Utilizing electromagnetic emanations for out-of-band detection of unknown attack code in a programmable logic controller

Nathaniel Boggs[a], Jimmy C. Chau[a], and Ang Cui[a]

[a]Red Balloon Security, 336 W. 37th St., New York, NY, USA

ABSTRACT

We propose using out-of-band emanations from embedded devices in order to detect malicious code execution. We passively monitor involuntary electromagnetic (EM) emissions from embedded devices to find and detect new signals. We demonstrate the efficacy and feasibility of an EM emanation based anomaly detection system using commercial off-the-shelf (COTS) software defined radio (SDR) hardware to detect code execution on an industrial control system (the Allen-Bradley 1756-EWEB module). We have developed a fully automated training and testing framework for this anomaly detection system. In this paper, we describe the system architecture, the cliff-detection algorithm used to process the received emanations, the testing setup and procedures, and our results. When trained on one set of EWEB modules and tested on a separate set, we present an experimental prototype capable of detecting unknown (attack) code execution with 98% accuracy at 100% detection rate. We present data supporting the robustness of these results across 16 physical device instances and with training recordings taken months apart from testing recordings.

Keywords: side-channel, anomaly detection, software-defined radio, programmable logic controller

1. INTRODUCTION

Programmable logic controllers (PLCs) and other embedded devices that make up large portions of critical infrastructure often contain vulnerabilities that attackers can exploit to steal information, sabotage operations, and even cause physical destruction. Past research into various defenses typically focuses on either network based defenses attempting to block or detect any attacks before they reach the device or on host defenses such as fortifying the code running on the device or additional security code running to detect any local attacks. All these defenses are inline with the attack path and therefore can themselves be exploited or disabled by the attacker.

Research over the past decades has demonstrated significant unintentional information leakage from devices via electromagnetic (EM) emanations and other side-channels. Such leakage comes from a combination of the CPU instruction executions, memory accesses, and other electronic components of the device. Researchers are now attempting to leverage this information leakage over side-channels to determine activities running on devices and to detect abnormal patterns that could indicate an attack. A defense capable of detecting a device being attacked without relying on any direct connection to the device or network where the attacker has control, would provide an extra layer of defense that attackers would not be able to directly compromise.

In this paper, we investigate using out-of-band side-channel EM emissions from the PLC to verify that the PLC is operating properly. We improve upon previous research by applying these techniques to a complex widely deployed Allen-Bradley 1756 PLC particularly the EWEB module of the PLC. Furthermore, we detail the new signal cliff detection algorithm we developed to flexibly detect and model device activity states.

We create a prototype system capable of interacting with the EWEB module in normal and abnormal scenarios, recording the EM emanations, training signatures, scoring test data, and mapping recorded ground truth

Author contact: E-mail: nathaniel@redballoonsecurity.com

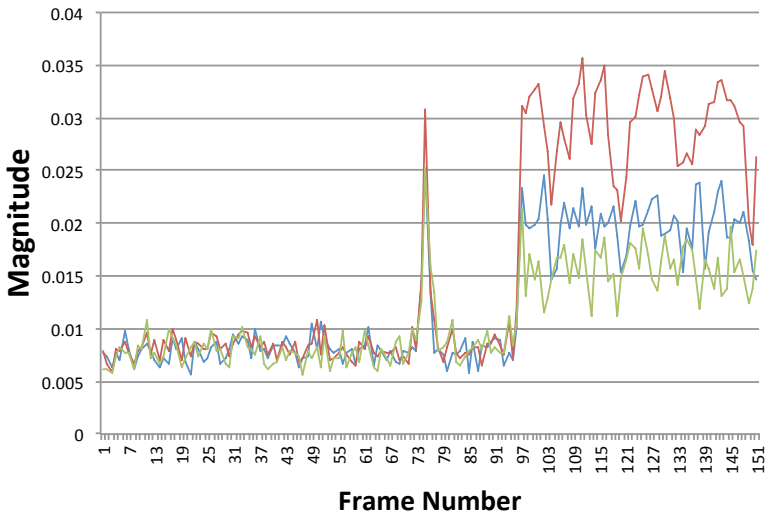

Figure 1. Graph of the magnitudes of adjacent frequency bins over time. When a new activity starts, a clear spike in magnitude appears and is sustained. This is the new signal that the cliff detection algorithm is designed to detect.

to evaluate the results. We conduct a variety of experiments using 16 separate hardware instances of the EWEB module and detect the unknown abnormal code execution with 98% accuracy at a 100% detection rate.

2. DETECTION ALGORITHM

The signal cliff detection approach we developed detects relative magnitude increases for each frequency using a sliding window that averages the before and after frames to detect a new sustained rise in signal magnitude. The core idea behind the cliff detection algorithm is that a change in device activity will correspond to a sustained new signal at a particular set of frequencies. Detecting relative increases allows for some inherent adaptation to any systematic magnitude shifts such as caused by an increase in distance. One disadvantage is the longer period required to determine that a new signal has appeared due to the required sliding window. This could make it impossible to quantify very brief events depending on how large the sliding window is.

See the graph in Figure 1 for a visual example of the type of cliff that is detected. Each data set is the magnitudes of a particular adjacent frequency. Note how the signal magnitude bleeds over into adjacent frequency bins. The short spike will be ignored, while the sustained magnitude increase on the far right of the graph will be detected as a new signal. The cliff detection algorithm detects the major increase that is sustained for some minimum window, in the case of this paper the window is 40 frames or about 13ms, as a new signal for that fast Fourier transform (FFT) frequency bin.

These new signals are then collected into a synchronization event signature where each column is a single FFT frequency bin with adjacent frequency bins that have new signals joined into a single range. We term these 'synchronization events' as a description of how the signature maps back to a specific activity occurring on the device, such that it is suitable to synchronize in time the external knowledge of the device state. See Figure 2 for a visual of how such new signals are captured from a waterfall and saved as new signals or synchronization events. Each of these synchronization event signatures is saved as a list of these ranges of FFT frequency bins that contain new signals.

When scoring new data with one of these signatures, we find the new signals in the same way as when creating the original signatures. Each range of FFT bins with new signals is determined to match the signature if it overlaps any of the ranges in the signature. The score is based on the percentage of signature FFT frequency bin ranges that overlap with the new data along with a penalty for new signals not in any of the signature's FFT frequency bin ranges. Additionally, in order to reduce noise, we discard any signals that are not significantly higher in raw magnitude than the bottom quartile of the FFT frequency bins.

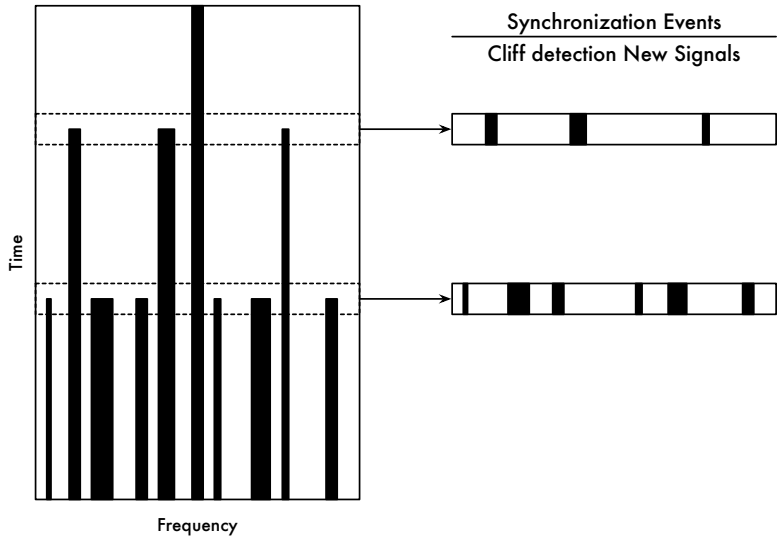

Figure 2. Example of simplified waterfall graph with examples of synchronization event signatures based on new signals. Note that the cliff detection only looks at new signals, automatically filtering out existing signals to reduce noise.

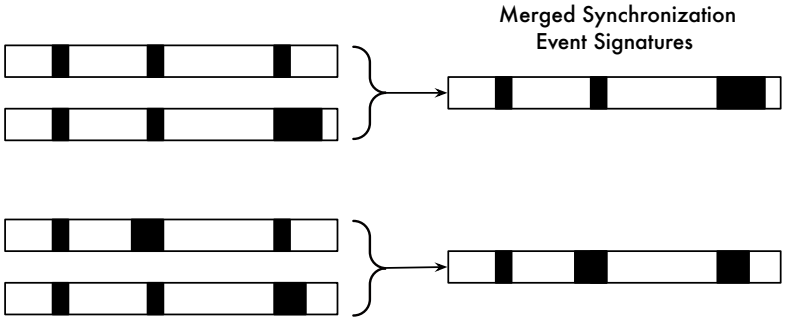

Figure 3. Similar synchronization event signatures can be merged together by a simple logical ‘or’ operation.

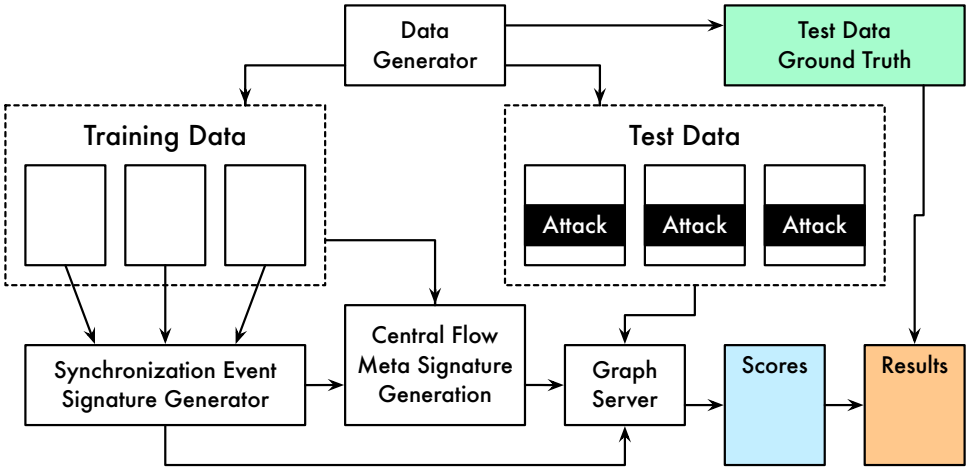

Figure 4. Overview of experiment including data generation, training to generate synchronization events, graph server that processes the test data via the cliff detection algorithm into scored frames, and finally evaluated results based on the logged ground truth data.

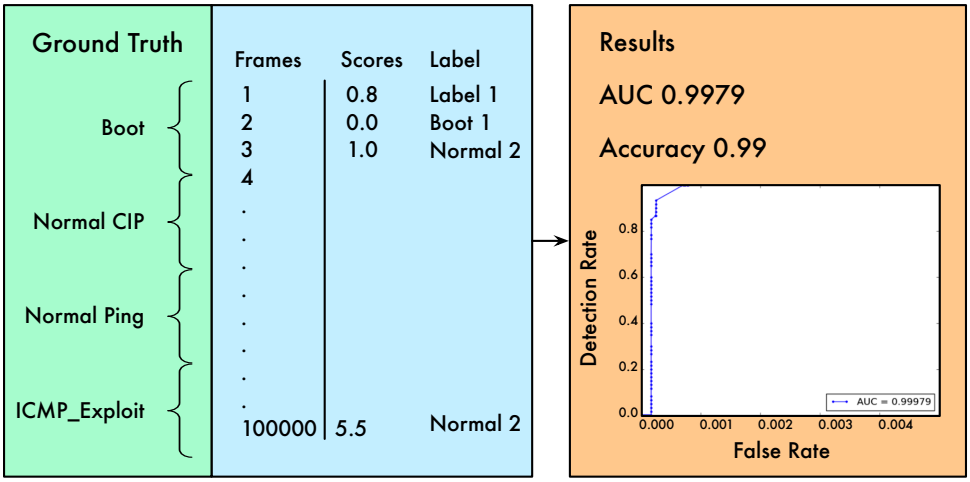

Figure 5. After the cliff detection algorithm computes its scores over the test data set, the ground truth is used to map the scores into results including accuracy and ROC curves.

These synchronization event signatures can also be automatically merged when significant overlap exists. Figure 3 demonstrates the merging process where overalpping signatures are joined with a logical 'or' operation. Synchronization event signatures are created from the training data and then used in the overall training and testing framework to detect the unknown code in the test data sets. The resulting scores associated with each frame are then evaluated against the ground truth collected during the initial recordings.

The meta event signatures are automatically created when the same synchronization events are seen in multiple training data sets at the same timing as part of a particular normal activity labeled via the ground truth. These meta event signatures are what the graph server uses to better understand the system state over time. See Figure 4 for an overview of how meta event signatures are created as part of the training. The automatic creation of these meta event signatures is done by taking the trained signatures and running the detection algorithm with them against the other training datasets. If the same pair of signatures detects synchronization events with the same time separating them across multiple training sets from different devices, then the pair of signatures plus the time difference is recorded as a meta signature.

Figure 6. The physical experiment layout. Note that the environment and distance are purposefully not carefully controlled in order to avoid any over fitting due to precise distance or noise avoidance. The ruler allows the researcher to confirm that the PLC is at least one foot away from the antenna.

The final results after the ground truth is applied can then be graphed as a receiver operator characteristic (ROC) curve. Figure 5 displays the mapping of recorded results to ground truth in order to compute the final accuracy and ROC curve.

3. EXPERIMENT DESIGN

3.1 Physical Setup

Rather than choosing a development board or consumer board such as a Raspberry Pi or Arduino, we choose to work with the Allen-Bradley 1756/EWEB module. This PLC module fits into the chassis of a PLC and acts as a gateway to the other modules on the backplane, which would typically control physical devices, by providing network connectivity. PLCs are particularly good candidate devices for out-of-band monitor as they have a small number of normal use cases for likely regular usage patterns while still being a high value target.

We use the log-periodic antenna pictured in Figure 6 at a range of one foot from the device. This antenna is designed for 900MHz to 2600MHz frequencies and thus suitable for the 1500MHz frequency centered recordings we take. The antenna connects to a USRP X310 software defined radio capable of recording up to 100 megasamples per second over a PCI-e or 10 gigabits per second network connection although for these experiments we recorded at 50 megasamples per second to keep the data volume more manageable. The 1500MHz frequency is useful as there is limited interference, but many harmonics of the EWEB components are still visible. Finding a more optimal frequency and data recording parameters is an exercise left for future work.

3.2 Device Interaction

In order to determine the accuracy of the cliff detection algorithm, normal activity on the device must be included to establish a proper baseline and false positive rate. The EWEB has distinctive code execution during the boot process and a default idle operation where no user interaction is established leaving just the default operating system processes running. All device interactions are conducted with the base device firmware, with only a small modification to provide the vulnerable interaction that will act as the unknown or attack interaction. Three normal user interactions with the device are modeled. The first is simply exercising its network stack via a simple ICMP ping command run from a remote host targeting the EWEB. As the EWEB's main purpose is

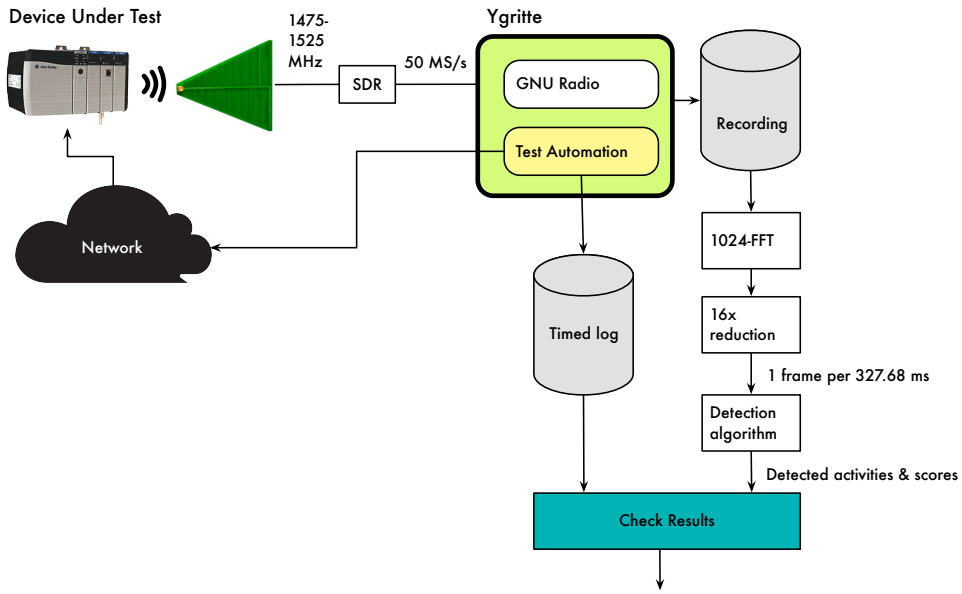

Figure 7. Summary of the experiment recording setup. Note that both the raw recordings and timed event log are saved so that the detection algorithm results can be evaluated with the recorded ground truth.

to facilitate network communication to the other devices on the PLC backplane, a sample of this backplane traffic using the CIP protocol is sent. The secondary purpose of the EWEB is to allow for various status and administrative activities via its built-in web server. A variety of HTTP requests are launched downloading files and requesting various web pages available.

For the unknown (attack) code, we have developed a memory scan payload. Such a memory scan is a crucial component of a variety of attack scenarios. For instance, an attacker will scan memory to find the layout in the presence of a memory randomization technique or when searching for a defensive program that must be disabled before the attacker can proceed. The firmware for the EWEB was modified to run the chosen unknown code when it receives an ICMP timestamp packet, which is a specialized ICMP packet not occurring during a simple Ping.

In summary, the following device states are considered in this paper:

- Boot
- Idle
- ICMP Pings
- Backplane traffic (CIP)
- HTTP Requests
- Unknown signals (memory scan)

3.3 Experiment Procedure

3.3.1 Recording

All data sets were collected and processed as described in Figure 7. We generate two sets of recordings: a training set with only normal device interactions and a "test" set, where the unknown (attack) code only executes during the "test" set of recordings. During each recording, an approximate timed event log of when each normal activity or unknown attack code executes is also recorded. In order to reduce noise and to reduce the algorithm running

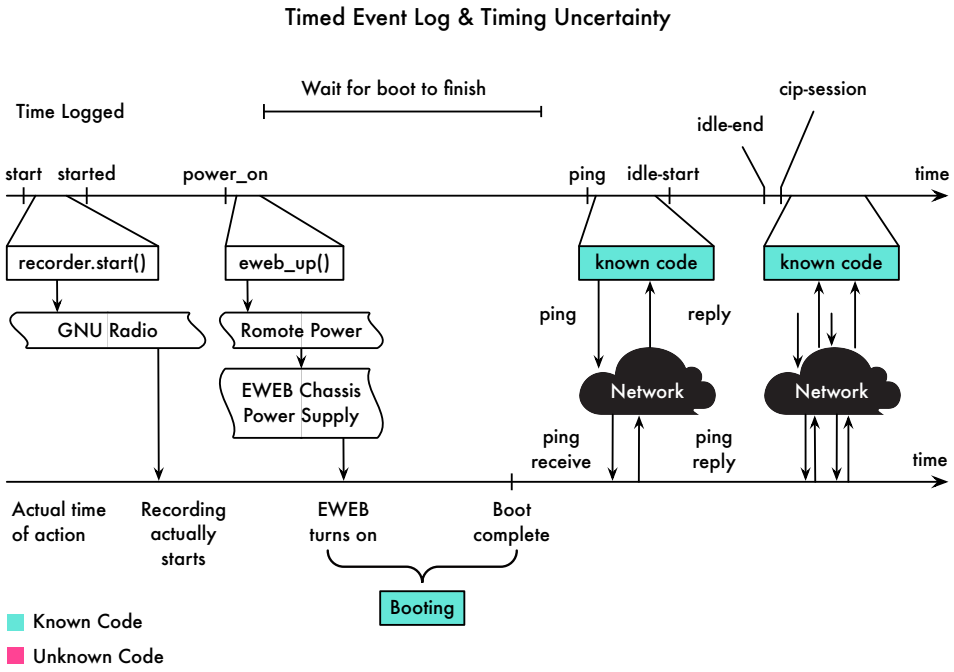

Figure 8. Visualization of how the ground truth timed event log is created. Each device interaction is time stamped with a start time of when the client starts the interaction and an end time of when the client receives a response or other known event conclusion with down time filled by an idle state label.

time, we take the mean of every 16 FFT frames and save it as a single frame thus reducing the data by a factor of 16. At the 50 megasamples per second, as shown in Figure 7, and a 1024 bin FFT each averaged frame represents 327.68 microseconds.

The time event log as seen in Figure 8 is the authoritative ground truth timing data for each device interaction. As these interactions occur over the network, perfect timing information of what the CPU on the EWEB module is actually executing at a given time is unavailable. While extreme measures such as profiling the CPU instructions executed during each normal activity and tracking precise timing of when such instructions are executed could in theory be achieved with a hardware debugger attached to the device, such a debugger itself affects the timing and could more critically affect the EM emanations coming from the device as it would require an additional cable attached.

Rather than go to these extreme measures with their downsides, we log the ground truth based on the start and end of an interaction on the remote host that is interacting with the EWEB module. By logging the time when the network packet is sent to the EWEB module and the time when a response is received, a time window within which the EWEB must have computed the normal activities triggered is created. Each of these time windows is then labeled as a normal or abnormal event in the evaluation phase.

3.3.2 Training

The goal of the training recordings is to provide the detection algorithm with the variety of expected normal data that can be used to build signatures. No attack or unknown code interactions take place during the training period. Only the recordings from the training period are used to generate signatures. Figure 9 provides an overview of the training period device interactions.

The training data sets include concentrated periods of a single activity as well as a period of random activities. These concentrated periods of each type of normal activity are designed to provide a stronger well known signal to train on. All activities are logged to the ground truth file.

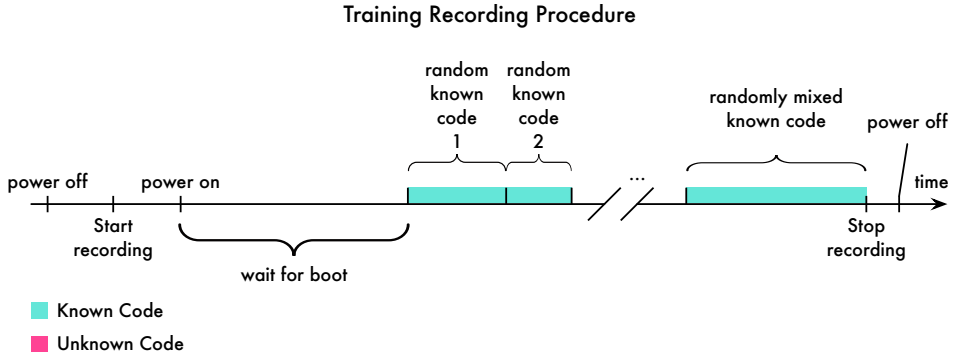

Figure 9. Time line and summary of device interactions recorded to create the training data sets that include only known interactions.

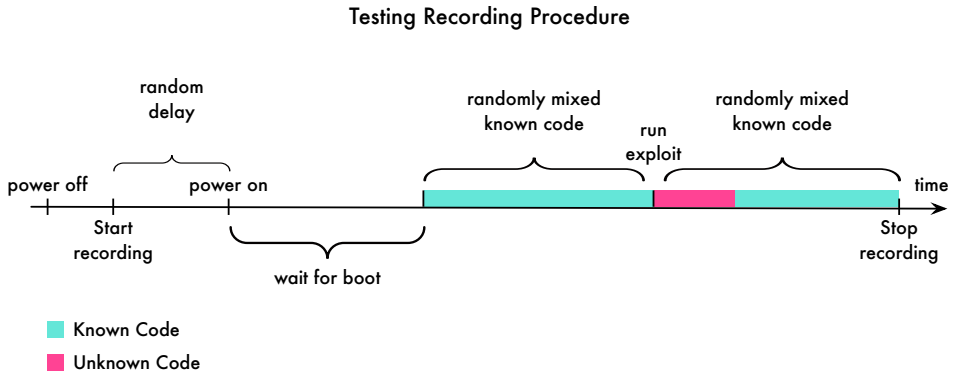

Figure 10. Time line and summary of device interactions recorded to create the test data sets that include the unknown (attack) code.

After the training data is recorded, the cliff detection algorithm is run on the data. Sets of new signals found at the same frame are then turned into a signature. A subset of these signatures are then saved to be used to detect abnormal signals in the test data. This subset of signatures is automatically chosen so that each signature is distinct in both timing, that is found many frames apart from one another, and set of FFT frequency bin ranges.

3.3.3 Testing

The test procedure only contains a randomized assortment of normal activities with the unknown code run at some point. This is designed to best simulate a normal workload on the device. Additional random delays are added to further avoid any results from being dependent on specific timing. All activities are logged to a ground truth log file in order to evaluate the results of the detection algorithm. For an idea of where the random delays are added and timeline of the test procedure see Figure 10.

Once the test data is recorded, the cliff detection algorithm is run against it with the set of signatures generated in the training step. Each frame's score is based on the signature that best matches that frame. This way as long as one signature marks the frame as normal, it will be considered normal. For each new signal in a frame that does not match the signature, a score penalty is assessed. These penalties where new signals are detected but do not match any signatures result in the detection of abnormal activity.

3.3.4 Evaluation

To evaluate the results of the trained signatures detecting the abnormal attack activity in the test data, the test data ground truth is parsed along with the per FFT frame scores generated by the trained signatures. See

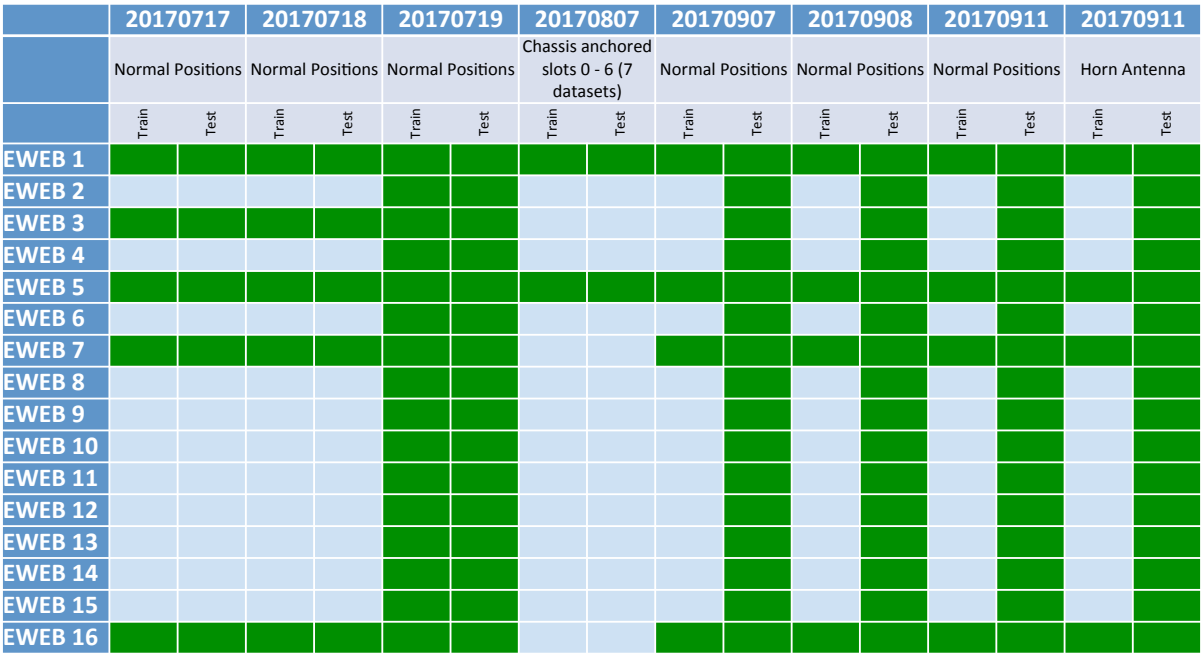

Figure 11. The 16 EWEB modules we aquired from a variety of sources were arbitrarily numbered for internal reference EWEB 1 through 16. The column is the day of the data collection with each dark green square indicating that data was collected on that date for that device with either training interactions and/or testing interactions recorded.

Figure 5 for a visualization. The timestamps of the ground truth log are mapped to the frame numbers based on the start time of the recording. Each event in the ground truth log is assigned a score that is the most abnormal of the scores of the frames it contains. This reflects the fact that in a production environment, in a particular time period any abnormal score will result in an alert.

Now that each event has an associated score based on the trained signatures, the false positives and true positives can be computed for any given threshold of abnormal. In a production environment, a fixed threshold would be set above which the frame would be labeled abnormal and an alert would be raised. To evaluate the trade off in terms of false positives and true positives, the threshold is varied to every possible score with the results logged. These logged results are then graphed as a ROC curve with the area under the curve (AUC) being computed as well to provide a single number comparision.

4. EXPERIMENT RESULTS

4.1 Datasets Recorded

With a single abnormal interaction, many data sets are required to compute accurate results. In total, over 7TB of compressed recordings were made over the course of three months. Table 11 indicates (in green) which date-device combinations are recorded as training and test sets.

These recordings vary in time and physical device used. While the same model number, each individual EWEB module has similar overall signal output but always with slight variations. The eleven EWEBs with the least data recorded were intentionally set aside to provide a pristine test once the experimental procedure was finalized and tested against signatures trained on the first five devices considered. The July recordings for these eleven EWEBs were only visually spot checked to confirm that the EWEBs were in fact properly operating. The chassis slot variation data set was only collected on two EWEBs as a training and testing data set was recorded with all seven slots resulting in a large amount of data beyond most recordings.

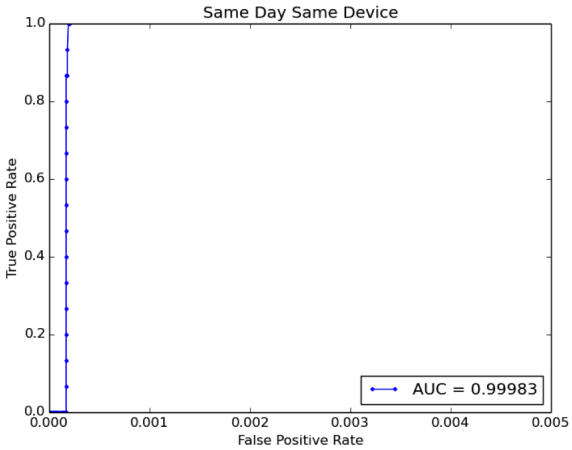

Figure 12.

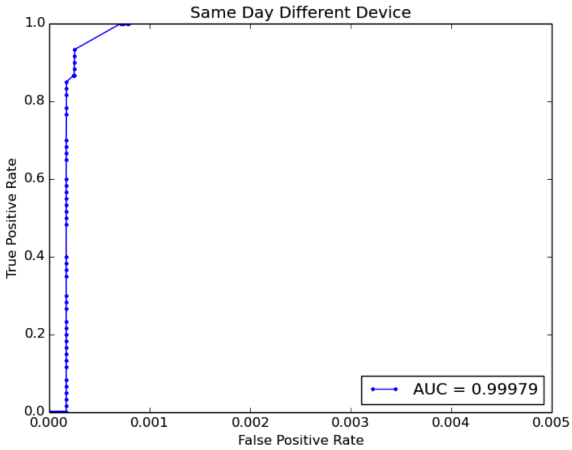

Figure 13.

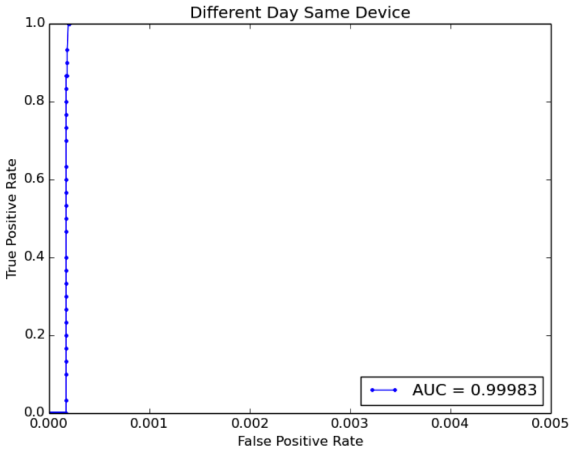

Figure 14.

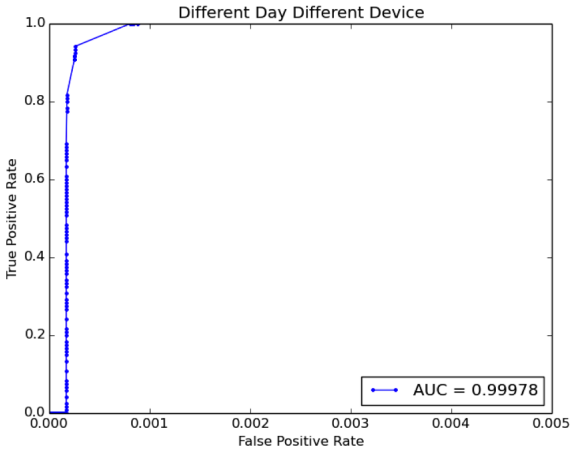

Figure 15.

4.2 Time and Device Difference

The first experiments use the data collected on the five devices across the first three days of data collection. We test whether training and testing against the same or different device and the same or different day impacts the results. This provides insight into how robust the training is to variations over time or separate hardware. Any random interference tied to a particular day or variation between hardware devices should be revealed.

Overall, the largest difference is found in hardware rather than major variations in time. The same device used in training and testing regardless of day had similar results. However, training on one device and testing against the other devices sees a significant increase in the false positive rate at higher true positive rates.

4.3 Meta Signatures

We now train with the meta event signatures added that were automatically detected by a merged set of training sets. See examples of these automatically generated meta signatures detecting the boot process in Figure 16. Figure 19 displays the ROC curves with the meta signatures integrated. The meta signatures do appear to significantly increase the accuracy mostly due to greatly reducing the false positives otherwise detected during the boot process due to the many signals present. See Section 2 for more details on how meta event signatures are automatically created.

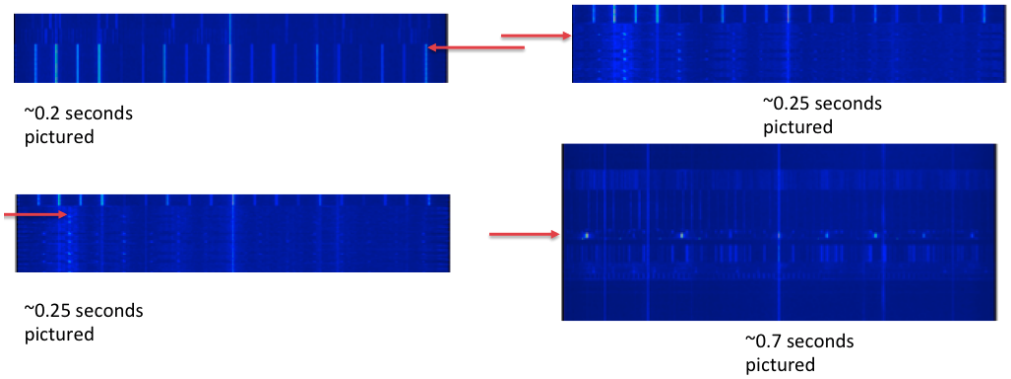

Figure 16. Waterfall plots of FFT frames with the red arrows indicating the synchronization events automatically found during the EWEB boot.

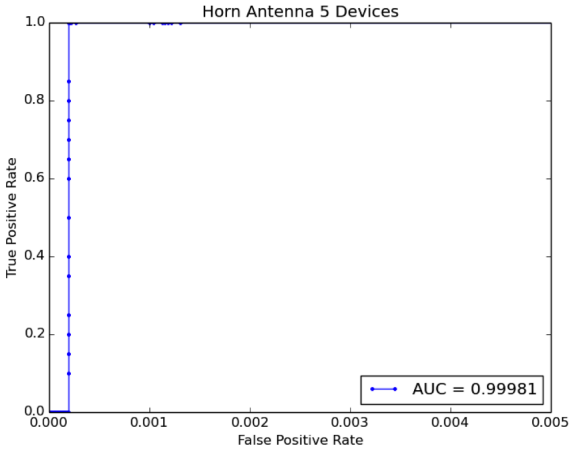

Figure 17.

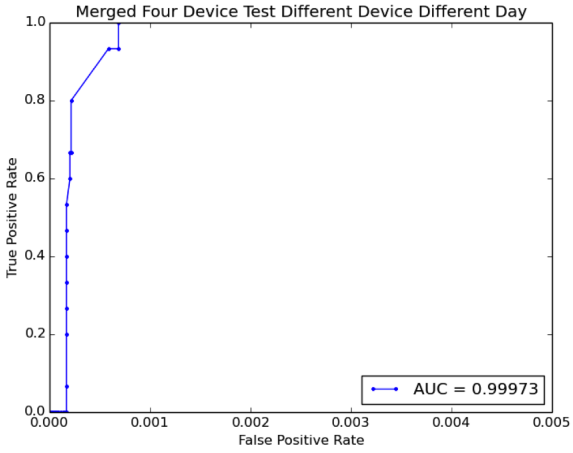

Figure 18.

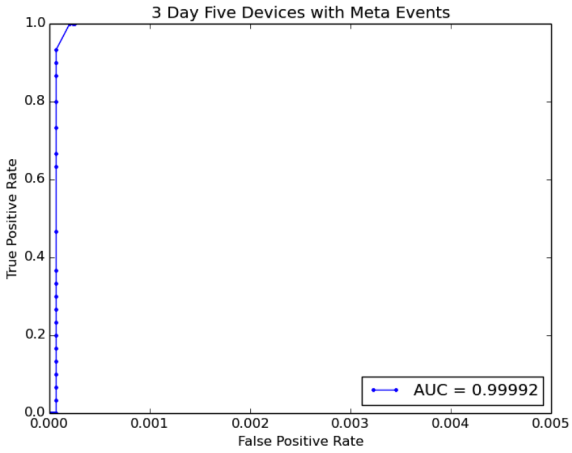

Figure 19.

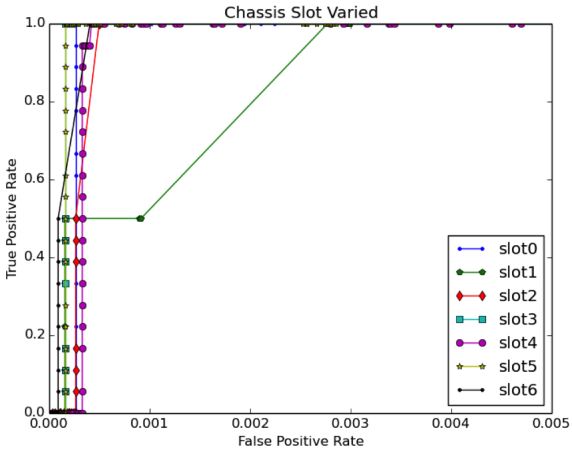

Figure 20.

See the results of using these meta event signatures in Figure 19. A clear improvement is seen as these meta event signatures are able to correctly identify the boot process normal activity in many cases. Without the meta event signatures, the boot event always appears abnormal as many strong signals of various frequencies appear during the boot process such that the existing signatures are never complete enough to find all such new signals normal.

4.4 Alternative Antenna

We collected an additional data set with a horn antenna rather than the log periodic antenna used for the rest of data collection to see if a different antenna would have a major impact on the results. Figure 17 displays these results. While more data would need to be collected to be sure, it appears that the horn antenna did not drastically change the results.

4.5 Merged Signatures

Merging signatures from multiple training data sets and only keeping signatures that are found in multiple data sets has the potential to reduce any overfitting that may occur. See Section 2 for more details on how signatures can be merged. Figure 18 visualizes the results of applying such a merging technique. These empirical results show that for the data considered here such merging does not seem particularly effective with similar results to the different day different device experiment.

4.6 Chassis Slot

We also tested changes in the chassis slot position on two devices. Slot 0 is the farthest slot from the antenna while slot 6 is the closest. The EWEB module is moved between slots, but the antenna remains one foot away from the edge of the chassis.

See Figure 20 for results. The slot does appear to have a dramatic effect at times with the slot 1 data showing a significant drop in performance. This is likely a fairly binary outcome where at some point the signals become weak enough that the cliff detection algorithm begins to fail to find new signals in the data. Further research into the maximum effective distance and and how an EWEB's position relative to the power supply affects performance will be required.

4.7 Fresh Devices

In order to further study how much variance there is between devices and to ensure that the techniques were not warped by the specific devices used in developing the algorithms, we conduct experiments with an additional 11 devices previously unused in any manner of testing or training. This helps to determine if any prior results are due to a over fitting to the five devices under test.

As seen in Figure 21, the initial results are significantly worse than previously seen. In investigating the difference with prior results, we ran the final thresholds separately on each device rather than taking a global score threshold to compute the true positives and false positives. The individual thresholding is seen in Figure 22. With this per device threshold, we see similar results as previously seen. Further investigation identified a couple devices that in general had a much higher amount of signals resulting in significant false positives when using the same scoring thresholds as other devices. However, when thresholded individually the unknown code still had even higher scores than the normal activities on these same noisier devices.

We also see a significant improvement when the meta signatures are applied to the 11 fresh devices. Figure 23 displays the results with meta signatures included for the 11 fresh devices. This is again due to the correct classification of the boot normal event whereas without the meta event signatures the boot event is always classified as abnormal resulting in a false positive.

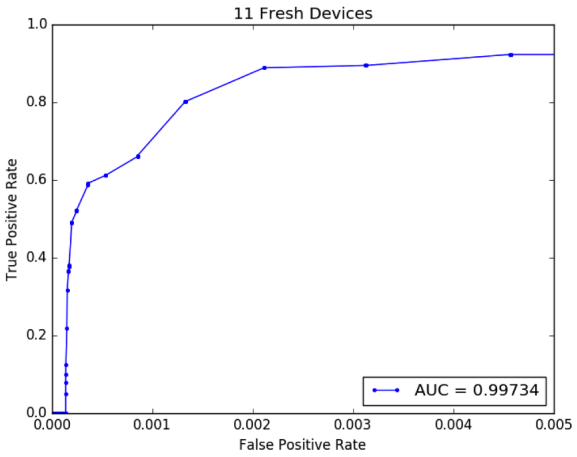

Figure 21.

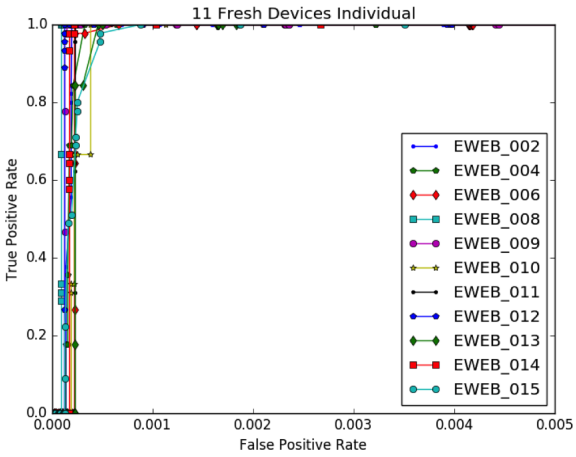

Figure 22.

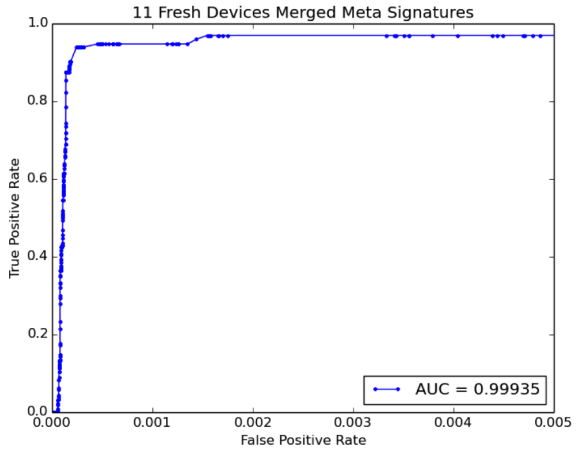

Figure 23.

Day	Device	AUC	Accuracy at 100% DR	True Positives at 100% DR	False Positives at 100% DR	Total Normal Events
Same	Same	0.999829	0.9998	15	17	88703
Different	Same	0.999829	0.9998	30	34	177406
Same	Different	0.999788	0.9993	60	252	354812
Different	Different	0.999783	0.9992	120	570	709624
Different	Merged different	0.999730	0.9993	30	122	177406
Different	Slot1 Different	0.998537	0.9972	18	318	114849

Figure 24. Summary results of the time / device variation, merge training set, and varying chassis slot experiments. Note that each row has a different number of total data sets so raw numbers should be normalized before direct comparisons.

Day	Device	AUC	Accuracy at 100% DR	True Positives at 100% DR	False Positives at 100% DR	Total Normal Events
Different	5 Device with Meta Signatures Different	0.999923	0.9998	30	36	177406
Same	Horn Antenna Different	0.999806	0.9998	20	20	102880
Different	11 Fresh Different	0.997337	0.9495	495	183470	3634275
Different	11 Fresh with Meta Signatures Different	0.999350	0.9831	495	61445	3634275

Figure 25. Summary results of the meta event signature, horn antenna, and 11 fresh device experiments. Note that each row has a different number of total data sets so raw numbers should be normalized before direct comparisons.

4.8 Result Summary

Figures 24 and 25 summarize the experiment results. Two key take aways are that significant detection of abnormal activity on the EWEB module is possible, but that while a low percentage, the false positive raw numbers are unacceptable for production use. Section 4.9 will go into further detail on identifying the primary causes of these false positives and future work that can potentially alleviate them. Note that as discussed above in Section 4.7, most of the false positives for the eleven fresh device experiments are caused by two particularly noisy EWEB modules and that such false positives are dramatically reduced when thresholds are considered on a per device basis.

4.9 False Positive Analysis

The above experiments identified three primary sources of false positives. The first is the boot event, which is due to large and rapid variations in signal frequency and overall high magnitude signals resulting in many cliffs unable to be fully modeled by enough signatures in training. The meta event signatures address longer events such as boot and are seen to be partially effective. The primary failure of meta event signatures is that they are for some data sets not automatically identified preventing their use in these automated experiments.

The second source of false positives is that two of the eleven fresh devices had significantly higher signal strength and noise pushing many more signals to be identified as abnormal for any given threshold. Simply thresholding on a per device basis rather than a global threshold for all devices does seem to correct this issue for the most part. Adapting per device thresholds does have production impact as it implies the requirement to have a per device calibration step for this cliff detection approach to be deployable.

The final source of false positives is a more subtle one where for some testing data sets the abnormal code would have only one or a handful of signals detected by the cliff algorithm at all. When only a single signal is detected for the abnormal event, any normal event that happens to have a bit of noise resulting in a signal unidentified by any signature will then also be classified as abnormal resulting in false positives. This hits upon a fundamental limitation of the cliff detection algorithm where attack code must generate significant new signals in order to be detectable likely excluding many attack types from possible detection with these techniques. Future work in additional detection algorithms and adaptations will be required to reduce these false positives. Additional metrics regarding the shape of signals or more sensitive signal detection may be promising research paths.

5. RELATED WORK

In,[1] the authors create an automatic approach to searching the spectrum to find EM emanations from computers that are frequency-modulated (FM) or amplitude-modulated (AM). Research has also been conducted on modeling the optimal antenna and EM side channel signal loss.[2] While identifying the optimal frequency range, signal information, and antenna type is important, it is beyond the scope of this work.

Significant research has been conducted on offensive usage of side channels. For example, side channel template attacks against cryptographic algorithms via EM emanations with up to 15 feet range[3] and power analysis[4] research has had good results in recovering key material. This offense oriented work and other direct research[5] has clearly demonstrated the significant information leakage problem that devices have.

In,[6] the authors profile the system activity via EM emanations to build a profile without the "Observer's Effect." Their focus is especially on tight loops. The work that most closely relates to this paper is,[7] in which the authors detect abnormal program execution using statistical tests to compare the signal distributions. This paper differs in the device and software considered here, the PLC EWEB module, which is a more complex device running the existing stock programs rather than researcher introduced code. The cliff detection algorithm for finding new signals also takes a fundamentally different approach with various tradeoffs.

6. FUTURE WORK

While an important first step, this work still has a long ways to go in order to be suitable for practical deployment as a defensive technology. Additional devices must be tested to determine how universally applicable the cliff detection algorithm is. As mentioned in Section 4.9 additional algorithms and approaches could be developed to reduce false positives. A survey of various attack scenarios could be conducted to determine, which classes of attacks are detectable by EM emanations.

Fundamental research on how and why devices emanate EM especially research towards understanding how different CPU and memory operations trigger different frequency components, will help in designing algorithms with higher information gain. Porting the algorithms described here into a real-time system is a vital step towards practical deployment as recording even a few minutes of data results in tens of gigabytes of information.

7. CONCLUSIONS

In conclusion, in this paper, we have leveraged the natural EM emanations of a PLC and in particular an EWEB module in this PLC to gain insight into its normal activities and detect abnormal activities via a novel signal cliff detection algorithm. The study of 16 different physical EWEB modules of the same model number acquired from a variety of sources indicates that these results are consistent across different hardware instances and not overfitted. While much future work lies before us before such techniques are ready for widespread deployment, these initial results demonstrate that such approaches are feasible to deploy for widely deployed critical infrastructure devices. In summary, the natural EM emanations of a device appear to be a promising source of data to detect at least certain attack types even on widely deployed PLC modules. Such out-of-band detection techniques can act as a critical layer of defense that does not increase the attack surface exposed to a host or network based attack and that the attacker will have no direct means of compromising as opposed to other defenses.

ACKNOWLEDGMENTS

We would like to thank our colleagues Timur Anthony and Joseph Pantoga for their help with the infrastructure and advice. This material is based upon work supported by the USAF, AFRL, and DARPA under Contract No. FA8650-16-C-7625. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the USAF, DARPA, or AFRL.

REFERENCES

[1] Prvulovic, M., Zajić, A., Callan, R. L., and Wang, C. J., "A method for finding frequency-modulated and amplitude-modulated electromagnetic emanations in computer systems," *IEEE Transactions on Electromagnetic Compatibility* **59**, 34–42 (Feb 2017).

[2] Zajic, A., Prvulovic, M., and Chu, D., "Path loss prediction for electromagnetic side-channel signals," in [*2017 11th European Conference on Antennas and Propagation (EUCAP)*], 3877–3881 (March 2017).

[3] Chari, S., Rao, J. R., and Rohatgi, P., [*Template Attacks*], 13–28, Springer Berlin Heidelberg, Berlin, Heidelberg (2003).

[4] Choudary, O. and Kuhn, M. G., [*Efficient Template Attacks*], 253–270, Springer International Publishing, Cham (2014).

[5] Zajic, A. and Prvulovic, M., "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," *Electromagnetic Compatibility, IEEE Transactions on* **56**(4), 885–893 (2014).

[6] Sehatbakhsh, N., Nazari, A., Zajic, A., and Prvulovic, M., "Spectral profiling: Observer-effect-free profiling by monitoring em emanations," in [*2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*], 1–11 (Oct. 2016).

[7] Nazari, A., Sehatbakhsh, N., Alam, M., Zajic, A., and Prvulovic, M., "Eddie: Em-based detection of deviations in program execution," in [*Proceedings of the 44th Annual International Symposium on Computer Architecture*], *ISCA '17*, 333–346, ACM, New York, NY, USA (2017).