Microsoft

Azure Commercial (Public) PCI Penetration Test Report

Prepared By:

KRATOS SECUREINFO

Kratos Technology & Training Solutions
10680 Treena Street, Suite 600
San Diego, CA 92131
Offered through its cybersecurity division:

Kratos SecureInfo
14130 Sullyfield Circle, Suite H
Chantilly, VA 20151
888.677.9351

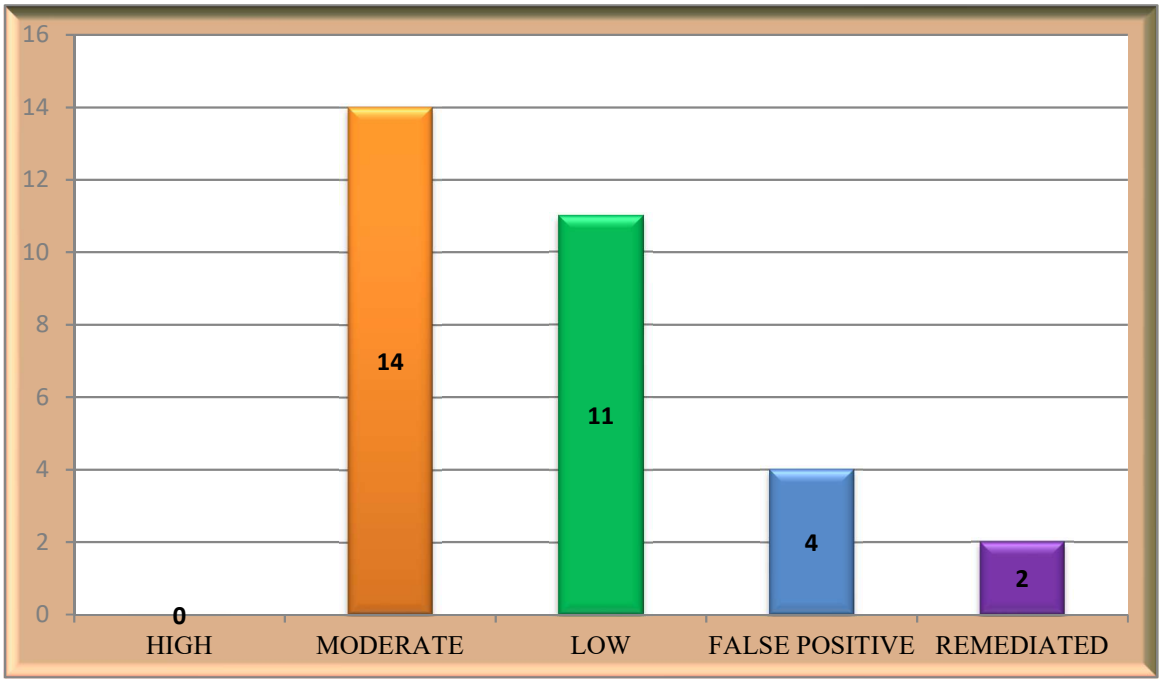
Executive Summary

Background

Microsoft retained Kratos SecureInfo to perform a PCI DSS (Payment Card Industry Data Security Standard) penetration test of their PCI ready cloud services via Azure. The Azure PCI penetration test is a representation of the security posture as of the end date of penetration testing, prior to any mitigation. This report provides the results of the activities performed and serves as a permanent record of the penetration testing activities. The effort was performed offsite remotely from the Kratos SecureInfo lab in Alexandria, Virginia between 11/29/2018 and 12/28/2018. The testing included automated and manual activities using the penetration testing guidance found in the "PCI DSS Penetration Testing Guidance, September 2017" document.

Findings

There were zero (0) **high**, fourteen (14) **moderate**, eleven (11) **low**, four (4) **false positive** identified during the penetration test. Two (2) other findings were remediated during testing. The findings by impact level chart summarize the findings by impact level. Detailed information about the findings is in "Appendix A – Findings".



Penetration Test Findings by Impact Level

Document Revision History

Date	Pages	Description	Author
1/4/2019	All	Draft Deliverable	Kratos SecureInfo
1/7/2019	All	Split report by cloud	Kratos SecureInfo
1/9/2019	9	Updated placeholder web numbers	Kratos SecureInfo
1/18/2019	All	Final Deliverable	Kratos SecureInfo

Table of Contents

Table of Figures

Table of Tables

1. Overview

The Payment Card Industry Data Security Standards (PCI DSS) is an industry coalition information security program that provides a standardized approach to security assessment, compliance, and continuous monitoring for Merchants and Service Providers that seek to store, process, or transmit credit card information. Testing PCI DSS mandated security controls, specifically through penetration tests, is an integral part of the process. Microsoft hereinafter referred to as "CSP" retained Kratos SecureInfo to perform penetration testing of the Azure Cloud Service hereinafter referred to as "CSP service" using current PCI DSS penetration testing guidance. Kratos SecureInfo conducted a proactive and authorized PCI DSS penetration test to validate PCI DSS security controls implemented on the CSP service. The primary goal for the penetration test includes:

- ✓ Gaining access to sensitive information
- ✓ Circumventing access controls and privilege escalation
- ✓ Exploiting vulnerabilities to gain access to systems or information
- ✓ Confirming that remediated items are no longer a risk

During the penetration test, Kratos SecureInfo attempted to identify exploitable security weaknesses of the CSP service including cloud service and application flaws, improper configurations, and end-user behavior to evaluate the CSP's security policy compliance, employees' security awareness, and the organization's ability to identify and respond to security incidents. Findings were then validated, documented, and given an appropriate impact rating which can be found in the "Appendix A – Findings". PCI testing was conducted according to the Kratos Penetration Testing Methodology document.

1.1. Timeline

The testing was performed remotely from the Kratos SecureInfo lab in Alexandria, Virginia between 11/29/2018 and 12/28/2018.

1.2. Scope

The scope included for penetration testing, as defined by the PCI DSS Requirement 11.3 and agreed upon by the service provider within the Rules of Engagement (RoE) included 4090 unique hosts. Microsoft provided IP addresses, an internal Unique Asset Identifier, the DNS name, and a build identifiers of each host. One host, 40.83.103.55, was later identified as have been included erroneously.

In order to facilitate testing within a reasonable time frame, sampling was employed during this test as described in the section below. Web testing was attempted on all endpoints within the sample which had web endpoints accessible. Additionally, testing was conducted to ensure that segmentation controls set in place to prevent out-of-scope systems from communicating or impacting the security of the CDE as implemented.

This PCI environment is unique in that rather than process PCI scoped data directly, Microsoft offers components which can be assembled and configured to create a PCI compliant environment for the end users of the cloud space. There is a great deal of customer responsibility for the correct selection and

configuration of these components and customers are responsible for arranging for a separate and unique penetration test of their specific instance before they are PCI certified.

Kratos SecureInfo, in collaboration with Microsoft, determined that Mobile Application testing DOES NOT apply to the PCI environment scope. In addition, Social Engineering was similarly EXCLUDED from the scope of testing. Finally, the above parties also determined that Physical penetration testing IS NOT applicable and IS NOT included in this report.

Testing efforts assumed the standards for a PCI data storage environment applied to the entire network scope provided.

1.2.1. Sampling

Sampling was deemed necessary for the large scope of network hosts in the environment. While normal statistical sampling attempts to approximate the state of a population within the sample as closely as possible, for the purposes of penetration testing a sampling methodology was chosen that would bias towards highlighting unique attack surface within the full population. Analysis was performed on the provided data and testers leveraged the build identifier to denote clusters of unique builds of devices. Leveraging a random quota sampling methodology the testers took a random selection of hosts within each cluster up to a quota of 1000. This random sampling process was repeated for each build identifier within the scope. This allowed for sufficient representation of all builds present within the environment and lead to using 100% of the population for many build IDs.

The Table 1-1: Testing Coverage denotes the full population of hosts within the provided scope and a percentage of coverage for hosts tested.

Total Population	Percent Coverage in Sample
4090	67%

Table 1-1: Testing Coverage

1.2.2. Restrictions and Alterations

During the engagement, Kratos SecureInfo did not perform any tests that would knowingly result in a denial of service (DoS) to operations, networks, servers, or telephone systems. Additional detail can be found in the service provider's penetration test RoE and Kratos testing methodology document.

During the engagement one (1) endpoint was removed from scope.

Endpoints Removed From Scope
40.83.103.55

2. Testing Narrative: Web Applications

2.1. Web Application Overview

The PCI penetration test of the Service Provider Cardholder Data Environment (CDE) included Internet-based attacks attempting to gain unauthorized access to the Service Provider CDE web applications and the underlying Application Program Interface (API). Specifically, One (1) test case was covered:

- ✓ A simulated Internet attack by an external un-credentialed entity (e.g. public) against the Service Provider CDE web application(s).

2.2. Web Application Discovery

2.2.1. Enumeration

Enumeration of the network endpoints was performed in an attempt to identify any web applications or API endpoints which have been made publically available. This enumeration was performed using a scanning utility against the entire sample and searching for machines with port 80 or 443 available. A screenshot demonstrating the command used is shown in Figure 2-1: Web Detection Scan.

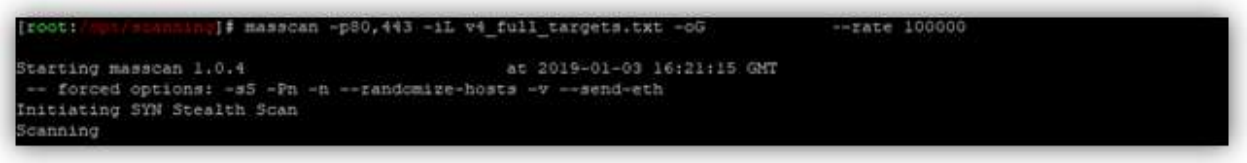


Figure 2-1: Web Detection Scan

After performing this detection of web applications the testers used a screenshotting utility to crawl each application and gather information about the contents of its pages. Testers then leveraged this information to further develop attack vectors for the web applications.

2.2.2. Vulnerability Identification

During testing server misconfigurations were identified through the use of automated tools and manual validation. Results of this effort are included in Table 2-1: Web Application Configuration .

Finding	Summary	Affected Hosts	Additional Information
PF-01 Unencrypted Communications (Low)	The application allows users to connect to it over unencrypted connections.	30 Affected Hosts	See Appendix A PF-01 including a full list of affected hosts.
PF-02 Strict Transport Security Not enforced (Low)	The application fails to prevent users from connecting to it over unencrypted connections.	49 Affected Hosts	See Appendix A PF-02 including a full list of affected hosts.

Table 2-1: Web Application Configuration Vulnerabilities

2.3. Web Application Findings

The Microsoft's Azure PCI scope consists of 2613 IP addresses with standard web port externally available. Of these 2613 IP addresses the testers identified 203 IP addresses serving content. These endpoints serving content were examined and analyzed for vulnerabilities that could serve as paths for exploitation.

Testing was performed in multiple phases beginning with discovery and identifying misconfigurations and ending with manual analysis including attempted circumvention of filtering. Testers identified a number of locations within the scope where user supplied input was being reflected into subsequent requests and into potentially dangerous locations.

Finding ID: N/A	Input Returned in response	Info CVSS: 0.0

Vector String: N/A

Description

During testing 61 locations were identified within the scope where input is being reflected into returned pages. This type of practice can be very dangerous as it provides an avenue for attacks to inject payloads into pages returned to users. This practice is not a vulnerability in of itself however it is a perquisite for many client side attacks.

Testing revealed that these reflections did have some sort of character or input filtering in place to prevent basic payloads from being exploitable.

Exploitation Proof of Concept

Testers attempted to leverage this input reflection at multiple locations throughout the scope in an effort to generate an exploitable payload. A sample of requests and responses are shown in the figures below.

```
GET /?xzxrqjzrlo=1 HTTP/1.1
Host: 51.4.140.42
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
Connection: close
Cache-Control: max-age=0
```

Figure 2-2: Sample request

```
<a id="lpkForgotPswd" class="login_link"
href="https [redacted]
[redacted]>Forgot Your Credentials?</a>
```

Figure 2-3: Response with input reflection

```
POST /Frames/Login.aspx?ReturnUrl=%2f HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: ASP.NET_SessionId=txewsjhduepszdc3455qp4clm;path=/;HttpOnly
Referer: https:/[redacted]Frames/Login.aspx?ReturnUrl=%2f
Content-Type: application/x-www-form-urlencoded
Content-Length: 692

__VIEWSTATE=%2fwEPDwULLTE3NjUw0DMwMDMPFgIeE1ZhbGlkYXRlUmVxdWVzdElvZGUCAWRkfvebn25III2MVjkbp0J1Fqkhbgg%3d&__V
IEWSTATEGENERATOR=7AFA5AE2&ctl00%24txtLoginBgIndex=login_bg1.jpg&ctl00%24__isOutlook=&ctl00%24phLogo%24cmbLa
ng=de-DE&ctl00%24phUser%24idStorage=5408xzz91%22%3e%3ca%3et5n6e&ctl00%24phUser%2412=XAvajlmo&ctl00%24phUser%
2413=xhzayinK&ctl00%24phUser%24txtUser5408=bqQGNtQA&ctl00%24phUser%24p4=fApUPNjw&ctl00%24phUser%24pl=u7C%21a
2g%21W6&ctl00%24phUser%24disableAutocomplete=TSMQzjYS&ctl00%24phUser%24txtPass5408=n5G%21q7v%21K4&ctl00%24ph
User%24txtVeryDummyPass=&ctl00%24phUser%24txtDummyCpny=&ctl00%24phUser%24btnLogin=Sign+In&ctl00%24phLinks%24
txtDummyInstallationID=[redacted]7
```

Figure 2-4: Sample Request 2

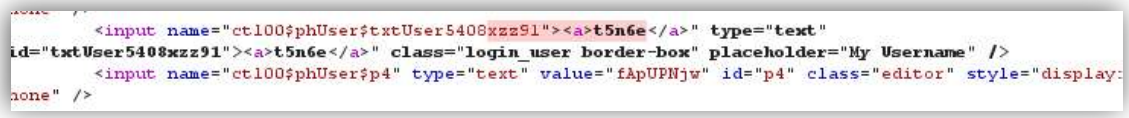


Figure 2-5: Sample Response 2

The testers wanted to specifically call out that throughout testing it was discovered that filtering was being used at the various endpoints however the filtering encountered was not consistent. The practice of inconsistent filtering can lead to issues being harder to troubleshoot and fix in the event that a flaw is identified.

Recommendation

Input should be validated upon submission. Additionally, when input is returned as elements of HTML proper HTML encoding should be applied.

2.4. Web Application Post Exploitation

No web application post exploitation activity was applicable under the scope at this time.

3. Testing Narrative: Network

3.1. Network Overview

The PCI penetration test of the Service Provider CDE included external public Internet testing the network infrastructure and external security posture. The focus was to gain unauthorized access to the Service Provider CDE via the network infrastructure. Specifically, tests simulated an external attack by an external un-credentialed entity (e.g. public) against the Service Provider CDE network infrastructure, as configured in a production environment. In addition to Network penetration test case(s), Kratos performed the following activities:

- ✓ Network discovery
- ✓ Network exploitation
- ✓ Network post-exploitation, if exploitation was successful

Successful exploitation of the external Service Provider CDE did not lead to new access path(s). PCI required post-exploitation activities to explore overall risk of a vulnerability to the Service Provider CDE as a whole. By performing post-exploitation, it was possible to assess confidence that any impact of the vulnerability is valid.

3.2. Network Discovery

3.2.1. Endpoint Enumeration

The external scope totaled at 4090 publicly routable IP addresses. A sample of the total population was used for testing as detailed in the scope, section 1.2 above. An NMAP scan was done to determine which hosts were communicating and had ports open to be able to test.

Description	Method of Scan	Result
Host Discovery (Service Ping Sweep)	NMAP	SEE ARTIFACT PT-02

Table 3-1: Externally Accessible Hosts

3.2.2. Service Enumeration

The services offered by the Service Provider CDE were mapped externally using various scanning techniques directed toward network service ports. The effort involved a complete probe and attempted fingerprint identification of reachable services offered by the common TCP and UDP network protocols. Table 3-2 shows the results for service probes on the Azure PCI network.

Description	Method of Scan	Result
Public identified services	NMAP	[REDACTED]

Table 3-2: Service Provider External Services

3.2.3. Vulnerability Identification

Prior to network exploitation, an un-credentialed vulnerability scan was conducted against the devices at the edge of the Service Provider CDE defined boundary perimeter. Validation was performed to

ensure that findings were correctly identified and exploitation was attempted where relevant. Table 3-3 shows the associated vulnerabilities and Service Provider CDE assets affected.

Finding	Summary	Affected Hosts	Additional Information
PF-03 Apache Tomcat Default Files (Moderate)	The default error page, default index page, example JSPs, and/or example servlets are installed on the remote Apache Tomcat server.	Affects 1 host(s). [REDACTED]	See Appendix A PF-03
PF-04 DNS Server Cache Snooping Remote Information Disclosure (Low)	The remote DNS server responds to queries for third-party domains that do not have the recursion bit set.	Affects 1 host(s). [REDACTED]	See Appendix A PF-04
PF-05 DNS Server Recursive Query Cache Poisoning Weakness (Low)	It is possible to query the remote name server for third-party names.	Affects 1 host(s). [REDACTED]	See Appendix A PF-05
PF-06 HTTP Proxy CONNECT Loop DoS (Low)	The proxy allows the users to perform repeated CONNECT requests to itself.	Affects 1 host(s). [REDACTED]	See Appendix A PF-06
PF-07 HTTP TRACE / TRACK Methods Allowed (Moderate)	The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods that are used to debug web server connections.	Affects 1 host(s). [REDACTED]	See Appendix A PF-07
PF-08 Multiple Vendor DNS Response Flooding Denial Of Service (Low)	The remote DNS server is vulnerable to a denial of service attack because it replies to DNS responses.	Affects 1 host(s). [REDACTED]	See Appendix A PF-08
PF-09 Nonexistent Page (404) Physical Path Disclosure (Moderate)	The remote web server reveals the physical path of the webroot when a nonexistent page is requested.	Affects 1 host(s). [REDACTED]	See Appendix A PF-09

Finding	Summary	Affected Hosts	Additional Information
PF-10 SSH Server CBC Mode Ciphers Enabled (Low)	The SSH server is configured to support Cipher Block Chaining (CBC) encryption. This may allow an attacker to recover the plaintext message from the ciphertext.	Affects 5 host(s). [REDACTED]	See Appendix A PF-10
PF-11 SSH Weak Algorithms Supported (Low)	Nessus has detected that the remote SSH server is configured to use the Arcfour stream cipher or no cipher at all. RFC 4253 advises against using Arcfour due to an issue with weak keys.	Affects 1 host(s). [REDACTED]	See Appendix A PF-11
PF-12 SSH Weak MAC Algorithms Enabled (Low)	The remote SSH server is configured to allow either MD5 or 96-bit MAC algorithms, both of which are considered weak.	Affects 1 host(s). [REDACTED]	See Appendix A PF-12
PF-13 SSL Anonymous Cipher Suites Supported (Low)	The remote host supports the use of anonymous SSL ciphers.	Affects 1 host(s). [REDACTED]	See Appendix A PF-13
PF-14 SSL Certificate Cannot Be Trusted (Moderate)	The server's X.509 certificate cannot be trusted.	Affects 125 host(s).	See Appendix A PF-14 including a full list of affected hosts.
PF-15 SSL Certificate Chain Contains RSA Keys Less Than 2048 bits (Low)	At least one of the X.509 certificates sent by the remote host has a key that is shorter than 2048 bits.	Affects 2 host(s). [REDACTED]	See Appendix A PF-15
PF-16 SSL Certificate Signed Using Weak Hashing Algorithm (Moderate)	The remote service uses an SSL certificate chain that has been signed using a cryptographically weak hashing algorithm (e.g. MD2, MD4, MD5, or SHA1).	Affects 2 host(s). [REDACTED]	See Appendix A PF-16
PF-17 SSL Medium Strength Cipher Suites Supported (Moderate)	The remote host supports the use of SSL ciphers that offer medium strength encryption.	Affects 2914 host(s).	See Appendix A PF-17 including a full list of affected hosts.
PF-18 SSL RC4 Cipher Suites Supported (Bar Mitzvah) (Low)	The remote host supports the use of RC4 in one or more cipher suites.	Affects 9 host(s). [REDACTED]	See Appendix A PF-18

Finding	Summary	Affected Hosts	Additional Information
PF-19 SSL Self-Signed Certificate (Moderate)	The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the-middle attack against the remote host.	Affects 20 host(s).	See Appendix A PF-19 including a full list of affected hosts.
PF-20 SSL Version 2 and 3 Protocol Detection (Remediated)	The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0.	Affects 3 host(s). [REDACTED]	See Appendix A PF-20
PF-21 SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam) (Low)	The remote host allows SSL/TLS connections with one or more Diffie-Hellman moduli less than or equal to 1024 bits.	Affects 6 host(s). [REDACTED]	See Appendix A PF-21
PF-22 SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE) (Low)	The remote host is affected by a man-in-the-middle (MitM) information disclosure vulnerability known as POODLE.	Affects 3 host(s). [REDACTED]	See Appendix A PF-22
PF-23 Unencrypted Telnet Server (Low)	The remote host is running a Telnet server over an unencrypted channel.	Affects 1 host(s). [REDACTED]	See Appendix A PF-23
PF-24 Web Server HTTP Header Internal IP Disclosure (Low)	This may expose internal IP addresses that are usually hidden or masked behind a Network Address Translation (NAT) Firewall or proxy server.	Affects 182 host(s).	See Appendix A PF-24 including a full list of affected hosts.

Finding	Summary	Affected Hosts	Additional Information
PF-25 Web Server Load Balancer Detection (Low)	The remote web server seems to be running in conjunction with several others behind a load balancer. Knowing that there are multiple systems behind a service could be useful to an attacker as the underlying hosts may be running different operating systems, patch levels, etc.	Affects 10 host(s). [REDACTED]	See Appendix A PF-25
PF-26 Web Server PROPFIND Method Internal IP Disclosure (Low)	The remote installation of IIS leaks a private IP address through the WebDAV interface. This may expose internal IP addresses that are usually hidden or masked behind a Network Address Translation (NAT) Firewall or proxy server.	Affects 1 host(s). [REDACTED]	See Appendix A PF-26

Table 3-3: External Vulnerability Identification Data

3.3. Network Findings

A network-level exploitation of Service Provider CDE was completed to analyze the risks of identified vulnerabilities. The penetration tests focused on external attacks against Service Provider CDE hosts to determine the sensitivity any information retrieved if exploitation is successful. Attack scenario(s) were created to exercise the security of Service Provider CDE with the intent of gaining access to the hosts/systems and elevating privileges, if possible. If exploitation of the scenario was unsuccessful, the scenario also discussed reasons why exploitation failed and what protections (if any) prevented the exploitation.

Initial tests were performed using a machine which was deployed into the CDE and attempted to gain access to devices which were not publically accessible. Testers performed activities such as ARP scanning, network discovery and packet sniffing to attempt to locate an avenue of attack. When a tenant was provided it became quickly apparent that devices are deployed into a private software defined network segment for each client. As shown in Figure 3-1 an ARP scan of the local network segment revealed that the segment was vacant outside of the gateway.

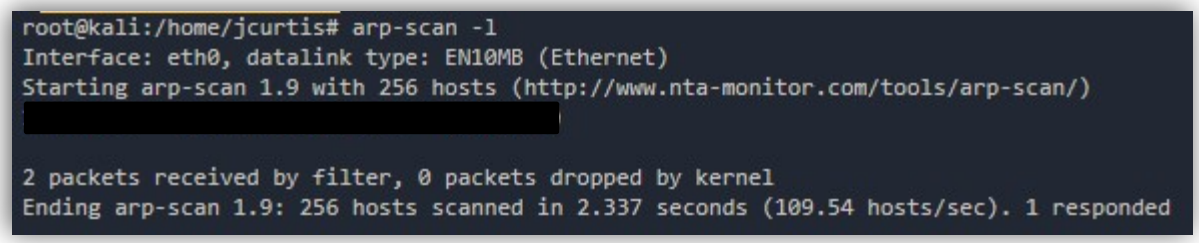


Figure 3-1: ARP Scan of SDN

A robust scan of the local network, including the gateway, was performed to determine if there were any services available for interaction however this revealed no such services were found. The only response received was that of the localhost's SSH service.

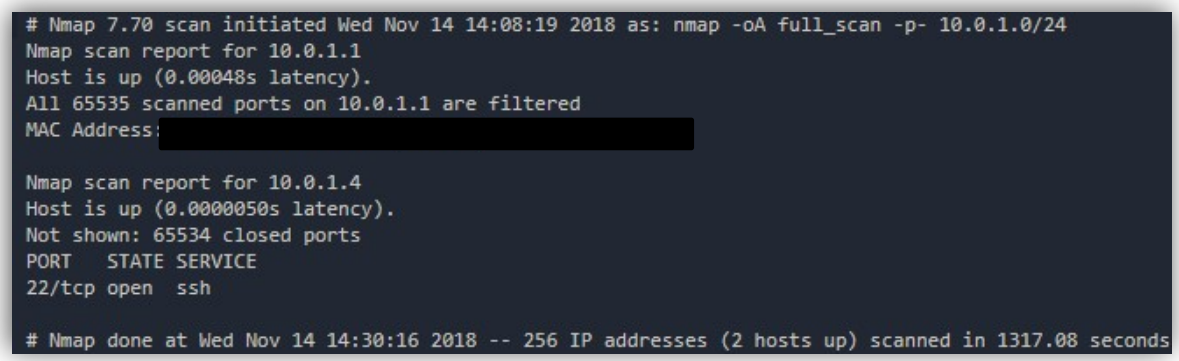


Figure 3-2: NMAP SDN Scan

Testers were not able to find a method of escape from the software defined network to facilitate additional access to devices outside of what is intentionally being made publically available.

Additional testing was performed against hosts externally facing ports. This testing attempted to identify any remotely exploitable vulnerability which would provide an attacker the ability for further access into the cardholder data environment. During testing the following man in the middle opportunities were identified:

- PF-22 SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE) (Low)
- PF-23 Unencrypted Telnet Server (Low)

A determined and well positioned attacker could be able to leverage these vulnerabilities to gain access to the data transmitted between client and server. Testers were not able to leverage these vulnerabilities at this point in time to gain additional access to the cardholder data environment.

Finding ID: PF-27	Unencrypted FTP Server	Remediated CVSS: 0.0

Original Vector String: CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N

Description

During testing, 28 unencrypted FTP servers were identified. A well-positioned attacker is able to intercept these communications and read, in plain text, the entire conversation between client and server. It is also possible for a well-positioned attacker to potentially modify communications between client and server thereby impacting integrity of information transmitted.

Testers did note that on the machines identified servers did offer AUTHTLS on port 21 and also the FTPS port 990 was available. When using the FTPS protocol port 21 has an explicit requirement for encryption. This means that a user my explicitly request encrypted communication. A man in the middle could intercept this request and "downgrade" it to prevent the encryption handshake from even occurring.

This vulnerability was remediated during testing.

Exploitation Proof of Concept

Testers connected to and verified that these servers were in fact offering unencrypted communications. A sample request was submitted and captured through the use of packet capturing software.

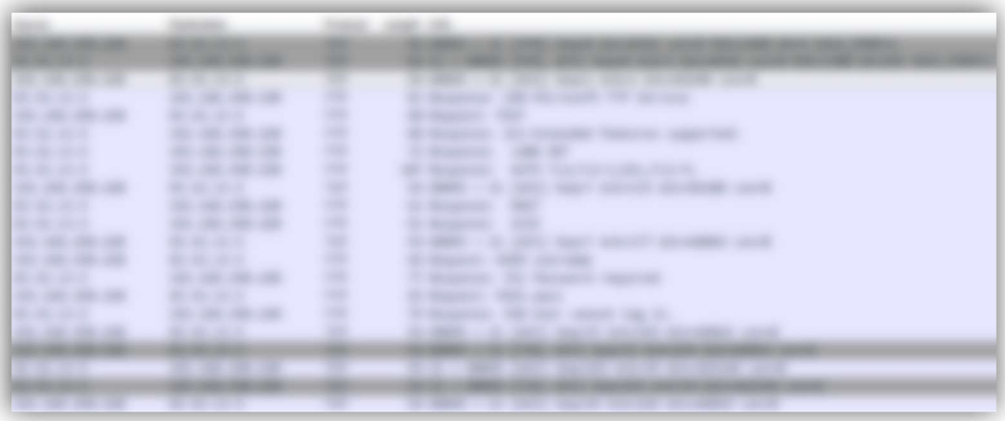


Figure 3-3: Packet Capture of FTP Traffic

After filtering all of the other packet data, the commands sent and received by the server are easier to see.

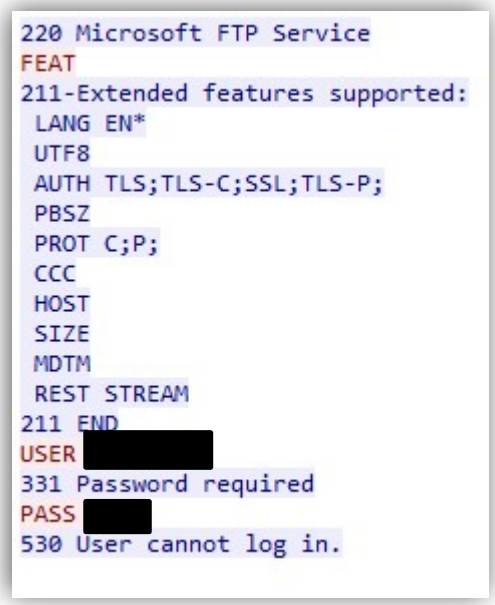


Figure 3-4: FTP commands

As shown in Figure 3-3: Packet Capture of FTP Traffic and Figure 3-4: FTP commands conversations happening over this port are not encrypted by default. The attacker was able to capture, in plain text, the users username and password as denoted by the red lines in Figure 3-4. Figure 3-4 also demonstrates the server has support for the AUTH TLS feature however this feature is not invoked until the user explicitly requests it.

Remediation Evidence

A rescan was performed and is vulnerability was no longer present. Rescan evidence can be found in Appendix C.

3.4. Network Post-Exploitation

No network post exploitation activity was applicable under the scope at this time.

Appendix A – Findings

Findings	File
This Excel Spreadsheet Contains Penetration Test Findings.	Please see "Appendix A – PCI Findings – Azure Public.xlsx"

Appendix B – False Positives

Please see the following table for the list of findings deemed false positive that were discovered during penetration testing.

Discovery Source	Vulnerability	Justification
Web Application	Open redirection (DOM-based)	Static variable made prevented exploitation.
Web Application	Session Token in URL	This incorrectly identified other tokens as a session token.
Web Application	Cookie Manipulation (Dom-Based)	Cookie manipulation was not possible.
Web Application	Content Type incorrectly Stated	The content type was correctly stated as a font file.

Appendix C – Evidence

Evidence ID	Description	Test Section	Artifact
PT-01	This artifact contains the Rules of Engagement (RoE) signed between Kratos SecureInfo and Microsoft.	Other	Please see "PT-01 Rules of Engagement Commercial.pdf"
PT-02	NMAP Discovery Scanning	Network	Please see "PT-02 NMAP Scanning Artifacts.zip"
PT-03	Vulnerability Scanning Results	Network	Please see "PT-03 Vulnerability Scan Public.html"
PT-04	Burp Web Application Report	Web Application	Please see "PT-04 Web Application Report.html"
PT-05	Retesting results	Network	Please see "PT-05 Retesting Results.zip"