Leveraging Cloud Shells (version 1.0)

**Cloud Service Label: PaaS, IaaS**

Description

Cloud Service providers like Azure and Google now offer command line interfaces directly from the web console. These interfaces are actually full fledged Docker containers in a CSP-maintained cluster. To avoid bearing the full cost of maintaining these containers, CSP’s place the container images within the customer’s own cloud storage. The Docker image as a result usually remains persistent and enables an adversary with even temporary access to the cloud console to leverage the container environment to place hidden executables and configurations that ensure access can be maintained even after an intrusion supposedly has been remediated. A reverse shell over the web protocol for example from a customer’s container will actually originate from a CSP controlled host and will not be visible in a customer’s logs nor can it be blocked by customer firewall or NSG rules.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| S2 Security | Demonstrated a proof of concept at Black Hat Training session on Google Cloud Platform. |
| Escaping the Google Cloud Shell Container | A blog on Offensi covers escaping a container hosted within Google cloud by exploiting open Docker unix sockets: one found in */run/docker.sock* and */google/host/var/run/docker.sock*. Once the container has been escaped, the Kubernetes pod can be re-configured to run all the containers in privileged mode. |
| GCP Git Exploits | When opening a cloud shell within GCP, the *cloudshell\_git\_repo* parameter in the URL can be used to pass in a GitHub repository that automatically launches the Python Language Server and executes malicious code hidden with the \_init\_.py file. Another example is pointing to a malicious Git repo with the following URL https*://ssh.cloud.google.com/console/editor?cloudshell\_git\_repo=https://github.com/offensi/git-poc&cloudshell\_git\_branch=master&cloudshell\_working\_dir=evilgitdirectory.* |
| GCP Go and Get | A Cloud Shell user could be tricked into executing malicious code by taking advantage of the go get cloud shell function through CVE-2019-3902 and navigating to the *https://ssh.cloud.google.com/cloudshell/editor?cloudshell\_go\_get\_repo=https://go.offensi.com/go.html.* |
| Azure Cross-Account Command Execution | A NetSPI blog covers the idea of abusing the “Contributor” role in Azure, which allows a user to download any cloud shell .IMG file, including user accounts that have the “Global Administrator” role. Malicious code could be embedded in the shell image file, re-uploaded to the Azure Storage Account, and run under the context of that said Global Admin. |

Mitigations

|  |  |
| --- | --- |
| **Mitigation** | **Description** |
| Delete IMG Files in Azure Customer Storage Accounts | Customers can reset their containers whenever they want in Azure by deleting the .IMG file within their cloud console storage account. Do this especially if it is suspected an account compromise for a user has occurred. |
| Locking down IP addresses and ports | Run container images with minimal functionality, whitelist specific IP addresses and ports for required services, and remove all unnecessary tools on the machine, being a Docker container or a virtual machine. |

Detection

Detection will be nearly impossible since activities occur on non-customer controlled cloud assets. A tool like Sysdig Secure or Falco, which both monitor and secure containers and kubernetes clusters within cloud environments, has the capability to detect reverse shells with out-of-the-box rules. For example, Sysdig Secure will generate alerts based on suspicious processes being run inside containers, like *sh* or *ls*.

References

1. Proprietary S2 training materials. Presented March 5, 2020.
2. <https://sysdig.com/blog/reverse-shell-falco-sysdig-secure/>. Accessed July 28, 2020.
3. <https://www.netsparker.com/blog/web-security/understanding-reverse-shells>. Accessed July 29, 2020.
4. <https://offensi.com/2019/12/16/4-google-cloud-shell-bugs-explained-introduction/>. Accessed July 29, 2020.
5. <https://offensi.com/2019/12/16/4-google-cloud-shell-bugs-explained-bug-4/>. Accessed July 29, 2020.
6. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-3902>. Accessed July 29, 2020.
7. <https://blog.netspi.com/attacking-azure-cloud-shell/>. Accessed July 30, 2020.