Laboratório de Segurança Cibernética para Análise de Malwares Reais em Infraestruturas Críticas

Otavio Augusto Maciel Camargo [*] **Elson Costa Gomes** [**]
Antônio Nascimento Filho [***]
Anderson Fernandes Pereira dos Santos [***]
Antonio Eduardo Carrilho da Cunha [***] **Paulo César Pellanda** [***]

[*] *Escola de Aperfeiçoamento de Oficiais, Rio de Janeiro, RJ*
e-mail: contact@macciel.com
[**] *Fundação Parque Tecnológico Itaipu – Brasil, Foz do Iguaçu, PR*
e-mails: elson.gomes@pti.org.br
[***] *Instituto Militar de Engenharia, Rio de Janeiro, RJ*
e-mail:{antonio.nascimento,anderson,carrilho,pellanda}@ime.eb.br

Abstract: Industrial Control Systems (ICS) are responsible for controlling critical infrastructures that often are targets of cyber attacks motivated by political, military, or financial interests. Supervisory Control and Data Acquisition (SCADA) systems are among the main components of ICS, are highly interconnected systems, and employ solutions common to conventional computer systems. Malware is among the top cyber threats to these systems. However, testing the cyber resilience of a real ICS requires testbeds and simulations to verify the harmful behavior of these threats without putting the original system at risk. This paper presents the results of an experiment that analyzed the impact of attacks with real malware on a SCADA system connected to a real-time simulated electrical system model in a hardware-in-the-loop testbed. The results show that even generic malware can impact the Modbus/TCP communication, causing interruptions and delays that can harm the SCADA system operation. This effect may affect the electrical system controls and protection actions, which require low latency reactions.

Resumo: Os Sistemas de Controle Industrial (ICS) são responsáveis pelo controle de infraestruturas críticas que muitas vezes são alvos de ataques cibernéticos motivados por interesses políticos, militares ou financeiros. Os sistemas de Controle Supervisórios e de Aquisição de Dados (SCADA) estão entre os principais componentes dos ICS, são altamente interconectados e empregam soluções comuns a sistemas computacionais convencionais. Os *malware* estão entre as principais ameaças cibernéticas a esses sistemas. No entanto, testar a resiliência cibernética de um ICS real exige plataformas de testes e simulações para verificar o comportamento nocivo dessas ameaças sem colocar em risco o sistema original. Este artigo apresenta os resultados de um experimento que analisou o impacto de ataques com *malware* reais em um sistema sistema SCADA conectado a um modelo de sistema elétrico simulado em tempo real em uma plataforma de testes *hardware-in-the-loop*. Os resultados mostram que mesmo *malware* genéricos podem impactar a comunicação Modbus/TCP, causando interrupções e atrasos que podem prejudicar a operação do sistema SCADA. Este efeito pode afetar os controles do sistema elétrico e as ações de proteção que requerem reações de baixa latência.

Keywords: Cyber attacks; Real-time simulations; ICS; SCADA; Cyber-physical systems.
Palavras-chaves: Ciberataques, Simulações em tempo real; ICS; SCADA; Sistemas ciberfísicos.

1. INTRODUÇÃO

Os Sistemas de Controle Industrial (ICS) são responsáveis pelo controle de infraestruturas estratégicas críticas. Seu uso vem crescendo devido ao aumento dos processos de automação, o qual é acompanhado também do crescimento das ameaças cibernéticas relacionadas. O número de vulnerabilidades em componentes de ICS aumentou 29%, tendo como base informações obtidas do ano de 2017 para 2018. Dentre essas, mais da metade foi classificada com nota CVSS [1] igual ou superior a 7, indicando níveis altos ou críticos (Kaspersky Lab, 2020).

Um fato importante a ser considerado na segurança de sistemas industriais é que falhas na proteção de infraestruturas críticas podem trazer consequências desastrosas. Devido às suas características peculiares, apenas um ataque de negação de serviço tradicional pode ocasionar danos significativos (Nazir et al., 2017). Em função disso, defesas contra ameaças cibernéticas para ICS se tornaram

[1] `https://www.first.org/cvss`.

prioridades em estratégias nacionais de defesa de muitos países. No Brasil, a Doutrina de Defesa Cibernética 2014 (BRASIL, 2014) estabelece que falhas nas infraestruturas estratégicas constituem sérias ameaças à soberania nacional. Alinhado a essa diretriz, a Defesa Cibernética foi colocada como uma das três prioridades da Defesa Nacional.

No ambiente cibernético, os *malware* são considerados as principais ameaças encontradas. Estima-se que são coletadas mais de um milhão de novas amostras de *malware* por dia (Dell Secureworks, 2017).

As redes de processos dos ICS, que contêm servidores SCADA – Sistemas de Controle Supervisórios e de Aquisição de Dados, são muito heterogêneas, sendo encontrados sistemas operacionais antigos e desatualizados. A razão disso decorre dos seguintes fatos: nem sempre é possível interromper os processos para atualizar os sistemas; muitas aplicações específicas não podem ser portadas para novos sistemas operacionais; e, muitas vezes, não são conhecidos os efeitos da aplicação de atualizações ou da execução de aplicativos antivírus em redes de ICS.

Estudos de *malware* requerem ambientes controlados para análise e coleta de dados. Esses ambientes ou modelos devem representar da forma mais fidedigna possível os sistemas reais. Para isso, são utilizados os *Testbeds*, plataformas para testes de sistemas ciberfísicos. As principais vantagens do seu uso é a economia de meios e a não exposição do sistema original a riscos cibernéticos (Nazir et al., 2017).

Tendo em vista que alguns *malware* podem ter mecanismos de ativação de acordo com o ambiente encontrado, é possível que o comportamento de determinados *malware* não se manifeste completamente caso o ambiente de teste não seja o mais adequado ou mais próximo do ambiente original (Egele et al., 2012).

A análise de *malware* de um sistema industrial possui as suas peculiaridades inerentes à arquitetura complexa necessária para a reprodução dos experimentos com a fidelidade dos sistemas ciberfísicos reais, acompanhando os avanços no desenvolvimento de *malware*.

Este trabalho objetiva realizar análise do impacto de *malware* reais introduzidos em um sistema SCADA ligado a um simulador de eventos elétricos de tempo real e *hardware in the loop*

Os ensaios realizados foram conduzidos durante o comissionamento do Laboratório de Segurança Cibernéticas de Sistemas Ciberfísicos (LaSC) no Instituto Militar de Engenharia (IME) e na Fundação Parque Tecnológico de Itaipu – Brasil (Fundação PTI – BR). A implantação do LaSC visa disponibilizar uma infraestrutura que permita a simulação *hardware-in-the-loop* em tempo real de sistemas elétricos ciberfísicos submetidos a ataques de *malware* reais, para estudos e pesquisas no campo da segurança cibernética de sistemas que integram camadas de Tecnologia da Informação e Comunicação (TIC), de sistemas de automação e da sua própria dinâmica física.

2. REFERENCIAL TEÓRICO

2.1 Sistemas Industriais

Sistema de controle industrial é um termo genérico que engloba diversos tipos de sistemas de controle como os sistemas SCADA, Controladores Lógico-Programáveis (PLCs), Unidades Terminais Remotas (RTU) etc. Esses sistemas são comuns em instalações industriais e infraestruturas críticas (Stouffer et al., 2015). Os ICSs têm duas camadas de controle, sendo uma física (i) que compreende os sensores, atuadores e *hardware* como os PLCs que atuam fisicamente no sistema, por exemplo, abrindo comportas, regulando a tensão, pressão etc.; e uma camada cibernética (ii), que compreende os dispositivos de comunicação e informação juntos e seus *softwares* para adquirir dados, elaborar processos, estratégias e enviar comandos para a camada física (Genge et al., 2012).

Os Sistemas Industriais têm algumas características importantes, como a distribuição geográfica ampla, sincronização constante, interação entre infraestruturas lógicas e físicas em operação contínua e vida útil dos componentes do sistema elevada. Um mal funcionamento ou ataque são mais tangíveis em ICS que em sistemas de TIC em geral (Cherdantseva et al., 2016).

Os sistemas SCADA estão entre as principais ferramentas dos ICS (Nazir et al., 2017). Eles coletam e armazenam os dados provenientes dos PLCs/RTUs, gerenciando, em todo ou apenas em parte, os processos, sendo essenciais para automação e controle do processo industrial.

A especificidade dos sistemas SCADA faz com que algumas medidas de segurança comuns em TIC não possam ser aplicáveis, devido a suas caraterísticas de ininterrupção. Soluções de segurança tradicionais de TIC não podem ser diretamente utilizadas em ICS sem uma análise prévia das consequências. Como exemplo, pode-se citar que a criptografia e autenticação devem ser usadas com cautela para não causar interrupções inadmissíveis. Por outro lado, algumas opções como redes privadas virtuais (VPN) e *firewalls* são comumente adotadas com sucesso em sistemas SCADA (Cherdantseva et al., 2016).

A Figura 1 exibe um exemplo de arquitetura de sistema industrial completo. A camada superior contém os locais de acesso externo através da Internet. Conectados por meio de um *firewall*, estão os demais componentes do sistema. O grupo da rede corporativa compreende as estações de trabalho, impressoras e servidores de aplicação, dentre outros. Anexo à rede corporativa, encontra-se uma rede de perímetro desmilitarizada (DMZ), a qual compreende servidores que requerem acesso externo. Separada por outra camada de segurança, a Rede de Controle é composta por dispositivos de interface homem-máquina (HMI), estações de trabalho e servidores SCADA. Estes últimos se conectam com os dispositivos em campo, representados por Campo 1 a 3, através de alguma interface de comunicação qualquer, seja ela uma linha de telefone, enlace de micro-onda, rede de satélites, dentre outras. Por fim, os PLCs e RTUs serão conectados diretamente a dispositivos físicos industriais para monitorar ou alterar o estado dos mesmos.

A maioria dos protocolos SCADA não foram concebidos tendo segurança como prioridade. O protocolo Modbus, por exemplo, não verifica a integridade dos comandos enviados e também não implementa métodos de autenticação (Fovino et al., 2009).

2.2 Análise de Malware

Malware, de ***mal****icious soft****ware***, é um termo genérico para descrever todo tipo de programas de computador com fins maléficos (Ye et al., 2017). Esses programas são utilizados para diversos fins, como para obtenção de vantagens financeiras com o uso de *botnets*, por exemplo. Um exemplo recente é o *ransonware Wannacry*, que infectou mais de 300 mil dispositivos nas primeiras 24 horas após o seu lançamento (Dell Secureworks, 2017).

Existem basicamente duas categorias de análise de *malwares*. A estática, que analisa a ameaça sem executá-la e a dinâmica, que executa e coleta as ações que ela realiza.

Para a análise estática, diversas técnicas podem ser aplicadas em diferentes representações do programa como o executável binário, o código-fonte e gráficos de chamadas. Geralmente, o código-fonte não está disponível, limitando a aplicação de técnicas ao binário do *malware*. Ainda assim, seus desenvolvedores utilizam técnicas de automodificação, retornando resultados ambíguos e confusos ao se tentar converter o binário para linguagem *Assembly*.

Devido às peculiaridades dos *malware* a análise dinâmica deve ser realizada em um ambiente controlado, monitorando as mudanças que ocorrem no sistema de arquivos, registros, processos em andamento e tráfego de rede (Guarnieri et al., 2019).

2.3 Testbeds

Simulações são sempre importantes, sendo impraticável conduzir experimentos de segurança em sistemas reais, em função dos custos e possíveis riscos à estabilidade do sistema com tais experimentos. Algumas vulnerabilidades como ataques de negação de serviço podem causar interrupções e atrasos ou, até mesmo, a queda do sistema inteiro.

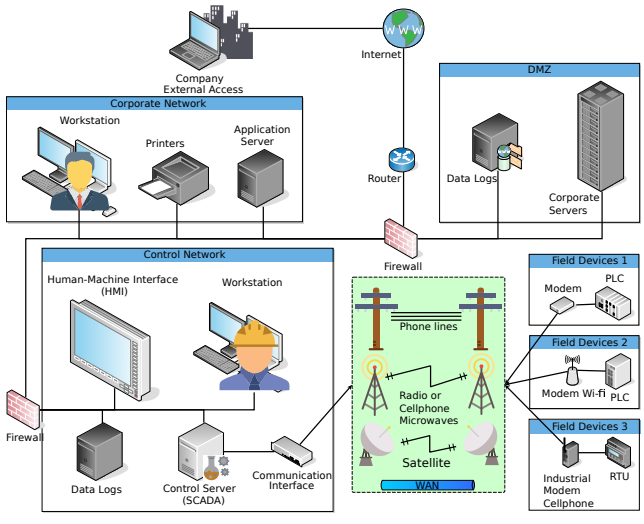

Figura 1. Arquitetura de um ICS-SCADA (Camargo, 2019).

3. ENSAIOS

Nesta seção, é discutida a abordagem implementada para análise de *malware* em sistemas industriais, com avaliação dos seus impactos na rede de comunicação. A arquitetura geral é descrita na Seção 3.1 e apresentada na Figura 2. Na Seção 3.2, são descritas as ferramentas utilizadas no sistema e, resumidamente, o motivo de cada escolha. Na Seção 3.4, são detalhadas as amostras utilizadas nos ensaios e, por fim, na Seção 4, é descrito como foram realizados os experimentos propriamente ditos e como foi feita a avaliação dos impactos na comunicação Modbus/TCP.

3.1 Arquitetura Geral

Na Figura 2, é apresentado um resumo da arquitetura do sistema implementado. Ele foi desenvolvido para ser hospedado em Linux e possui um virtualizador, o sistema de análise de *malware* e uma camada de processos real, com dados simulados pelo equipamento Real Time Digital Simulator (RTDS), cujos dados são transmitidos para o sistema SCADA na rede de controle virtualizada com máquinas *Windows*. Os binários submetidos a análise retornam *logs* de comportamento, arquivos com tráfego de rede e imagens de memória e mais inúmeros outros artefatos que podem ser inspecionados para realizar avaliação do impacto nas comunicações, verificar a detecção de *malware*, dentre outros fins úteis.

Ainda nesta figura, podem-se verificar as máquinas virtuais vinculadas ao VMWare: vCenter, responsável pelo gerenciamento do virtualizador; pfSense, responsável pelo firewall e roteamento das subredes do sistema; wintools, VM com o *software* RSCAD instalado, responsável por carregar o modelo elétrico no RTDS; labrtds, VM para uso de alunos de pós-graduação que usam o RTDS remotamente; cklasc, VM principal do sandbox com o *software* Cuckoo Sandbox e algumas ferramentas adicionais instaladas; ckextras, ferramentas auxiliares para funcionamento do Cuckoo Sandbox como bancos de dados, simulador de rede, entre outros. Também, representados na rede, estão os equipamentos: Real Time Digital Simulator (RTDS), equipamento físico de simulação de eventos elétricos digitais em tempo real que simula o modelo do sistema elétrico supervisionado e controlado pelo sistema SCADA e um rack com diversos equipamentos da Schweitzer Engineering Laboratories (SEL), equipamentos físicos com os relés de proteção que são conectados ao RTDS. Vale ressaltar que, neste ensaio, somente o RTDS foi utilizado ligado ao sistema SCADA.

3.2 Ferramentas Utilizadas

Virtualizador: O sistema todo foi virtualizado utilizando o sistema VMWare VSphere [2]. A escolha do Virtualizador foi feita por quatro principais motivos: o escasso tempo destinado ao cumprimento da missão aliado à *expertisse* do LaSC; o baixo custo das licenças para uso futuro tendo em vista o IME ser uma instituição de ensino; a ampla

[2] `https://www.vmware.com/br/products/vsphere.html`.

utilização da solução VMWare no meio corporativo, sendo consolidada como uma das melhores soluções em virtualização do mercado. Por ser uma solução importante no ambiente corporativo, esta acaba por ser alvo de *malware* modernos, entretanto, não é uma solução comum para análise da sua dinâmica, o que favorece a manifestação do seu comportamento.

Máquinas Virtuais: Na máquina ckextras, que disponibiliza ferramentas auxiliares para o funcionamento do sandbox, foi utilizada a versão Linux Debian 10 para seu uso, pela sua estabilidade e por ter ampla compatibilidade com as versões das ferramentas utilizadas.

Na máquina virtual que foi analisada, da rede de controle, que compreende o HMI e o servidor SCADA, foi utilizado o sistema *Microsoft Windows*. Esse sistema foi escolhido por representar mais de 35% dos sistemas operacionais em uso atualmente contra 0,7% do Linux (StatCounter Global Stats, 2019). Entre sistemas SCADA, não se tem conhecimento de uma estatística atual representativa.

Todas as máquinas virtuais permaneceram durante a análise em uma rede do tipo *host-only*. O roteamento foi feito através do *Cuckoo Sandbox*, o que permite definir rotas diferentes para cada análise, podendo ser por meio do simulador *InetSim* ou diretamente pela Internet. Neste caso, foi liberado o acesso completo à internet para os *malware*.

Nas máquinas a serem analisadas, foram desativados os sistemas básicos de segurança como *firewalls*, *anti-vírus* e foi desabilitado o controle de acesso do usuário (UAC), conforme recomendado na documentação do sandbox.

Sandbox para Análise Dinâmica de Malware: A ferramenta escolhida para análise dinâmica de *malware* foi o *Cuckoo Sandbox*. Esta ferramenta gera diversos *logs* sobre o comportamento do binário em execução como: tráfego de rede, registro de alteração de arquivos, modificação nos registros do sistema, imagens da memória RAM, dentre outros.

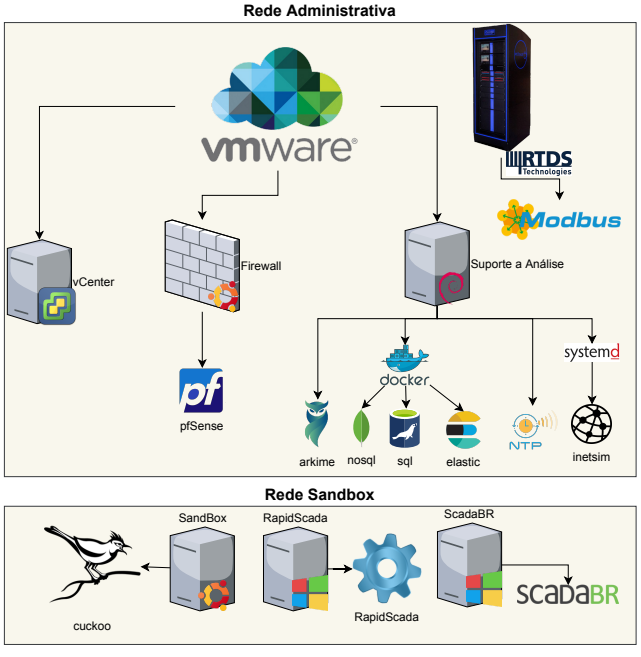

Figura 2. Arquitetura geral do sistema.

Outras ferramentas foram consideradas, conforme já explorado em (Camargo, 2019). Contudo, tais ferramentas, em sua maioria, não foram escolhidas por terem sido descontinuadas, limitarem a configuração do sistema ou por serem focadas em outras plataformas. Porém, além desses motivos, a necessidade de sigilo e a importância política, econômica e militar das infraestruturas a serem protegidas impedem o uso de soluções proprietárias, cujos dados, em qualquer momento, podem deixar de ser do controle de quem os está demandando.

Com relação à análise técnica da ferramenta, o *Cuckoo Sandbox* está em pleno desenvolvimento, é de fácil instalação e configuração e, por ser escrita em *python* com código aberto, permite customização. Esta é a ferramenta mais utilizada atualmente nos trabalhos acadêmicos relacionados a detecção de *malware* utilizando técnicas de aprendizado de máquina, como explorado em (Camargo, 2019). Trata-se de um referência na área acadêmica e corporativa, pois a solução líder de mercado para análise de Malware VirusTotal, adquirida pela Google em 2012, a utiliza para análise dinâmica [3].

No Sandbox, a máquina utilizada para controlar as simulações foi configurada com o sistema operacional Linux Ubuntu 18.04, que foi escolhido por ser inerte às ameaças compiladas para *Windows* 32 e 64 bits utilizadas como amostras, por estar em desenvolvimento ativo, com repositórios atualizados e, principalmente, por ter suporte amplo a todas as ferramentas adotadas, o que não se verifica nas versões subsequentes.

Apesar de cada análise ter um tempo definido, a ferramenta *Cuckoo Sandbox*, utilizada na sua versão 2.0.7, permite que esse tempo possa ser acelerado, no caso de possíveis atrasos decorrentes de métodos anti-análise, ou interrompidos prematuramente em caso de encerramento do processo principal.

A configuração de roteamento foi feita de modo a permitir acesso total a internet pelos *malware*, uma abordagem arriscada com o objetivo de obter maior manifestação de ações pelas amostras executadas.

Servidor SCADA: Diversos sistemas SCADA foram considerados e avaliados para o teste de comissionamento:

- *SCADA-LTS* [4]: uma ramificação do SCADABR na qual diversas funcionalidades foram inseridas; contudo, encontra-se ainda na versão beta e não recomendada para ambiente de produção conforme instruções oficiais, sendo necessárias várias adaptações manuais nos arquivos de configuração para que o mesmo funcione;
- *openScada* [5]: aplicação bem cotada na comunidade; possui código aberto e licença GNU GPL; o servidor funciona em Linux apenas e está em versão beta;

[3] https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors.
[4] https://github.com/SCADA-LTS/Scada-LTS.
[5] http://oscada.org/.

- SCADABR [6]: solução de fácil uso, com ampla documentação desenvolvida pela própria comunidade, além de possuir uma versão estável e possuir uma grande quantidade de clientes em produção no Brasil; possui alguns *bugs* que dificultam a análise dinâmica devido a problemas de sincronismo quando utilizado o protocolo ModbusTCP e há *bugs* não solucionados que impedem o uso do protocolo Modbus serial;
- *IndigoScada* [7]: projeto menor que o SCADABR mas bem completo, com licença livre, código aberto e documentação completa; o desenvolvimento e suporte da comunidade são consideravelmente menores do que o SCADABR e não há suporte em português;
- *Argos* [8], *FreeScada* [9], *pyScada* [10], *szarp* [11] são projetos mais simples, funcionam apenas para sistemas Linux, foram abandonados ou ainda estão em suas versões beta; e
- *RapidScada* [12]: é o projeto mais completo de código aberto atualmente e que funciona em plataformas *Windows*; é de fácil configuração e instalação com ampla documentação, extensos recursos e com versão estável; não funciona em *Windows XP* e não tem documentação em português.

De todas as soluções consideradas, foi selecionado o RapidScada como primeira opção e o SCADABR como opção alternativa, porém não chegou-se a utilizar a segunda opção no estudo em questão.

Foi utilizada a versão 5.8.3 para Windows 7 do *RapidScada* que foi configurado para utilizar uma linha de comunicação ModbusTCP. Na área de administração do sistema, foram criados: um objeto, uma linha de comunicação, um dispositivo, um canal de entrada e um de saída. O dispositivo foi configurado para obter dados Modbus da interface Modbus configurada no RTDS, que rodava em tempo real. O servidor SCADA foi configurado como "mestre", ou seja, ele é o responsável por enviar requisições para o dispositivo simulado e aguardar sua resposta.

A Figura 3 mostra o esquema final apresentado pelo HMI do servidor SCADA, que foi criado e configurado exclusivamente para este trabalho, como uma representação gráfica do controle das linhas elétricas do modelo simulado em tempo real no RTDS. Ressalta-se que essa imagem é do HMI do SCADABR, contudo é praticamente a mesma do RapidScada, mudando apenas a posição dos botões e alguns outros detalhes gráficos que não interferem no funcionamento do servidor SCADA.

Sistemas Auxiliares: Algumas ferramentas e tecnologias foram auxiliares ao desenvolvimento dessa plataforma:

- *Python*: como linguagem de programação para desenvolvimento de *scripts* e necessária para o funcionamento do sistema *Cuckoo Sandbox*;
- PyShark [13]: biblioteca *Python* para manipular arquivos .pcap utilizada na análise de impacto;

[6] https://www.scadabr.com.br/.
[7] https://sourceforge.net/projects/indigoscada/.
[8] http://www.cintal.com.ve/.
[9] https://sourceforge.net/projects/free-scada/.
[10] https://pypi.org/project/PyScada/.
[11] https://szarp.org/.
[12] https://rapidscada.org/.
[13] https://kiminewt.github.io/pyshark/.

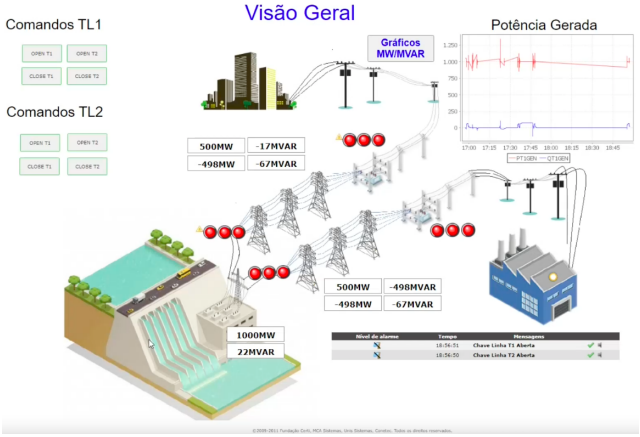

Figura 3. HMI do Sistema SCADA.

- Volatility: ferramenta de extração de dados em imagens de memória volátil;
- MariaDB e MongoDB: bancos de dados SQL e NoSQL respectivamente;
- Docker [14]: aplicação para abstração e virtualização em nível do sistema operacional para instalação de diversos serviços; e
- SAMBA: servidor para compartilhamento de arquivos em redes *Windows*.

Ferramentas Desenvolvidas: Foram utilizadas ferramentas desenvolvidas por Camargo (2019) para cumprir as tarefas-chave deste trabalho. Dentre elas, desta-se a principal ferramenta para manipular tráfego de rede coletado em formato .pcap, em *Python*, que analisa o mesmo e verifica se houve impacto na comunicação ModBus TCP durante aquela análise.

Além desta, foram criadas outras ferramentas de backup e gerenciamento do ambiente de simulação para automatizar as tarefas.

3.3 Modelo do Sistema Elétrico Simulado

O sistema dinâmico físico (sistema elétrico) é representado por modelos simulados em tempo real no RTDS. O objetivo é analisar os efeitos ciberfísicos de eventos cibernéticos por meio de simulação e, com isso, gerar proposta de contramedidas. Os ambientes interligados proporcionados pelo LaSC permitem também a conexão de outros equipamentos físicos comumente utilizados em campo na camada de Tecnologias de Automação (TA).

O modelo de sistema elétrico de potência utilizado nos ensaios é composto por uma unidade geradora (representando uma usina), uma subestação elevadora de tensão, duas linhas de transmissão com medições por meio de transformadores de tensão e corrente, dois disjuntores por linha e um equivalente dinâmico (geração-carga) de um sistema maior.

Essa configuração foi ilustrada de forma lúdica no sinótico do SCADA, conforme ilustrado na Figura 3. As indicações dos valores de variáveis do sistema e a sinalização do status de dispositivos são coletados da simulação em tempo real executada no RTDS.

[14] https://www.docker.com.

3.4 Amostras

Malwares Os *malware:* utilizados neste ensaio foram provenientes majoritariamente do repositório Virus Share [15], @ytisf [16] e do antigo VxHeaven [17]. Foram escolhidos 10 *malwares* e 10 goodwares aleatoriamente, do conjunto amostral dos experimentos realizados em (Camargo, 2019). Detalhes das amostras estão descritos na Tabela 1.

Tabela 1. Principais famílias de *Malwares* utilizados (Akbanov et al., 2019; McAfee Labs, 2017)

Análise	Família	md5
6	forclivnt	b89c7866a999404748fc5247dea6060f
12	hala	d85861cd64ae7419f68c0eb4e94b9d39
4	wannacrypt	84c82835a5d21bbcf75a61706d8ab549
9	vbinject	ecec4e416ab3e65948fbdecc15fd3da5
14	conficker	bd35d4d98fcbb1ec0e090fd2c631baa5
8	vbinject	e0e5ec97f635316e2c728d11368a547f
3	locky	b06d9dd17c69ed2ae75d9e40b2631b42
5	wannacrypt	3d34b5e5dd5b3e3876253b153f6d2308
7	sefnit	2254ca97c7625d720032551e3235ae80
16	veedna	4021d28f0eaa4f63e36edc60160db3b3

Goodware: Muitos dos trabalhos acadêmicos, bem como o trabalho de Camargo (2019), utilizaram como amostras benignas arquivos oriundos do sistema operacional, grande parte deles encontrados no diretório `C:/Windows/System32`. Entretanto, algumas amostras não executavam continuamente e tinham pouca similaridade com aplicativos de uso comum. Foram, então, selecionados apenas alguns aplicativos dessa fonte, sendo os demais coletados manualmente do site Major Geeks [18]. Os critérios de seleção foram: ter menos de 20 MB, não depender de instalação, preferencialmente realizar alguma operação ao iniciar, não exigir clique em caixas de diálogo para prosseguir a execução e não possuir nenhuma incompatibilidade com o sistema operacional. Foram coletados 10 *software* benignos, cujas amostras estão detalhadas na Tabela 2.

Tabela 2. Amostras de *Goodware* utilizadas.

Nr	Nome do arquivo	md5
17	bintext.exe	6a3d209ea00cdf67e2d2d1a721db65a6
26	ipnetinfo.exe	dda22b0cee0e7f742a5c4696e3b5a804
25	freecell0_68-0_68.exe	310c896c63be98c273d8a0c0c41609df
20	cleanmgr0_68-0_68.exe	2c4e4027e418eb4f0ed1e3793a4834df
28	ntsd0_67-0_67.exe	43c797488aed00ae5170b0531f8fc6e9
27	notepad0_66-0_66.exe	e8f945b7f0a938fedb44ae9996898f6c
18	BlueScreenView.exe	6126f1221d29712c069ee28ef4186e24
24	eventvwr0_60-0_60.exe	f636fd7e97ab17b8ff9d3ff593833301
23	DownTester.exe	4deefa52a7aadbab2db3288b23826ef5
22	ddeshare0_67-0_67.exe	4c7dc46c27d2bf288726bac3f8fe34ec

4. EXPERIMENTOS

Para avaliar a eficiência do sistema no cumprimento dos objetivos propostos, foram realizadas diversas análises de *malware* envolvendo todas as camadas de um sistema industrial, conforme apresentado na Figura 2 que apresenta a arquitetura completa do LaSC com todas as suas camadas que, por sua vez, correspondem às camadas de um sistema industrial padrão.

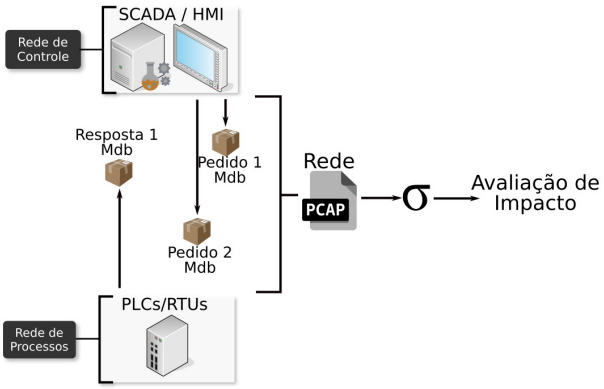

Figura 4. Esquema da análise de impactos.

No sistema SCADA, foram analisados os impactos na comunicação Modbus após a infecção com *malware* reais utilizando a mesma metodologia aplicada em (Camargo, 2019).

Foi feita uma análise com o objetivo de verificar os efeitos de *malware* na comunicação Modbus de um ICS. Todas as análises foram feitas com o tempo limite de 120 segundos e acesso irrestrito à Internet. Foram utilizados 10 *goodware* e 10 *malware*, totalizando 20 amostras.

Existem incontáveis maneiras de se avaliar um possível dano causado por um ataque cibernético a um ICS. Contudo, na literatura não há uma padronização de avaliação, dada a diversidade de sistemas físicos e cibernéticos existentes. Alguns autores mensuram os valores de alguns dispositivo físico, que demanda conhecimento específico sobre eles.

Neste experimento, o método de avaliação de impacto foi a mensuração do tempo entre pacotes ModbusTCP na comunicação entre o SCADA Master e o SCADA Escravo. Os tempo de todos os pacotes em cada análise foram submetidos a análises estatísticas para melhor definir se houve alterações.

A Figura 4 apresenta um esquema do funcionamento do experimento, partindo da análise dinâmica, coleta dos pacotes Modbus em formato .pcap, extração dos tempos e medidas estatísticas e, por fim, interpretação dos resultados.

O tráfego de rede capturado em cada análise foi armazenado em um arquivo no formato *pcap*. Cada um desses arquivos foi processado por um *script* responsável em localizar cada pacote Modbus e extrair duas métricas: o tempo entre um pacote de **requisição** feita pelo servidor SCADA mestre (Q) e outro de resposta enviada pelo dispositivo simulado SCADA Escravo (R) e o tempo entre duas **requisições** do servidor SCADA mestre (QQ).

[15] `http://www.virusshare.com`.
[16] `https://github.com/ytisf/theZoo/tree/master/malwares/Binaries`.
[17] `https://github.com/opsxcq/mirror-vxheaven.org/tree/master/vxheaven.org`.
[18] `https://www.majorgeeks.com/`.

Um exemplo das métricas extraídas encontra-se na Tabela 3, que apresenta quatro registros de pacotes ModBus nas primeiras 4 linhas, com o tempo de registro contado a partir do início da análise, e o tempo total, apresentado na Tabela 4, que durou entre a requisição 1 e 2, e entre a requisição 1 e resposta 1, em milissegundos.

Tabela 3. Métricas Extraídas.

Tempo (ms)	Pacote
1,21	Modbus ***Requisição*** *(Q)* 1
4,30	Modbus *Resposta (R)* 1
6,04	Modbus ***Requisição*** *(Q)* 2
8,33	Modbus *Resposta (R)* 2

Tabela 4. Métricas Calculadas.

Variação	Tempo (ms)
Tempo ***Requisição*** *(Q) 1-**Requisição** (Q) 2*	4,83 ms
Tempo ***Requisição*** *(Q) 1-Resposta (R) 1*	3,09 ms

O conjunto com os tempos (QQ) e (QR) para cada análise foi submetido à medida estatística do desvio padrão, o qual representa a dispersão dos dados em torno da média.

5. RESULTADOS

Os resultados coletados com as medidas estatísticas aplicadas, descritas na Seção 4, são apresentados na Tabela 5. A partir desses resultados, foram gerados os gráficos das Figuras 5 e 6.

Tabela 5. Análise de Impacto.

Anl	σ qr	σ qq
[6]	2,4915	0,0065
[12]	2,5677	0,0691
[4]	2,7146	0,0070
[9]	2,8649	0,0276
[14]	3,2242	0,0719
[8]	3,2469	0,0212
[3]	3,4621	0,0068
[5]	3,6181	0,0078
[7]	3,8647	0,0170
[16]	11,6677	0,0250
17	0,9013	0,0052
26	1,8902	0,0059
25	2,1318	0,0072
20	2,3985	0,0086
28	2,5548	0,0058
27	2,7013	0,0071
18	2,7137	0,0053
24	2,8295	0,0122
23	3,1506	0,0060
22	3,2404	0,0054

Legenda
qr: *query-response* (pedido-resposta) — σ: desvio padrão
qq: *query-query* (pedido-pedido) — [x]: análise de *malware*

A Figura 5 apresenta os valores do desvio padrão dos tempos entre duas requisições (QQ) Modbus feitas pelo sistema SCADA durante as análises realizadas com amostras de *goodware* e *malware*. É possível observar que, nas análises com *malware*, o desvio padrão foi visivelmente maior, indicando maior dispersão do tempo entre as requisições.

A Figura 6 apresenta os valores do desvio padrão dos tempos entre pacotes Modbus de pedido e resposta (QR) para amostras de *goodware* e *malware*, respectivamente.

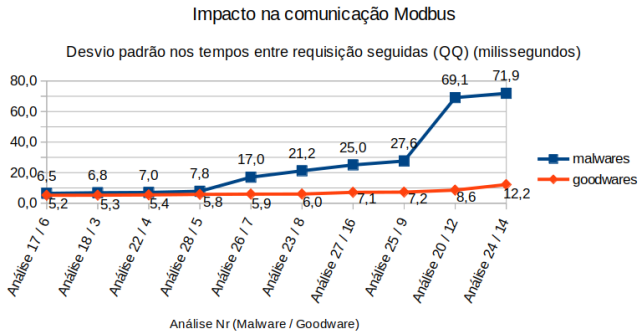

Figura 5. Desvio padrão dos grupos pedido-pedido (QQ).

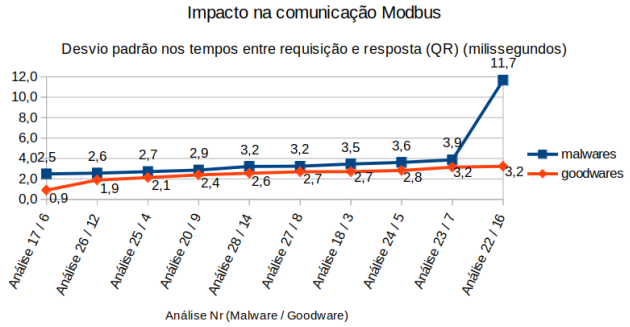

Figura 6. Desvio padrão dos grupos pedido-resposta (QR).

As análises com *malware* tiveram um desvio padrão visivelmente maior em parte das análises, enquanto que, com *goodware*, os valores foram mais baixos e com pouca variação.

O comportamento observado no desvio padrão se repete com as medidas de variância e amplitude, pois, de forma geral, elas também refletem a dispersão dos dados. Por isso, foram suprimidos os resultados de outras medidas estatísticas neste artigo.

Pode se concluir, pelos resultados dos experimentos realizados, que *malware* genéricos (não desenvolvidos especificamente para ICS) podem influenciar negativamente na comunicação Modbus/TCP, provocando atrasos e, possivelmente interrupções, que podem prejudicar a operação do sistema SCADA e impactar consideravelmente na operação do sistema elétrico, que requer reações de proteção de baixa resiliência, em nível de poucos milissegundos ou mesmo de microssegundos, em casos mais específicos. Tal resultado corrobora os resultados encontrados por Camargo (2019), que utilizou um simulador ModbusTCP por *software*. Contudo, a quantidade de ensaios realizadas foi limitada, devendo ser ampliada em futuros estudos dedicados a modelos de ICS de interesse, o que permitirá o alcance de resultados mais amplos e relevantes.

6. CONCLUSÃO

A análise de *malware* reais em sistemas industriais é capaz de mostrar vulnerabilidades e demandas de segurança desconhecidas. Caso ocorra um ataque, é importante ter a dimensão dos seus impactos. Testes com *malware* reais em sistema industriais não são comuns na literatura, contudo somente esse tipo de abordagem pode mostrar o verdadeiro impacto dos ataques.

Há um grande número de trabalhos que utilizam abordagens que se baseiam em componentes físicos e cibernéticos simulados por *software*. Estas são mais baratas e com quantidade razoável de recursos, contudo não suportam as funcionalidades chave para habilitar a experimentação com *malware* e SCADA reais. Sistemas de rede simulados podem modelar operações normais, mas falham em capturar a complexidade de interações entre *software*, *malware* e *hardware* reais. Por isso, foi criado o LaSC, através do qual realizaram-se os ensaios aqui relatados. O LaSC poderá ser adaptado e ampliado para simular eventos cibernéticos em outros tipos de sistemas ciberfísicos, como os sistemas de comunicações baseados em rádios definidos por *software*.

Visando auxiliar o preenchimento dessas lacunas, objetivou-se avaliar o impacto entre a comunicação de um servidor SCADA Mestre e um Escravo reais via simulação *hardware in the loop* em tempo real de um sistema elétrico.

Houve impacto significativo na comunicação quando comparadas análises entre *malware* e *goodware*. Com relação ao desvio padrão do tempo de duas requisições Modbus, observa-se que as amostras de *goodware* variaram pouco, aproximadamente 7 milissegundos entre os extremos. Por outro lado, as amostras com *malware*, em geral, provocaram maior sobrecarga da rede e consequentemente prejudicaram os tempos de resposta, chegando a ser superior a 13 vezes a amostra de menor valor de *goodware*. Com relação à avaliação da métrica entre um pedido e resposta Modbus, observou-se que a dispersão de tempo entre os diversos pacotes foi cerca de 40% maior no grupo com *malware*, chegando a ser também mais do que 13 vezes maior se comparado com os extremos entre *goodware* e *malware*. Sistemas industriais não são resistentes a grandes variações nos tempos de recebimento e envios de informações pelos sistemas SCADA e tais resultados possivelmente provocariam falhas nos sistemas de proteção elétrica.

É importante ressaltar que os esses ensaios foram executados apenas para verificar a comunicação por uma das vias no servidor SCADA e não incluíram os casos em que as aplicações impediam a comunicação entre o SCADA e os dispositivos de sensoriamento e controle do sistema eléetrico simulado em tempo real (RTDS). Contudo, tal efeito foi verificado em um ensaio realizado à parte por meio da aplicação do artefato *WannaCry*. Em determinado momento desse teste, o servidor SCADA passou a não responder aos comandos do operador, que ficou impedido, por exemplo, de operar os disjuntores das linhas de transmisão, mas continuou a receber dados do Modbus Slave RTDS.

Para futuros ensaios, sugere-se aumentar o tamanho das amostras e realizar análises com protocolos diferentes, assim como uma avaliação mais detalhada dos impactos com a utilização de novas métricas.

AGRADECIMENTOS

Este estudo é um resultado do Acordo de Parceria para PD&I nº EME 18-040-00/01 estabelecido entre a União, representada pelo Exército Brasileiro por intermédio do Departamento de Ciência e Tecnologia, a Itaipu Binacional e a Fundação Parque Tecnológico Itaipu – Brasil. Também houve apoio financeiro do Programa Estratégico de Defesa Cibernética do Exército Brasileiro ao Instituto Militar de Engenharia. Para apresentação deste artigo, foi recebido apoio da Escola de Aperfeiçoamento de Oficiais.

REFERÊNCIAS

Akbanov, M., Vassilakis, V.G., and Logothetis, M.D. (2019). WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms. *Journal of Telecommunications and Information Technology*, 1(1), 113–124. doi:10.26636/jtit.2019.130218.

BRASIL (2014). *Doutrina Militar de Defesa Cibernética - MD31- M-07*. Ministério da Defesa.

Camargo, O.A.M. (2019). *Impacto de Malwares Reais em Sistemas Industriais com Classificação Supervisionada Usando Aprendizado de Máquina*. mathesis, Instituto Militar de Engenharia. doi:10.13140/RG.2.2.29795.58407. URL `https://cutt.ly/maldomedoi`.

Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., and Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. *Computers and Security*, 56, 1–27. doi:10.1016/j.cose.2015.09.009.

Dell Secureworks (2017). State of Cybercrime Executive Summary. Technical report, Secureworks.

Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys*, 44(2), 1–42. doi:10.1145/2089125.2089126.

Fovino, I.N., Carcano, A., Masera, M., and Trombetta, A. (2009). An experimental investigation of malware attacks on SCADA systems. *International Journal of Critical Infrastructure Protection*, 2(4), 139–145. doi:10.1016/j.ijcip.2009.10.001.

Genge, B., Siaterlis, C., Fovino, I.N., and Masera, M. (2012). A cyber-physical experimentation environment for the security analysis of networked industrial control systems. *Computers & Electrical Engineering*, 38(5), 1146–1161. doi:10.1016/j.compeleceng.2012.06.015.

Guarnieri, C., Tanasi, A., Bremer, J., and Schloesser, M. (2019). Cuckoo Sandbox. URL `https://cuckoosandbox.org/`.

Kaspersky Lab (2020). Threat Landscape for Industrial Automation Systems. Technical report, Kaspersky.

McAfee Labs (2017). Threat Report: The WannaCry malware attack infected. Technical report, McAfee, Santa Clara, CA, USA.

Nazir, S., Patel, S., and Patel, D. (2017). Assessing and augmenting SCADA cyber security: A survey of techniques. *Computers and Security*, 70, 436–454. doi:10.1016/j.cose.2017.06.010.

StatCounter Global Stats (2019). Operating System Market Share Worldwide. URL `http://gs.statcounter.com/os-market-share`.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., and Hahn, A. (2015). Guide to Industrial Control Systems (ICS) Security. Technical report, National Institute of Standardsand Technology. doi:10.1103/PhysRevE.70.056118.

Ye, Y., Li, T., Adjeroh, D., and Iyengar, S.S. (2017). A Survey on Malware Detection Using Data Mining Techniques. *ACM Computing Surveys*, 50(3), 41:1–41:40. doi:10.1145/3073559.