L.U.I.S.S. Guido Carli
Relatore: Paolo Spagnoletti
Candidato: Samuele Sambuca
23 dicembre 2017

Indice

Introduzione

Nella seconda metà del XVIII secolo, con l'introduzione della macchina a vapore che meccanizzava per la prima volta la produzione industriale avveniva quel fenomeno che oggi identifichiamo come prima rivoluzione industriale; nella seconda metà del XIX secolo con il sempre più diffuso utilizzo dell'elettricità, con l'introduzione del motore a scoppio e con l'utilizzo del petrolio come fonte energetica, avveniva la seconda rivoluzione industriale; nella seconda metà del XX secolo, infine, con la nascita dell'informatica, avveniva la terza rivoluzione industriale.

In questa tesi, abbiamo non solo esaminato il fenomeno della quarta rivoluzione industriale e l'avvento dell'industria 4.0, ma ci siamo focalizzati sui rischi relativi alla cyber-sicurezza: la connessione ad un sistema informatico mette a rischio, infatti, non solo l'azienda stessa, ma anche potenzialmente i suoi consumatori.

Vogliamo quindi proporre, con questa tesi, un approccio di analisi sia economico, che informatico.
Abbiamo descritto nel primo capitolo l'industria 4.0 definendola ed identificandola nelle tecnologie abilitanti per poi trattare alcune strategie che le aziende possono applicare per consolidare l'impresa nella nuova era; abbiamo in seguito esaminato le manovre del governo Italiano per modernizzare il tessuto produttivo ed infine abbiamo dimostrato i benefici ed i rischi che essa comporta.
Rimanendo sul tema dei rischi abbiamo poi esaminato nel secondo capitolo quelli relativi alla cyber-sicurezza, definendo il concetto e le caratteristiche di una cyber-arma, l'impatto che può avere un'arma digitale su un ambiente fisico e le fasi di progettazione, sviluppo ed implementazione delle stesse.
Quanto prima ci servirà ad analizzare al meglio il caso proposto al terzo capitolo dove analizzeremo il malware Stuxnet, un worm usato da Stati Uniti ed Israele per sabotare il programma nucleare Iraniano.

1. L’innovazione tecnologica nell’industria 4.0

In questo capitolo verrà trattato il ruolo della tecnologia nell’impresa; un ruolo così determinante che si parla di industria 4.0 e di rivoluzione industriale. Le aziende stanno effettivamente cambiando diventando sempre più interconnesse e digitalizzate e anche in Italia, il Governo ha varato un piano speciale per sviluppare questo tipo di impresa.
I riferimenti per questo capitolo sono le fonti da [1] a [9].

1.1 La quarta rivoluzione industriale

La quarta rivoluzione industriale è il processo che porterà alla produzione industriale del tutto automatizzata ed interconnessa. Il motore di questo processo sono le tecnologie digitali che, rese più accessibili e più efficienti, avranno impatto su quattro direttrici di sviluppo: la prima riguarda l’utilizzo dei dati, la potenza di calcolo e la connettività con l’avvento dei big data, dell’Internet of Things, sistemi machine-to-machine e cloud computing per la centralizzazione e conservazione delle informazioni.
La seconda direttrice di sviluppo è quella delle tecnologie di analisi dei dati: infatti solo una minima parte dei dati raccolti viene effettivamente analizzata, ma grazie al diffondersi e al perfezionamento delle tecnologie basate sul machine learning le imprese potrebbero ottenere diversi vantaggi; infatti grazie al machine learning le macchine possono perfezionare la propria resa “imparando” dai dati raccolti e analizzati real-time.
La terza direttrice di sviluppo è l’interazione tra uomo e macchina, che coinvolge le interfacce “touch” divenute popolari negli ultimi anni, e la realtà aumentata divenuta molto più accessibile.
Infine c’è tutto il settore che si occupa del passaggio dal digitale al “reale”: la manifattura additiva, cioè la stampa in 3D, la robotica, le comunicazioni, le interazioni machine-to-machine e le nuove tecnologie per immagazzinare e utilizzare l’energia in modo mirato, razionalizzando i costi e ottimizzando le prestazioni.
L’espressione “Industria 4.0” viene usata per la prima volta alla fiera di Hannover in Germania nel 2011. Un anno più tardi viene creato un gruppo di lavoro dedicato a questo tema per presentare al governo federale tedesco una serie di raccomandazioni sulla sua implementazione, questo report verrà poi diffuso alla fiera di Hannover nel 2013.
Nel report si legge che l’introduzione dell’Internet of Things and Services nel settore manifatturiero sta dando inizio alla quarta rivoluzione industriale; le imprese del futuro, infatti, stabiliranno complessi network globali, che incorporeranno macchinari, sistemi di

immagazzinamento e strutture di produzione, sotto la forma di Sistemi Cyber-Fisici (CPS dall'inglese Cyber-Physical Systems) capaci di scambiarsi informazioni, intraprendere azioni e di controllarsi reciprocamente.
Prima di procedere chiarifichiamo il concetto di CPS: per Sistema Cyber-Fisico si intende un sistema di elementi computazionali collaborativi che controllano entità fisiche; si tratta quindi di un sistema informatico che interagisce in modo continuo con un sistema composto da elementi fisici ciascuno dotato di propria capacità computazionale (capacità di elaborare dati) gestendo la comunicazione e il controllo tra le parti riuscendo quindi a formare un sistema distribuito che interagisce dinamicamente con il mondo reale che lo circonda.
Queste "smart factories" impiegano un approccio alla produzione completamente nuovo reso possibile grazie all'impatto delle tecnologie digitali.
Fonti: [1], [4], [5]

1.2 Le tecnologie abilitanti

Così come è stato il motore a vapore per la prima rivoluzione industriale, il petrolio, i prodotti chimici e l'elettricità per la seconda e l'elettronica e l'IT per la terza, anche la quarta rivoluzione industriale ha le sue cosiddette "tecnologie abilitanti". Queste ultime sono tecnologie in grado di "rivitalizzare" il tessuto produttivo portando innovazione e diversi vantaggi specifici a seconda del caso.
Il Boston Consulting Group ne ha individuate ben nove definendoli i "nove pilastri dell'avanzamento tecnologico"; molte di queste tecnologie sono già presenti da diversi anni, ad esempio si dispone della stampa 3D già dagli anni 80', ma non sono mai state tanto accessibili quanto oggi anche grazie alla aumentata potenza di calcolo dei dispositivi elettronici e all'abbattimento dei prezzi.

Il primo pilastro rappresenta i Big Data e i software di analisi e di auto-apprendimento; come già accennato prima, infatti, l'analisi dei Big Data ottimizza la qualità della produzione, risparmia energia e migliora il lavoro complessivo delle macchine. Nel contesto dell'Industria 4.0 la raccolta e la valutazione dei dati provenienti da vari sorgenti diventerà presto lo standard per il supporto alle decisioni real-time.

Ad avere grande importanza saranno poi i Robot Autonomi; le imprese manifatturiere hanno già da molti anni usato i robot per risolvere problemi complessi, ma nell'industria 4.0 essi avranno un'utilità ancora maggiore.
I nuovi robot saranno più autonomi, flessibili e cooperativi: presto potranno interagire l'uno con l'altro e addirittura lavorare fianco a fianco con gli esseri umani, imparando da loro.

Le tecnologie di Simulazione sono il terzo pilastro che nella fase di progettazione offre senza dubbio un importante contributo; simulazioni 3D di prodotti, materiali e processi produttivi sono già usati, ma sarà innovativo il fatto che useranno dati real-time per riprodurre fedelmente il mondo reale in un modello virtuale e questo permetterà agli operatori di testare le impostazioni delle macchine per la produzione rilevando eventuali problemi futuri o inefficienze e di ottimizzarle per l'uso preposto. Questo porterà ad un aumento della qualità dei prodotti e del servizio offerto dalle macchine.

Il quarto passo fondamentale sarà l'integrazione sia verticale che orizzontale dei sistemi informatici, infatti, oggi molti non sono completamente integrati: imprese, fornitori e clienti raramente sono collegati tra loro e nella stessa impresa difficilmente lo sono i reparti di ingegneria, produzione e servizio al cliente. Con l'industria 4.0 la condivisione dei dati tra dipartimenti diversi e, addirittura, tra diverse imprese riuscirà a creare una catena del valore automatizzata.

L'Industrial Internet of Things è la quinta importante innovazione tecnologica, oggi sono pochi sensori e macchine sono collegati tra loro e fanno uso di sistemi integrati di computazione; generalmente questi sistemi sono collegati tra loro seguendo uno schema statico e piramidale di produzione. Con l'Industrial Internet of Things più dispositivi faranno uso di sistemi integrati di computazione collegati tra loro attraverso tecnologie standard di comunicazione. Questo permetterà una interazione machine-to-machine più articolata ed efficiente, inoltre decentralizzerà i processi di analisi e di decision making, permettendo risposte real-time.

La sesta tecnologia abilitante è la manifattura additiva, che comprende le tecniche di stampa in 3D; usate oggi maggiormente per la realizzazione di prototipi e componenti individuali, con l'industria 4.0 esse saranno ampiamente usate per la produzione rapida di prodotti personalizzati.

Settimi protagonisti della rivoluzione industriale sono i sistemi basati sulla realtà aumentata che offrono una grande varietà di servizi come ad esempio: la selezione di merci dal magazzino, l'invio di istruzioni di riparazione tramite smartphone, ma anche il training dei dipendenti attraverso l'uso di occhiali a realtà aumentata; inoltre le imprese potranno usare la realtà aumentata per fornire ai dipendenti informazioni real-time e migliorare il processo decisivo e le procedure lavorative.

Il cloud è l'ottava tecnologia abilitante, infatti, con lo svilupparsi dell'Industria 4.0 ci sarà sempre più richiesta di condivisione dati attraverso diversi siti e compagnie e allo stesso

tempo ci sarà un miglioramento delle tecnologie cloud in modo da offrire soluzioni di upload e download sempre più veloci.
Il risultato sarà una maggiore presenza di servizi per la produzione data-driven.

La nona ed ultima tecnologia abilitante è la cybersicurezza di cui ci occuperemo maggiormente dal capitolo 2.
Molte compagnie fanno ancora uso di sistemi di produzione non connessi tra loro via network, ma con la crescente connettività dei macchinari il passaggio agli strumenti di Industria 4.0 sarà inevitabile e il bisogno di proteggere i sistemi industriali critici crescerà drasticamente.

La nuova rivoluzione industriale sarà guidata da innovazioni in questi 9 importanti settori e le imprese dovranno adottare nuove strategie e, come è sempre stato per quelle di successo, adattarsi al mondo che cambia.
Fonte: [5]

1.3 Consigli strategici per le imprese

L'industria 4.0 porterà allo sviluppo di nuovi modelli di business e partnership orientati a soddisfare rapidamente i bisogni di clienti individuali.
Per riuscire a beneficiare dall'imminente digitalizzazione le imprese manifatturiere dovrebbero agire su tre ambiti: adottare nuove tecnologie di produzione, adattare i modelli di business e realizzare la trasformazione digitale.

Il primo ambito permetterà alle imprese di rispondere in modo più efficace ad una domanda in continua evoluzione anche attraverso impianti industriali di nuova generazione.
Questi ultimi possono essere divisi in tre categorie: gli impianti completamente automatizzati e digitalizzati per la produzione di massa, gli impianti per la produzione personalizzata su vasta scala per rispondere agilmente alle esigenze del mercato e gli impianti di tipo "e-plant in a box" adatti alla produzione di nicchia e facilmente dislocabili a seconda della domanda.

Il secondo ambito suggerisce alle aziende di adattarsi al cambiamento dello scenario competitivo in modo da trarre il maggior vantaggio possibile dalla modifica che l'impatto del digitale avrà sulla catena del valore.
Le imprese dovrebbero valorizzare gli asset esistenti, fare leva sui fattori del proprio vantaggio competitivo e garantire la flessibilità dell'intera organizzazione.
Lo sviluppo verso Industria 4.0 dipende dall'utilizzo sempre più massivo delle tecnologie digitali e delle tecnologie abilitanti richiamate in precedenza, di conseguenza si può immaginare un miglioramento in termini di efficienza e la nascita di nuove strategie di

mercato che avvicinino il bene prodotto al consumatore rendendo possibile anche un modello di business in cui il produttore anziché vendere il bene lo affitta al consumatore e ne cura la manutenzione.
Si parla a questo proposito del cosiddetto “modello Xerox”: la proprietà della fotocopiatrice non è dell’utente finale che paga invece un canone di locazione con una componente fissa e una variabile legata al numero di copie fatte; il proprietario della fotocopiatrice si fa carico della manutenzione del bene e della sostituzione delle parti usurate.
Tutto ciò potrà influire anche sulle modalità di produzione e progettazione dei beni medesimi, alcuni dei quali dovranno essere sostituiti più frequentemente altri invece saranno soggetti ad usura in modo limitato con evidenti implicazioni positive per l’ambiente.

Un altro modello di business che è possibile adottare è quello che applica la cosiddetta circular economy in cui i materiali e l’energia utilizzati mantengono il loro valore il più a lungo possibile, i rifiuti sono ridotti al minimo e si utilizza il minimo possibile di risorse, questo risponde alla duplice esigenza di ridurre l’impatto ambientale delle attività economiche e, sul piano economico, di conseguire risparmi evitando sprechi e riducendo i costi di approvvigionamento delle materie prime. La Dell ad esempio, in vista del crescente impatto della cosiddetta e-waste (spazzatura elettronica), causata dalla rapida innovazione tecnologica che riduce la vita dei prodotti, ha annunciato l’applicazione di questo tipo di strategia.
Non solo useranno plastica riciclata e fibre di carbonio recuperate, ma, parte del programma “2020 Legacy of Good Plan”, ha l’obiettivo di tagliare l’e-waste, usando 1.400 tonnellate di materiali riciclati e recuperando 57.000 tonnellate di materiali per il 2020.[18]

Nel quadro dei nuovi modelli di business si colloca anche la sharing economy che favorisce pratiche basate sul riuso invece che sull'acquisto e sull’accesso, piuttosto che sulla proprietà, in forma sincrona o differita. Due esempi sono Uber e Airbnb, che hanno sviluppato le proprie piattaforme per permettere ai propri utenti di fornire un servizio o accedervi.

La maker economy è un altro nuovo modello di business connesso all’Industria 4.0. Si tratta di forme di autoproduzione artigiana che però sfruttano ed integrano tecnologie ed idee innovative.
Esempi della diffusione della maker economy possono essere la presenza di sempre più attività che offrono un servizio di stampa 3D, in cui il cliente può portare il proprio progetto e farlo stampare, o ancora, la vendita dei progetti di stampa 3D.

Il terzo ed ultimo ambito riguarda la capacità delle imprese di realizzare la trasformazione digitale, la strategia vincente è integrare i sistemi e favorire la condivisione di dati tra

dipartimenti e persone. Una delle principali barriere tecniche risiede nel fatto che molti sistemi tradizionali aziendali non comunicano tra loro ed è molto difficile estrarne le informazioni. Infine è necessario dotare il personale con gli strumenti necessari per collaborare e condividere conoscenze e iniziative.

Raggiungere gli obiettivi per l'implementazione dell'Industria 4.0 non è facile per le aziende e per questo motivo, determinante è l'aiuto da parte dei governi che si stanno mobilitando con piani per modernizzare il tessuto produttivo.
Fonti: [3], [7]

1.4 L'approccio del governo Italiano

Il piano del governo per l'Industria 4.0 è stato approvato da entrambe le camere del parlamento ed è in vigore già dall'inizio del 2017. La manovra punta a mobilitare investimenti privati aggiuntivi per 10 miliardi, a raggiungere gli 11 miliardi in più di spesa privata in ricerca, sviluppo ed innovazione e ad incrementare di 2,6 miliardi gli investimenti privati early stage.
Questo sarà possibile grazie ad un mix di strategie che comprendono incentivi fiscali, sostegno al venture capital, diffusione della banda ultra larga e formazione dalle scuole alle università sui temi dell'Industria 4.0.
Il piano prevede un superamento del superammortamento al 140% con un iperammortamento al 250% per i beni legati all'industria 4.0 che avrà tempi più lunghi; è, inoltre, previsto un rifinanziamento del Fondo Centrale di Garanzia.
L'attuazione del piano prevede di operare lungo cinque direttrici:
- Attrarre gli investimenti innovativi di cui sopra;
- Sviluppare le competenze necessarie alle imprese per adottare e realizzare il cambiamento. Il piano prevede la diffusione della cultura 4.0 attraverso la scuola digitale, l'Alternanza Scuola Lavoro, le Università e gli istituti Tecnici Superiori dedicati, ma anche finanziando la ricerca;
- Accrescere la sensibilità al tema e creare una governance pubblico-privata che assuma il ruolo di guida verso la digitalizzazione;
- Creare le strutture per facilitare la digitalizzazione delle imprese: questo comprende sia la creazione di infrastrutture di rete che consentano l'accesso alla banda Ultra Larga, sia la collaborazione alla definizione di standard e criteri per l'interoperabilità dell'Internet of Things;
- Fornire strumenti pubblici di supporto con le tecniche di cui sopra.

Il piano Industria 4.0 ha fatto parte sia della legge di bilancio 2017, anno in cui è stato introdotto, sia dell'ultima legge di bilancio e contribuisce allo sviluppo del paese, ma anche all'acquisizione delle competenze necessarie per rendere funzionante l'impresa 4.0.
Vedremo nel prossimo paragrafo i vantaggi della rivoluzione industriale e i nuovi rischi a cui espone le aziende.
Fonte: [2]

1.5 Trade-off tra benefici e rischi attesi

Vediamo ora concretamente quali sono i benefici della rivoluzione che sta modificando l'ecosistema produttivo in questo momento:

- Maggiore flessibilità attraverso la produzione di piccoli lotti ai costi della grande scala;
- Maggiore velocità dal prototipo alla produzione in serie attraverso tecnologie innovative;
- Maggiore produttività attraverso minori tempi di set-up, riduzione errori e fermi macchina;
- Migliore qualità e minori scarti mediante sensori che monitorano la produzione in tempo reale;
- Maggiore competitività del prodotto grazie a maggiori funzionalità derivanti dall'Internet delle cose;

Tramite il collegamento di tutti gli asset coinvolti nella filiera logistico-produttiva, il vantaggio primario della nuova industria è sicuramente la disponibilità di tutte le informazioni real-time.
È chiaro che questo collegamento mette a rischio l'azienda dal punto di vista informatico, un rischio a cui prima era esposta solo limitatamente.
In questo contesto il ruolo dei CIO (Chief Informatic Officer) è strategicamente molto importante quando si tratta di adottare innovazioni tecnologiche e il numero delle aziende che decidono di affidarsi a queste figure manageriali è in crescita; inoltre secondo uno studio condotto dal MIT[10] la percentuale di tempo che i CIO investono in "Innovazione" è correlato fortemente e positivamente con la performance generale dell'impresa.
Secondo lo studio i CIO delle aziende che compongono l'ultimo 25% in termini di margine di profitto, hanno speso il 19% del proprio tempo nell'implementazione di tecnologie innovative, mentre quelli delle aziende che compongono il primo 25% ne hanno speso in media il 53%; inoltre anche la differenza tra i due valori è un dato statistico significativo e dimostra l'importanza che il comportamento dei CIO ha sulla performance aziendale.
Se da un lato restare al passo con l'avanzamento tecnologico è un fattore rilevante, dall'altro i rischi che questo comporta costringono i CIO incrementare le spese per la cybersicurezza; ciò è dimostrato dalla presenza in sempre più aziende della figura del CISO (Chief Information Security Officer), figura responsabile della cybersicurezza che affianca il CIO.

Si percepisce l'importanza di tutelarsi dal punto di vista informatico, dal fatto che le perdite relative alla cybersicurezza impattano sulla performance negativamente quasi quanto l'innovazione lo fa positivamente.
Per chiarire meglio questo concetto cerchiamo di quantificare il danno dividendolo in due semplici categorie: il danno diretto ed il danno indiretto.

La prima categoria è molto più facile da quantificare ed è dato dal numero di violazioni conseguite con successo dagli hacker.
Secondo un report di investigazione di Verizon[10] sulla cybersicurezza le 70 aziende intervistate hanno subito 79.790 incidenti tra cui 2.122 violazioni di dati confermate; un altro report sponsorizzato da IBM security ed eseguito dal Ponemon Institute[11] ha intervistato 419 compagnie in 13 paesi diversi e tutte quante hanno subito violazioni che hanno compromesso da un minimo approssimato di 2.600 ad un massimo di quasi 100.000 record.
Prima di andare avanti chiarifico che per record compromesso si intende l'identificativo della persona i cui dati sono andati persi o rubati.
Secondo lo stesso studio, il costo medio del furto di un solo record è di $141 ed è, invece, di $3,6 milioni il costo medio di una violazione dati: questi costi sono stati calcolati tenendo conto del numero dei record rubati e della percentuale di clienti persi per questa ragione.

La seconda categoria, quella dei danni indiretti, è più difficile da quantificare ed è causata da: risorse allocate in modo inefficiente, cautela aumentata e non necessaria nell'adozione di innovazioni tecnologiche e da inefficienze causate dagli aumentati, seppur necessari, controlli di cybersicurezza.

Il ragionamento ci porta a considerare la cybersicurezza una delle sfide delle industrie del futuro: è fondamentale che queste ultime non sottovalutino questo aspetto, ragion per cui nel prossimo capitolo cercheremo di acquisire le conoscenze informatiche necessarie a comprendere la minaccia.

Fonti: [6], [8]

2. La minaccia cyber

In questo secondo capitolo acquisiremo alcuni elementi che ci aiuteranno ad attaccare il problema del terzo capitolo: per prima cosa scopriremo che cos'è e come si identifica una vulnerabilità informatica, per poi focalizzarci sui vari tipi di malware e in modo particolare sui worm (poiché Stuxnet è un malware di questo tipo). Cercheremo, poi, di dare una definizione alla locuzione "cyber-arma" ed infine sarà illustrato il processo di sviluppo della stessa.

I riferimenti per questo capitolo sono le fonti da [10] a [14].

2.1 Comprendere le vulnerabilità: il Threat Model

Un sistema si dice sicuro quando garantisce: confidenzialità, disponibilità ed integrità.

Questa definizione introduce tre concetti chiave che sono alla base della sicurezza informatica:

- Confidenzialità: Questo termine copre due concetti correlati:
  - Data confidentiality: Assicura che informazioni private o confidenziali non siano disponibili o divulgate ad individui non autorizzati.
  - Privacy: Assicura che gli individui controllino o influenzino quali delle loro informazioni vengano raccolte, conservate e/o divulgate.
- Integrità: Questo termine copre due concetti correlati:
  -Data integrity: Assicura che le informazioni e i programmi vengano modificati solo in un modo specifico ed autorizzato.
  -System integrity: Assicura che un sistema esegua la sua esatta funzione in modo perfetto, senza subire deliberate o improvvise manipolazioni.
- Disponibilità: Assicura che il sistema funzioni prontamente e che il servizio non blocchi gli utenti autorizzati.

Questi tre concetti formano quella che spesso è chiamata triade CIA (dall'inglese).
I tre concetti incorporano gli obiettivi fondamentati della sicurezza informatica sia per i dati che per i servizi di rete.
Il FIPS 199 (un documento che fornisce standard per la categorizzazione di informazioni e dei sistemi che le gestiscono) fornisce un'utile caratterizzazione di questi tre obiettivi in termini di requisiti per ciascuna categoria:

- Confidenzialità: preservare restrizioni di autorizzazione riguardo l'accesso e la divulgazione di informazioni anche allo scopo di preservare la privacy e la proprietà delle informazioni. Una mancanza in confidenzialità è la divulgazione non autorizzata di informazioni.
- Integrità: Proteggere contro modiche o distruzioni improprie di informazioni; ciò include assicurarsi la non ripudiabilità e l'autenticità delle informazioni. Una mancanza in integrità è la non autorizzata modifica o distruzione di informazioni.
- Disponibilità: Assicurare un accesso affidabile e duraturo alle informazioni e il loro utilizzo. Una mancanza di disponibilità è il discontinuo o impossibile accesso alle informazioni o al loro utilizzo.

Inoltre, quando si tratta di sicurezza (informatica e non) è bene ragionare tenendo conto che un attaccante utilizzerà qualsiasi mezzo in suo possesso per raggiungere lo scopo che si è preposto.

Per meglio identificare le vulnerabilità informatiche facciamo uso del cosiddetto "Threat Model" che ci consente di stabilire tre condizioni ritenute necessarie e sufficienti perché, prima o poi, avvenga un attacco con successo.

Le tre condizioni sono:

1) Un difetto o una debolezza del sistema (in questo senso un difetto è un errore di programmazione, mentre una debolezza è un errore logico intrinseco nella complessità del sistema che produce effetti indesiderati);

2) L'accessibilità al difetto o debolezza di cui al punto uno;

3) La capacità della minaccia di sfruttare il difetto o la debolezza;

Analizziamo più nel particolare le tre condizioni:

La prima condizione è detta "Suscettibilità del sistema" ed è caratterizzata dal fatto che confidenzialità e disponibilità non possono essere simultaneamente completamente raggiunte. Questo perché le due caratteristiche sono antitetiche e l'architetto di sistema dovrà necessariamente stabilire un trade-off che soddisfi le esigenze del cliente.
Nonostante ciò è raro che un sistema non contenga debolezze di progettazione o implementazione e un attaccante le cercherà per sfruttarle e compromettere elementi critici o funzioni del sistema.

La seconda condizione è detta "Accessibilità della minaccia" ed è verificata se la minaccia è in grado di raggiungere le suscettibilità del sistema e conseguentemente sfruttarle per

raggiungere il suo scopo. Generalmente, la minaccia userà i normali accessi previsti per gli utenti standard o i servizi previsti per gli utenti legittimi.

La terza condizione è detta “Capacità della minaccia” ed è verificata se l’attaccante ha le conoscenze e i mezzi per portare a termine l’attacco con successo; infatti, se le prime due condizioni sono verificate, l’attaccante può ottenere un’adeguata conoscenza della vulnerabilità. Gli hacker migliori spendono gran parte del loro tempo ad “allargare la superficie di attacco” e questo implica anche l’utilizzo di metodi di “reverse engineering” in modo da poter osservare i comportamenti del sistema che si vuole attaccare.
Non è da sottovalutare questo aspetto poiché maggiore è il livello di conoscenza del target, migliori saranno gli strumenti sviluppati per attaccarlo: come vedremo nel terzo capitolo, il team che ha sviluppato Stuxnet aveva una profonda conoscenza del sistema operativo Windows e degli ICS (Industrial Control Systems) prodotti dalla Siemens.

Stabiliamo quindi che solo in caso di coesistenza di queste tre condizioni esiste una vulnerabilità.
In particolare quelle che vengono sfruttate, ma prima di allora non erano pubblicamente conosciute vengono chiamate “vulnerabilità 0-day”, chiamate così proprio perché lo sviluppatore ha 0 giorni di tempo per riparare la falla.

Fonti: [9], [12]

2.2 Gli attacchi informatici

Per soddisfare efficacemente il bisogno di sicurezza di un’organizzazione e per valutare e scegliere i prodotti e le politiche di sicurezza il manager responsabile ha bisogno di un metodo sistematico per definire i requisiti target e caratterizzare gli approcci per raggiungerli. La OSI Security Architecture fornisce un utile, prospettiva di molti dei concetti fondamentali della sicurezza informatica, in particolare si focalizza sui concetti di attacco e meccanismo e servizio di sicurezza:

Un attacco alla sicurezza è ogni azione che compromette la sicurezza delle informazioni di un’organizzazione;
Un meccanismo di sicurezza è un processo (o un dispositivo che incorpora tale processo) che è progettato per individuare o prevenire un attacco informatico o per riprendersi dallo stesso;
Un servizio di sicurezza, è ,invece, un processo o servizio di comunicazione che garantisce la sicurezza dei sistemi di processione dei dati e di trasferimento delle informazioni di un’organizzazione. I servizi sono creati per difendersi da un attacco informatico e sfruttano uno o più meccanismi di sicurezza per provvedere allo scopo.

Un utile metodo per classificare gli attacchi informatici è in termini di attacchi passivi ed attivi.
Con i primi si intende classificare quegli attacchi che hanno lo scopo di apprendere o usare informazioni dal sistema, ma non danneggiare le risorse del sistema mentre con "attacchi attivi" si intende classificare gli attacchi che hanno lo scopo di alterare le risorse contenute in un sistema o influenzare le sue operazioni.

Gli attacchi passivi, in particolare, implicano l'intercettazione o il monitoraggio delle trasmissioni. Lo scopo dell'attaccante è di ottenere le informazioni che si stanno trasmettendo.
Due tipi di attacchi passivi sono: la diffusione dei contenuti dei messaggi e l'analisi del traffico.
La diffusione dei contenuti dei messaggi, come si può capire, è il rilascio di informazioni sensibili o confidenziali.
L'analisi del traffico è, invece, più sottile. Anche se si riuscisse a criptare l'informazione un potenziale attaccante potrebbe comunque riuscire ad individuare, analizzando il traffico dei dati, la posizione e l'identità degli host e la lunghezza e la frequenza dei messaggi: informazioni che possono aiutare a determinare la natura della conversazione.
Questo tipo di attacchi è generalmente difficile da individuare, poiché non coinvolgono l'alterazione dei dati. Di solito i messaggi vengono inviati e ricevuti normalmente e niente fa sospettare le parti coinvolte della presenza di una terza.
Quando si tratta di affrontare gli attacchi passivi, si parla infatti molto più di prevenzione (tramite ad esempio la crittografia) che di individuazione.

È bene ricordare che un attacco passivo può anche essere messo in atto per apprendere di più sul sistema da attaccare e che raramente un attacco attivo può essere lanciato con successo senza nessuna conoscenza riguardo il sistema informatico vittima.

Gli attacchi attivi, invece, implicano la modifica di un flusso di dati o la creazione di un falso flusso e possono essere suddivisi in quattro categorie: masquerade, replay, modification of messages e denial of service.
Una "masquerade" ha luogo quando un'entità con bassi o nessun privilegio finge di esserne un'altra con privilegi superiori ottenendoli.
Il "Replay" implica la cattura passiva di un'unità di dati e la loro conseguente ritrasmissione per produrre un effetto indesiderato.
La modifica dei messaggi è un attacco che ha lo scopo di alterare il contenuto o parte del contenuto di un messaggio o di ritardarlo per produrre un effetto dannoso.
Il denial of service blocca o rallenta il normale uso delle reti di comunicazione. Questo attacco può avere un bersaglio specifico, prendendo di mira le comunicazioni verso un

particolare destinatario o può colpire un intero network disabilitandolo, sovraccaricandolo o riducendo le performance. Gli attacchi attivi, a differenza di quelli passivi, sono difficili da prevenire, a causa della vasta gamma di vulnerabilità potenziali.
L'obiettivo generalmente è infatti quello di individuarli e ripristinare il corretto funzionamento del sistema.
Se l'individuazione ha un effetto deterrente, essa può contribuire alla prevenzione.
Fonte: [9]

2.3 Tipi di malware

Un malware, diminutivo della locuzione malicious software, è un software che può essere usato per compromettere le funzioni di un computer o una rete, per rubare dati, bypassare uno o più sistemi di sicurezza, o per causare qualsiasi altro tipo di danno ad uno o più computer. Quindi "malware" è un termine ampio per definire una grande varietà di programmi malevoli, i più comuni sono gli spyware, gli adware, i bot, i bug, i rootkit, i Trojan Horse e, ovviamente, i virus ed i worm.

Gli spyware sono un tipo di software malevolo che spia un utente ignaro; le sue attività di spionaggio includono (ma non si limitano a) il monitoraggio delle attività, la raccolta di informazioni personali e di tutto quello che viene inserito dalla tastiera o registrato da qualsiasi altro strumento di input del computer (webcam, microfono, sensore di impronte e altri). Gli spyware possono però avere altri tipi di capacità utili al software per nascondersi e mandare dati senza essere scoperto da un amministratore o rilevato dal firewall.
Si diffondono tramite l'exploit di vulnerabilità di altri software o legandosi a software legittimi o tramite Trojans.

Gli adware sono un tipo di malware che automaticamente permette la visualizzazione di pubblicità indesiderate. Esempi comuni, che includono le pubblicità pop-up sui siti web o quelli mostrati dalle versioni gratuite di alcuni software, seppur chiaramente fastidiosi non sono realmente dannosi o particolarmente difficili da eliminare, ma non è raro che questi software abbiano alcuni tratti degli spyware.

I bot sono programmi che possono essere creati anche senza scopi malevoli, essi servono ad automatizzare specifiche operazioni e il loro uso va dal gaming, alle aste online, ma sono menzionati perché è possibile crearli anche per scopi più sinistri.

Sta diventando comune, infatti, creare botnet (reti di bot) in computer di terze parti, generalmente utenti ignari, che possono essere usate per eseguire attacchi DDoS (Distributed

Denial of Service), ma non solo: essi possono essere usati come spambot con caratteristiche simili agli adware o per distribuire software maliziosi.

Un bug non è esattamente un malware, ma lo diventa quando viene sfruttato a scopi malevoli. In informatica, un bug, è un'imperfezione in un software causata da un errore umano, che causa risultati indesiderati, e che esiste nel codice sorgente di un programma.
Bug minori possono causare qualche comportamento inaspettato del programma, altri maggiori possono causare chiusure inaspettate (dette crash) o blocchi del programma, ma i bug più problematici sono quelli relativi alla sicurezza che possono permettere ad un utente malintenzionato di bypassare i protocolli di identificazione, fare l'override dei privilegi di accesso (agire come un amministratore usando un account configurato per agire come utente), o, ancora, rubare dati.
È bene tenere a mente che, tecnicamente, ogni bug può essere prevenuto con una adeguata conoscenza dell'ingegneria del software, con controlli della qualità o con strumenti di analisi del codice.

Un Ransomware è un software particolarmente insidioso, famoso per la recente diffusione di WanaCrypt0r 2.0 (conosciuto come WannaCry), che limita l'accesso dell'utente al computer, criptando i dati dell'hard drive o bloccando il sistema, e che in seguito mostra un messaggio che intende obbligare l'utente a pagare una determinata somma al creatore del malware per poter riottenere l'accesso al computer.
I ransomware si diffondono generalmente tramite download o tramite altre vulnerabilità in una rete network.

Un rootkit è un tipo di software malevolo ideato per ottenere il controllo remoto ad un computer senza essere rilevati da amministratori, utenti o programmi di sicurezza.
Una volta che il rootkit è installato, una terza parte può rubare informazioni, installare o disinstallare software, modificare le configurazioni di sistema, e anche alterare o eseguire altri software. A causa della sua capacità di nascondersi continuamente, la prevenzione, il rilevamento e la rimozione di questo tipo di malware è molto difficile e ci si affida quindi a metodi manuali più che automatici per l'identificazione e la rimozione.
Si può prevenire l'infezione da rootkit applicando patch alle vulnerabilità software rilevate, evitando download sospetti e tenendo aggiornato il sistema operativo e i software di sicurezza.

Un Trojan horse, comunemente noto semplicemente come trojan, è un tipo di malware che facendo credere all'utente di essere un programma o un file lecito, si fa scaricare ed

installare. Un Trojan una volta installato, se non rilevato, può eseguire tutte le attività tipiche di un rootkit, inoltre, a differenza di altri tipi di malware non è programmato per auto-replicarsi e diffondersi (questa caratteristica lo renderebbe un worm).

Gli ultimi due tipi di malware comuni da analizzare sono molto famosi: i virus ed i worm che differiscono in termini di funzionamento e diffusione, ma hanno simili capacità potenziali.

Infatti potenzialmente sia i virus che i worm hanno la capacità di permettere un accesso remoto costante e totalmente anonimo, di rubare dati personali e finanziari, di danneggiare fisicamente un computer o una rete e di eseguire qualsiasi tipo di codice.

Le differenze in termini di funzionamento sono varie: la più evidente è sicuramente quella che un virus non sempre infetta il computer automaticamente dopo l'installazione, ma ha bisogno di essere avviato.
L'avvio può avvenire quando alcune condizioni (ad esempio una certa ora di una certa data) si verificano o anche quando un altro programma (ad esempio chrome.exe) è avviato, inserendosi nello stack di esecuzione del programma.
Il virus, inoltre, cerca di diffondersi replicandosi e copiandosi in altri programmi o file e questo facilita la sua diffusione.

Si può venire infettati da un virus anche attraverso vulnerabilità di applicazioni web come quelle che permettono il XSS (Cross-Site Scripting) che permette l'esecuzione di codice Javascript da remoto.

Un worm, invece, entra in funzione dal momento in cui l'infezione è avvenuta e può diffondersi velocemente su un network senza aver bisogno di interazione con il creatore.
Il worm di per se è innocuo, la parte dannosa è il cosiddetto "payload", la parte di codice che equipaggiata al worm lo rende in grado di fare potenzialmente qualsiasi tipo di danno.

Questo tipo di malware infetta i computer sfruttando le vulnerabilità dei software ed è in grado di nascondersi per anni e all'evenienza aggiornarsi; nel terzo capitolo vedremo che Stuxnet è stato creato nel 2005 ed ha infettato la rete della centrale nucleare Iraniana nel 2007 prima di essere usato, addirittura, nel 2010.

2.3 Cyber-armi: determinanti, struttura e caratterizzazione

Le cyber-armi hanno fatto parte dell'arsenale bellico delle nazioni sviluppate già dal 1990, è stato, però, solo nel 2010 che il vero potenziale strategico di un codice malevolo è stato rivelato al mondo con il sabotaggio della centrale nucleare Iraniana di Natanz.
Pur tenendo presente che non esiste nessuna definizione internazionalmente riconosciuta, per i nostri scopi di analisi definiamo una cyber-arma come un codice informatico che è usato, o

progettato per essere usato, con l'intento di minacciare o causare danni di tipo fisico, funzionale o mentale a strutture, sistemi o esseri viventi.
Per sottolineare la difficoltà nel definire e concettualizzare le cyber-armi teniamo presente alcune considerazioni.
La prima è che la maggior parte delle cyber-armi manca della componente fisica: sono formate da codice informatico, parte del mondo sebbene non percepite come tale finché il loro effetto non si manifesta.
Questo crea problemi non solo perché, ovviamente, rende l'arma molto difficile da tracciare ed intercettare, ma anche perché rende ostico il tentativo di regolarizzarla o bandirla in quanto la giurisprudenza generalmente concepisce le armi come entità fisiche piuttosto che immateriali. L'eccezione è quando dell'hardware viene modificato per essere usato come cyber-arma o quando è progettato per esserne parte.

La seconda considerazione riguarda l'ultima parte della definizione che annovera il danno o l'intento dello stesso come condizione per distinguere un qualsiasi codice informatico da una cyber-arma. Viene quindi da chiedersi se malware che spiano un sistema o estraggono dati, senza comprometterlo, possano essere definiti cyber-armi.
Stabilire la natura ed il grado del danno è un processo complesso poiché dipende direttamente dall'obiettivo dell'attaccante, infatti, a differenza di un'arma convenzionale, una informatica non dispone di mezzi propri per causarlo (si pensi ad un ordigno esplosivo o ad un gas); il danno, almeno quello diretto, esiste solo nella misura in cui la modifica del comportamento degli asset coinvolti è in grado di procurarlo.

Prima di analizzare il ciclo di vita di una cyber-arma analizziamo la sua struttura che dividiamo in tre elementi: lo strato di accesso , lo strato di trasporto e il payload.

Il primo strato è basato sullo sfruttamento di una vulnerabilità ed è, praticamente, ciò che permette alla cyber-arma di penetrare il sistema.

La natura della vulnerabilità può essere di tipo software se si sfrutta un bug non ancora risolto, o di tipo hardware se ha a che fare con difetti dello stesso, o di configurazione se è causata da errori nell'installazione o nell'aggiornamento del sistema, o ancora di tipo umano, se è causata da persone interne all'organizzazione.
La parte di codice di un malware che è responsabile della sua installazione è, generalmente, chiamata "dropper" e l'analisi di questi è molto importante, in modo particolare se sfruttano vulnerabilità.

Lo strato di trasporto, invece, rappresenta il meccanismo di consegna e propagazione delle componenti del software di una cyber-arma nel sistema attaccato.
Questo strato può essere realizzato a livello logico attraverso siti web, certificati, phishing o altri strumenti di questo tipo o a livello fisico se si usano dispositivi esterni come chiavette USB, CD e DVD.

Il terzo strato è il payload che è un applicazione software o uno script progettato, creato o usato per compromettere un sistema o i dati contenuti.
Il payload può avere due tipi di architetture: singola, nel caso in cui bisogna compromettere un sistema semplice o di singola funzione della cyber-arma, o multipla, nel caso di un sistema complesso o di più funzioni del malware.

Da un punto di vista strategico-militare è chiaro che una cyber-arma viene costruita per avere un vantaggio rispetto all'avversario, raggiungendo con successo degli obiettivi in una situazione di conflitto.

Tenendo questo a mente e da ciò che è stato già esposto possiamo definire alcuni elementi caratterizzano le armi informatiche (e le distinguono, in alcuni casi, da quelle convenzionali) che saranno utili, anche, per analizzare Stuxnet nel terzo capitolo.

La prima caratteristica di una cyber-arma è che viene progettata specificatamente per aggredire il target: non è possibile, quindi, crearne una in grado di colpire sistemi di diverso tipo a differenza di armi convenzionali. Un attacco convenzionale può colpire e danneggiare allo stesso una centrale elettrica e una diga, mentre uno informatico progettato per colpire la prima, difficilmente potrà colpire anche la seconda.

Altre caratteristiche è che le cyber-armi sono intangibili e la loro progettazione richiede una profonda conoscenza del target, inoltre, sono, generalmente, molto meno costose da costruire.
Un'arma informatica è configurabile e possono esistere varie versioni che aggrediscono diverse vulnerabilità per adempiere allo stesso scopo, ma nonostante questo restano non riutilizzabili poiché una volta scoperta l'arma si viene a conoscenza anche della vulnerabilità usata e vengono prese appropriate contromisure.
Un'ultima caratteristica che possiamo definire è che le cyber-armi esattamente come quelle convenzionali hanno natura violenta e il loro impatto sull'ambiente fisico può essere pericoloso.

Fonti: [10], [11]

2.4 Cyber-armi: ciclo di vita

Una cyber-arma viene progettata e costruita allo scopo di ottenere un vantaggio su un avversario all'interno o all'esterno del cyberspazio e questo implica la partecipazione di una componente umana.

Quest'ultima può essere di tipo:

1) Governativo se a richiederne la progettazione è uno stato o una sua istituzione e, nonostante la presenza di organizzazioni molto poco elastiche dilunghi i tempi necessari allo sviluppo dell'arma stessa, grazie all'accesso a più risorse (di intelligence, di personale o di equipaggiamento), una componente governativa porta a produrre armi più sofisticate;
2) Non-Governativo in caso di progettazione da parte di un gruppo o di organizzazioni di persone che decidono di implementare una cyber-arma senza associarsi a nessuno stato. Le motivazioni possono essere varie: di tipo economico, ideologico o etico, ma è certo che la flessibilità di questo tipo di organizzazioni porta a produrre armi più velocemente anche se meno sofisticate;
3) Ibrido se è rappresentata da attori Statali che si servono di esperti non appartenenti ad alcuna agenzia o da persone che sono supportate da uno Stato.

Indipendentemente dalla componente umana, l'organizzazione definirà gli obiettivi che vuole raggiungere all'interno o all'esterno del cyberspazio per poi selezionare il bersaglio o i bersagli adatti al raggiungimento dello scopo.
L'impatto dell'arma è l'effetto fisico o digitale che essa produce e possiamo distinguere:

1) In base agli intenti dell'organizzazione:
   a) la categoria degli effetti desiderati che contribuirà a raggiungere gli obiettivi della missione;
   b) la categoria degli effetti non desiderati che influenzeranno negativamente il raggiungimento degli obiettivi.
2) In base alle aspettative dell'organizzazione:
   a) la categoria degli effetti attesi che racchiude tutti quelli che era possibile prevedere che fossero desiderati o meno;
   b) la categoria degli effetti non attesi che racchiude quelli non previsti avuti su altre dimensioni non prettamente bersaglio dell'organizzazione.

Procediamo ora ad analizzare il completo ciclo di vita di una cyber-arma che, al fine di esporre una descrizione esaustiva di ogni aspetto, è divisa in dieci fasi:

- Definizione del progetto: in questa fase la cyber-arma viene concettualizzata sia dal punto di vista strategico che manageriale, quindi viene stabilita l'architettura dell'arma e le sue funzioni principali.
- Ricognizione: in questa fase è effettuata una ricerca riguardo il bersaglio al fine di trovare vulnerabilità che possano essere sfruttate per raggiungere lo scopo dell'attaccante. È una fase di ricerca e di apprendimento sui sistemi da attaccare in modo da allargare la superficie di attacco.
- Progettazione: in questa fase vengono descritte all'intera organizzazione le specifiche, le funzionalità e ogni dettaglio utile riguardo l'arma, comprese le deadline di ciascun modulo e componente.
- Sviluppo: in questa fase gli ingegneri si occupano sia di scrivere il codice usando uno o più linguaggi di programmazione o di script sia di definire le opportune simulazioni che serviranno nella prossima fase.
- Testing: in questa fase gli ingegneri creano le simulazioni definite nella fase precedente preparando l'ambiente per i test e sfruttando al meglio le informazioni ottenute durante la fase di ricognizione, cercando quindi di creare un ambiente che sia il più simile possibile a quello da attaccare. Questa fase è molto importante poiché un'arma non testata quasi certamente fallirebbe e potrebbe allertare gli amministratori del sistema da attaccare.
- Validazione: in questa fase i risultati della fase di testing vengono raccolti e confrontati con gli obiettivi della fase uno e tre: in caso questi ultimi siano raggiunti si procede alla prossima fase, altrimenti si ritorna alle fasi di sviluppo, testing ed, eventualmente, ricognizione.
- Intrusione e controllo: se nella fase precedente si sono raggiunti gli obiettivi preposti l'arma è validata e pronta all'uso.
  Questa fase coinvolge due processi: il primo è quello di intrusione e riguarda il momento in cui la cyber-arma entra effettivamente nel sistema bersaglio; come abbiamo visto l'intrusione può essere remota o fisica.
  Il secondo processo riguarda l'ottenimento del controllo del sistema in modo da monitorarlo e decidere quando lanciare l'attacco.
- Attacco: in questa fase l'attacco viene lanciato attivando il payload equipaggiato all'arma che cercherà di completare l'attacco.
- Mantenimento: in questa fase l'attacco viene monitorato in modo da accertarsi di provocare gli effetti desiderati, in caso contrario alcune misure potrebbero essere ancora essere applicate per cercare di risolvere il problema o si può passare alla decima ed ultima fase.
- Esfiltrazione: in questa ultima fase il ciclo di vita dell'arma termina ed essa viene rimossa dal sistema bersaglio. Questa fase non è obbligatoria e molto spesso fallisce poiché può

essere veramente molto complicato rimuovere ogni traccia di un attacco, ma sarebbe fondamentale se l'organizzazione avesse deciso di usare l'arma anche su altre strutture simili.

C'è da considerare, infine, che in alcuni casi l'esfiltrazione può essere un obiettivo dell'organizzazione in quanto quest'ultima non vuole rendere nota la sua partecipazione all'aggressione o, al contrario, a volte questa fase non viene realizzata proprio perché l'organizzazione vuole che la responsabilità dell'attacco gli venga riconosciuta.

Fonte: [13]

3.Il worm Stuxnet

Quattro vulnerabilità zero-day identificate e sfruttate, un rootkit per Windows, una interfaccia di comando e controllo, il primo rootkit mai creato per attaccare le PLC, tecniche per l'evasione degli antivirus, complessi processi di iniezione ed occultamento del codice, routine per infettare network e aggiornamenti peer-to-peer: questo è Stuxnet.
In questo ultimo capitolo esporremo prima degli accenni sugli ICS, per poi passare al worm vero e proprio per il quale illustreremo l'architettura, identificheremo gli elementi caratterizzanti e spiegheremo il funzionamento.
I riferimenti per questo capitolo sono le fonti da [14] a [20].

3.1 Gli Industrial Control Systems

Industrial Control Systems è un termine generico per indicare diversi tipi di sistemi di controllo, inclusi i sistemi Supervisory Control And Data Acquisition (da qui in avanti SCADA), i Distributed Control Systems (da qui in avanti DCS) e altri sistemi configurati per il controllo industriale che fanno uso delle Programmable Logic Controller (da qui in avanti PLC).
Anche se i sistemi di controllo, generalmente, operano in maniera molto simile, essi differiscono in alcuni aspetti; la differenza più evidente sta nel fatto che i DCS e altri tipi di sottosistemi controllati da PLC sono localizzati in un'area ristretta, mentre i sistemi SCADA sono progettati per acquisire dati da posti geograficamente distanti.
Per questo motivo non è difficile intuire che le comunicazioni tra DCS e PLC avvengono attraverso tecnologie Local Area Network (LAN) che sono più sicure e veloci di quelle basate sulla comunicazione a lunga distanza dei sistemi SCADA che sono, infatti, progettati specificamente per affrontare le diverse sfide che quest'ultima implica.

Analizzeremo ora questi tre sistemi più approfonditamente in modo da comprendere meglio il problema che Stuxnet ha costituito e che altri malware simili possano potenzialmente costituire.
I sistemi SCADA sono usati per controllare asset dispersi geograficamente dove l'acquisizione dati è importante quanto il controllo; essi sono progettati per acquisire informazioni e a trasferirle ad una struttura centrale dove vengono esposti ad un operatore testualmente o graficamente che in questo modo può supervisionare e controllare i processi in tempo reale. Gli SCADA integrano, infatti, i sistemi di acquisizione dati, i sistemi di trasmissione dati e un software Human-Machine Interface (da qui in poi HMI) per fornire un sistema centralizzato di monitoraggio e controllo di diversi processi di input e output.

I sistemi SCADA sono costituiti sia da una parte hardware che da una parte software: la prima generalmente include una Master Terminal Unit (da qui in avanti MTU), vario equipaggiamento per la comunicazione e una o più Remote Terminal Unit (da qui in avanti RTU) o PLC, che controllano attuatori e/o sensori per il monitoraggio. La MTU conserva e processa le operazioni e i risultati di input ed output delle RTU e delle PLC che controllano i processi locali.
L'hardware dedicato alle comunicazioni permette lo scambio di informazioni tra la MTU e le RTU e/o PLC.

La parte software, invece, è programmata per gestire il sistema dicendogli cosa monitorare, quale range di valori è accettabile e cosa fare in caso questi ultimi fuoriuscissero dall'insieme di valori accettabili.

Un'altra componente fondamentale per un sistema scada sono gli Intelligent Electronic Devices (da qui in avanti IED), che, come si intuisce in parte dalla parola, sono dei sensori o degli attuatori intelligenti: in grado, quindi, di acquisire informazioni, comunicarle ad altri dispositivi e eseguire processi e controlli in locale; il loro uso permette l'automazione di diversi processi a livello locale.

I Distributed Control Systems sono usati per il controllo dei sistemi di produzione all'interno della stessa area geografica; questi sistemi fanno uso di meccanismi e routine di controllo per mediare un gruppo localizzato di controllori preposti al completamento di un intero processo di produzione.
Inoltre i DCS moderni sono interfacciati con la rete aziendale per fornire ai manager aggiornamenti e feedback riguardo la produzione.

I Programmable Logic Controllers possono essere usati dai sistemi SCADA che dai DCS come componenti di controllo di sistemi gerarchici in modo da provvedere alla gestione degli asset locali tramite il controllo dei feedback; possono, però, anche essere implementati come componente primario in sistemi di controllo più piccoli.

I PLC hanno una memoria programmabile per conservare istruzioni per l'implementazione di specifiche funzioni, queste, infatti, sono caricate con blocchi di codice scritti in vari linguaggi di programmazione come STL o SCL che sono poi eseguiti dal controllore per avviare, controllare e monitorare i processi industriali. Questo ultimo punto è bene ricordarlo perché sarà fondamentale più avanti.

La centrale nucleare colpita da Stuxnet implementava questo tipo di tecnologia per l'automazione di molti processi, ma non è il solo tipo di struttura a farne uso; gli ICS

vengono usati per la produzione e la distribuzione di energia elettrica, gas, acqua e petrolio e sono asset chiave per il monitoraggio ed il controllo di condotti petroliferi, navi, furgoni e sistemi ferroviari e di purificazione delle acque.
Il danno potenziale che un hacker può provocare attraverso questi sistemi è elevato ed è per questo necessario eseguire continui controlli per garantirne la sicurezza.
Fonte: [19]

3.2 Contesto storico e scenario di attacco

Il sabotaggio della centrale nucleare Iraniana, sebbene avvenuto sotto l'amministrazione Obama, fa parte di un piano sviluppato sotto la presidenza di George W. Bush nel 2006. L'operazione, nome in codice Olympic Games, sembrava essere l'unico modo per bloccare il programma nucleare dell'Iran.
Al tempo, infatti, gli alleati Europei degli Stati Uniti erano preoccupati per gli effetti che le sanzioni sull'Iran avrebbero potuto avere sulle proprie economie e, avendo da poco accusato falsamente Saddam Hussein di ricostruire il proprio arsenale nucleare in Iraq, il presidente Bush non godeva di molta credibilità nell'ambiente internazionale.
L'Iran avvertendo questo clima di incertezza, ostacolò le negoziazioni nonostante avesse, effettivamente, ripreso ad arricchire l'uranio. Quest'ultimo è un processo con il quale si cerca di ottenere maggiori concentrazioni di uranio 235 (partendo dall'isotopo uranio 238), poiché questo ha maggiori probabilità di dividersi e dare il via alla "fissione nucleare".
Il presidente Iraniano, Mahmoud Ahmadinejad, disse di voler installare, nella centrale nucleare di Natanz, oltre cinquantamila centrifughe a scopi civili; la dichiarazione, però, preoccupò l'amministrazione Bush e persino il Vice Presidente Dick Cheney chiese al Presidente di considerare l'ipotesi di un'attacco militare contro la struttura.
Di seguito, dopo un'attenta valutazione si concluse che le opzioni militari avrebbero tutte avuto esiti incerti e il piano del sabotaggio informatico nacque quando il generale James Cartwright la presentò al Presidente Bush; questa idea avrebbe portato alla creazione della più sofisticata cyber-arma mai vista.
Bisognava accedere ai controlli industriali della centrale nucleare, il che era complicato perché questi sistemi erano, ovviamente, air-gap, cioè fisicamente non collegati ad Internet; il codice avrebbe poi infettato i componenti che controllavano le centrifughe.
La prima parte dello sviluppo di una cyber-arma, come abbiamo visto, è la ricognizione, cosa che venne fatta o sviluppando uno spyware che, inserito nei dispositivi progettati dalla Siemens, avrebbe comunicato allo spionaggio Statunitense le configurazioni delle PLC e le schematiche degli ICS all'interno della centrale nucleare o facendole rubare ad un insider.
Una volta terminata la ricognizione e sviluppato il worm, vi è la fase di testing e al team di sviluppo serviva replicare perfettamente le condizioni in cui il worm si sarebbe trovato

all'interno della struttura di Natanz.
In particolare, era necessario costruire una replica esatta delle P-1, vecchie e mal progettate turbine per l'arricchimento che gli Iraniani avevano comprato sul mercato nero dal ex-responsabile del programma nucleare Pakistano: Abdul Qadeer Khan.
Fortunatamente, gli Stati Uniti possedevano quel tipo di turbine, che gli erano state consegnate nel 2003 dal dittatore Libico Mu'ammar Gheddafi quando aveva rinunciato al suo programma nucleare.
I test vennero superati con successo e quindi venne deciso di usare il worm; il problema adesso era far entrare Stuxnet nella centrale nucleare di Natanz.
Gli Stati Uniti ed Israele si affidarono ad ingegneri, tecnici della centrale nucleare e altri, che fossero spie o meno, che avevano accesso fisico alla struttura.
Quando l'attacco colpì la centrale per la prima volta nel 2008, gli Iraniani, ignari dei reali motivi del guasto alle turbine, incolparono il personale e ci furono anche dei licenziamenti.
Questo primo tentativo, però, danneggiò relativamente poco la centrale e quando Bush lasciò il suo posto ad Obama nessuna turbina era ancora stata distrutta.
Il nuovo presidente sembrò essere molto interessato al programma di cyber-attacco che, quindi, proseguì fino al 2010, quando la nuova versione del worm venne inserita nella centrale.
La nuova versione riuscì effettivamente a distruggere oltre 1000 turbine, ma un bug nel codice avrebbe presto portato all'identificazione ed eliminazione del malware; infatti il malware, verosimilmente, si replicò nel computer portatile di un ingegnere che sarebbe poi tornato a casa e lo avrebbe connesso ad internet.
Il bug portò Stuxnet a non riconoscere il cambiamento di ambiente, per cui, credendo di trovarsi ancora nella centrale, inizio a replicarsi e a diffondersi tramite internet.
Secondo la fonte [14] il 29 Settembre del 2010, vi erano approssimativamente 100.000 host infetti, di cui il 60% solo in Iran, il che portò gli investigatori a credere che il bersaglio del worm era proprio in quel paese.

Gli analisti scoprirono che lo scopo finale di Stuxnet, era quello di riprogrammare gli ICS modificando il codice delle PLC per fare in modo che si comportassero secondo il volere dell'attaccante e, allo stesso tempo, per nascondere le modifiche agli operatori della centrale.
Gli investigatori hanno poi ricostruito il possibile scenario di attacco del worm, ragionando sulle sue componenti. Per ora esponiamo lo scenario nel suo complesso, senza focalizzarci sul come, ma dal prossimo paragrafo verrà fatta un analisi più tecnica del malware.
Stuxnet è entrato attraverso una chiavetta USB infetta portata all'interno della struttura da una persona ignota, il malware ha, poi, iniziato a duplicarsi e replicarsi sulla rete locale in cerca di un field-PG, un tipo di computer, che ha installato il software Step 7, che serve a

programmare le PLC e che non è mai connesso ad Internet; proprio per questa loro ultima caratteristica, il centro di comando e controllo di Stuxnet non poteva gestire il worm e tutto il necessario per il sabotaggio doveva essere già integrato nel suo payload.
Si poteva però aggiornare il malware grazie ad una rete peer-to-peer che Stuxnet implementava nei computer infetti.
Una volta trovato un field-PG, con software Step 7, il malware provvedeva a modificare il codice delle PLC in modo da sabotare la centrale, ma anche in modo da nascondere le modifiche, così da non far accorgere gli Iraniani dell'attacco in atto.

Fonti: [19], [14]

3.3 Architettura del malware

Stuxnet ha una complessa architettura che consiste principalmente in un grande file dll (dynamic-link library), cioè una libreria che contiene codice e dati che possono essere usati da più programmi contemporaneamente, e da due blocchi di configurazione. Il file dll di Stuxnet contiene diversi export e risorse: le tabelle 3 e 4 della fonte [14], riportate di seguito, forniscono una buona panoramica per iniziare l'analisi del malware.

DLL Exports (tabella 3 fonte [14])	
Export #	**Funzione**
1	Infetta i dispositivi rimovibili ed avvia il server RPC dopo 60 secondi
2	Modifica le API per infettare i file di progetto Step 7
4	Avvia il processo di rimozione e chiama l'export 18
5	Verifica che l'installazione sia avvenuta correttamente
6	Verifica la versione del worm
7	Chiama l'export 6
9	Aggiorna Stuxnet dai progetti Step 7
10	Aggiorna Stuxnet dai progetti Step 7 (in modo differente da Export 9)
14	Infetta i progetti Step 7
15	Prepara al processo di installazione
16	Avvia il processo di installazione del malware
17	Rimpiazza i normali dll usati dai programmi Step 7
18	Disinstalla Stuxnet
19	Infetta i drive rimovibili (ad esempio chiavette USB e Hard-Disk esterni)
22	Contiene le routine di propagazione via Network
24	Controlla se è presente la connessione ad internet
27	Implementa il server RPC
28	Routine di comando e controllo
29	Routine di comando e controllo
31	Aggiorna Stuxnet dai progetti Step 7 nello stesso modo degli Export 9 e 10
32	Come l'export 1, ma avvia il server RPC all'istante

DLL Resources	(tabella 4 fonte [14])
Resource ID	**Funzione**
201	Driver Kernel-mode Mrxnet.sys, firmato da Realtek
202	File dll per l'infezione dei progetti Step 7
203	File .cab contenente file di tipo dll per l'infezione di WinCC (una HMI di Siemens)
205	File contenenti Dati di configurazione per la risorsa 201
207	Versione auto-eseguibile di Stuxnet
208	File dll di rimpiazzo per software Step 7
209	Dati criptati inseriti in %WINDIR%\help\winmic.fts
210	Template del PE-file, usato per creare il dropper di Stuxnet (WTR4132.TMP)
221	Modulo per la distribuzione del worm sfruttando la vulnerabilità MS08-067
222	Modulo per la distribuzione del worm sfruttando la vulnerabilità MS10-061
231	Modulo per il controllo della connessione ad Internet
240	Template del file .LNK, usato per sfruttare la vulnerabilità MS10-046
241	WTR4141.TMP: Template del file dll usato per caricare il dropper WTR4132.TMP
242	Driver Kernel-mode MrxCls.sys, che fornisce al worm funzionalità da rootkit
250	Modulo che sfrutta la vulnerabilità MS10-073 per la Privilege Escalation

Anche se alcune di queste funzioni non sono del tutto chiare adesso, lo saranno quando saranno descritte più avanti. Il dropper di Stuxnet contiene sia il file dll, che i blocchi di configurazione in una grande sezione chiamata "stub" che è essenziale per il funzionamento del worm: quando quest'ultimo viene eseguito, il file dll viene estratto dalla stub e mappato nella memoria del computer; da qui viene chiamato un export.
Quando ciò avviene, gli viene passato come parametro un puntatore alla sezione di stub (un puntatore, è un tipo di dato che rappresenta la posizione degli elementi di un programma nella memoria di un dispositivo elettronico). L'export, alla fine del suo stack di esecuzione, estrarrà il file dll dalla sezione di stub a cui punta, lo mapperà in memoria ed estrarrà un altro

export a cui verrà passato a sua volta il puntatore alla sezione di stub. In questo modo viene garantito che ogni strato del malware abbia accesso al principale file dll e a tutti i blocchi di configurazione.
Stuxnet ha anche un altro modo per chiamare gli export dal dll principale: questa tecnica consiste nel richiamare dalle sue risorse il template di un eseguibile e nel configurarlo con alcune informazioni, tra cui ad esempio quale file dll caricare e quale export chiamare, e, successivamente, iniettare questo eseguibile in un processo per farlo eseguire.

Parte fondamentale dell'architettura di Stuxnet è il modo in cui l'arma riesce a bypassare i meccanismi di sicurezza basati su tecnologie che monitorano le chiamate al metodo "LoadLibrary" che deve essere sempre chiamato per poter caricare i file dll di cui il malware necessita.
Il worm chiama il metodo LoadLibrary con un nome speciale, creato appositamente che generalmente causerebbe il fallimento della chiamata, però, Ntdll.dll, una libreria dinamica di sistema, è stato modificato per monitorare le richieste di caricamento con nomi speciali. Questo forza il caricamento dei file dll che servono a Stuxnet facendoli caricare non dall'hard-disk, come di solito avviene, bensì dalla memoria RAM dove sono stati allocati, precedentemente dall'arma stessa durante l'installazione.

Il metodo GetProcAddress viene, poi, invocato per ottenere il nuovo indirizzo memoria dell'export da chiamare.
Preciso che sia LoadLibrary che GetProcAddress non fanno parte di Ntdll.dll, ma di Kernel32.dll, che, come vedremo in seguito, sarà anch'esso oggetto di modifiche da parte del malware.

Prima di passare al paragrafo sull'installazione è bene parlare di altri due elementi fondamentali che caratterizzano Stuxnet: i blocchi di configurazione a cui abbiamo fatto riferimento prima e la tecnica di iniezione del codice.

I blocchi di configurazione sono insiemi di dati che contengono tutti i valori usati per controllare come Stuxnet agirà sul computer compromesso.

Per quanto riguarda l'iniezione di codice, ogni volta che un export viene chiamato, Stuxnet inietta l'intero dll in un altro processo e chiama l'export in particolare. L'iniezione può avvenire sia in un processo già esistente o in uno nuovo arbitrariamente o specificatamente scelto. Una volta iniettato il codice in un processo, Stuxnet, può poi decidere se mantenerlo o istruire quest'ultimo ad iniettarlo in un altro processo ancora.

La scelta del processo in cui iniettare il codice è presa, considerando sia i processi di default di Windows, sia i prodotti che, eventualmente, proteggono il computer.

Quindi in primo luogo, Stuxnet cerca il processo dell'antivirus configurato e in particolare cerca la presenza dei seguenti tre: Kaspersky Anti-Virus versione da 6 a 9 (da qui in poi KAV), McAfee e Trend PcCillin; se rilevati Stuxnet estrae informazioni sulla versione e basandosi su quella deciderà se continuare l'iniezione di codice o se eseguire prima la privilege escalation per evitare di essere rilevato.
La tabella 5 della fonte [14], qui riportata, offre un'ottima panoramica della scelta del processo in base al prodotto di difesa configurato.

Process Injection	**Tabella 5 fonte [14]**
KAV da v1 a v7	Lsass.exe
KAV da v8 a v9	Processo di KAV
McAfee	Winlogon.exe
AntiVir	Lsass.exe
BitDefender	Lsass.exe
ETrust da v5 a v6	Fallisce l'iniezione di codice
ETrust (altre versioni)	Lsass.exe
F-Secure	Lsass.exe
Symantec	Lsass.exe
ESET NOD32	Lsass.exe
Trend PC Cillin	Processo di Trend PC Cillin

Per eseguire l'iniezione del codice Struxnet crea un nuovo processo in modalità sospesa (usando il metodo CreateProcess di Kernel32.dll) dell'applicazione scelta. Fatto questo il modulo scelto (ad esempio Lsass.exe) viene rimosso dalla memoria e al suo posto viene caricato un PE (un eseguibile portatile, tipico dei programmi che si avviano anche da dispositivi rimovibili) estratto dalle risorse del worm. Prima di caricarlo, però, l'arma aggiunge una sezione chiamata .verif che rende il file PE della esatta dimensione del modulo originale rimosso; inoltre, dove prima vi era l'indirizzo di memoria del codice sorgente del processo originale, Stuxnet scrive un "jmp" (metodo in Assembly che esegue un "salto" da un indirizzo memoria ad un altro) che porta al file PE.

Fonti: [14], [15], [16]

3.4 L’installazione

Quando il file dll principale viene eseguito per la prima volta, l’export 15 viene invocato per primo; in sintesi nella tabella v’è scritto che il 15 prepara al processo di installazione, questo vuol dire che: controlla se l’arma è stata eseguita in una macchina windows compatibile, controlla se il computer è già infetto, eleva i privilegi se non possiede il livello più alto, controlla la presenza di eventuali antivirus, decide in quale processo iniettare il codice e, usando la tecnica vista poc’anzi, esegue l’iniezione e chiama l’export 16.
Il flusso di esecuzione dell’export 15 inizia controllando se i dati di configurazione sono aggiornati, questi possono essere conservati in due sezioni; il worm controlla quale delle due contiene quelli più recenti e procede con quei blocchi. In seguito l’arma controlla se il sistema operativo su cui si sta installando ha un'architettura a 32bit o a 64bit. Se la macchina è a 64bit l’esecuzione termina, altrimenti procede controllando il sistema operativo; Stuxnet procede con l’installazione solo se si trova su: Windows 2000, XP, 2003, Vista, 7, Windows Server 2008 o Windows Server 2008 R2.

In seguito Stuxnet controlla se possiede i privilegi da amministratore sul computer, se così non fosse sfrutterebbe una vulnerabilità per ottenerli; il worm contiene due modi per aumentare il proprio livello di privilegi: uno che funziona sfruttando una vulnerabilità dell’utilità di pianificazione (task Scheduler) presente su Windows Vista, Windows 7 e Windows Server 2008 R2 e una che sfrutta la vulnerabilità MS10-073, su Windows XP e 2000.
Se sfruttate con successo il dll principale di Stuxnet verrà eseguito da un nuovo processo (il csrss.exe) se viene sfruttata la MS10-073 o come una nuova task se viene sfruttata la vulnerabilità del Task Scheduler.
Infine, cerca la presenza di eventuali software anti-virus per capire, come esposto sopra, in quale processo iniettare il codice e invoca l’export 16 per avviare il processo di installazione.
L’export 16 è il file di installazione principale dell’arma e, esattamente come 15, esegue vari controlli per assicurarsi di non fallire e di non essere individuato.
Per prima cosa controlla che i dati di configurazione ricevuti da 15 siano validi, dopo controlla il valore “NTVDM TRACE” nella chiave di registro: “HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\MS-DOS Emulation” e se il valore equivale a 19790509 l’installazione fallisce. Questo è molto importante perché questo è chiaramente un marker messo dagli sviluppatori per evitare che una diffusione incontrollata del worm avesse infettato e danneggiato anche i propri sistemi, ma non solo; infatti, per alcuni investigatori il numero non è casuale, ma corrisponde ad una data: il 9 Maggior 1979, giorno in cui Habib Elghanian è stato giustiziato a Tehran. È stato uno dei primi civili di religione Ebraica ad essere stato giustiziato dal nuovo governo Islamico e ciò ha causato l’esodo dall’Iran della comunità Ebraica che contava circa

centomila membri. Ovviamente, che sia casuale o meno, chiunque avrebbe potuto usare quella data come marker e ciò non collega ufficialmente Israele a Stuxnet.

In seguito il worm controlla che la data corrente non sia dopo il 24 Giugno 2012, in quel caso l'installazione fallisce, altrimenti legge dalla sezione stub alcuni file, li cripta e li scrive sul disco, questi file sono: il dll principale di Stuxnet, un file contenente dati di 90 byte, i dati di configurazione di Stuxnet e un file di log. Fatto questo, l'arma controlla di nuovo la data e se il giorno 24 Giugno 2012 non è ancora arrivato prosegue con l'installazione decriptando e caricando in memoria il file dll principale appena creato per controllare, chiamando l'export 6, che le due versioni siano uguali. Dopo questo passaggio Stuxnet estrae e avvia due risorse: la 201 e la 242 che vengono scritte sul disco rispettivamente come Mrxnet.sys e MrxCls.sys. Entrambi i driver hanno la firma digitale della Realtek e alcune versioni precedenti di Stuxnet avevano il driver MrxCls firmato da JMicron; anche questo è molto importante dal punto di vista investigativo, poiché entrambe le società hanno uffici al Hsinchu Science Park a Taiwan e suggerisce l'impiego di spie per rubare fisicamente i certificati digitali.
Detto questo è bene specificare che il primo driver, MrxCls.sys, permette a Stuxnet di essere eseguito ogni qual volta si accenda il computer permettendo quindi all'arma di continuare il proprio lavoro in maniera persistente.
Il secondo, Mrxnet.sys, fornisce all'arma alcune funzionalità da rootkit che gli permettono di nascondere i file maliziosi sui computer compromessi, di rimpiazzare le parti che vengono rimosse e di copiarsi sui driver rimovibili inseriti per potersi diffondere.
A questo punto Stuxnet si inietta nel processo services.exe ed invoca l'export 32 per avviare il Server RPC e per infettare i dispositivi rimovibili e, poi, si inietta nel processo S7tgtopx.exe e chiama l'export 2 per infettare i file di progetto Step 7.
Infine il worm aspetta che l'RPC server funzioni correttamente e chiama l'export 9 per ricevere informazioni e salvarle in un file di log.
A questo punto tutte le routine per la diffusione e l'attacco sono state attivate.

Fonti: [14], [15], [16]

3.5 Routine di comando e controllo

Dopo aver eseguito le istruzioni descritte poc'anzi il worm cerca di contattare il server di comando e controllo tramite la porta 80, quindi usando il protocollo HTTP.
Due sono i server di comando e controllo noti: www.mypremierfutbol[.]com e www.todaysfutbol[.]com; questi URL che puntavano a server in Malesia e Danimarca ora sono stati reindirizzati per evitare che l'attaccante possa compromettere altri computer.
Stuxnet inizia a raccogliere dati sul sistema infetto tramite l'export 28, questi dati comprendono: informazioni complete sul sistema operativo, nome ed indirizzo di rete locale del computer e, sopratutto, informazioni riguardo la presenza o meno dei software di

programmazione degli ICS oggetto dell'attacco. Raccolte le informazioni e preparato il payload che le contiene, viene chiamato l'export 29 per mandarle al server iniettando il codice nel processo di un qualsiasi browser attivo. L'export 29 è programmato per mandare il payload ai server maliziosi specificati nei file di configurazione di Stuxnet, ma prima controlla se la connessione ad internet è attiva collegandosi a: www.msn.com o www.windowsupdate.com; quindi a siti perfettamente legittimi. Se il test da esito positivo il pacchetto network viene costruito e mandato tramite il protocollo HTTP.
È fondamentale annoverare che il server è programmato per mandare anche una risposta che contenga un modulo eseguibile, questa importante funzione, da a Stuxnet le funzionalità di una backdoor, poiché rende possibile aggiornare o eseguire qualsiasi tipo di codice sulla macchina compromessa.

Fonte: [14]

3.6 Routine di propagazione

Come abbiamo precedentemente introdotto, Stuxnet usa vari metodi per diffondersi, infatti, può farlo: tramite network, tramite dispositivi rimovibili o copiandosi nei progetti Step 7 e usando tecniche per auto-eseguire il worm all'avvio di questi ultimi.
Iniziamo l'analisi considerando i metodi di propagazione network; l'export 22 implementa la maggioranza delle routine di propagazione network e lo fa compilando una grande classe che contiene cinque sottoclassi, ognuna con metodi differenti, queste sono:

1. Una sottoclasse che gestisce le comunicazioni e gli aggiornamenti Peer-to-Peer;
2. Un metodo per infettare le macchine WinCC via database che usano hardcoded password, queste ultimo è un nome che definisce tutte quelle password o informazioni riservate che vengono inserite nel codice sorgente di un software senza essere criptate. È una pratica che qualsiasi manuale o bollettino di sicurezza informatica suggerisce di evitare;
3. Un metodo che permette la propagazione attraverso le risorse network condivise;
4. Un metodo che permette la propagazione attraverso la vulnerabilità MS10-061;
5. Un metodo che permette la propagazione attraverso la vulnerabilità MS08-067;

Li analizzeremo uno ad uno partendo dal primo metodo.
Il termine Peer-to-Peer indica un'architettura logica in cui ogni nodo, al pari di un altro nodo può avere sia funzionalità di client che di server. Stuxnet avvia questo tipo di comunicazione tramite l'implementazione di server e client RPC sui computer infetti. Qualsiasi computer infetto collegato alla rete può comunicare con il server RPC; in questo modo Stuxnet riesce ad aggiornarsi anche sui computer air-gap, cioè, ricordiamo, non connessi ad Internet.

Guardando i metodi implementati possiamo iniziare a capirne il funzionamento; la tabella che abbiamo ricostruito a partire dalla fonte [14] offre una buona panoramica.

Lato server	
Id metodo	**Funzione**
0	Acquisisce il numero della versione di Stuxnet installata
1	Riceve un file .exe e lo esegue tramite iniezione di codice
2	Carica un modulo ed esegue un export
3	Inietta del codice nel processo lsass.exe e lo esegue
4	Compila l'ultima versione del worm e lo invia ad un computer compromesso
5	Crea un processo
6	Legge un file
7	Scarica un file
8	Cancella un file
9	Scrive record di dati

Dal lato Client il computer compromesso chiama la funzione 0 per poi determinare se la versione del server è diversa o meno.
In caso la versione del server RPC sia aggiornata allora viene eseguita una chiamata alla funzione 4 che compila ed invia un eseguibile del worm.
Il lato client la riceve e successivamente la installa tramite iniezione di codice come abbiamo già visto.
Se, invece, la versione del server RPC fosse antiquata sarebbe il client a preparare un eseguibile dell'arma e ad inviarlo al server, chiamando poi la funzione 1.

Il secondo metodo viene invocato quando viene trovato un sistema che sta eseguendo il software di un database WinCC. Il worm si collega al server del database usando una hardcoded password tramite, appunto, il software che si sta eseguendo.
Una volta avvenuta la connessione Stuxnet invia del codice SQL malizioso che permette ad una versione dell'arma di essere trasferita ed eseguita, infettando il computer sul quale è eseguito il software del database. Inoltre modifica una tabella esistente facendo in modo che del codice venga eseguito ogni volta che la tabella viene visualizzata.
Dopo aver mandato questa query di configurazione, Stuxnet crea una tabella e vi inserisce un

valore binario e un blocco di dati di configurazione; il primo è una stringa esadecimale che rappresenta il dll principale del worm come un file eseguibile.
Se questo ha successo Stuxnet aggiunge il file appena creato come “stored procedure” e lo esegue. Una stored procedure in SQL è un insieme di comandi SQL che si usano frequentemente e che quindi vengono memorizzati per essere usati in modo più rapido.
Una volta eseguita, la stored procedure creata da Stuxnet viene eliminata insieme al file dll principale.

Il terzo metodo permette a Stuxnet di diffondersi attraverso le risorse condivise via network usando il meccanismo degli scheduled job o usando le Windows Management Instrumentation.

Il quarto metodo, invece, sfrutta una delle due vulnerabilità zero-day usate dall’arma per diffondersi via network. Il codice per sfruttare la vulnerabilità MS10-061 è contenuto nella risorsa 222 di Stuxnet e permette di inviare e salvare un file nella cartella %System% attraverso il meccanismo dello Spooler di stampa, cioè quel software che, seguendo una logica di tipo FIFO, gestisce le richieste di stampa ad una stampante condivisa.

Il quinto ed ultimo metodo di propagazione attraverso il network funziona attraverso lo sfruttamento della vulnerabilità MS08-067, che sfrutta una debolezza del protocollo SMB per eseguire codice arbitrariamente da remoto.

Uno dei principali modi in cui Stuxnet si diffonde è tramite l’infezione di dispositivi rimovibili, questa è una caratteristica fondamentale, poiché, come già detto, i computer che configurano gli ICS sono air-gap e gli operatori sono soliti scambiarsi informazioni e dati attraverso dispositivi rimovibili.
Le versioni più vecchie di Stuxnet usano un file autorun.inf per fare in modo che Windows esegua il worm automaticamente una volta inserita la chiavetta, ma versioni più aggiornate sfruttano una vulnerabilità note come CVE-2010-2568.
Spiegheremo in particolare il funzionamento di quest’ultima; il codice per rilevare ed infettare i dispositivi rimovibili è contenuto negli export 1, 19 e 32.
Gli export 1 e 32 implementano le routine che permettono al worm di capire quando è stato inserito un dispositivo rimovibile, mentre l’export 19 implementa la routine per copiare e diffondere l’arma e anche per cancellarla dal drive. Alcune condizioni, possono, infatti, portare Stuxnet ad eliminarsi come ad esempio l’aver infettato, con quel drive, già tre computer.
Per capire se deve infettare il drive, il malware controlla alcune condizioni:

- Il drive non è già infetto;
- Il worm deve avere il permesso (da un flag nei file di configurazione) di infettare i dispositivi rimovibili;
- L'infezione ha meno di 21 giorni;
- Il drive ha almeno 5MB di spazio disponibile;
- Il drive ha almeno tre file.

Se queste condizioni vengono soddisfatte, Stuxnet crea i seguenti file:

- %DriveLetter%\WTR4132.tmp: questo file di 500Kb contiene il dll principale di Stuxnet nella sezione di Stub e deriva dalla risorsa 210;
- %DriveLetter%\WTR4141.tmp: questo file di 25Kb deriva dalla risorsa 241;
- %DriveLetter%\Copy of Shortcut to.lnk;
- %DriveLetter%\Copy of Copy of Shortcut to.lnk;
- %DriveLetter%\Copy of Copy of Copy of Shortcut to.lnk;
- %DriveLetter%\Copy of Copy of Copy of Copy of Shortcut to.lnk;

Tutti i file .lnk sono creati a partire dal template contenuto nella risorsa 240 e ne sono necessari quattro poiché ognuno di essi colpisce specificamente un sistema operativo ed in questo modo Stuxnet può replicarsi, dal drive sui sistemi operati Windows 2000, XP, Vista, 7 e Windows Server 2003.
Questi file .lnk contengono un exploit che permette l'esecuzione automatica WTR4141.tmp quando un operatore, semplicemente apre la cartella del drive.
Il file per prima cosa cerca di nascondere i file di Stuxnet sul drive modificando le API di due librerie di sistema che abbiamo già incontrato: Kernel32.dll e Ntdll.dll.
In particolare modifica il codice originale delle funzioni FindFirstFileW, FindNextFileW, FindFirstFileExW di Kernel32.dll e le funzioni NtQueryDirectoryFile e ZwQueryDirectoryFile di Ntdll.dll.
La modifica fa in modo che il codice controlli le seguenti caratteristiche nei file:

- Se sono file .lnk controlla che il loro peso sia di 4171 byte;
- Se sono file con nome del tipo WTRxxxx, con peso tra i 4Kb e gli 8Mb, dove xxxx è un numero con 4 cifre decimali la cui somma modulo 10 è 0 (esempio 0145: 0 + 1 + 4 + 5 = 10 che mod10 fa 0; si può notare che i file di questo tipo di cui sopra hanno questa caratteristica).

Se una di queste due condizioni si verifica la risposta della API viene alterata, nascondendo il file. Subito dopo WTR4141.tmp esegue WTR4132 ed installa Stuxnet con le modalità che abbiamo già visto.

L'ultimo metodo di trasmissione è tramite la replica e l'esecuzione attraverso i file di progetto Step 7; vengono usati per questo scopo i metodi dell'export 2 che viene usato per modificare alcune API che servono al processo s7gtopx.exe ad aprire i file di progetto.
I dll modificati sono s7apromx.dll, mfc42.dll e msvcrt.dll dove CreateFileA viene rimpiazzato da CreateFilaA_hook, inoltre in ccprojectmgr.exe, StgOpenStorage viene rimpiazzato da StgOpenStorage_hook.

CreateFileA viene usato per aprire i file di progetto .S7P, con la nuova routine, però, viene anche chiamato l'export 9 che viene usato per conservare il percorso in cui è salvato il progetto aperto ed eventualmente per infettare quella cartella.
StgOpenStorage, invece, viene usato per aprire i file .MCP che possono essere revocati all'interno dei file di progetto Step 7; esattamente come CreateFile_Hook, la nuova routine chiama l'export 9 quando uno di questi file viene aperto per conservare il percorso ed eventualmente infettare la cartella.
In entrambi i casi, se il worm decide di infettare la cartella, viene chiamato l'export 14; la decisione è presa, come sempre, a seconda del verificarsi di alcune condizioni:

- Nel caso di file .S7P la prima condizione è che il file non deve essere troppo vecchio, cioè deve essere stato aperto negli ultimi 3,5 anni, la seconda è che deve contenere la cartella "wincproj" con almeno un file MCP valido ed infine, il file non deve essere un progetto-esempio di default.
- Nel caso invece di file .MCP esistono solo due condizioni, ma molto simili alle precedenti: il file deve essere stato aperto almeno una volta negli ultimi 3,5 anni e deve contenere la cartella GracS con almeno un file .pdl al suo interno.

Con questo termina la nostra esposizione dei metodi di trasmissione di Stuxnet e, già solo l'analisi di questi porta a capire ancora più a fondo la complessità di questo malware.
Fonte: [14]

3.7 Modifica del comportamento delle PLC

In questo ultimo paragrafo andiamo ad esaminare quella che è la funzione ultima di Stuxnet, quello per cui è stato creato, cioè modificare il comportamento dei controllori a logica programmabile (PLC) per sabotare una infrastruttura. Questa parte sarà molto tecnica e ci saranno molti passaggi specifici che potrebbero risultare poco chiari, ma questi sono ciò che rende Stuxnet la minaccia più sofisticata mai riscontrata. Ogni valore ricercato, ogni controllo effettuato, ogni riga di codice esiste per identificare specifici tipi di hardware che compongono il sistema che l'arma vuole attaccare e nessun altro. Stuxnet non è un worm creato per attaccare il più gran numero di macchine possibili, ricordiamo, infatti, che gli ICS

controllano gasdotti, impianti ferroviari e altre infrastrutture critiche. Per questo Olympic Games è un operazione chirurgica e in quanto tale deve essere il più preciso possibile.
Iniziamo l'esposizione ricordando che l'attacco è contro una centrale nucleare Iraniana che all'epoca aveva il compito di arricchire l'uranio anche a gradazione sufficiente per produrre armi nucleari. Questo era possibile grazie a centinaia di turbine la cui velocità era controllata da un motore elettrico, a sua volta controllato da una PLC. I computer che controllavano i PLC erano, come già specificato, air-gap quindi l'attacco era completamente automatizzato ed impossibile da fermare una volta avviata la procedura poiché il centro di comando e controllo non poteva comunicare con il worm in quel sistema. Vi era la possibilità di aggiornarlo secondo le modalità che abbiamo già visto, ma non di controllarlo; chiarito questo, procediamo con la trattazione.
L'infezione dei controllori a logica programmabile inizia con l'invocazione all'export 17 che utilizza la risorsa 208.
Una PLC, come introdotto nel paragrafo relativo, esegue dei blocchi di codice e la libreria responsabile della modifica di questi ultimi è la s7otbxdx.dll infatti è preposto proprio alla comunicazione tra il computer con il software Step 7 e l'oggetto fisico che la PLC controlla.
L'export 17 sostituisce l'originale libreria, con una malevola contenuta nella risorsa 208, fatto questo Stuxnet ha la possibilità di: monitorare i blocchi di codice che vengono scritti e letti dal controllore a logica programmabile, infettare quest'ultimo scrivendo i propri blocchi di codice, rimpiazzando quelli esistenti e di mascherare l'infezione a qualsiasi operatore.
Stuxnet, furbamente, non elimina la vecchia libreria, ma la rinomina in s7otbxsx.dll, in questo modo tutte le chiamate a metodi che un operatore esegue normalmente vengono reindirizzate da s7otbxdx.dll alla vera libreria e tutto sembra funzionare normalmente, se non fosse che a questo punto il worm può controllare qualsiasi segnale in ingresso ed uscita e modificarli. Addirittura 93 dei 109 export che contiene il falso s7otbxdx.dll vengono gestiti in questo modo, semplicemente richiamando il vero metodo da s7otbxsx.dll.
Una volta sostituito il dll, il worm usa l'API s7ag_read_szl per determinare il tipo di PLC, che deve essere della serie 6ES7-315-2 per proseguire; il passaggio successivo è enumerare ed analizzare i blocchi di dati di sistema (da qui in poi SDB, dall'inglese System Data Block). Stuxnet cerca un SDB con una DWORD all'offset 50h che sia uguale a 0100CB2Ch: questo specifica che il sistema usa un processore di comunicazione a PROFIBUS modello CP 342-5. Quest'ultimo è un bus per la creazione di network industriali, creato per gestire Input ed Output distribuiti. Fatto questo, cerca due valori e ne quantifica i ritrovamenti, i due valori sono 7050h e 9500h. Questo controllo fa passare allo stack successivo se, e solo se, il numero dei ritrovamenti di questi due valori è pari o superiore a 33. Questi sono codici identificativi assegnati a dei convertitori di frequenza, usati per controllare la velocità di altri dispositivi, come ad esempio i motori che facevano girare le turbine della centrale di Natanz.
Ricapitolando, Stuxnet cercava nel sistema infettato uno specifico tipo di PLC con uno

specifico processore di comunicazione che a sua volta comunicava con almeno 33 specifici convertitori di frequenza: il livello di conoscenza dell'architettura informatica della centrale e di precisione dell'attacco è veramente molto alto, il che rende ancora più evidente il coinvolgimento di almeno un attore nation state; insomma, un gruppo di hacker non è solito usare vulnerabilità zero-day dal valore di quasi mezzo milione di dollari sul mercato nero, per attaccare un solo obiettivo.
Il passo successivo è rimpiazzare il blocco di funzioni DP_RECV che serve per ricevere e mandare pacchetti network tramite il PROFIBUS. Anche questo blocco di funzioni non viene eliminato, ma rinominato in FC1869 e ogni volta che il blocco malevolo riceve un pacchetto, dopo averlo controllato, lo invia al blocco originale per l'esecuzione delle normali operazioni.
Infine Stuxnet infetta i cosiddetti Organization Blocks (da qui in poi OB) che corrispondono ad entry point di specifiche funzioni, o blocchi delle stesse, eseguiti ciclicamente dalla CPU. Il metodo di infezione è piuttosto elementare: il worm scrive semplicemente il suo codice prima di quello originale dell'OB in modo che venga eseguito per primo ad ogni chiamata. Enumerare tutti gli OB compromessi va al di là degli obiettivi della trattazione, ma due in particolare sono degni di nota: l'OB1 che è l'entry point principale dei programmi delle PLC e l'OB35 che è l'entry point di un programma eseguito ogni 100 millisecondi e la cui funzione è quella di monitorare gli input in modo da rispondere automaticamente e velocemente in caso di problemi critici che potrebbero danneggiare il sistema. Adesso l'arma è pronta ad attaccare, vediamo quindi cosa succede quando l'infezione è completa.
Arrivato sull'obiettivo il payload di Stuxnet viene eseguito automaticamente; il worm aspetta tredici giorni prima di fare qualcosa, perché quello è il tempo che ci vuole a riempire di esafluoruro di uranio tutte le turbine, e in questo tempo registra e salva tutte le normali operazioni. Una volta passati i 13 giorni, avviene l'attacco: le turbine ruotano a circa 1000Hz, cioè a circa 60.000 rotazioni al minuto, Stuxnet poteva agire in due modi: o far aumentare la velocità di rotazione fino a 1400Hz, cioè 84.000 rotazioni al minuto o ridurla a 2Hz, cioè 120 al minuto. Nel primo caso le turbine avrebbero subito la cosiddetta "frequenza di risonanza", a cui il metallo vibra fuori controllo; nel secondo caso un po' come in una trottola che ruota troppo lentamente, la differenza di velocità tra la parte più in alto e la punta fa ampiamente oscillare la turbina; in entrambi i casi il risultato è la rottura della centrifuga ed il rilascio di gas di uranio nell'area. Mentre questo avveniva, Stuxnet avrebbe inviato agli operatori i dati registrati nei precedenti 13 giorni, ma ad un certo punto anche questo non sarebbe bastato; una turbina accelerata a 1400Hz produce un suono assordante, molto più forte di una che gira a 1000Hz, gli operatori se ne sarebbero accorti e avrebbero cercato di arrestare il processo, ma l'arma, preparata, avrebbe intercettato anche questo segnale, bloccandolo. Il worm ha danneggiato circa un migliaio di centrifughe prima che il bug di un aggiornamento che aveva

reso il dropper più aggressivo lo facesse diffondere in fretta fuori dalla centrale ed individuare.

Fonte: [14]

Conclusioni

L'uomo ha iniziato a combattere sulla terra, per poi conquistare e combattere anche tra i mari, nello scorso secolo l'aviazione ha aggiunto una terza dimensione alla guerra e in questo, Stuxnet potrebbe aver aggiunto la quarta. Lo stesso presidente Obama aveva espresso dubbi riguardo l'uso di cyber-armi nel corso di Olympic Games, poiché erano incerti gli effetti che avrebbero avuto in ambito internazionale: l'attacco informatico avrebbe potuto giustificarne altri simili contro gli U.S.A. da parte, ad esempio, di Cina, Russia o Nord Korea; insomma, era un ambito inesplorato nel diritto internazionale. L'operazione Stuxnet ha rallentato poco il programma nucleare Iraniano a causa della sua prematura identificazione, ma questo è stato abbastanza per cambiare il modo in cui verranno combattute le guerre del futuro. Oggi l'Iran ha uno dei più grandi cyber-eserciti al mondo e alcuni attacchi a banche ed altre infrastrutture Americane sono riconducibili ad hacker Iraniani.
Con l'implementazione dell'Industria 4.0, minacce come questa potranno mettere a rischio la sicurezza di aziende e persone, cosa sarebbe potuto accadere se le vulnerabilità trovate a Natanz fossero state sfruttate per chiudere valvole nei gasdotti, per causare il deragliamento di treni o, ancora, per sabotare centrali elettriche o dighe ?
Nonostante questo non bisogna vedere con timore il problema della cyber-sicurezza; questo può e deve essere affrontato informando sul tema, aumentando gli investimenti e cercando persone competenti da inserire nel settore, poiché, non si può fermare il mondo per paura del cambiamento.

Bibliografia

[1] Articolo “Che cos’è l’industria 4.0 e perché è importante saperla affrontare.”

[2] Articolo “Industria 4.0 tutti i miliardi che il governo vuole mobilitare.”

[3] Articolo del Sole 24 Ore “Tre assi per consolidare l’impresa 4.0.”

[4] Report “Recommendations for implementing the strategic initiative INDUSTRIE 4.0.”

[5] Report del Boston Consulting Group “Industry 4.0.”

[6] Working Paper “Trade-offs between digital innovation and cyber-security.”

[7] Indagine conoscitiva della Decima Commissione Permanente su Industria 4.0 del 2016.

[8] Ponemon Institute LLC report - Cost of Data Breach Study

[9] Cryptography and Network Security - William Stallings

[10] Cyberweapons: power and the governance of the invisible - Tim Stevens

[11] Cyberweapons: an emerging global governance architecture - Tim Stevens

[12] Three Tenets for Secure Cyber-Physical System Design and Assessment - Jeff Hughes and George Cybenko

[13] Cyber Weapons: a Profiling Framework - Clara Maathuis, Wolter Pieters and Jan van den Berg

[14] W32.Stuxnet Dossier - Nicolas Falliere, Liam O Murchu, Eric Chien

[15] Stuxnet under the microscope - Aleksandr Matrosov, Eugene Rodionov, David Harley, Juraj Malcho

[16] Stuxnet Malware Analysis Paper - Amr Thabet

[17] Guide to Industrial Control Systems (ICS) Security - NIST publication - Keith Stouffer, Joe Falco, Karen Scarfone

[18] Articolo “8 companies to watch in circular economy” al link: https://www.greenbiz.com/article/8-companies-watch-circular-economy

[19] Articolo “Obama order sped up wave of Cyberattacks against Iran” al link: http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?pagewanted=all&_r=0

[20] Dettagli sulla vulnerabilità dell’utilità di pianificazione su Windows Vista, 7 e Server 2008 R2 https://www.cvedetails.com/cve/cve-2010-3338