Threat Assessment and Remediation Analysis (TARA) Workbook for Industrial Control Systems / Supervisory Control and Data Acquisition (ICS/SCADA)

Version 1.4

Jackson Wynn

15-May-18

Approved for Public Release
Distribution Unlimited. Case Number 18-1773.

Background

The International Atomic Energy Agency (IAEA) Coordinated Research Project (CRP) J02008 Enhancing Computer Security Incident Analysis at Nuclear Facilities[1] seeks to improve computer security capabilities at nuclear facilities by expanding the common vocabulary that network defenders, incident responders, analysts, and investigators use when describing cyber-attacks. This effect is achieved in part by taking an existing, widely-used catalog of cyber-attack patterns—the Common Attack Pattern and Enumeration Catalog (CAPEC)—and expanding it to include attack patterns targeting the Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA) systems. This spreadsheet contains open source threat vector and countermeasure data useful to development of cyber threat scenarios that target ICS/SCADA components commonly found in Nuclear Power Plant (NPP) control systems. Data collected in this spreadsheet comes from a variety of public domain sources, including CAPEC[2], ATT&CK[3], CVE[4], CWE[5], and ICS-SCADA Alerts[6].

Industrial Control Systems / Supervisory Control and Data Acquisition (ICS/SCADA)

ICS are typically used in industries such as electric, water and wastewater, oil and natural gas, transportation, chemical, pharmaceutical, pulp and paper, food and beverage, and discrete manufacturing (e.g., automotive, aerospace, and durable goods.) SCADA systems are generally used to control dispersed assets using centralized data acquisition and supervisory control. These control systems are vital to the operation of the U.S. critical infrastructures that are often highly interconnected and mutually dependent systems.[7]

Organization of Document

This compilation is based on the TARA[8] catalog format. Threat Assessment and Remediation Analysis (TARA) is a MITRE-developed methodology for evaluating the susceptibility of a system to cyber attack, assess the risk, and identify countermeasures effective at preventing or mitigating that risk. The TARA methodology utilizes a catalog of threat vectors, countermeasures, and mapping data. A TARA catalog was developed for the J02008 CRP that contains threat vectors and countermeasures applicable to a wide range of cyber and cyber-physical systems containing ICS/SCADA components. The catalog tool export function was used to generate an Excel workbook that includes tabs for Threat Vectors, Countermeasures, and Mappings, which are contained in this document. In TARA, a threat vector is a sequence of steps performed by an adversary in the course of conducting a cyber attack. Countermeasures include actions, devices, procedures, or techniques that meet, oppose or counter an attack by preventing it, by minimizing the harm it can cause, or by discovering and reporting it in order that corrective action can be taken. A mapping is an association representing the type of effect that a countermeasure has on an threat vector.

Threat Vector	***Sequence of steps performed by an adversary in the course of conducting a cyber attack***
AV ID	*Unique ID assigned to threat vector*
Vector Name	*Headline description of threat vector*
Description	*Detailed description of threat vector*
Category	*Type of cyber attack: [Cyber, Cyber Physical, Supply Chain, Social Engineering]*
Objective	*Adversary's intended objective*
Prerequisite	*Condition that must occur in order for the attack to be successful*
Reference	*External reference(s) for additional information*
Countermeasure	***Actions, devices, procedures, or techniques that meet, oppose or counter an attack by preventing it, by minimizing the harm it can cause, or by discovering and reporting it in order that corrective action can be taken.***
CM ID	*Unique ID assigned to countermeasure*
Countermeasure Name	*Headline description of countermeasure*
Description	*Detailed description of countermeasure*
Effect	*Type(s) of countermeasure effects: [Prevent, Detect, Respond]*
Lifecycle	*Phase of system lifecycle countermeasure would be deployed in*
Maturity	*Maturity level of countermeasure: [High, Medium, Low]*
Cost	*Lifecycle cost of ownership of countermeasure: [High, Medium, Low]*
Reference	*External reference(s) for additional information*
Mapping	***An association representing the type of effect that a countermeasure has on an threat vector***
Prevent High (PH)	*Preventative effect - High confidence (Verified through testing)*
Prevent Moderate (PM)	*Preventative effect - Moderate confidence (Cyber SME best judgement)*
Prevent Low (PL)	*Preventative effect - Low confidence (Unverified effect)*
Respond High (RH)	*Responsive effect - High confidence (Verified through testing)*
Respond Moderate (RM)	*Responsive effect - Moderate confidence (Cyber SME best judgement)*
Respond Low (RL)	*Responsive effect - Low confidence (Unverified effect)*
Detect High (DH)	*Detective effect - High confidence (Verified through testing)*
Detect Moderate (DM)	*Detective effect - Moderate confidence (Cyber SME best judgement)*
Detect Low (DL)	*Detective effect - Low confidence (Unverified effect)*

Threat Vectors

AV ID	Vector Name	Description	Category	Objectives	References
T000001	BIOS replaced with version that allows unsigned updates	The adversary replaces existing BIOS with a version that accepts unsigned BIOS updates or updates with invalid checksums.	Cyber Physical	Disrupt	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
T000002	Bypass secure BIOS update	The adversary triggers a buffer overflow by executing an unsigned portion of a firmware update package before write-protections are enabled	Cyber Physical	Degrade; Disrupt	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
T000003	Trusted insider installs malicious BIOS for future exploitation	Trusted user intentionally installs known bad BIOS image containing exploitable vulnerabilities that can be later used to target the device.	Cyber Physical	Disrupt; Implantation	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
T000004	Malware reflashes device with malicous BIOS	Malware infects device and exploits vulnerabilities in system BIOS to reflash or modify the system BIOS	Cyber Physical	Degrade; Disrupt	http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf
T000005	Device rolled back to vulnerable BIOS image	The adversary initiates BIOS roll back that reloads previous image containing vulnerabilities.	Cyber Physical	Deceive; Degrade	http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf
T000006	Insider uploads malicious BIOS to update server for enterprise-wide distribution	The adversary loads a malicious BIOS image into the update server used to distribute new BIOS within the enterprise, and pushes it out to devices to bypass image checksum verification and trusted distribution path controls.	Supply Chain	Degrade; Disrupt	http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf
T000007	Targeting devices that use unsecure versions of SNMP protocol	The adversary targets devices that are managed using SNMPv1 and v2 protocols, in which community strings are sent in the clear. Community strings provide basic authentication of SNMP management platforms and managed devices. SNMP v1/v2c protocols transmit community strings as cleartext, which can be sniffed and spoofed to gain unauthorized access to the managed device.	Cyber	Disrupt; Ex-filtrate	http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/service/8_5_1/admin/sasnmpv1.pdf
T000008	Unsecured SNMP agent	The adversary gains access to device configuration data through its SNMP agent, which is enabled by default. For many SNMP-managed devices, default SNMP community strings are factory set to 'public' and 'private', respectively, and/or are reset to values that can be easily guessed. SNMP community strings used in one managed device are often used in other managed devices on the same network.	Cyber Physical	Degrade; Ex-filtrate	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2005-0834

T000009	Predicting a session credential to gain unauthorized access	This attack targets predictable session ID in order to gain privileges. The attacker can predict the session ID used during a transaction to perform spoofing and session hijacking.	Cyber	Disrupt; Ex-filtrate; Penetration	http://capec.mitre.org/data/definitions/59.html;
					http://cwe.mitre.org/data/definitions/290.html;
					http://cwe.mitre.org/data/definitions/330.html;
					http://cwe.mitre.org/data/definitions/331.html;
					http://cwe.mitre.org/data/definitions/346.html;
					http://cwe.mitre.org/data/definitions/488.html

T000010	HTTP Request Smuggling	HTTP Request Smuggling results from the discrepancies in parsing HTTP requests between HTTP entities such as web caching proxies or application firewalls. Entities such as web servers, web caching proxies, application firewalls or simple proxies often parse HTTP requests in slightly different ways. Under specific situations where there are two or more such entities in the path of the HTTP request, a specially crafted request is seen as two different sets of requests. This allows requests to be smuggled through to a second entity without the first one realizing it.	Cyber	Deceive; Penetration	http://capec.mitre.org/data/definitions/105.html ; http://capec.mitre.org/data/definitions/33.html ; http://secappdev.org/handouts/2009/advanced%20web%20application%20security.pdf
T000011	Lifting Data Embedded in Client Distributions	An attacker can ex-filtrate sensitive data embedded or cached in client code. This information may include PII or account access information that could be leveraged in future attacks.	Cyber	Ex-filtrate	http://capec.mitre.org/data/definitions/37.html
T000012	Postfix, Null Terminate, and Backslash	If a string is passed through a filter of some kind, then a terminal NULL may not be valid. Using alternate representation of NULL allows an attacker to embed the NULL midstring while postfixing the proper data so that the filter is avoided. One example is a filter that looks for a trailing slash character. If a string insertion is possible, but the slash must exist, an alternate encoding of NULL in midstring may be used.	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/53.html

T000013	Exploiting Trust in Client	An attack of this type exploits a programs' vulnerabilities in client/server communication channel authentication and data integrity. It leverages the implicit trust a server places in the client, or more importantly, that which the server believes is the client. An attacker executes this type of attack by placing themselves in the communication channel between client and server such that communication directly to the server is possible where the server believes it is communicating only with a valid client.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/22.html;
					http://cwe.mitre.org/data/definitions/20.html;
					http://cwe.mitre.org/data/definitions/200.html;
					http://cwe.mitre.org/data/definitions/287.html;
					http://cwe.mitre.org/data/definitions/290.html;
					http://cwe.mitre.org/data/definitions/693.html

T000014	Accessing, Intercepting, and Modifying HTTP Cookies	This attack relies on the use of HTTP Cookies to store credentials, state information and other critical data on client systems. The first form of this attack involves accessing HTTP Cookies to mine for potentially sensitive data contained therein. The second form of this attack involves intercepting this data as it is transmitted from client to server. This intercepted information is then used by the attacker to impersonate the remote user/session. The third form is when the cookie's content is modified by the attacker before it is sent back to the server. Here the attacker seeks to convince the target server to operate on this falsified information	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/31.html; http://cwe.mitre.org/data/definitions/113.html; http://cwe.mitre.org/data/definitions/302.html; http://cwe.mitre.org/data/definitions/311.html; http://cwe.mitre.org/data/definitions/539.html; http://cwe.mitre.org/data/definitions/565.html
T000015	Cross Site Request Forgery (Session Riding)	An attacker crafts malicious web links and distributes them (via web pages, email, etc.) typically in a targeted manner, hoping to induce users to click on the link and execute the malicious action against some third-party application. If successful, the action embedded in the malicious link will be processed and accepted by the targeted application with the users' privilege level. This type of attack leverages the persistence and implicit trust placed in user session cookies by many web applications today. In such an architecture, once the user authenticates to an application and a session cookie is created on the user's system, all following transactions for that session are authenticated using that cookie including potential actions initiated by an attacker and simply "riding" the existing session cookie.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/62.html; http://cwe.mitre.org/data/definitions/306.html; http://cwe.mitre.org/data/definitions/352.html; http://cwe.mitre.org/data/definitions/664.html; http://cwe.mitre.org/data/definitions/716.html; http://cwe.mitre.org/data/definitions/732.html; https://ics-cert.us-cert.gov/advisories/ICSA-14-073-01; https://ics-cert.us-cert.gov/advisories/ICSA-14-079-02
T000016	Simple Script Injection	An attacker embeds malicious scripts in content that will be served to web browsers. The goal of the attack is for the target software, the client-side browser, to execute the script with the users' privilege level. This attack exploits a programs' vulnerabilities that are brought on by allowing remote hosts to execute code and scripts. Web browsers, for example, have some simple security controls in place, but if a remote attacker is allowed to execute scripts (through injecting them in to user-generated content like bulletin boards) then these controls may be bypassed. Further, these attacks are very difficult for an end user to detect.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/63.html
T000017	Escalating privileges through subversion of code-signing	Because languages use code signing facilities to vouch for code's identity and to thus tie code to its assigned privileges within an environment, subverting this mechanism can be instrumental in an attacker escalating privilege.	Cyber	Disrupt	http://capec.mitre.org/data/definitions/68.html; http://cwe.mitre.org/data/definitions/325.html; http://cwe.mitre.org/data/definitions/328.html

T000018	Using Unicode Encoding to Bypass Validation Logic	An attacker may provide a unicode string to a system component that is not unicode aware and use that to circumvent the filter or cause the classifying mechanism to fail to properly understanding the request. That may allow the attacker to slip malicious data past the content filter and/or possibly cause the application to route the request incorrectly. Suspicious: Unicode encoded data is passed to API	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/71.html

T000019	Using slashes, escaped slashes, or UTF-8 encodings to bypass input validation	The adversary uses alternate encodings to bypass input validation capabilities. Slashes can be used to exploit filtering vulnerabilities to gain access to directory resources on a target host. Use of backslash as a leading character can causes a parser to believe that the next character is special, i.e., an escape character. UTF-8 can be used to encode potentially harmful input and submit it to applications, potentially exploiting vulnerabilities common to older UTF-8 encoders.
		Suspicious: IDS alerts of suspicious URLs; suspicious network activity attributable to fuzzing attacks.
			Cyber	Degrade	http://capec.mitre.org/data/definitions/267.html;
					http://capec.mitre.org/data/definitions/78.html;
					http://capec.mitre.org/data/definitions/79.html;
					http://capec.mitre.org/data/definitions/80.html;
					http://cwe.mitre.org/data/definitions/171.html;
					http://cwe.mitre.org/data/definitions/172.html;
					http://cwe.mitre.org/data/definitions/181.html

T000020	Xquery Injection	This attack utilizes XQuery to probe and attack server systems: in a similar manner that SQL Injection allows an attacker to exploit SQL calls to RDBMS, XQuery Injection uses improperly validated data that is passed to XQuery commands to traverse and execute commands that the XQuery routines have access to. XQuery Injection can be used to enumerate elements on the victims environment, inject commands to the local host, or execute queries to remote files and data sources.	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/84.html;
					http://cwe.mitre.org/data/definitions/707.html;
					http://cwe.mitre.org/data/definitions/713.html;
					http://cwe.mitre.org/data/definitions/74.html

T000021	Man in the Middle (MitM) attack	This type of attack targets the communication between two components (typically client and server). The attacker places himself in the communication channel between the two components. Whenever one component attempts to communicate with the other (data flow, authentication challenges, etc.), the data first goes to the attacker, who has the opportunity to observe or alter it, and it is then passed on to the other component as if it was never intercepted. This interposition is transparent leaving the two compromised components unaware of the potential corruption or leakage of their communications. The potential for Man-in-the-Middle attacks yields an implicit lack of trust in communication or identify between two components.	Cyber	Deceive; Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/94.html

T000022	Cryptanalysis	Cryptanalysis is a process of finding weaknesses in cryptographic algorithms and using these weaknesses to decipher the ciphertext without knowing the secret key (instance deduction). Sometimes the weakness is not in the cryptographic algorithm itself, but rather in how it is applied that makes cryptanalysis successful. An attacker may have other goals as well, such as: 1. Total Break - Finding the secret key 2. Global Deduction - Finding a functionally equivalent algorithm for encryption and decryption that does not require knowledge of the secret key. 3. Information Deduction - Gaining some information about plaintexts or ciphertexts that was not previously known 4. Distinguishing Algorithm - The attacker has the ability to distinguish the output of the encryption (ciphertext) from a random permutation of bits The goal of the attacker performing cryptanalysis will depend on the specific needs of the attacker in a given attack context. In most cases, if cryptanalysis is successful at all, an attacker will not be able to go past being able to deduce some information about the plaintext (goal 3). However, that may be sufficient for an attacker, depending on the context	Cyber	Ex-filtrate	http://capec.mitre.org/data/definitions/97.html;
					http://cwe.mitre.org/data/definitions/327.html;
					http://cwe.mitre.org/data/definitions/693.html;
					http://cwe.mitre.org/data/definitions/719.html

T000023	Cross Site Tracing	Cross Site Tracing (XST) enables an attacker to steal the victim's session cookie and possibly other authentication credentials transmitted in the header of the HTTP request when the victim's browser communicates to destination system's web server. The attacker first gets a malicious script to run in the victim's browser that induces the browser to initiate an HTTP TRACE request to the web server. If the destination web server allows HTTP TRACE requests, it will proceed to return a response to the victim's web browser that contains the original HTTP request in its body. The function of HTTP TRACE, as defined by the HTTP specification, is to echo the request that the web server receives from the client back to the client. Since the HTTP header of the original request had the victim's session cookie in it, that session cookie can now be picked off the HTTP TRACE response and sent to the attacker's malicious site. XST becomes relevant when direct access to the session cookie via the "document.cookie" object is disabled with the use of httpOnly attribute which ensures that the cookie can be transmitted in HTTP requests but cannot be accessed in other ways. Using SSL does not protect against XST. If the system with which the victim is interacting is susceptible to XSS, an attacker can exploit that weakness directly to get his or her malicious script to issue an HTTP TRACE	Cyber	Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/107.html
T000024	Malicious Software Update	An attacker uses deceptive methods to cause a user or an automated process to download and install malicious code that is believed to be an valid and authentic software update or patch. There are several variations to this type of attack, all of which rely on the ability of an attacker to position and disguise malicious content such that it masquerades as a legitimate software update.	Supply Chain	Deceive; Implantation	http://capec.mitre.org/data/definitions/185.html; http://capec.mitre.org/data/definitions/186.html
T000026	Accessing Functionality Not Properly Constrained by ACLs	In applications, particularly web applications, access to functionality is mitigated by the authorization framework, whose job it is to map ACLs to elements of the application's functionality; particularly URL's for web apps. In the case that the application deployer failed to specify an ACL for a particular element, an attacker may be able to access it with impunity. An attacker with the ability to access functionality not properly constrained by ACLs can obtain sensitive information and possibly compromise the entire application. Such an attacker can access resources that must be available only to users at a higher privilege level, can access management sections of the application or can run queries for data that he is otherwise not supposed to.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/1.html

T000027	Manipulating Input to File System Calls	An attacker manipulates inputs that the target software passes to file system calls, with the objective of gain access to, and perhaps modify or delete, files and directories that are otherwise not accessible.	Cyber	Destroy; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/76.html
T000028	Manipulating User-Controlled Variables	This attack exploits user controlled variables (DEBUG=1, PHP Globals, and So Forth). An attacker can override environment variables leveraging user-supplied, untrusted query variables directly used on the application server without any data sanitization. In extreme cases, the attacker can change variables controlling the business logic of the application. For instance, in languages like PHP, a number of poorly set default configurations may allow the user to override variables. Suspicious: Abnormal application server behavior; Evidence of probing behavior	Cyber	Degrade; Disrupt	http://capec.mitre.org/data/definitions/77.html
T000029	Session Sidejacking	The attacker uses a network sniffer tool to monitor the wireless traffic at a WiFi hotspot while examining it for evidence of transmittal of session tokens in unencrypted or recognizably encrypted form. An attacker applies his knowledge of the manner by which session tokens are generated and transmitted by various target systems to identify the session tokens. The attacker sniffs on the wireless network to detect unencrypted traffic that contains session tokens.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/102.html
T000030	JSON Hijacking (aka JavaScript Hijacking)	An attacker targets a system that uses JavaScript Object Notation (JSON) as a transport mechanism between the client and the server (common in Web 2.0 systems using AJAX) to steal sensitive information transmitted from the server back to the client inside the JSON object. This exploit takes advantage of the loophole in the browser's Single Origin Policy that does not prohibit JavaScript from one website to be included and executed in the context of another website.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/111.html

T000031	Choosing a Message/Channel Identifier on a Public/Multicast Channel	Attackers aware that more data is being fed into a multicast or public information distribution means can 'select' information bound only for another client, even if the distribution means itself forces users to authenticate in order to connect initially. Doing so allows the attacker to gain access to possibly privileged information, possibly perpetrate other attacks through the distribution means by impersonation. If the channel/message being manipulated is an input rather than output mechanism for the system, (such as a command bus), this style of attack could change its identifier from a less privileged to more so privileged channel or command.	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/12.html
T000032	XPath Injection	An attacker can craft special user-controllable input consisting of XPath expressions to inject the XML database and bypass authentication or glean information that he normally would not be able to. XPath Injection enables an attacker to talk directly to the XML database, thus bypassing the application completely. XPath Injection results from the failure of an application to properly sanitize input used as part of dynamic XPath expressions used to query an XML database. Suspicious: Evidence of application errors resulting from malformed XPath queries	Cyber	Degrade; Ex-filtrate	http://capec.mitre.org/data/definitions/83.html
T000033	Command Delimiters	An attack of this type exploits a programs vulnerabilities that allows an attacker's commands to be concatenated onto a legitimate command with the intent of targeting other resources such as the file system or database. The system that uses a filter or a blacklist input validation, as opposed to whitelist validation is vulnerable to an attacker who predicts delimiters (or combinations of delimiters) not present in the filter or blacklist. As with other injection attacks, the attacker uses the command delimiter payload as an entry point to tunnel through the application and activate additional attacks through SQL queries, shell commands, network scanning, and so on.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/15.html
T000034	OS Command Injection	An adversary injects operating system commands into existing application functions. An application that uses untrusted input to build command strings is vulnerable. An adversary can leverage OS command injection in an application to elevate privileges, execute arbitrary commands and compromise the underlying operating system.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/88.html

T000035	Reflection attack on the authentication protocol	An attacker can abuse an authentication protocol susceptible to reflection attack in order to defeat it. A reflection attack is a method of attacking challenge-response authentication when the same protocol is used in both directions, i.e., each peer uses the protocol to authenticate the other. This attack can provide the attacker illegitimate access to the target system, without possessing the requisite credentials.	Cyber	Degrade	http://capec.mitre.org/data/definitions/90.html; http://en.wikipedia.org/wiki/Reflection_attack
T000036	Log Injection-Tampering-Forging	The attacker injects, deletes, manipulates or forges malicious log entries in the log file, allowing him to mislead a log audit, cover traces of attack, or perform other malicious actions. The target host is not properly controlling log access. As a result tainted data in the log files leads to a failure in accountability, non-repudiation and incident forensics.	Cyber	Deceive; Disrupt	http://capec.mitre.org/data/definitions/93.html
T000037	Accessing, modifying or executing executable files	An attack of this type exploits a system's configuration that allows an attacker to either directly access an executable file, for example through shell access; or in a possible worst case allows an attacker to upload a file and then execute it. Web servers, ftp servers, and message oriented middleware systems which have many integration points are particularly vulnerable, because both the programmers and the administrators must be in synch regarding the interfaces and the correct privileges for each interface. Suspicious: Evidence of unauthorized access; corrupted executable files	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/17.html
T000038	Manipulation of resources loaded by a software application	An attack of this type exploits a system's trust in configuration and resource files. First the attack must modify the resource, e.g., a configuration or image file, to contain malicious code. When the application loads the resource the malicious code is executed. A configuration file can be similarly altered to contain parameters that compromise security functions or otherwise cause the application to function in an unsafe manner, i.e., crashing or corrupting application data.	Cyber	Disrupt	http://capec.mitre.org/data/definitions/35.html

T000039	Exploitation of Session Variables, Resource IDs and other Trusted Credentials	Attacks on session IDs and resource IDs take advantage of the fact that some software accepts user input without verifying its authenticity. For example, a message queuing system that allows service requesters to post messages to its queue through an open channel (such as anonymous FTP), authorization is done through checking group or role membership contained in the posted message. However, there is no proof that the message itself, the information in the message (such group or role membership), or indeed the process that wrote the message to the queue are authentic and authorized to do so.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/21.html
T000040	File System Function Injection, Content Based	An attack of this type exploits the host's trust in executing remote content including binary files. The files are poisoned with a malicious payload (targeting the file systems accessible by the target software) by the attacker and may be passed through standard channels such as via email, and standard web content like PDF and multimedia files. The attacker exploits known vulnerabilities or handling routines in the target processes. Vulnerabilities of this type have been found in a wide variety of commercial applications from Microsoft Office to Adobe Acrobat and Apple Safari web browser. When the attacker knows the standard handling routines and can identify vulnerabilities and entry points they can be exploited by otherwise seemingly normal content. Once the attack is executed, the attacker's program can access relative directories such as C:\Program Files or other standard system directories to launch further attacks. In a worst case scenario, these programs are combined with other propagation logic and work as a virus.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/23.html
T000041	Exploit race conditions and/or deadlock conditions in software	This attack targets a race condition occurring when multiple processes access and manipulate the same resource concurrently and the outcome of the execution depends on the particular order in which the access takes place. The attacker can leverage a race condition by "running the race", modifying the resource and modifying the normal execution flow. For instance a race condition can occur while accessing a file, the attacker can trick the system by replacing the original file with his version and cause the system to read the malicious file.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/26.html

T000042	Leveraging Time of Check and Time of Use (TOCTOU) Race Conditions	This attack targets a race condition occurring between the time of check (state) for a resource and the time of use of a resource. The typical example is the file access. The attacker can leverage a file access race condition by "running the race", meaning that he would modify the resource between the first time the target program accesses the file and the time the target program uses the file. During that period of time, the attacker could do something such as replace the file and cause an escalation of privilege.	Cyber	Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/29.html
T000043	Use of fraudulent PKI certificates	A fraudulent PKI certificate is used by an attacker to masquerade as a trusted website. This can occur when a certification authority signs PKI certificates on behalf of a third party without sufficiently validating its identity.	Cyber	Deceive; Penetration	https://ics-cert.us-cert.gov/advisories/ICSA-12-263-01; https://nvd.nist.gov/vuln/detail/CVE-2012-3037

T000044	[Spear] Phishing	Phishing is a social engineering technique where an attacker masquerades as a legitimate entity with which the victim might do business in order to prompt the user to reveal some confidential information or perform some action that can later be exploited. Phishing attacks will bait users using URLs, emails or other content, that result in implantation of malware, keylogger, etc. Spear Phishing is an enhanced version of the Phishing attack targeted to a specific user or group.	Social Engineering	Deceive; Degrade; Disrupt; Ex-filtrate	http://capec.mitre.org/data/definitions/163.html;
					http://capec.mitre.org/data/definitions/98.html;
					http://www.wired.com/threatlevel/2011/04/oak-ridge-lab-hack/

T000045	Exploiting SNMP Trap Handling	The adversary exploits a vulnerability in SNMP trap handling that causes the SNMP agent to crash, hang, or leads to privilege escalation.	Cyber Physical	Disrupt	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2006-2444;
					http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-4350;
					http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-1402;
					http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-1746;
					http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-2214

T000046	Device DoS using crafted SNMP messages	The adversary manipulates the configuration or triggers a crash, hang, or reload by sending crafted or malformed SNMP messages to a device.	Cyber Physical	Disrupt	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2005-0835; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-5846; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-0680; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-2946; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-4309; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-6976; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-2141; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-2267; https://ics-cert.us-cert.gov/advisories/ICSA-16-327-01
T000047	Exploitation of hidden or undocumented SNMP community strings	The adversary utilizes a hidden or undocumented community string to gain access to a device with SNMP agent enabled.	Cyber Physical	Degrade; Exfiltrate	http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-1999-0254; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2001-0380; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2001-0711; http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2002-1448
T000049	Buffer Overflow	Buffer overflow attacks target improper or missing bounds checking on buffer operations, typically triggered by input injected by an attacker. As a consequence, an attacker is able to write past the boundaries of allocated buffer regions in memory, causing a program crash or potentially redirection of execution as per the attacker's choice	Cyber	Degrade; Destroy; Disrupt	http://capec.mitre.org/data/definitions/100.html ; https://cwe.mitre.org/data/definitions/787.html; https://ics-cert.us-cert.gov/advisories/ICSA-11-243-03A; https://ics-cert.us-cert.gov/advisories/ICSA-11-279-01; https://ics-cert.us-cert.gov/advisories/ICSA-13-050-01A; https://ics-cert.us-cert.gov/advisories/ICSA-15-041-01; https://ics-cert.us-cert.gov/advisories/ICSA-17-187-04; https://ics-cert.us-cert.gov/alerts/ICS-ALERT-11-080-01
T000050	Forced Integer Overflow	This attack forces an integer variable to go out of range. The integer variable is often used as an offset such as size of memory allocation or similarly. The attacker would typically control the value of such variable and try to get it out of range. For instance the integer in question is incremented past the maximum possible value, it may wrap to become a very small, or negative number, therefore providing a very incorrect value which can lead to unexpected behavior. At worst the attacker can execute arbitrary code.	Cyber	Degrade	http://capec.mitre.org/data/definitions/92.html

Countermeasures

CM ID	Countermeasure Name	Description	Effect(s)	Lifecycle	Maturity	Cost	References
C000001	Verify secure BIOS update non-bypassability	Conduct testing on a non-production device to verify that the no other approach besides the secure BIOS update process can be used to overwrite or modify the current BIOS image, including physical device intervention by a user.	Prevent	Fielding; Operation	Medium	Medium	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000002	Verify BIOS image write protection	Conduct testing on a non-production device to verify that the currently installed BIOS image cannot be overwritten or modified except through a secure BIOS update process.	Detect; Prevent	Fielding; Operation	Medium	Low	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000003	Verify recovery process to restore last-known-good BIOS image	Conduct testing on a non-production device to verify that BIOS recovery capabilities cannot be used to install an unapproved BIOS image on the device. Verify through testing that user authentication is required to initiate BIOS image recovery.	Prevent; Respond	Fielding	Medium	Medium	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000005	Institute secure BIOS update capabilities using RTU	An authenticated BIOS update mechanism relies on digital signatures to ensure the authenticity of a BIOS update image. In order to update BIOS using a cryptographically protected update mechanism utilize Root of Trust for Update (RTU) that contains a signature verification algorithm and a key store that includes the public key needed to verify the BIOS update image.	Prevent	Fielding; Implementation; Operation	Medium	Medium	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000007	Perform in-lab testing of BIOS update mechanism	Conduct in-lab testing of BIOS update mechanism to verify that image update does not result in buffer overflow condition.	Detect	Fielding; Methodology	Medium	Medium	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000010	Restrict physical access to device	Utilize physical barriers, locks, etc., to limit physical device access to trusted administrators only.	Prevent; Respond	Fielding; Operation	High	Low	http://resources.infosecinstitute.com/physical-security-managing-intruder/; http://www.intelligentedu.com/computer_security_for_everyone/7-computer-access.html
C000012	Enforce the 2-man rule when performing critical administrative functions	Under 2-man rule all access and actions required the presence of two authorized people at all times.	Prevent	Fielding; Operation	High	High	http://en.wikipedia.org/wiki/Two-man_rule

C000013	Conduct ISVV of safety-critical software	For safety-critical systems, employ independent software verification and validation (ISVV) of software configurations.	Detect; Prevent	Methodology	Medium	Medium	https://en.wikipedia.org/wiki/Independent_software_verification_and_validation
C000015	Verify BIOS implemented security controls after BIOS image update	Conduct testing on a non-production device to verify that BIOS implemented security controls function as expected after a BIOS update is performed. Verify that device's security configuration is not altered after update is performed. Verify that administrative passwords, etc., are not reset to default values after a new BIOS image is installed.	Detect; Prevent	Fielding; Operation	Medium	Low	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000018	Use checksums to verify the integrity of downloaded BIOS image updates	Use checksums to verify the integrity of a downloaded BIOS image before it is distributed as device updates.	Detect; Prevent	Fielding; Operation	Medium	Medium	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000020	Restrict access to the BIOS update server	Enforce user authentication to restrict access to servers used to distribute BIOS updates.	Prevent; Respond	Fielding; Operation	Medium	Low	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf
C000021	Upgrade to SNMP v3	Upgrade to using SNMPv3 protocol, which provides encryption capabilities that are not present in previous versions of the protocol, including capabilities to encrypt community strings that prevent attackers from reading them off the wire.	Prevent	Design; Fielding; Implementation; Operation	Medium	Medium	http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-20/snmpv3.html; https://www.webnms.com/simulator/help/sim_network/netsim_conf_snmpv3.html
C000022	Isolate SNMP traffic to internal network	Configure the network to restrict SNMP management traffic to limited access network segments dedicated for administrative use.	Prevent; Respond	Fielding; Implementation; Operation	Medium	Medium	https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface; https://www.stigviewer.com/stig/network_infrastructure_policy/2016-07-11/finding/V-17772
C000023	Change default SNMP community string values	Replace default 'public' and 'private' SNMP community string values with ones with sufficient complexity to prevent an attacker from easily gaining control of an SNMP managed device.	Prevent	Fielding; Operation	Medium	Low	http://www.tech-faq.com/snmp.html; https://www.giac.org/paper/gcih/44/default-snmp-community-strings-set-public-private/100366
C000025	Enforce strict HTTP and XML parsing	Many web servers, including Apache and IIS, can be configured to perform strict parsing on HTTP requests, cookies, etc.	Detect; Prevent	Fielding; Operation	Medium	Low	http://cwe.mitre.org/data/definitions/444.html

C000027	Terminate client sessions after each request	Terminate the client session after each request in order to prevent possible replay attacks. Note that this can lead to potentially significant performance impacts.	Prevent	Design; Fielding; Implementation; Operation	Medium	High	http://cwe.mitre.org/data/definitions/444.html
C000028	Mark web pages containing sensitive content as non-cacheable	Use the HTTP no-cache pragma to mark web pages containing sensitive content as non-cacheable.	Prevent	Fielding; Implementation	Medium	Low	http://cwe.mitre.org/data/definitions/444.html
C000030	Conduct threat modeling	Conduct threat modeling during system design to identify potential attack vectors and potential impacts resulting from security compromises. Threat Susceptibility Analysis (TSA) is an example methodology that applies threat modeling to systems early in their acquisition lifecycle.	Detect	Methodology	Medium	Medium	https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/cyber-threat-susceptibility-assessment
C000034	Identify, assess, and manage attack surfaces	Identify risk by analyzing changes to a system's baseline attack surface. An attack surface describes all of the different points where an attacker could get into the system, and where they could get data out. Mitigate risk by reducing the attack surface thorough eliminating subsystems, components, functions, capabilities, etc., that are not required to meet mission objectives and requirements.	Prevent	Design; Requirements	Medium	Medium	https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet
C000039	Convert input data into the data format in which it is used	Convert input data into data type(s) appropriate for processing, such as converting a numeric text string into a numeric data type, e.g., signed or unsigned integer value. Verify that the input value is within the expected numeric range.	Prevent	Design; Implementation	Medium	Low	http://cwe.mitre.org/data/definitions/20.html
C000041	Use same character encoding	When exchanging data between components, ensure that both components are using the same character encoding. Ensure that the proper encoding is applied at each interface. Explicitly set the encoding you are using whenever the protocol allows you to do so.	Prevent	Design; Implementation	Medium	Low	http://cwe.mitre.org/data/definitions/20.html
C000045	Utilize high quality session IDs	Utilize session IDs that are long enough to discourage guessing and incorporate random data obtained from a high quality random number generator. Do not encode details about a user into a session ID that can be known or guessed by an adversary.	Prevent; Respond	Design; Fielding; Implementation; Operation	High	Medium	http://capec.mitre.org/data/definitions/59.html
C000047	Encrypt session IDs stored in cookies	Encrypt session IDs before storing them in cookies that may be exposed to a user.	Prevent	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/59.html

C000049	Enforce client authentication	Ensure that client authentication is performed prior to message data transfer to prevent receipt of unauthenticated/anonymous message data.	Prevent; Respond	Fielding; Operation	High	Medium	http://capec.mitre.org/data/definitions/22.html
C000051	Use digital signatures/checksums to authenticate source of changes	Apply a digital signature or Message Authentication Code (MAC) to ensure the integrity on message data. Utilize cryptographic checksums based on PKI certificates whenever both message integrity and non-repudiation is required.	Detect; Prevent; Respond	Design; Fielding; Implementation; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/22.html
C000058	Use a cryptographic token to bind an action to a request	Use cryptographic tokens to associate a request with a specific action. Regenerate a new token for every request. Discard requests containing invalid tokens, i.e., token that arrive with a request for an action other than the action it was supposed to be associated with.	Prevent	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/62.html
C000059	Enable use of the HTTP Referrer header field	The HTTP referrer is the URL of the previous webpage from which a link was followed, and can be used to identify where people are visiting them from for security purposes. This value can be used in conjunction with white list and/or black list validation to enforce access control to a web page.	Detect; Respond	Design; Fielding; Implementation; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/62.html
C000061	Require user confirmation when action involves sensitive data	Prompt the user to confirm an action concerning potentially sensitive data. This way, even if the attacker manages to get the user to click on a malicious link and request the desired action, the user has a chance to recover by denying confirmation.	Respond	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/62.html
C000062	Disable client side scripting	Disable client side scripting in browsers as the last line of defense against scripting attack. Note that the cost of this CM is the functional impact that comes with disabling browser functionality.	Prevent	Fielding; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/63.html
C000064	Do not deploy content proxies that mask where data originates from	XMLHttpRequest (XHR) is used within many Ajax libraries, but till the release of browsers such as Firefox 3.5 and Safari 4 has only been usable within the framework of the same-origin policy for JavaScript. This meant that a web application using XMLHttpRequest could only make HTTP requests to the domain it was loaded from, and not to other domains. If a HTTP proxy for remote content is setup on the server side, the client's browser has no way of discerning where the data is originating from.	Detect; Respond	Fielding; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/63.html

C000065	Sanitize outbound content	Verify that sensitive content is sanitized/removed from all outbound data streams in accordance with security policy. Provide a level of assurance that is commensurate with the sensitivity of the data.	Respond	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/63.html
C000067	Avoid use of modifiable environmental variables in signing and verification process	An crypto implementation that uses environment variables may be compromised through manipulation of those variables. Minimize or avoid use of environmental variables that can be modified by users. Verify that variable changes are recorded to the syslog.	Detect; Prevent	Design; Fielding; Implementation; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/68.html
C000073	Limit filesystem access	Use file system directory and file access permissions to restrict access in accordance with the principle of least privilege.	Prevent; Respond	Fielding; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/78.html
C000074	Perform security checks after decoding	Apply security checks after input data has been decoded and validated as correct data format. If a decoding or validation error occurs, fail the validation process.	Detect	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/78.html
C000075	Verify file contents before making file processing decisions	Verify the content of all files to be processed, and avoid making file processing choices based solely on the file's name or extension.	Prevent; Respond	Design; Implementation	Medium	Medium	http://capec.mitre.org/data/definitions/78.html
C000077	Perform XML parsing with minimal privileges	Run XML parsing and query infrastructure with minimal privileges so that an attacker is limited in their ability to probe other system resources from XQL	Prevent; Respond	Operation	Medium	Low	http://capec.mitre.org/data/definitions/84.html
C000079	Only accept PKI credentials from a trusted certificate authority	Ensure PKI credientials are signed by a certificate authority that is trusted either directly on indirectly through certificate chaining. Prohibit use of self-signed certificates. Utilize certificate chaining to validate signatures of certificates not directly issued by the certificate authority.	Prevent; Respond	Fielding; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/94.html
C000081	Use strong mutual authentication	Use strong mutual authentication based on PKI certificates to verify the identities of both ends of a communications channel	Prevent; Respond	Fielding; Operation	Medium	Medium	http://capec.mitre.org/data/definitions/94.html
C000083	Use cryptography that is sufficient strong	Use proven/certified FIPS certified cryptographic algorithms and implementations. Choose initialization vectors with sufficiently random numbers. Generate key material using good sources of randomness and avoiding known weak keys. Use proven protocols and their implementations. Pick the most appropriate cryptographic algorithm for your usage context and data	Prevent; Respond	Design; Fielding; Implementation; Operation	High	Medium	http://capec.mitre.org/data/definitions/97.html

C000084	Disable HTTP TRACE support	Disabling HTTP TRACE in web servers eliminates a vulnerability exploited by Cross Site Tracing.	Prevent	Fielding; Operation	Medium	Low	http://capec.mitre.org/data/definitions/107.html ; https://access.redhat.com/solutions/198813
C000086	Treat each API as an attack surface feature	Treat each API as a potential attack vector that an adversary could use to probe or penetrate the system. Design the API to restrict privileged access; verify through testing that the API cannot be used to escalate privileges, exploit race conditions or exhaust system resources. Do not expose APIs that cannot be so evaluated.	Prevent	Design; Fielding; Implementation; Methodology; Operation	Medium	Low	http://cwe.mitre.org/data/definitions/648.html; https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1.pdf
C000087	Accept hyperlinks/attachments from trusted sources only	Enforce user policies that restrict or prohibit download of files or clicking of hyperlinks from unknown or untrusted sources.	Prevent; Respond	Operation	Medium	Low	http://capec.mitre.org/data/definitions/185.html; http://capec.mitre.org/data/definitions/186.html
C000089	Validate the range of numeric input	Perform input validation on numeric input by checking that it is within the expected range. Verify that user input meets both the minimum and maximum requirements for the expected range.	Detect; Prevent; Respond	Design; Implementation	Medium	Low	http://www.regular-expressions.info/numericranges.html
C000090	Validate input fields use of NULL, escape, backslash, meta, and control characters	Verify that input validation correctly handles NULL, escape, meta and/or control characters or sequences anywhere within the input data stream.	Detect; Prevent	Design; Implementation	Medium	Low	http://cwe.mitre.org/data/definitions/113.html; http://cwe.mitre.org/data/definitions/129.html; http://cwe.mitre.org/data/definitions/138.html
C000091	Apply blacklist and whitelist validation in combination	Use a combination of blacklist and whitelist input validation to ensure that only valid, expected and appropriate input is accepted by the system	Detect; Prevent; Respond	Design; Implementation	Medium	Medium	http://cwe.mitre.org/data/definitions/74.html; http://cwe.mitre.org/data/definitions/75.html; http://cwe.mitre.org/data/definitions/76.html; http://cwe.mitre.org/data/definitions/97.html
C000092	Apply parser-based validation for structured data	Validate XML data against XML Schema or DTD declarations	Detect; Prevent	Design; Implementation	Medium	Medium	http://cwe.mitre.org/data/definitions/112.html
C000093	Merge data streams prior to validation	When data from multiple sources is combined, perform the validation step after the sources have been combined to ensure that individual data elements that pass individual validation step do not violate intended restrictions after they have been combined.	Prevent	Design; Implementation	Medium	Medium	http://cwe.mitre.org/data/definitions/20.html; http://cwe.mitre.org/data/definitions/88.html
C000094	Validate data exchanges across language boundaries	Validate all data produced when code is invoked that crosses language boundaries, such as from an interpreted language to native code, e.g., Java Native Interface (JNI) invocation of native C or C++ code.	Detect; Prevent; Respond	Design; Implementation	Medium	Medium	http://cwe.mitre.org/data/definitions/129.html; http://cwe.mitre.org/data/definitions/20.html

Countermeasures	Threat Vectors																												
CM ID	T000001	T000002	T000003	T000004	T000005	T000006	T000007	T000008	T000009	T000010	T000011	T000012	T000013	T000014	T000015	T000016	T000017	T000018	T000019	T000020	T000021	T000022	T000023	T000024	T000026	T000027	T000028	T000029	T000030
C000001	PM;	PM;	PM;	PM;	PM;	PM;																							
C000002	PM;	PM;	PM;	DM; PM;	PM;	PM;																							
C000003	PL; RM;	PL; RM;																											
C000005	PM;																												
C000007		DM;																											
C000010	PM;		PM;		PM;																								
C000012			PM;																										
C000013			DM; PM;																										
C000015	DM;	DM;	DM;	DM; PM;	DM;																								
C000018	DM;																												
C000020			PM;																										
C000021							PM;																						
C000022							PM; RM;																						
C000023								PM;																					
C000025										PM;									DM;	PM;									
C000027										PM;																			
C000028										PM;																			
C000030																													
C000034																													
C000039												PM;						PM;											
C000041																PL;		PM;											
C000045									PM;					RM;															PM;
C000047									PM;					PM;	PM;													PM;	
C000049									RM;				PM;								RM;								
C000051													DM; PM;	DM; PM;										DM; RM;					
C000058															PM;														
C000059														RM;	DM;														
C000061															RM;														
C000062																PM;							PM;						
C000064																DL; RL;													
C000065																RL;													
C000067																	DL; PM;												
C000073																													
C000074																			DM;										
C000075																			PM;										
C000077																				PL; RM;									
C000079																					PM;								
C000081									RM;				PM;								PM;								
C000083											PM; RM;				PM;		PM;				PM; RM;	PM;							
C000084																							PM;						
C000086																							PM;						
C000087															RM;									PM;					
C000089																													
C000090												DM; PM;	DM; PM;			DM; PM;			DM; PM;	DL; PM;						DM; PM;			
C000091																		PM; RM;	PM;	PM;							PM; RM;		
C000092																													
C000093												PM;						PM;											
C000094												DL; PL; RM;						RM;											
C000095												PM;						PM;											
C000096											PM;																		
C000097																													
C000100												PM;																	
C000101																													
C000102																													
C000103																													
C000104												DM;																	
C000105																													
C000106												DM; PM;												DM; PM;					

References

[1] https://www.iaea.org/projects/crp/j02008

[2] http://capec.mitre.org/

[3] https://attack.mitre.org/wiki/Main_Page

[4] http://cve.mitre.org/

[5] http://cwe.mitre.org/

[6] https://ics-cert.us-cert.gov/

[7] https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf

[8] https://www.mitre.org/sites/default/files/publications/pr-2359-threat-assessment-and-remediation-analysis.pdf