



# SECURING CLOUD DEPLOYMENTS

A Red Team Perspective





# About Matt

- Senior Red Team Lead – Microsoft
- Author: Pentesting Azure Applications
- GPEN, GWAPT, SEC545, CCSK, ...



@mattburrough

[linkedin.com/in/mburrough](https://www.linkedin.com/in/mburrough)

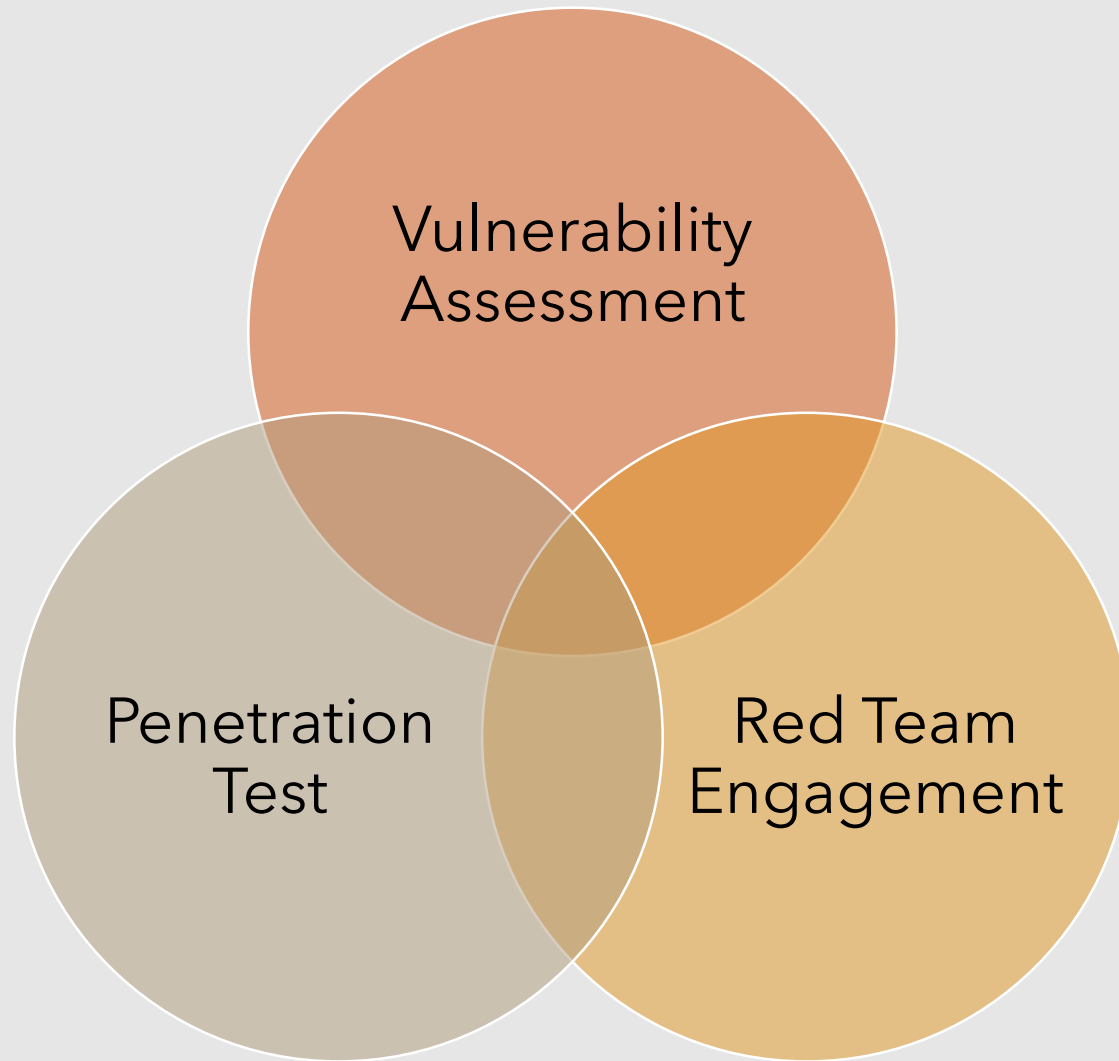


# RED TEAM 101

A Quick Review/Primer

# What is Red Teaming?

- Offensive Security Research
- Model Real World Attacks
- Objective-Oriented
- Complementary to other security controls (code reviews, SDL, auditing)




# Red Team Goals

- Make the Blue Team Better
- Find Flaws Before Attackers Do
- Prove Threats have Real World Impact

# How Does Red Teaming Change in the Cloud?

- Permission, Authorization
- Scope/Rules of Engagement
  - Shared Resources
  - Services, Infrastructure, Metastructure
  - Limits on tools?
- Reporting



# FREQUENT FINDINGS



# Lift and Shift Gone Wrong

- Controls that used to be sufficient aren't anymore
- Miss out on Cloud-First benefits like scaling
- Taking for granted previous controls, like patching, monitoring, security policies?

# Improperly Configured Storage

- Did you mean to leave that blob open to the world?
- Doing key management properly?
- Accounts and permissions probably not what you're used to...
- Encryption at rest? In transit?
- Data retention?
- Just because you can doesn't mean you should.

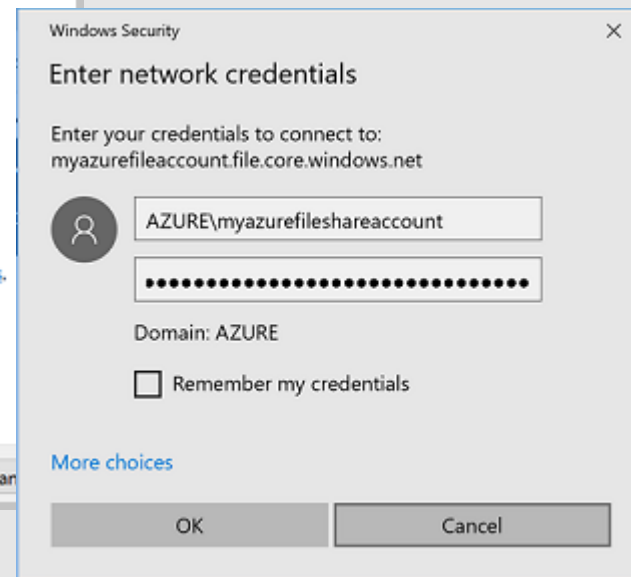
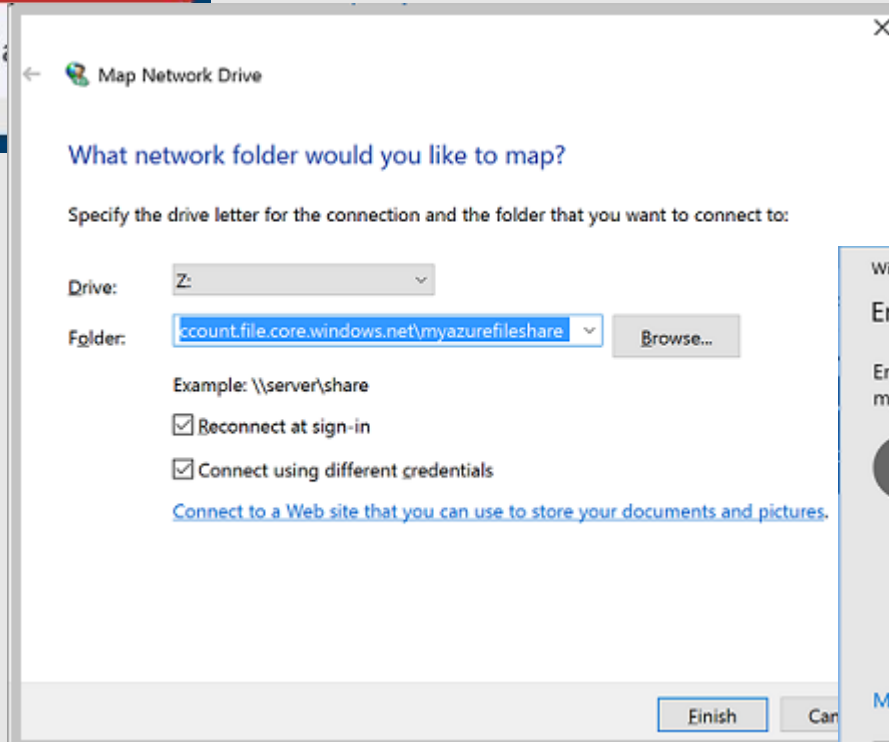
# Connect

myazurefileshare

## Connecting from Windows

To connect to this file share from a Windows computer, run this command:

```
> net use [drive letter]  
\\myazurefileaccount.file.core.windows.net\myazurefiles  
/u:AZURE\myazurefileaccount  
/r:mehLWRwJkxSZTBfs8QFd7Xl3qjwF8Tojea2Eu4BfT0e4/
```



# Secrets in Source

- Code, Configs are moving to the cloud, too.
- Accessing Cloud APIs means developers may be putting more secrets into code.
- Are unredacted secrets exposed?
- Encoding isn't Encryption
- Attackers can now find them at scale.

storagekey ext:config



Showing 50 out of 100 results

[Provide feedback](#)**App.c...** 4 matches

3 matches

2 matches

2 matches

2 matches

2 matches

2 matches

2 matches

2 matches

1 match

1 match

1 match

## App.config

Preview pane Right

[Contents](#) [History](#) [Compare](#)

Download



```
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60 <appSettings>
61   <!-- PAF settings | AppName | DEV / TEST / PROD -->
62   <add key="AppName" value="..." />
63
64
65
66
67   <!--<add key="AppServiceURL" value="..." />
68   <add key="AppServiceURL" value="..." />
69   <!-- PAF settings | Client Thumbprint | DEV_TEST / Prod -->
70   <add key="ClientThumbprint" value="222BE..." />
71   <!--<add key="ClientThumbprint" value="8..." />
72   <!--<add key="StorageAccount" value="..." />
73   <add key="StorageKey" value="IKF..." w="..." />
74   <add key="StorageAccount" value="..." />
75   <add key="StorageKey" value="uV1..." w="..." />
76   <!--
77   <add key="StorageAccount" value="..." />
78   <add key="StorageKey" value="jZV..." g="..." />
79   <!--
80   <!--<add key="StorageAccount" value="..." />
81   <add key="StorageKey" value="Bl..." i0="..." />
```



Thread: Today, I am going to show you how much data is leaked on [@github](#) and how easy it is to query that data by malicious actors using services such as Google BigQuery.

[@mzack9999](#) [@emgeekboy](#) [#infosec](#) [#BugBounty](#) [#DevOps](#) [#Security](#)

9/8/18, 11:08 PM

135 Retweets 211 Likes



**Ice3man** @Ice3man543 · 1d

Our journey begins with a simple idea - to query all possible AWS keys leaked on Github. Thankfully, [@Google](#) gives \$300 free credit on BigQuery, so not a problem. We start by creating a regex for AWS Keys.



**Ice3man** @Ice3man543 · 1d

And bam! In a fraction of seconds, we get 9922 AWS Access Keys. These are all from various projects, some may work and some may not. After removing the keys that are invalid, we are left with 7953 keys. Not bad I'd say. We are not disclosing any valid keys for security reasons.

```
ucketGETcors", "c_c_pkey  
.json.php", "c_c_pkey": "  
EXAMPLE"}  
XAMPLE"}  
SFODNN7EXAMPLE"}  
iam/iam_test.go", "c_c_pl  
KIAIOSFODNN7EXAMPLE"}  
}
```

1


1

15




# Insecure Network Settings


- Excessively permissive firewall rules
- Management ports exposed to the Internet
- Firewall exceptions for home IPs





## Firewall settings

██████████ (SQL server)

 Save

 Discard

 Add client IP

Deny public network access 


Yes

No




Allow Azure services and resources to access this server

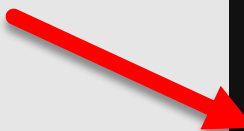
Yes

No

 Connections from the IPs specified below provides access to all the databases in ██████████

Client IP address ██████████

Rule name	Start IP	End IP	
<input type="text"/>	<input type="text"/>	<input type="text"/>	...
Allow All	0.0.0.0	255.255.255.255	...
Home 	<span>██████████</span> 	<span>██████████</span> 	...



# Social Engineering

- Phishing for users', administrators' credentials
- Brand Impersonation
- Improper service cleanup/deprovisioning (Ips, DNS, service names) can let an attacker claim them

# Confusing Authentication for Authorization

- Just because someone has an account doesn't mean they belong here...
- Determining a user's role through user-controlled fields.
- This applies to other identity fields as well.



# Gray Clouds

- Are security standards/policies followed?
- Security monitoring?
- Compliant for regulatory compliance?
- Using an unapproved vendor?



# STOPPING A RED TEAMER

and attackers, too!

# Monitoring

- The best offense is a good defense (who are keeping a close watch on the offense).
- Not just see, but act.
- Need visibility across the whole graph.
- Alerting on a single pane of glass.

# Multifactor Authentication

- Much harder to steal, guess, brute force...
- Not impossible, but attackers like low hanging fruit.
- Make sure you're using across all services. Better yet...

# Use a Unified Identity Solution

- In the cloud, Identity is the new Network Edge
- Single Sign On eliminates a patchwork of user accounts and password policies
- Makes provisioning and deprovisioning simpler, consistent
- Central source of logging, monitoring
- Security features like Conditional Access



# Administrator Account Hygiene

- Just Enough Admin
- Just In Time Access
- Alternate Accounts
- Privileged Access Workstations
- Password Diversification

# Exercise Zero Trust

- Assume Breach
- How much has your network changed in 3 months?
- Defense in Depth

# User Education

- Regular security training for all users
- Emphasize importance of unique passwords, MFA
- Provide phishing simulation exercises
- It's okay to make a mistake, but report it!



THANK YOU!