Type of Penetration Test		White, Black, Grey				
Task Categories		**Penetration Testing Tasks**	**Level of Effort:**	**Task Description:**	**Task Goal:**	**Required Submittal**
Architecture Review		Documentation Review	Low			Y
		Onsite Interviews	Low			Y
		Online Recon	Low			Y
4.0 Embedded Device Penetration Tasks	**4.1 Electronic Component Analysis**	4.1.1 Device Dissassembly	Low	Disconnect power from the device and disassemble the device to gain access to the embedded electronic components. Attempt to do a non--destructive disassembly if possible. Document the entire process to later facilitate reassembly. Identify the existence and function of any physical tamper mechanisms protecting the device.	Gain physical access to embedded components and electronic buses for further testing. Identify any methods that could be used to bypass the tamper mechanisms.	Y
		4.1.2 Circuit Analysis	Low	Document the electronic circuit by taking pictures, reading chip IDs, tracing buses, and identifying major electronic functionality.	Gain information about the embedded hardware and identify potential electronic components for attack.	Y
		4.1.3 Datasheet Analysis	Medium	Find, download, and analyze all pertinent datasheets and related documentation for each major electronic component inside the device, to identify possible security weaknesses and attack angles.	Gain information about the function of each component and how to interface directly with each component. Identify target components and buses for following tasks.	Y
		4.1.4 Dumping Data at Rest	Medium	Using the datasheets, identify the pins necessary to perform data dumping. With the device powered off, connect your testing tools and perform the dump. If needed, be sure to disable any other component by triggering reset pins or by using other methods. Review the dumped data to determine if you were successful. Attempt multiple dumps and compare the results if you are doubtful about your success.	Obtain all data from unprotected storage devices for later analysis.	Y

		4.1.5 Snooping Data in Motion	Medium	Using the datasheets previously obtained, identify the pins and traces needed to perform bus snooping. With the device powered off, connect the testing tools and begin capture. Power on the device and capture sufficient data samples from each target bus. Review dumped data to identify if you were successful. Attempt multiple dumps and compare results if you are doubtful about your success.	Obtain data samples from all major buses for later analysis.	Y
		4.1.6 String Analysis	Low	Use tools and multiple decoding methods to decode each obtained data. Within the logical context of the data source, identify human readable strings and other anomalies. Other identifiers may be byte patterns signifying where firmware image files begin and end.	Identify symmetric cryptographic keys, firmware images, and other items of interest.	Y
		4.1.7 Entropy Analysis	Low to Medium	Analyze obtained data sets for blocks of data that portray high levels of entropy. Small data blocks with high entropy often signify asymmetric cryptographic keys and usually correspond to common key length sizes. Larger data blocks with high levels of entropy often signify encrypted data. Attempt to use suspected cryptographic keys to decrypt encrypted data blocks or encrypted communications traffic.	Identify asymmetric cryptographic keys and encrypted data objects.	Y
		4.1.8 Systematic Key Search	Low	Use tools to identify cryptographic keys by attempting to use possible blocks of data from each obtained data set as the cryptographic key. For instance, if the tool is trying to identify a 128 bit symmetric key, the tool will systematically attempt to use each 128 bit data block as a potential cryptographic key to decrypt a known block of encrypted data or a known capture of encrypted communications traffic. In this case, the tool will try bits 0 through 127 as a potential cryptographic key, then try bits 1 through 128, then bits 2 through 129, and so on.	Identify symmetric and asymmetric cryptographic keys.	Y

		4.1.9 Firmware Carving	High	Reverse engineering of the data in an attempt to understand its purpose. For instance, testers could attempt to understand the captured data blocks to determine what each set of bytes represent in the serial bus protocol or the data stored in the flash/EEPROM chips. This could be done by sending known commands or setting known configurations and attempting to identify in the data blocks where those commands and configurations are transmitted and stored.	Identify the purpose of blocks of data that could be used in exploitation attempts.	
		4.1.10 Decoding Retrieved Data	High to Extremely High	Based on the findings from previous tasks, determine feasible attacks that can be launched on the embedded components.	Create proof of concept attacks to demonstrate the feasibility and business risk created by the discovered vulnerabilities.	
	4.2 Field Technician Interface Analysis	4.2.1 Interface Functional Analysis	Low	Obtain required software and hardware to establish an appropriate connection to the field device, be it a serial port, infrared port, or digital display. Identify the intended functionality and features of the interface. Identify any unprotected or high -risk functions that attackers may be interested in exploiting, such as firmware updates, configurations, or security table reads.	Gain an understanding of the interface feature set and identify functions that should be targeted for later tasks.	Y
		4.2.2 Field Technician Interface Communications Capture	Low	Use a hardware or software tool to intercept normal communications on the interface. Capture all identified target functions from previous tasks.	Obtain low--level capture of targeted functions.	Y
		4.2.3 Field Technician Interface Capture Analysis	Medium	Analyze interface captures, identifying weaknesses in authentication, authorization, and integrity controls. Gain an understanding of how data is requested and commands are sent. If the protocol uses authentication, attempt to identify the passwords or keys being sent before a session is established. For example, in the case of protocols such as C12.18 for AMI meters, attempt to identify the different levels of passwords being sent before each command.	Identify potential vulnerabilities and attacks.	Y

	4.2.4 Field Technician Interface Endpoint Impersonation	Low to Medium	Use an attack tool to impersonate either end of the field technician interface. For instance, this attack tool could simulate the field technician tool while communicating with the field device interface, or the attack tool could simulate the field device interface while communicating with the field device tool.	Obtain a usable attack point to perform later tasks.	Y
	4.2.5 Field Technician Interface Fuzzing	Medium to High	Use or create a fuzzing tool to send both valid and invalid communications to the target interface, analyzing the results and identifying anomalies. This task includes items such as password guessing, invalid input testing, data enumeration, etc.	Identify vulnerabilities in the interface implementation and obtain data not otherwise available from any field device vendor tool provided to the utility.	
	4.2.6 Field Technician Interface Exploitation	High to Extremely High	Based on the findings from previous tasks, determine feasible attacks that can be launched on the field technician interface. Attempt to use any authentication or cryptographic keys retrieved from one meter on different meters to identify shared passwords and cryptographic keys.	Create proof of concept attacks to demonstrate the feasibility and business risks created by the discovered vulnerabilities.	
4.3 Firmware Binary Analysis	4.3.1 Firmware Binary Disassembly	Medium	If firmware is successfully retrieved and the tester has sufficient time and skill, disassemble the firmware and attempt to identify vulnerabilities in the instruction calls. Warning, this task often proves very difficult as many microprocessors do not have publicly available decompilers. Consequently, one may need to be created first would could result in this becoming an “Extremely High” level of effort.	Obtain a human readable version of the firmware for later analysis.	Y
	4.3.2 Firmware Binary Code Analysis	High to Extremely High	Identify weaknesses in memory use, loop structures, cryptographic functions, interesting functions, etc. This could also include the extraction of cryptographic keys or algorithms hardcoded into the firmware.	Identify vulnerabilities that can be exploited.	Y

		4.3.3 Firmware Binary Exploitation	High to Extremely High	Based on the findings from previous steps, determine feasible attacks which can be launched at the firmware. For instance, cryptographic materials found in the firmware could be used to access protected networks and devices, or buffer overflow like attacks could be leveraged to run arbitrary code on remote devices.	Create proof of concept attacks to demonstrate the feasibility and business risk created by the discovered vulnerabilities. Create proof of concept attacks to demonstrate the feasibility and business risks created by the discovered vulnerabilities.	Y
5.0 Network Communications Penetration Tasks	**5.1 Network RF Testing**		Medium	Use a tool (such as a USRP2) to capture the RF communications of the target field device. Discover of the frequencies used are usually straightforward by referencing the FCC or other regulatory license IDs printed on the outside of the transmitting device, through vendor documentation, or even patent filings.	Obtain data for following tasks.	
			Extremely High	If Spread Spectrum (SS) techniques are used on the signal, knowledge of the SS algorithm must be obtained either from documentation, through recovery in the disassembled firmware, or through capture of all signal components in the used spectrum. Use of a tool such as GNU Radio to capture and discover the algorithm is possible, but very time consuming.	Obtain data for following tasks.	
			Medium	Use a tool such as GNU Radio to demodulate the signal. If spread spectrum technologies are used, this greatly increases the level of effort of this task.	Obtain data for following tasks.	
			Medium	Use a tool to decode and extract communications payload from RF capture.	Obtain data for following tasks.	
			Medium to High	Use a tool to transmit RF signals at the appropriate frequencies and hopping patterns to either replay captured data, impersonate the target field device, or attempting to cause denial of service scenarios.	Identify vulnerabilities in the RF signaling.	
	5.2 Network Protocol Testing	5.2.1 Network Protocol Traffic Capture	Low	Use a tool to capture sample communications. Attempt to cause known actions that result in communications between devices, such as firmware updates, and capture this communication individually to facilitate later analysis. Obtain samples of all target functionality.	Obtain data for the following tasks.	Y

	5.2.2 Network Protocol Cryptographic Analysis	Medium	If the traffic capture uses a known protocol, identify the negotiated cryptographic algorithm and key length to determine if any known vulnerabilities exist. If traffic capture is using an unknown protocol and is not readable, extract payloads from the captured network traffic and perform an entropy analysis to determine if the data is encrypted. High levels of entropy among the payload bytes often signify that encryption is being used, and weaknesses in cryptographic implementations can often be determined by variations in that entropy.	Determine if cryptography is being used and identify any vulnerabilities.	
	5.2.3 Unknown Protocol Decoding	High to Extremely High	If traffic capture is using an unknown protocol, reverse engineer the network captures in an attempt to understand the protocol. Analyze each capture in light of the actions performed to initiate that traffic. For instance, if analyzing a traffic capture of a firmware update, try to identify the firmware being sent in the payload. Additionally, analyze actions such as initial registration between devices to determine if an authentication mechanism is being used.	Identify the purpose of blocks of data that could be used in later analysis.	Y
	Protocol Enumeration				Y
	5.2.4 Network Protocol Fuzzing	Medium to High	Use a tool to send both valid and invalid communications to both end points of the communications link individually, analyzing the results and identifying anomalies. This task includes items such as password guessing, invalid input testing, data enumeration, replaying data, susceptibility to Man--in--the--Middle (MitM) attacks, etc.	Identify vulnerabilities in the network protocol implementation.	

		5.2.5 Network Protocol Exploitation	High to Extremely High	Based on the findings from previous tasks, determine feasible attacks which can be launched on the field technician interface. For example, if devices are not required to authenticate themselves when joining a field area network, it may be possible to insert a ‘rogue’ node in the network or to harvest controlled devices away from their management server such as AMI headends or synchrophasor managers. Another example might be spoofing a firmware update or disconnect signal or perform an active MitM attack.	Create proof of concept attacks to demonstrate the feasibility and business risk created by the discovered vulnerabilities.	
6.0 Server OS Testing	**6.1 Information Gathering**	6.1.1 DNS Interrogation	Low	Use tools to attempt zone transfers and perform queries from target Domain Name Service (DNS) servers.	Identify targets, verify ownership, and detect anomalies.	Y
		6.1.2 Port Scanning	Low	Use tools that send requests to possible application layer services (such as scanning TCP and UDP ports to discover services like HTTP and SSH).	Identify all listening services and possible firewall rules.	Y
		6.1.3 Fingerprinting	Low	Use tools to examine listening services.	Identify the nature and function of all listening services.	Y
		6.1.4 SNMP Enumeration	Low	Use tools to attempt to examine SNMP services.	Identify insecure SNMP services, extract information about the endpoints, and identify vulnerabilities that allow attackers to reconfigure endpoints.	Y
		6.1.5 Packet Sniffing	Low	Capture various samples of network communications.	Collect samples for later analysis.	Y
	6.2 Vulnerability Analysis	6.2.1 Unauthenticated Vulnerability Scanning	Medium	Use automated tools without credentials to identify known vulnerabilities in network services and their respective systems.	Identify vulnerabilities in the operating system and the network services	
		6.2.2 Authenticated	Medium	Use automated tools that use valid credentials to	Identify vulnerabilities in the operating system and	
		6.2.3 Vulnerability Validation	Medium	Manually validate findings from automated tools where possible. Merge and combine findings where applicable.	Consolidate findings and remove any false positive findings that you identify.	
		6.2.4 Packet Capture Analysis	Low to Medium	Examine network traffic samples and look for protocols with known vulnerabilities such as session hijacking, weak authentication, or weak/no cryptographic protections.	Identify vulnerabilities in network protocols and network communications.	Y

	6.3 Exploitation	6.3.1 Identify Attack Avenues	Medium	Review all findings and outputs from previous tasks and identify plausible attacks that have a moderate chance of success. Prioritize these possible attacks by likelihood and the tester's ability to execute them.	Organize and plan next steps.	
		6.3.2 Vulnerability Exploitation	Low to Medium	Create proof of concept attacks to demonstrate the feasibility and business risk created by the discovered vulnerabilities. Once a vulnerability has been exploited, attempt to pivot and identify additional vulnerabilities to exploit.	Validate the assumed business risk created by the identified vulnerabilities and identify additional targets of opportunity.	Y
		6.3.3 Post Exploitation	Low to Medium	Remove any code, data, or configurations that were added to the system as a part of the assessment.	Return the systems to their pre--assessment state.	Y
7.0 Server Application Penetration Tasks	**7.1 Application Mapping**	7.1.1 Application and Platform Fingerprinting	Low	Use tools to query the application service to identify the platform type and version hosting the application. (Such as Apache and Tomcat)	Identify the application server and technologies used to host the application.	Y
		7.1.2 Functional Analysis	Low	Gain an understanding of the application from the user's perspective. Explore the application and identify major functionality and features exposed to the user. Identify major sections and portions of the application, including the user roles.	Gain a better understanding of the application for later analysis.	Y
		7.1.3 Process Flow Modeling	Low	Model the process flows that users must follow while using the application. Identify dependencies between actions and requirements to get to each portion of the application.	Gain a better understanding of the application for later analysis.	Y
		7.1.4 Request/Resource Mapping	Low	Attempt to map, execute, and record every possible request in the application. Examine the requests and responses to understand how the application works from the developer's perspective. Identify parameter names and values that are reflected back to the user or appear to be used in a database query.	Identify requests that have a higher probability of containing vulnerabilities. Prioritize for later analysis.	Y
	7.2 Application Discovery	7.2.1 Default Configuration Testing	Low	Test the platform and application server configuration, such as SSL/TLS testing, file extension handling, method handling, and the existence of administrative interface and unreferenced links.	Identify vulnerabilities in the application.	Y

		7.2.2 Authentication Testing	Low	Test the application authentication for flaws such as user enumeration, guessable passwords, authentication bypass, flawed password reset, race conditions, multifactor authentication, and CAPTCHA implementation weaknesses.	Identify vulnerabilities in the application.	Y
		7.2.3 Session Management Testing	Low	Test the application for session management flaws such as session fixation, session hijacking, unprotected session keys, and Cross Site Request Forgery (CSRF).	Identify vulnerabilities in the application.	Y
		7.2.4 Authorization Testing	Low	Test the application for authorization flaws such as path traversal, authorization bypass, and privilege escalation.	Identify vulnerabilities in the application.	Y
		7.2.5 Business Logic Testing	Low	Test the business logic flow and user process flow to verify steps that cannot be skipped or re--ordered.	Identify vulnerabilities in the application.	Y
		7.2.6 Code Injection Testing	Low	Test the application for data validation flaws such as XSS, SQL Injection, LDAP injection, XPath Injection, overflows, format string issues, and HTTP Splitting.	Identify vulnerabilities in the application.	Y
		7.2.7 Denial of Service Testing	Low	Test the application for flaws that may cause denial of service vulnerabilities either on the service platform, in the application logic, or on the backend systems and databases.	Identify vulnerabilities in the application.	Y
		7.2.8 Client--Side Code Testing	Low	Test the application for flaws in the use of mobile or client--side code.	Identify vulnerabilities in the application.	Y
	7.3 Application Exploitation	7.3.1 Identify Attack Avenues	Medium	Review all findings and outputs from previous tasks and identify plausible attacks that have a moderate chance of success. Prioritize these possible attacks by likelihood and the tester’s ability to execute them.	Organize and plan next steps.	Y
		7.3.2 Vulnerability Exploitation	Low to Medium	Create proof of concept attacks to demonstrate the feasibility and business risk created by the discovered vulnerabilities. Once a vulnerability has been exploited, attempt to pivot and identify additional vulnerabilities to exploit.	Validate the assumed business risks created by the identified vulnerabilities and identify additional targets of opportunity.	Y
		7.3.3 Post Exploitation	Low to Medium	Remove any code, data, or configurations that were added to the system as a part of the assessment.	Return systems to their pre--assessment state.	Y

8.0 End--to--End Penetration Test Analysis		8.1 Gap Analysis	Low	The final task in any penetration test should be a gap analysis of communications that span the entire system. This should include a review of input and output from external systems that may not be in scope for this assessment. For instance, when testing an AMI meter system, a tester might have performed tests on all components from the meter to the headend. However this final end--to--end task should ensure that all possible inputs from external systems to in--scope systems have been tested and evaluated as possible attack angles, such as an out--of--scope backend systems dependent on data from the in--scope system. Also, malicious data from out--of--scope systems that is accepted and used by in--scope systems, such as public key infrastructure (PKI) servers, should be considered in this part of the assessment. Penetration testers should also identity if any vulnerabilities found later in the testing process affect components tested earlier or by other testing teams.		
9.0 Result Interpretation and Reporting		9.1.1 Executive Summary	Low	A brief 1--2 page section discussing the overarching root causes for the vulnerabilities and high level business strategies to address these root causes.		Y
		9.1.2 Introduction	Low	A short section describing the goals of the tests, components that were in and out of scope, any special restrictions on the tests, and the team involved with the testing.		Y
		9.1.3 Methodology	Low	A short section of the report focuses on the technical reasons for the test as well as the methodology used.		Y
		9.1.4 Findings and Recommendations	Low	this section of the report is traditionally the longest, most detailed, and highly technical. This is the core of the report for future use and reference. This section may also discuss the likelihood and impact of each vulnerability within the context of the proposed or existing deployment.		Y

	9.1.5 Conclusion	Low	A section similar to the executive summary but at a more technical depth summarizing the major findings and recommendations. This section should also discuss any major questions or goals of the assessment such as the team’s recommendations of a go no--go purchase of a product.		Y