

CloudTeam - Final Report

Azure Cloud Security Review

June 1st, 2020 - Version 1.0

Prepared for:

Aaron Moreno
Robert Sterne
Dj. Forbes

Prepared By:

Andre Barjesteh
Tanzila Sharin
Shane Scarbrough
Marvin Thai
Thomas Yamasaki

Synopsis



During the months of May and June 2020, CloudTeam conducted a security assessment of XCorp's Azure Cloud Infrastructure that was created for training purposes of their employees. The goal of this was to create a threat model for XCorp's security team and to work with them to enact defense in depth on their cloud architecture. We were given access to their architecture, and their team, to ensure this was completed successfully.

What is being built?

The primary purpose of this threat model is focusing on XCorp's Cloud Infrastructure using Azure, and not necessarily the WebApps that are deployed with them. Below are a list of assets we were able to

- Network Security Group x 1
- Load Balancer x 1
- Availability Set x 1
- MySQL DB
- Docker container to with application
- Virtual Network x 1
- Virtual Machines x 3 running Ubuntu 18.04
- PHP 7.0

The purpose of this setup is to deploy a web application for training the XCorp's Red Team employees to ensure that they are able to identify gaps in the company's security.

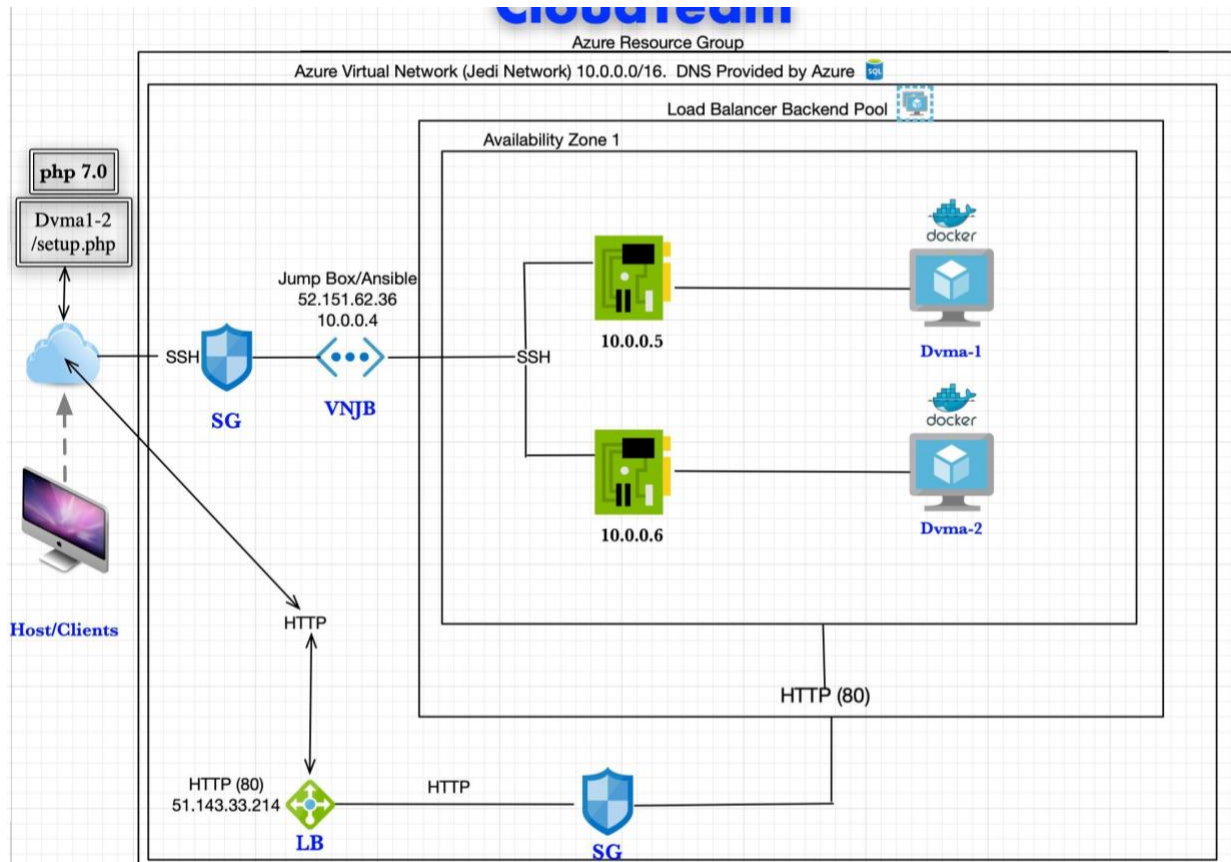


Figure 1: Overview of Cloud Structure

Key Findings

During our investigation of the above architecture and software being used, the CloudTeam utilized the STRIDE model to help us understand and classify possible threats to our system. The following summarizes our key findings:

Spoofing:

The infrastructure is vulnerable to DNS spoofing and malicious users can spoof the training app the Red Team is using.

- Since we have a public IP address within our Azure network, our network is vulnerable to *IP Address spoofing* and *DNS spoofing*.
- Our public IP address has SSH access to the Jumpbox server. If an attacker is able to spoof our public IP address, they can gain the ability to SSH to the Jumpbox server.
- An attacker can also redirect traffic to an alternate website by hijacking our DNS entry. This is most commonly carried out by cache poisoning (e.g. when an attacker injects a forged entry into the DNS server).
 - If the appropriate DNS spoofing detection software is not implemented, such as XArp, thus making our website vulnerable to this sort of attack.

- We should ensure our web app uses end-to-end encryption which will help decrease the chances of being compromised.
- We should also use digitally signed DNS records to help determine data authenticity.

Tampering:

Due to its nature, the web application allows for XSS and SQL injection, which can lead to data loss or control of the system's architecture if security measures are not properly implemented.

- Since the DVWA is a web application, it is generally vulnerable to tampering attacks such as *cross site scripting (XSS)* and *SQL injection*.
- Our web application has several forms and so we must be careful to deny JavaScript, HTML, Flash, or any other type of code that the browser may be able to execute. We should perform a security review of the code and search for all places where input from an HTTP request could possibly make its way into the HTML output.
- Using SQL Injection, an attacker can gain access to the most valuable asset - user and application data. This is a very serious threat that can cause repudiation issues such as voiding transactions or changing balances, or allowing the complete disclosure of all data on the system. Our web app uses PHP, which is especially vulnerable to such attacks due to the prevalence of older functional interfaces.

Repudiation:

From the standpoint of threats to Non-Repudiation, our system focused on flexibility has exposed our system to this threat, specifically:

- Host/Client Terminal: Does not have a digital signature to prevent an attacker from gaining access to the Jumpbox, reference Figure 2 below.
- We cannot adequately determine access to terminal is authorized (is who he/she claims to be)
- This unauthorized entry point may compromise various log files and transactions

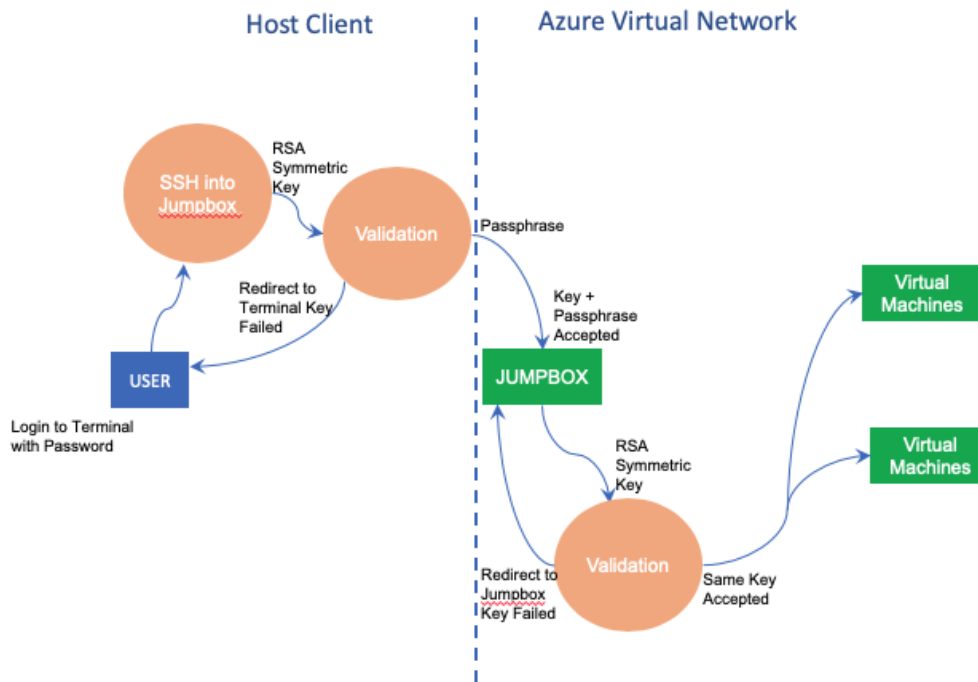


Figure 2: User login process flow

Information Disclosure:

From the standpoint of threats to confidentiality, we have not adopted encryption of our data sites to address mobile or static threats. We have identified the following issues that impact confidentiality threats, shown in Figure 2:

- Host/Client Terminal: No encryption of hardware and only a simple password to unlock the terminal. Theft of this terminal will be a significant breach of our entire architecture.
- Jump Box and VMs: While we implemented key based SSH logins into the Jump Box and VMs via RSA digital Signature Algorithm, only the Jump Box implemented a passphrase.
- VMs do not have a passphrase implemented, which is a possible threat to confidentiality if the Jump Box is compromised.

Denial of Service:

Azure has some advantages related to preventing Dos attacks due to its sheer size and scale regarding Absorption, Detection, and Mitigation. Absorption happens before detection, and detection happens before mitigation. Absorption is the best defense against DoS attacks. If the attack cannot be detected, it cannot be mitigated. But if even the smallest DoS attack can't be absorbed, then services are not going to survive long enough for the attack to be detected. In this context, we have some issues regarding our rate of Absorption:

- Load Balancer: This is limited to only addressing the load between the two VM machines, which can easily be overwhelmed because each instance has only a 4GB RAM capability, which is just enough to support X Corp's training purposes only.
- A coordinated Dos Attack will quickly overwhelm our capacity of absorption of an aggressive attack. To counteract our limited budget regarding our infrastructure, we should consider decreasing our time-to-detection at key points in our system.

- At key points at the VMs and Load Balancer, we do not have a proactive Intrusion Prevention System (IPS) to buy us key time in case of threats to availability.

Escalation of Privileges:

Our current architecture allows the greatest flexibility to meet XCorp demands for flexibility. Despite this ease of adaptability, we did not implement any tiered authorization scheme in our system. We currently do not have a tiered restriction in place for:

- Tier 0: Administrators who manage the identity store
- Tier 1: Administrators, who manage enterprise servers, servers, and applications
- Tier 2: Administrators, who manage devices like desktop, laptops, and printers
- Tier 3: Users

We essentially have a system-- if compromised-- immediately leads to entire ownership by the attacker. Additionally, another threat has recently emerged which does not have a patch issued.

Our system is also vulnerable to a DLL side-loading attack. In this case, an attacker can take advantage of "improperly or vaguely specify a required DLL," which may expose our system to an unintended DLL loaded into a program--specifically:

- An attacker could replace the DLL mstscax.dll in the folder c:\windows\system32, which requires local administrative privileges.
- Attackers could copy mstsc.exe to an external folder, placing the DLL in the same folder and running mstsc from there. This does not require admin privileges, Microsoft says the mstsc should not be used outside the folder c:\windows\system32; however, this is not enforced.

Both scenarios let an attacker bypass security controls because malicious code runs under the context of mstsc.exe, which is a Microsoft-signed executable (Sheridan 2020).

Recommendations

There are a few key areas we can cover to help prevent some of the above listed issues.

We have several ways we're already handling authentication over SSH, but there are a few more practices we can put into place to help prevent any unauthorized authentication. Threat actors generally will be scanning for ports 22, 222, or 2222 as these are commonly used for SSH (Jevtic 2020). In our infrastructure for each VM, we can use a non-default and non-common SSH port, ensuring that it is documented for trainees to use.

To do this, we can change our /etc/ssh/sshd_config on our JumpBox (Figure 3).

```
GNU nano 2.9.3 /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Port 5455
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m

[ Read 125 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo     M-A Mark Text  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line  M-E Redo     M-G Copy Text  M-_ WhereIs Next
```

Figure 3: Set our port to something other than 22 or a common SSH port.

Once this is set, we are able to restart the ssh process, then exit from our box. We can then try to see if our normal SSH login should work. Once we confirm that it doesn't we can try using the new port.

```

theassyrian@DESKTOP-D8R0GUV:~$ ssh RedTeamTrainee@52.250.65.118
ssh: connect to host 52.250.65.118 port 22: Connection refused
theassyrian@DESKTOP-D8R0GUV:~$ ssh -p 5455 RedTeamTrainee@52.250.65.118
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1022-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Jun  4 01:01:01 UTC 2020

System load:  0.05               Processes:            109
Usage of /:   9.1% of 28.90GB     Users logged in:     0
Memory usage: 23%               IP address for eth0: 10.1.0.4
Swap usage:   0%

 * MicroK8s gets a native Windows installer and command-line integration.

   https://ubuntu.com/blog/microk8s-installers-windows-and-macos

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

26 packages can be updated.
0 updates are security updates.

Last login: Thu Jun  4 00:59:59 2020 from 71.204.138.118
RedTeamTrainee@RedTeamTrainingMachine:~$ exit
logout
Connection to 52.250.65.118 closed.

```

Figure 4: SSH over port 22 is now blocked, but works on 5455 when configured.

To make this work, we also need to add a new inbound rule on our Network Security Group. We can add the port, and for added security, we can take the public IPs of users that are associated with the employees (Figure 5).

Inbound security rules



Priority	Name	Port	Protocol	Source	Destination	Action	
100	SSH_Permissions	5455	Any		10.1.0.4	 Allow	...

Figure 5: Inbound security rule for an employee.



There's many other linux hardening techniques we can use. We can edit our sudoer file to allow for RedTeam to use specific sudo commands, but try to disable any escalation of privilege they may have by running sudo su. We can also set our sudo log file using visudo to be its own specific log file to ensure easier monitoring of sudo commands being used. This will help gather information on other areas we can harden.

Another way we can harden the infrastructure is to add a load balancer in front of our jumpbox. While this seems unnecessary, it is our single point to gain access to our deployments. To prevent potential attacks, adding a load balancer in front of the jumpbox, as well as creating redundancies for our jumpbox can help to ensure availability of our system. This does cause other issues in terms of management and attack surface, however, we can have these under a single managed security group to ensure that our rules apply across the board.

Our final high risk recommendation is to ensure that we enable HTTPS for our application VMs. We want to ensure that our boxes have secure and encrypted connections to prevent any potential snooping that could occur. This can also be a good starting point for mitigating DNS Spoofing by ensuring that we have an SSL certificate properly configured on our applications. Azure allows us to set up certificates within the app for 70 to 300 dollars a year depending on the type of certificate we want. We can also set up custom domain names and use Cloudflare as part of our overall infrastructure, at a free cost ("Getting Started with Cloudflare SSL" 2020).

Is it enough?

XCorp should do more to protect and secure their assessments. XCorp infrastructure uses Microsoft Azure as a platform to deploy their infrastructure and provide web services.

XCorp needs to understand that cloud uses Shared Security model; By that, it means Cloud Service Providers protect the datacenter but RedTeam is responsible for safeguarding its own data.

XCorp needs to understand that Cloud Service Providers' responsibility is concerned with keeping up with vulnerabilities and data exploits behind the interfaces and services that are exposed and consumed by their customers.

However, it is XCorp's responsibility to understand cloud architecture and is responsible for securing XCorp cloud infrastructure and securing the data.

It is XCorp's responsibility to protect and secure the environment in order to fence off security threats that imperil customer data in XCorp environment.

The goals are to do enough to strengthen XCorp infrastructure:

For this, CloudTeam recommends the followings:

Rating: High

Description: XCorp does not have a Cloud environment management function.

Impact: XCorp does not have a group of personnel or personnel who is tasked with responsibility for strategic planning, architecting infrastructure, and maintaining XCorp's assets.

Remediation: Hire or delicate an Architect who will be leading the effort and be responsible for

- Architecting the XCorp cloud environment.
- Define, oversee and approve cloud setup prior to deployment.

- Layout plan for securing data, and assets in the cloud.
- Implement best practices as well as technology to monitor and safeguard data in the cloud.
- Conduct regular cyber risk assessments with help from cybersecurity specialists if needed.
- Monitor XCorp public facing interfaces and Network's inbound and outbound using Monitoring function tools (Network Security Monitoring tools), and automate monitoring software tools.

Rating: High

Description: Immediate stop using all default login credentials.

Impact: The web login credentials for customer facing is using default login credentials (admin/admin). It has been known to take only a second of time for Cybercriminals to again access that interface and start the exploits beyond that.

Remediation: Immediate conduct an assessment on all login credentials that are created and used in the environment and change all default web login credentials. Moreover, make it a policy that no default login credentials for services/interfaces are allowed. Automate setup script to detect such violations.

Rating: Medium

Description: Increase Robustness and recovery ability of XCorp's cloud infrastructure.

Impact: Current environment only includes one Load Balancer.

There are no backup nodes/servers for the Load Balancer node. In the event any of these servers malfunction, there will be no recovery and XCorp's cloud environment would be totally inoperable. On top of that, XCorp's assets could be totally up to Cyber Criminals merciness.

Remediation: Create some backup nodes/servers that are ready to failover to ensure the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.

Rating: Medium

Description: Foster a cloud risk and awareness work environment

Impact: Insufficient cyberthreat knowledge and insecure Interfaces and APIs as well as poorly APIs call create vulnerabilities and allow exploits. Cybercriminals can be posing as legitimate users, and operators or developers to exploit vulnerabilities.

Remediation: Conduct code review and design review for application/deployment that use Cloud provided APIs and services. Conduct cybersecurity Training for organization to rise/Change employee awareness and behavior. Foster a workplace environment that enables employees to acquire the skills needed to keep cyber-threats at bay.



Citations:

1. Jevtic, Goran. "5 Linux SSH Security Best Practices To Secure Your Systems." Knowledge Base by PhoenixNAP, 28 Jan. 2020, phoenixnap.com/kb/linux-ssh-security.
2. Updated March 04. "Getting Started with Cloudflare SSL." Cloudflare Help Center, support.cloudflare.com/hc/en-us/articles/360023792171-Getting-Started-with-Cloudflare-SSL.
3. Sheridan, Kelly. "Researchers Use Microsoft Terminal Services Client in New Attack Method." 21 Apr. 2020, <https://www.darkreading.com/endpoint/researchers-use-microsoft-terminal-services-client-in-new-attack-method/d/d-id/1337614>.