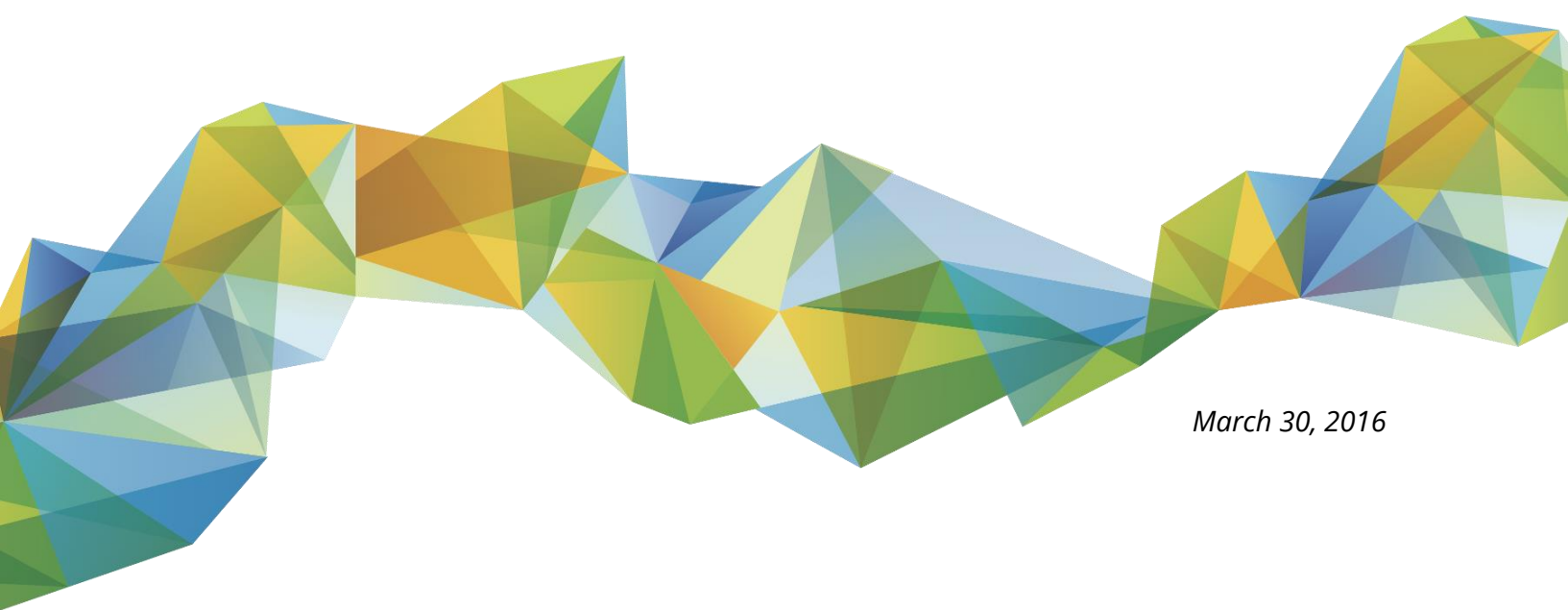


Essence Field Testing Report



March 30, 2016

Copyright © 2016 by Cigital, Inc.® All rights reserved. No part or parts of this documentation may be reproduced, translated, stored in any electronic retrieval system, transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the copyright owner. Cigital, Inc. retains the exclusive title to all intellectual property rights relating to this documentation.

The information in this documentation is subject to change without notice and should not be construed as a commitment by Cigital, Inc. Cigital, Inc. makes no representations or warranties, express or implied, with respect to the documentation and shall not be liable for any damages, including any indirect, incidental, consequential damages (such as loss of profit, loss of use of assets, loss of business opportunity, loss of data or claims for or on behalf of user's customers), that may be suffered by the user.

Cigital and the Cigital logo are trademarks of Cigital, Inc. Other brands and products are trademarks of their respective owner(s).

Cigital, Inc.

21351 Ridgetop Circle
Suite 400
Dulles, VA 20166
United States
+1 (703) 404-9293

Cigital Cooperatief UA

Officia I
DeBeolalaan 7
1083 HJ Amsterdam
The Netherlands
+31 2 03 01 91 50

Cigital Ltd

Riverbridge House
Leatherhead, Surrey
KT22 9AD
United Kingdom
+44 1372 365 700

<https://www.cigital.com/>

Table of Contents

1Executive Summary	4
2Participants	4
3Deployment Architecture	5
4Onsite Testing Activities	5
5Results and Conclusion	8
About Cigital, Inc.....	12

1 Executive Summary

Essence phase II field testing was conducted at the Central Electric Membership Corporation (Central EMC), located at 128 Wilson Road Sanford, NC 27332 from March 21 to March 22, 2016.

We successfully deployed the Essence appliance in both virtualized and non-virtualized environments, captured all the Multispeak® traffic as expected, passed all six test cases as planned, and engaged real operators to interact with the console; the operators provided valuable feedback. The virtual appliance was left to continuously operate for two more weeks with daily monitoring and statistics gathering.

Matt Gardner, our target operator, was able to assign hosts and endpoints, create detection rules, and navigate the network map. Both Matt and Angela were very impressed by the network map. Angela mentioned that she wanted to show it to her CEO in a presentation.

Initially, we encountered significant challenges while attempting to capture data, but we were able to overcome the difficulties and successfully captured all the Multispeak® traffic. This caused the onsite testing to be extended one more day than originally planned. We identified a few improvement areas for handling live traffic. Details can be found in Section 4, Onsite Testing Activities.

Overall the new feature of building the network map from live traffic showed great promise and the operators seemed to like it. This will guide us to develop more intuitive and visually compelling user interfaces. Machine learning is a powerful tool that expands the possibilities of detecting anomalies and it complements Essence's rule based detection mechanisms well. However, generating proper traffic pattern to test targeted problems is hard, and the user interface needs more improvement before operators can use it.

2 Participants

Central EMC

Angela Hare – Manager of Information Technology

Matt Gardner – System administrator

Cigital, Inc.

Ping Ning

Bob Wintenberg

3 Deployment Architecture

The following deployment diagram depicts how each Essence appliance was installed. Note that each Essence appliance was connected to a switch (one virtual and one physical), and the two deployment approaches were used to cover all available network topologies with Multispeak® nodes.

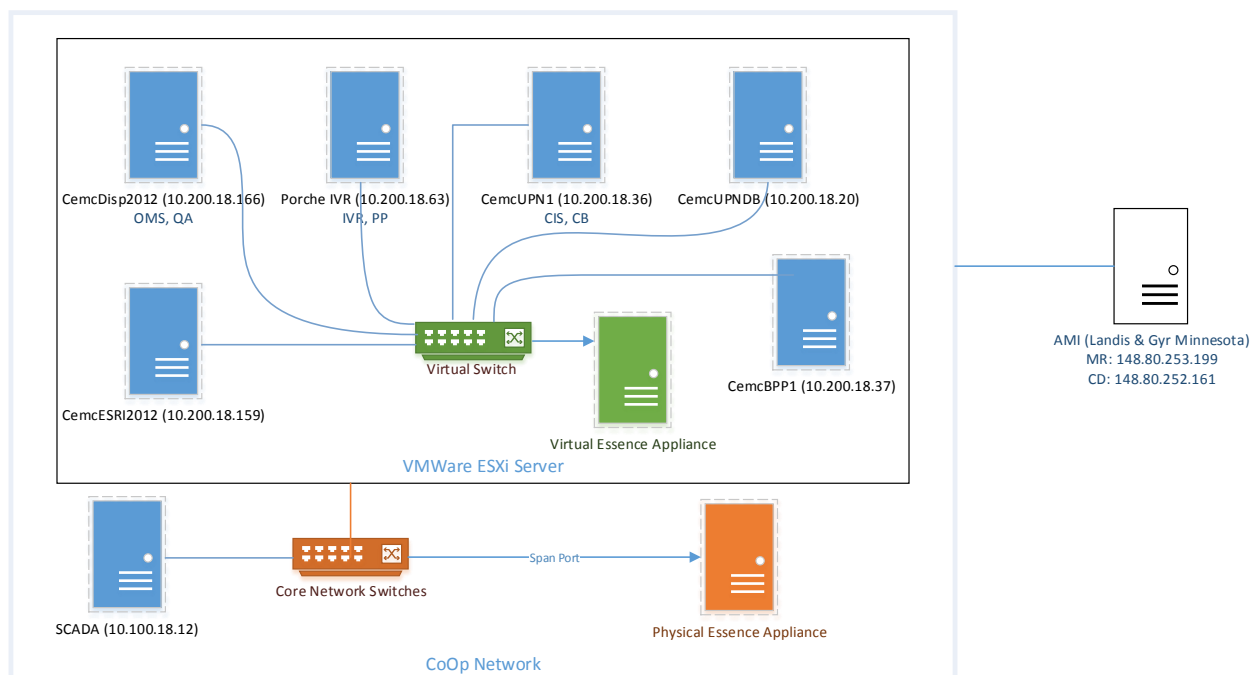


Figure 1 Deployment Architecture

This deployment architecture covered the following Multispeak® endpoints during the field testing:

Virtual Essence Appliance

- All TCP traffic between the AMI (MR, CD), CIS (CB), IVR/PP, and OMS/OA.

Physical Essence Appliance

- All TCP traffic going to the AMI and from the AMI.

Note that it was not possible for the physical appliance to capture traffic sent between the OMS, CIS and IVR since it was only routed via the virtual switch (i.e. east-west traffic) in the VMWare ESXi server; it was able to capture traffic sent from the AMI and to the AMI.

4 Onsite Testing Activities

The onsite field testing of the Essence device was conducted over two days. Following is the log of events that include observations and comments. The next section will summarize the results, observations, and improvements.

March 21, 2016

08:15 Bob and Ping arrived onsite and met with Matt Gardner; since we only had the technical team we proceeded to perform the physical installation in the server room

09:00 Physical installation completed

09:30 Tried to set up the device on the local network so we could access it from the conference room; Matt was trying a few available ports but none seemed to work; we decided to skip the remote set up and test the physical deployment inside the server room.

09:45 Connected to the SPAN port of the hardware switch and began to see network traffic on the SPAN port, including some Multispeak® packets

10:00 Successfully built the network map from live production traffic

10:15 Moved into a conference room to test the virtual deployment, which was installed before we got onsite; Angela Hare joined us. We discussed the overall test plan for the day.

10:30 Matt told business users to start normal connect/disconnect Multispeak® traffic which had been put on hold to get us the real data; Angela also sent a stream of PingURL messages to create volume of traffic.

10:45 While Essence was capturing data, we asked Matt to play with the network map and create rules; the following feedback was received from Matt and Angela:

- Matt liked the auto refresh feature of the map and saw the map updated with new traffic over time.
- On the labeling of the network nodes, use hostname instead of IP address if available.
- Consider adding a user preference whether to display node name or IP addresses to Settings
- When saving an endpoint configuration, the JavaScript popup can have a better presentation
- The node labeling from search result on the map was not aligned properly with the dot of the target node in the result on the map
- When displaying an alert, consider adding timestamp of the related packet
- Add consolidation of duplicate alerts (alerts with the same description and/or other data) and display one entry with number of occurrences
- Angela wished to see deep packet inspection for DNP3 as well because there were so many things happening through DNP3
- Currently each detection rule is defined for only one IP address. Consider using IP address patterns to define rules, so that a rule can be applied to a whole subnet
- Consider adding a feature to allow importing a list of endpoints in endpoint configuration

12:00 Lunch break

13:00 Started validating captured data and found discrepancy between the amount of traffic Angela sent and the amount we captured

13:30 Ran tcpdump at the same time as the capturer while Angela re-generated the traffic;

- During the investigation we replayed the captured data inside the VM, which flooded the virtual switch because it was on promiscuous mode, and part of the traffic went out to the network; that caused the physical switch to shut down the port to the vSwitch environment. This incident caused the Multispeak® endpoints to be down for a short period of time. Matt investigated and concluded that no harm was done. He restarted the connectivity and everything worked fine. There was no impact on business except for the short down time.

- We learned the lesson and subsequently performed investigation only in our local non-production VM

13:45 Investigated the issue of not capturing all Multispeak® messages; we found a few causes:

- The capturer expected traffic to be sent to typical HTTP ports such as 80, 8080, 8088, 9090, 5000, and 5555, which we had identified from all past experiences and used to improve the performance of the capturer. In Central, there was a port 16504 that had the most traffic, and another port 17005. We missed the traffic on both.
- The regex patterns created to identify Multispeak® messages from the captured traffic and validated using our past data samples failed to capture a new namespace pattern that contains a “-“, such as “SOAP-ENV:body”.
- We also identified that some newline characters in the messages caused some messages to be dropped by the capturer, and some captured messages had UTF-8 encoded special characters in them. Later investigation concluded that bugs in the capturer caused both.

16:30 We had identified most of the causes and went back to the hotel to make adjustment and fixes to the capturer code.

March 22, 2016

8:15 We arrived onsite and started verifying the adjustments and fixes from the day before

- We were able to capture 418 out of 455 messages
- We continued the investigation and identified the root cause for the rest of the missing packets. They were using persistent HTTP connection in the client server interaction with the HTTP 100-continue header. This allowed the client and server to maintain a TCP connection for a fairly long time without ending the TCP stream. The capturer used the popular NIDS library which depended on the end of stream marker to delineate HTTP requests and responses.
- We implemented a partial fix to the capturer and were able capture 100% of the Multispeak® messages. It was partial because in some cases we captured the same HTTP request twice, which did not impact the traffic analysis in a significant way.

13:00 Quick lunch break

14:30 After we validated that the capturer could capture all Multispeak® messages, we went back to execute the test cases

15:00 Successfully tested all rule based detection test cases; Matt played with Essence and discussed his impression; Bob began testing machine learning with live production data

15:30 Successfully demonstrated supervised and non-supervised machines learning test cases with live production data. Discussed the concept. They felt it was an interesting idea and could be productive in discovering problems. Here are some more feedbacks:

- When displaying alerts, aggregate the same finding with counts instead of displaying every individual instance
- Make the map more distinguishable for endpoints inside the network and put endpoints outside of the network further away from the inside network nodes

16:00 Successfully exercised all test cases with live production data and engaged real field operator; provided brief training on continuous monitoring; un-install the physical device and packing

16:30 Conclusion of the onsite field testing

- Angela asked for permission to demonstrate the network map to the CEO in her presentation.

5 Results and Conclusion

We set out to evaluate the operation readiness of the Essence device in field testing after we had accumulated and digested the input from the previous field tests. Also, we tested the network map with live production data and Essence's machine learning based anomaly detection in supervised and non-supervised modes; the two aforementioned features were built during cycle two.

We successfully deployed the Essence appliance in both virtualized and non-virtualized environments, captured all the Multispeak® traffic as expected, passed all six test cases as planned, and engaged real operators to interact with the console who provided valuable feedback. The virtual appliance was left to continuously operate for two more weeks with daily monitoring and statistics gathering.

The combination of virtualized and non-virtualized deployments proved that the Essence device can be deployed to cover any network topology. With live production data we have successfully passed the following test cases. Lessons learned and improvements are also captured in the following table:

Test Case Executed	Results	Observation and Lessons Learned	Improvement
1. Building Network Graph from Network Packet Stream	Passed all steps	<ol style="list-style-type: none"> 1. Operator liked the auto refresh feature of the map and saw the map updated with new traffic and grow over time 2. On the labeling of the network nodes, the operator would've liked to see hostnames instead of IP address if available 3. Operator suggested adding a user preference whether to display node name or IP addresses to Settings 4. When saving an endpoint configuration, the JavaScript popup could have a better look 5. The node labeling from search results on the map was not aligned properly with the dot of the target node in the results on the map. That made it harder for operator to identify which node was the selected one. 6. When displaying an alert, there was no time stamp for the packet. Time stamp could help correlate what happened. 	<ol style="list-style-type: none"> 1. Support hostname labeling on the map and allow user to set preferences on whether to display IP address or hostname. 2. Aggregate same alerts for same source and destination with an instance count and display only once 3. Align search results on the map with the nodes more precisely to make it easy for operator to find the correct nodes 4. Add time stamp of packets to alert details 5. Improve popup box presentation on Save endpoint configuration

		<ol style="list-style-type: none"> When displaying alerts, aggregating the same finding with counts instead of displaying individual instances so there weren't that many repetitive items displayed. Operator suggested making the map distinguishable for endpoints inside the network and putting endpoints outside of the network further away from the inside network nodes 	<ol style="list-style-type: none"> Support distinguishable boundary between internal network and external network in the map
2. Defining and Executing Value Out Of Bound Rules	Passed all steps	<ol style="list-style-type: none"> Angela wished to see deep packet inspection for DNP3 because there were so many things happening through DNP3 Currently detection rules are per IP basis. Operator suggested using IP address patterns to define rules, so a rule can be applied to a whole subnet 	<ol style="list-style-type: none"> Enhanced data capturer to handle unknown situation easier using configurable port numbers and regex expressions.
3. Defining and Executing Denial of Service Rules	Passed all steps	<ol style="list-style-type: none"> Regarding endpoint configuration, operator suggested adding a feature to allow importing a list of endpoints 	<ol style="list-style-type: none"> Support wild card rules for IP addresses in rule definition and execution
4. Defining and Executing Connectivity Violation Rules	Passed all steps	<ol style="list-style-type: none"> The capturer expected traffic to be sent to typical HTTP ports such as 80, 8080, 8088, 9090, 5000, and 5555, which we had identified from all past experiences and used to improved capturing efficiency. In Central, there was a port 16504 that had the most traffic, and another port 17005. We missed the traffic on both. The regex patterns created to identify Multispeak® messages from the captured traffic and validated using our past data samples failed to capture a new namespace pattern that contains a “-“, such as “SOAP-ENV:body”. We also identified that some newline characters in the messages caused some messages to be dropped by the capturer, and some captured messages had UTF-8 encoded special characters in them. Later investigation concluded that bugs in the capturer caused both. There was a problem related to the use of persistent HTTP connection in the client server interaction using the HTTP 100-continue header. This allowed the client 	<ol style="list-style-type: none"> Research a library or find a solution to support persistent HTTP stream Ensure regex pattern is cases insensitive and can handle special characters including ‘-‘, newlines, etc. Add DNP3 support (out of scope)

		<p>and server to maintain a TCP connection for a fairly long time without ending the TCP stream. The capturer used the popular NIDS library which depended on the end of stream marker to delineate HTTP requests and responses.</p> <p>8. In future testing, verify data capturing with controlled traffic first before executing the test cases.</p>	
5. Anomaly Detection: Unsupervised Machine Learning	Passed all steps	<p>1. Creating the proper thresholds to detect the target anomaly pattern was difficult. The user interface required deeper knowledge than the operator would normally have.</p>	<p>1. Provide a more intuitive user interface so operators can use it to tune the detection in their environment.</p>
6. Anomaly Detection and Root Cause Identification: Supervised Machine Learning	Passed all steps.		
7. Post Monitoring		<p>1. Matt reported that the network map drawing had become non-responsive. Further investigation pointed to the fact that too many packets were in the database and the processing had been too slow. The Cassandra database could provide scalability and generally handle time series data well. Unfortunately as we discovered, the behavior of sliding time window over packet streams is not a strength of Cassandra because data cannot be ordered by timestamp due to the cluster key mechanism. We need to consider alternative solution to handle the Essence behavior.</p>	<p>1. Use MySQL to support the time window sliding needs of Essence so that Essence can handle stream data for continuous operation. Given the short time to the end of the project, it's not feasible to evaluate another NoSQL database use. We have significant knowledge about MySQL.</p>

Table 1 Results, Observations and Improvements

Matt Gardner was able to assign hosts and endpoints, create detection rules, and navigate the network map. Both Matt and Angela were very impressed by the network map. Angela mentioned that she wanted to show it to her CEO in a presentation.

Initially, we encountered significant challenges while attempting to capture data, but we were able overcome the difficulties and successfully captured all the Multispeak® traffic. This caused the onsite testing to be extended one more day than originally planned. We identified a few improvement areas for handling live traffic

Overall the new feature of building the network map from live traffic showed great promise and the operators seemed to like it. This will guide us to develop more intuitive and visually compelling user interfaces. Machine learning is a powerful tool that expands the possibilities of detecting anomalies and it complements Essence's rule based detection mechanisms well. However, generating proper traffic pattern to test targeted problems is hard, and the user interface needs more improvement before operators can use it.

About Cigital, Inc.

Cigital, Inc. is the leading software security and quality consulting firm. Established in 1992, Cigital plans and implements initiatives that help organizations ensure their applications are secure and reliable while also improving how they build and deploy software. Our recognized experts apply a combination of proven methodologies, tools, and best practices to meet each client's unique requirements, providing resources and knowledge to deliver value to their business.

Cigital has enabled some of the most well-known organizations world-wide in financial services, communications, insurance, online gaming, hospitality, e-commerce and government to reduce their mission-critical software business risks. Our offices in the UK and Northern Europe broaden our reach and allow us to support trans-Atlantic clients more completely with European consulting, assessment and training operations.

Our expert advisors are recognized thought leaders in software security and have written the books on software security and quality with such publications as the Web Security Testing Cookbook, Exploiting Online Games, Software Security and Building Secure Software.

Cigital is headquartered near Washington, D.C. with regional offices in the U.S., London, Amsterdam and India.