may be triggering. If you move fast with automated tools you can stay well ahead of the defenders.
- **Azure:** Please minimize the lag between the event occurrence and the time it is available in the logs

# Logging Tax

- A number of logging event types are only available with higher-paid tiers (e.g. E5, P2)
  - For example: MailItemsAccessed events, risk details in sign-ins log, etc.
- Dilemma for customers - choosing between saving \$\$\$ and visibility
  - *"Charging people for premium features necessary to not get hacked is like selling a car and then charging extra for seatbelts and airbags,"* U.S. Senator Ron Wyden
- Most recently – a problem in [Storm-0558 breach](#), IR was complicated by unavailable logging
- Microsoft agreed with CISA to [expand access to logging](#) in Sept. (great!)
- Its still a problem until then, and there will be a time lag before they are adopted widely

# Logging Tax: Impact

- **Defenders:** Not much can be suggested here, it's a balancing act between affordability and visibility
  - You could supplement some of the missing data (e.g. IP reputation) through 3<sup>rd</sup> party sources and your own analytics
- **Attackers:** Take advantage of the information that does not get logged for common users, most likely won't get noticed
- **Azure:** Hopefully with new changes will become less of an issue
  - Consider not charging for fundamental security functionality



# Deficient Documentation

- While the APIs are documented pretty well, the log event documentation is lacking
- For many events the only thing that's documented is the operation name
  - Individual fields and their meaning are a mystery
  - You have to hunt through blogs and discussion boards for hints
- There are a number of enum values that appear that are poorly documented:
  - Authentication types
  - Status codes
  - Operation subtypes

# Deficient Documentation: Impact

- **Defenders:** Can't adequately monitor some of the events; with no knowledge of what some obscure IDs mean you will likely ignore them in your queries
- **Attackers:** Through experimentation you can find the obscure cases and take advantage of them, defenders most likely won't know how to interpret them
- **Azure:** Fill in the blanks by documenting all logged information fully

# Unannounced Changes

- New event types may appear unannounced
  - Example – new error code added for passwordless sign-in failures
- What's more troubling is that existing events may *disappear* or their format may change
  - Example:
    - 'MailboxLogin' event disappeared from Exchange logs
    - Sign-in attempts with incorrect username are no longer logged
- This will make your monitoring queries stop catching events without a warning
  - Worst kind of failure – when you don't know you have a problem



# Unannounced Changes: Impact

- **Defenders:** You may need to invest into logic to monitor log integrity, and keep a close eye on changing documentation.
  - This only partially helps, and most defenders don't have that time to invest
- **Attackers:** Subtle changes to logs may help you hide if defenders are slow to learn about them
- **Azure:** Please document and announce all changes

# Some Events Are Never Logged

- Cloud is a recon paradise – “Get”-type events are not logged (with rare exceptions)
  - (and this is not just an Azure problem)
- When attacker gets access, they can enumerate the environment: users, services, configuration – without being seen
- Possibly due to space saving reasons, but no option to have such logging enabled
- I ran a whole bunch of open source M365/AAD red team tools and most of them don't leave a trace in the logs

# Some Events Never Logged: Impact

- **Defenders:** Be aware of this, you are essentially blind to someone snooping around until they try to make changes
- **Attackers:** Have a ball with recon – you are invisible (with some exceptions, experiment to see if your TTP leave a trace)
- **Azure:** Please consider at least giving us an option of getting enumeration events logged, defenders will find them very useful



# ID Inconsistencies

- User IDs are not always stable, same user can show up in the logs as:
  - `john.smith@company.com`
  - `John.Smith@company.com`
  - `AN045789@company.com`
- Exchange throws a wrench in the works by adding names like:
  - `"NAMPR07A006.PROD.OUTLOOK.COM/Microsoft Exchange Hosted Organizations/company.onmicrosoft.com/0cf179f8-0fa4-478f-a4cc-b2ea0b18155e"` on behalf of `"NAMPR07A006.PROD.OUTLOOK.COM/Microsoft Exchange Hosted Organizations/company.onmicrosoft.com/John Smith"`
  - ...and others
- Different names in different logs: `user_principal_name` vs `user_id`, `user_id` vs `user_key`
- Occasionally user ID is “Not Available”
  - Correlation may be better by unique user key, but you need human readable ID nonetheless

timestamp	operation	user_id	user_key
2023-06-07 09:13:07.000 UTC	UserLoginFailed	g[REDACTED]e	9be3c3ec-ff48-[REDACTED]
2023-06-13 15:07:26.000 UTC	UserLoggedIn	Not Available	9be3c3ec-ff48-[REDACTED]

# ID Inconsistencies: Impact

- **Defenders:** Be ready for inconsistencies, develop queries thoughtfully
- **Attackers:** Experimentation is required, but inconsistencies in user identification can be used to fool unsophisticated monitoring logic
- **Azure:** Please make things consistent and easy to understand across the board

# IP Inconsistencies

- IP addresses are a mix of plain IPv4 and IPv6 most of the time, but there are some exceptions:
  - IPs with port numbers (both 4 & 6)
  - Empty IPs
  - Local & private IPs: 127.0.0.1, 192.168.\*, 10.\*
  - All zero IPs: 0.0.0.0
  - All 255 IPs: 255.255.255.255
- Some IPs are for Azure infrastructure, not the user performing the operation – give no information on the user location
- IPs sometimes cannot be correlated to sign-in records
- IP reputation is available *sometimes (subject to logging tax)*

```
"ip_address": "255.255.255.255"}}
```



# IP Inconsistencies: Impact


- **Defenders:** Expect these problems in your hunting queries and alerts, remove port numbers, do not expect IP to always be correct
- **Attackers:** Study what events don't log IPs correctly, this could be used to your advantage (e.g., to break correlation between events)
- **Azure:** Make things consistent please, and provide IP reputation for all

# Geolocation Inconsistencies

- Microsoft does provide geolocation, but only for sign-in records
  - It would have been useful for other tables too, you cannot always correlate to sign-in IP
- Occasionally there are hiccups:
  - In March all IPs geolocated to Uzbekistan

## Microsoft breaks geolocation, locking users out of Azure and M365

Customers banished to an IP address in Uzbekistan that Redmond's cloud did not recognize

 [Simon Sharwood](#)

Fri 24 Mar 2023 / 05:57 UTC

- Other imperfections are more stable and expected:
  - Missing geolocation information

### location

```
▶ {"city": "", "state": "", "country_or_region": "", "geo_coordinates": {"altitude": 0, "latitude": 0, "longitude": 0}}
```

- Same IP resolving to multiple locations (most likely by design, a mobile device)
- Geolocation is easily fooled by TOR, VPNs, proxies, cloud provider IPs – take with a grain of salt

# Geolocation Inconsistencies: Impact

- **Defenders:** Useful tool, but consider abovementioned caveats, maybe invest in 3<sup>rd</sup> party source of data
- **Attackers:** Geolocation could be used to discover your activities, consider hiding behind VPN/proxies (TOR easily discoverable)
- **Azure:** Please provide higher quality geolocation, and provide it for all logs



# Device Information Inconsistencies

- User Agent is hit and miss:
  - Some logs parse it and provide piecemeal (but don't give you the original string)
  - Some log the full UA string, but you have to interpret yourself
- Device information is not available for all logs
  - Most likely determined based on User Agent
  - Can sometimes give you OS and Browser version; for registered devices – Device ID, Device Name, Trust Type, etc.
  - Actual values occasionally inconsistent (e.g. “Windows” vs “Windows 10”)
- This information is generally unreliable and can be spoofed, but useful in some investigations
  - Not all attackers are careful to obfuscate their true user agent, and not all can spoof the same UA as the user

# Device Information Inconsistencies: Impact

- **Defenders:** Be aware that this information can be easily faked, but only by careful attackers, it can reveal information about attacker origin and help differentiate them from legitimate user actions
- **Attackers:** Hide your true UA and try to spoof the original user UA to stay stealthy, various strategies are possible
- **Azure:** Please provide all the information you have about the client (i.e. parsed and unparsed) and clean up inconsistencies

# Broken Values

- Some of the log field values have complex structure (e.g. JSON)
  - In hunting and monitoring queries we need to parse them to find specific sub-fields and values
  - Occasionally, due to size limits, values get cut off
  - Example:  
`Set-ConditionalAccessPolicy`
- The resulting queries fail to find events, creating blind spots

```
G\","NetworkId\":"d39dbe05-d99d-4677-ae1f-79427ad6e715\","CidrIpRanges\":null,"CountryIsoCodes\":[\ "AF\","\ "AX\","\ "AL\","\ "DZ\","\ "AS\","\ "AD\","\ "AO\","\ "AI\","\ "\ "AU\","\ "AZ\","\ "AT\","\ "BH\","\ "BD\","\ "BB\","\ "BS\","\ "BY\","\ "\ "BO\","\ "BQ\","\ "BA\","\ "BW\","\ "BV\","\ "BR\","\ "IO\","\ "BN\","\ "\ "CM\","\ "CA\","\ "KY\","\ "CU\","\ "CR\","\ "CK\","\ "KM\","\ "CN\","\ "\ "CO\","\ "HR\","\ "CI\","\ "CW\","\ "CY\","\ "CZ\","\ "CD\","\ "DK\","\ "\ "EH\","\ "GQ\","\ "SV\","\ "ER\","\ "EE\","\ "ET\","\ "FK\","\ "FO\","\ "\ "TF\","\ "GA\","\ "GM\","\ "GE\","\ "DE\","\ "GH\","\ "GI\","\ "GR\","\ "\ "GG\","\ "GN\"..."]
```

# Broken Values: Impact

- **Defenders:** Be aware of this problem. As an alternative to parsing do substring searches
- **Attackers:** Seek out operations and values that may cause this situation, e.g. by triggering an intentional overflow with large values
- **Azure:** Fix this – extend field sizes, log values selectively, log multiple records, log legal JSON



# Log Pollution

- Beyond issues with log quality, logs could also be used for active attacks
- **Log pollution** is a type of attack that allows adversary to control what gets written into the logs, with malicious intent
- Attacker can control some of the field values by presenting a user agent string of their choice, or by choosing command parameters that will be then logged
- Because logs are frequently exported as CSV files, this can be combined with **CSV Injection** (payloads that can trigger command execution in Excel)
- Log pollution with CSV Injection are not new in the cloud
  - Rhino Security Labs demonstrated such attacks in [AWS](#) and [Azure](#) in 2018

# CSV Injection in Sign-in Log

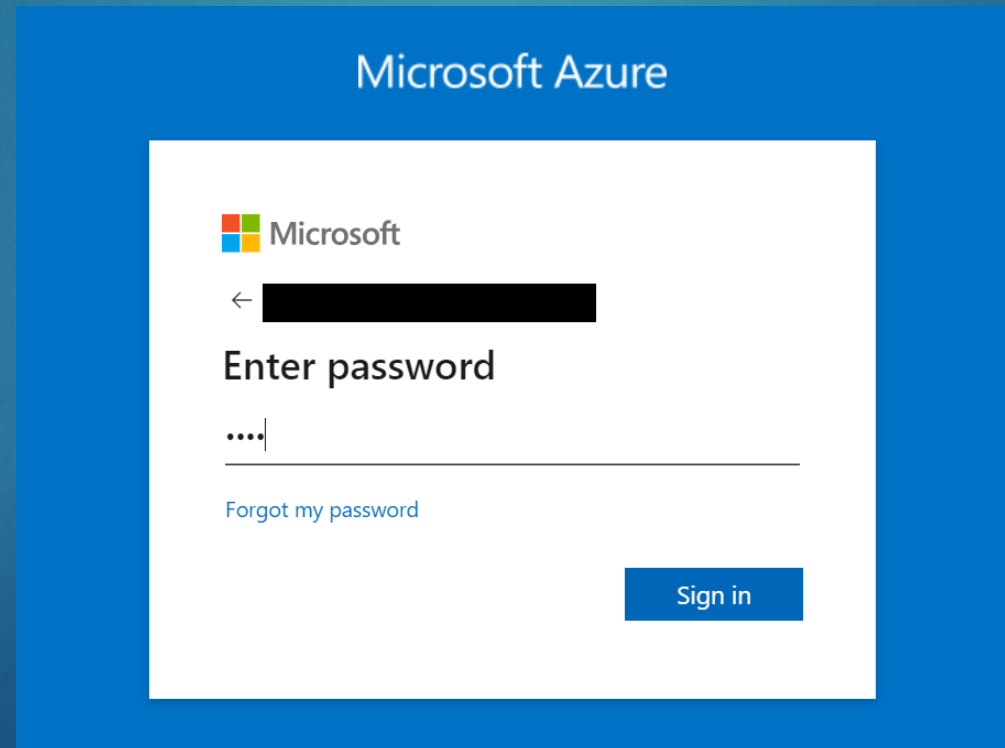
- We found a new (pre-authentication!) instance of this attack in Azure Sign-in log last year
- The idea is to get a CSV injection string through a browser user agent into the logs, and then social engineer an admin to open corresponding CSV file in Excel
- Requires finding a victim with old version of Excel installed or tricking them into disabling Excel “Enable DDE Server” setting:

 Enable Dynamic Data Exchange Server Launch (not recommended)

- Timeline:
  - 2022-09-13 - Reported to MSRC
  - 2022-09-21 – MSRC responded:
    - “...vulnerability really exists in how Microsoft Excel opens files. It also requires clicking-through a warning in Microsoft Excel. We could consider blocking CSV injection payloads as a DND precaution as a fix in Next Version releases.”

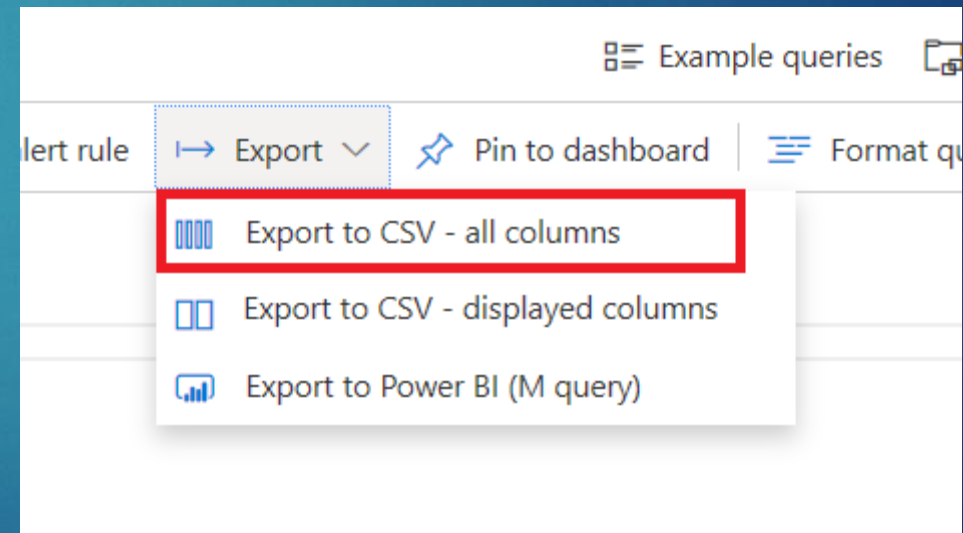
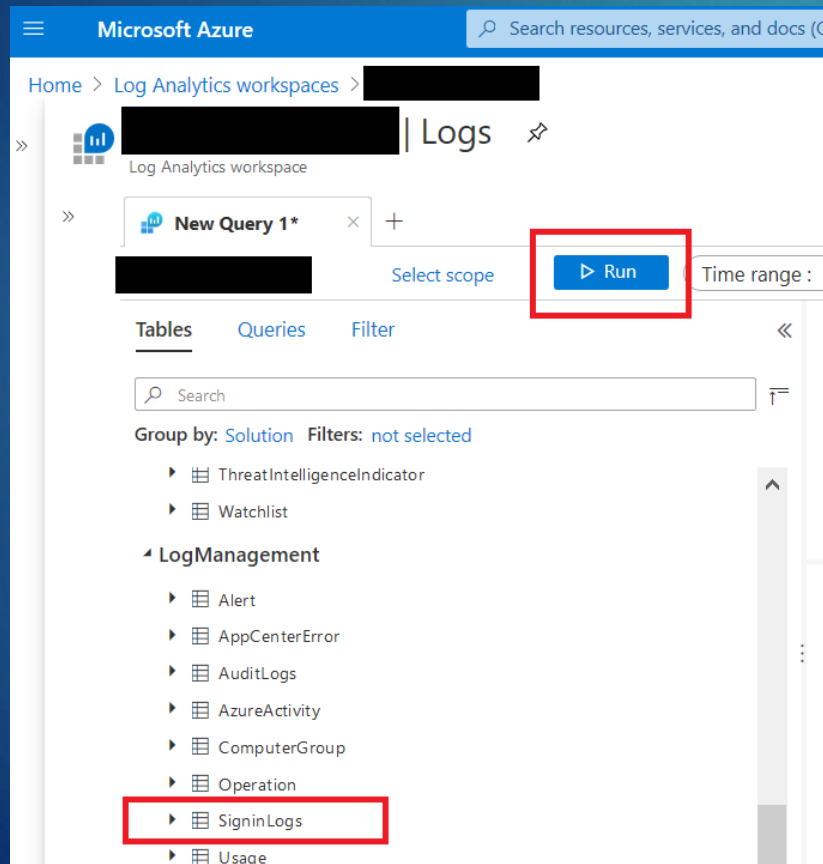
# CSV Injection: Walk-Through

- Set up your favorite browser with a custom User Agent like so:
  - `=msexcel|'\\..\\..\\..\\Windows\\System32\\cmd.exe /c calc.exe'!'A1'`
- Attempt to sign into an existing Azure account with any password:



# CSV Injection: Walk-Through (cont.)

- Social-engineer target to open Sign-in logs and then download them as CSV
- If all goes well the payload will execute





# CSV Injection: Demo

Demo of CSV Injection Attack in Azure Logs

# CSV Injection: Impact

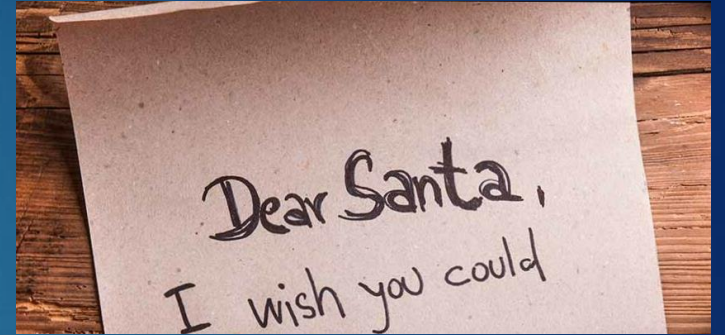
- **Defenders:** Be aware of possibility of log pollution, limit access to who can access logs (logs can be recon goldmines). Social engineering prevention training will be of value to all your staff, including IT.
- **Attackers:** This attack may not be as easy to pull off because of security improvements made in the Office suite, but does not hurt to try.
- **Azure:** Please consider fixing this. Defense in depth is important, and the payload may be found for other tools that handle CSVs, e.g. LibreOffice.

# Why is This Happening

- *(Speculating...)*
- Logs are usually low priority in development (sometimes an afterthought)
- Fixes to known issues are de-prioritized in favor of primary service functionality
- Some of the logs come with legacy products integrated into the environment – messy
- Provider has no strong incentive to make changes – no alternative
- Whatever the real reason is – things need to improve
  - Quality cloud logs are our only hope for catching attacks

# What I Think Should Happen

- At a minimum, non-intrusive changes:
  - Full documentation for all events, fields, enum values
  - Log structure and contents should be treated like an API
    - Versioned
    - No announced changes (additions or removals)
  - Logs are timely (available in seconds), no delayed or reordered events
  - All values are clean, correct, non-ambiguous
  - Logs contain information helpful for IR (e.g., IP reputation)
- At a maximum, logging will benefit from a refactoring
  - It is possible to do, AWS and Okta seem to have more streamlined solutions
- I just want for the log to provide robust and correct signal
  - It's hard enough to try and detect attacks without having to work around all these issues





# Conclusion

- Logging in the cloud environment has a lot of advantages in security monitoring if done robustly
- Lack of alternatives is by design and puts additional importance on log quality
- Logging in Azure has a number of issues:
  - **Defenders** need to be aware of them and sometimes work around them
  - **Red-teamers** could take advantage of them to confuse defense and complicate discovery
  - **Azure** needs to pay attention to them and fix them in order to help their customers stay secure

# Q&A

@0xd13a on Discord/Twitter <https://www.linkedin.com/in/beryozad>

Slides can be found at <https://github.com/0xd13a/presentations>

Thank you!

