Magdeburger Journal zur Sicherheitsforschung

Gegründet 2011 | ISSN: 2192-4260
Herausgegeben von Stefan Schumacher
Erschienen im Magdeburger Institut für Sicherheitsforschung
http://www.sicherheitsforschung-magdeburg.de/publikationen/journal.html

This article appears in the special edition »In Depth Security – Proceedings of the DeepSec Conferences«.
Edited by Stefan Schumacher and René Pfeiffer

Defense Informs Offense Improves Defense

How to Compromise an Industrial Control Systems Network – and How to Defend it

Joseph Slowik

ICS attacks have an aura of sophistication, high barriers to entry, and significant investment in time and resources. When looking at the situation from a defender's perspective, nothing could be further from the truth. Attacking and potentially taking down an ICS network requires - and probably operates best - via permutations of ›pen tester 101‹ actions combined with some knowledge of the environment and living off the land.

In this paper, we will explore some concrete ICS attack examples to explore just what is needed to breach and impact this environment. More importantly, using malware and data captured from recent attacks - specifically TRISIS and CRASHOVERRIDE - we'll see how the attackers ›messed up‹ their attacks and why a more simplified and direct approach to achieving offensive goals would not only be more effective, but likely far more difficult for defenders to catch as well. To close the conversation, we'll explore what defensive measures can be applied - and are necessary - to detect and stop such attacks in their tracks.

Keywords: ICS, Industrial Control Systems, TRISIS, CRASHOVERRIDE, malware, pentest, penetration testing, BLACKENERGY

Citation: Slowik, J. (2019). Defense informs offense improves defense: How to compromise an industrial control systems network – and how to defend it. *Magdeburger Journal zur Sicherheitsforschung*, *18*, 960–968. Retrieved September 15, 2019, from http://www.sicherheitsforschung-magdeburg.de/uploads/journal/MJS_066_Slowik_ICS.pdf

1 Introduction

Industrial control system (ICS) attacks grab headlines, rattle politicians, and scare observers. Yet underlying such attacks are increasingly commodity methods, shifted only at late stages to deploy custom, attack-specific malware. By understanding the nature of ICS attacks (of which there are thankfully few publicly known), network defenders and other stakeholders can gain appreciation for the necessary steps to respond to, defend against, or perform tests on such networks. At present, observers should expect continued interest in these systems either to maximize potential damage or deliver potent messages to witnessing populations. In response, those responsible for maintaining, defending, or testing the defenses of such networks must continue to observe the threat landscape and adapt accordingly to changes in adversary behaviors.

2 Defining ICS and Relevant Terms

ICS is frequently a term that follows the following format: »I know it when I see it, but I can't really define it.« This murkiness of definition results in an overly broad conception of what an industrial system truly is, leading to diffusion of effort and misconceptions on the actual attack space.

For the purposes of this paper, »Industrial Control Systems« are defined as: »A collection of personnel, hardware, and software that can affect or influence the safe, secure, and reliable operation of an industrial process.« [1] This definition encompasses traditional industrial systems and operations, while avoiding »Internet of Things« (IoT), building control systems, and other ancillary devices not directly related to industrial operations.

While some may view this as limiting, such circumscription allows for more focused, directed discussion of threats and responses. Additionally, given the exploding scope of IoT and related fields, expanding our definition to include such devices threatens »scope creep« in our analysis, introducing far too much »noise« from which we must derive a useful »signal«. For example, the burgeoning field of IoT devices provides an extensive array of technologies, services, and protocols of interest to security researchers, but distracts from the unique nature and implications of an industrial-focused attack, such as against electric power distribution.

On another point of specificity, for the scope of this paper »attack« is limited to the following definition: »the employment of cyberspace capabilities to destroy, deny, degrade, disrupt, deceive, corrupt, or usurp« the legitimate, intended operation of the targeted system [2].

3 History of ICS Attacks

Cyber-enabled ICS events are thankfully relatively few in number compared to the vast number of IT-based events. As shown in Figure 1, the number of ICS disruptive events is relatively small – only five publicly known events of interest – while those instances of attacks caused directly by malicious software (malware) even fewer still, at only three.

Prior to STUXNET's reveal, [3] the field of ICS security largely lived in the realm of theoretical possibilities and PhD theses. Since then, the number of events has slowly increased, from the discovery of HAVEX [4] and BLACKENERGY2 [5] as ICS-aware malware to actual disruptive events beginning with the BLACKENERGY3-enabled attack on Ukrainian electric distribution in 2015 [6].

Over time, ever-greater numbers of potential adversaries have become interested in this space, and the number and frequency of attacks, although still small, continues to increase [7]. While events continue to grab headlines when discovered – due to a combination of legitimate interest and aggressive vendor marketing – most public discussion and analysis focuses only on the final, observable impact of such events while paying little attention to how such attacks were executed.

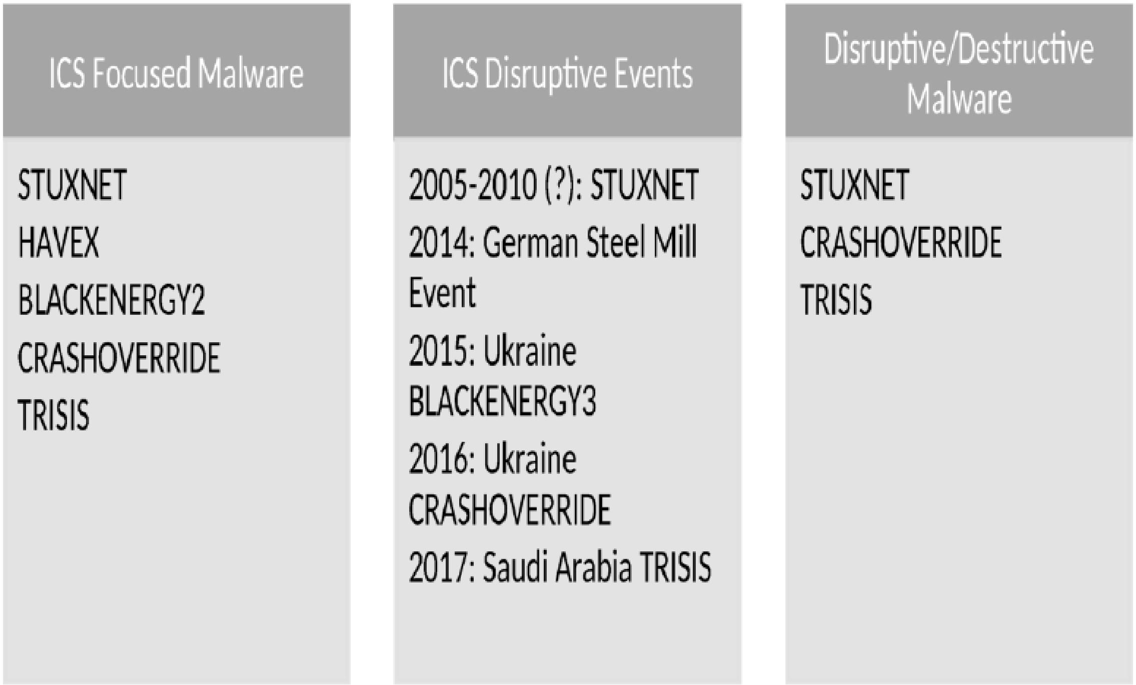
4 ICS Attack Tradecraft

ICS attacks do not manifest themselves as »bolts from the blue« where an adversary can pivot from initial intrusion to ICS disruption within minutes, or even hours. Rather, the design and architecture of modern networks means that even in those cases where control systems are (unfortunately) externally-accessible in some form, actual impacts require significantly greater access and knowledge of the controlled process to produce. Therefore, aside from some untargeted or autonomous infection events, such as a Wannacry infection propagation to ICS networks, [8] truly effective ICS intrusions require significant investment in time and resources to execute.

The best way to illustrate this concept is by referencing the series of interdependent steps required to execute such an intrusion. For this paper, the SANS ICS Cyber Kill Chain, shown in Figure 2, provides a reasonably accessible and accurate representation of what an ICS attacker must succeed in to successfully execute an attack [9].

Based on this model, observers should note several items:

1. ICS attacks are seldom (if ever) »direct«, but involve multiple operational stages from initial access before reaching ultimate objectives.
2. ICS intrusions typically must navigate the enterprise IT environment and identify mechanisms to pivot from IT networks to enclaved control system networks.



ICS Focused Malware	ICS Disruptive Events	Disruptive/Destructive Malware
STUXNET	2005-2010 (?): STUXNET	STUXNET
HAVEX	2014: German Steel Mill Event	CRASHOVERRIDE
BLACKENERGY2	2015: Ukraine BLACKENERGY3	TRISIS
CRASHOVERRIDE	2016: Ukraine CRASHOVERRIDE	
TRISIS	2017: Saudi Arabia TRISIS	

Figure 1: ICS Attacks and Disruptive Events

3. The required investment in time, resources, and effort to achieve these steps means that attack lifecycles are typically measured in months (or longer).

Thus, the popular conception of a network intrusion followed by the immediate delivery of an impact is not only inaccurate, but ignores the fundamental nature of an ICS-centered intrusion – or at least, an intrusion where the attacker wishes to maintain some level of control over events. Based on this view of attacks as a sequence of events over time rather than a sudden intrusion followed by effect, it is important for defenders (and those wishing to probe or otherwise test defenses) to note how this actually plays out in practice in light of recent intrusions.

5 ICS Attack Examples

As noted earlier, ICS attacks are thankfully few in number, but the previous three years have presented several informative examples from which defenders and other stakeholders can extract significant information. Most relevant to this discussion are the two major events from 2016 and 2017: CRASHOVERRDIE and TRISIS, respectively.

5.1 CRASHOVERRIDE

CRASHOVERRIDE, also referred to as Industroyer, was responsible for an electric distribution attack in 2016 that interrupted the flow of energy to consumers in parts of Kiev, Ukraine in 2016 [10] [11] [12]. Initial analysis and media attention focused on the final stage of this event: compromising electric distribution operations via purpose-built malware, the CRASHOVERRIDE framework. While this is academically interesting and represents a concerning development in malware evolution, focusing on the »final stage« of the attack obscured (if not outright ignored) the necessary prerequisites to execute the attack.

The final aspect of CRASHOVERRIDE focused on encoding ICS device manipulation in malware – essentially, changing the state of systems controlling breakers from »closed« (allowing power to flow) to »open« (interrupting the flow of electricity) – or more basically, altering a binary state between 0 and 1. While seemingly simple, CRASHOVERRIDE nonetheless succeeded in at least attempting to codify specific ICS protocol communication methods in software to enable this action – abstracting the specifics of the ICS system away from the individual executing or scheduling the attack.

In 2018, additional information on the attack became available enabling a deep-dive into exactly how CRASHOVERRIDE was executed through a long-running intrusion into the victim's network [13]. This further analysis revealed multiple items of interest for those seeking to either defend against or emulate such an attack. First, the overall »dwell time« of the incident was at least several months (initial intrusion no later than October 2016, with impact in late December 2016), and possibly up to a year. Second, while CRASHOVERRIDE itself represents a piece of custom malware purpose-build for the infection event, an analysis of log data and other artifacts associated with the intrusion indicate an absence of custom tools and software.

Rather than building and deploying custom mal-

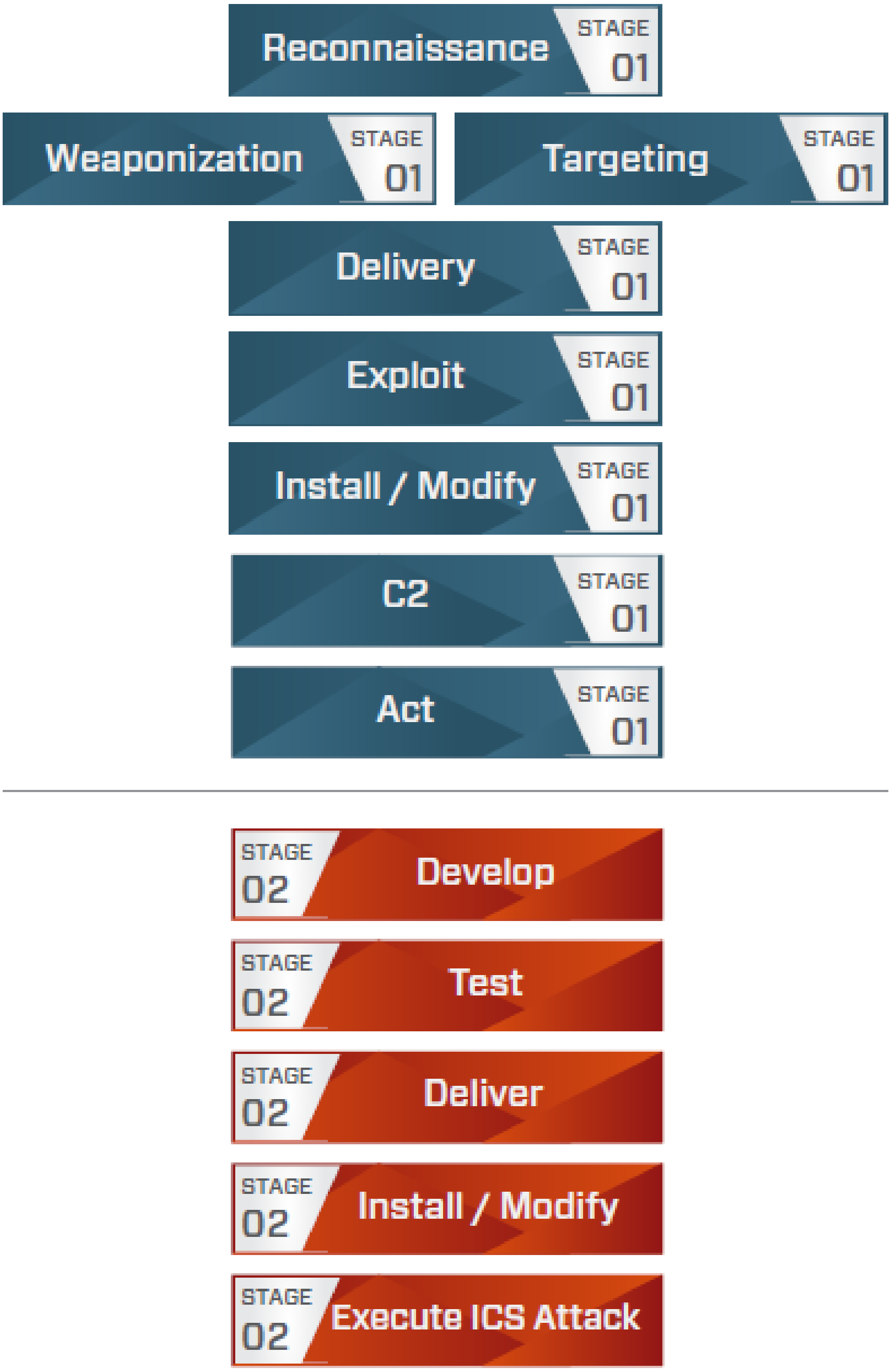


Figure 2: ICS Cyber Kill Chain

ware, the adversary responsible for CRASHOVERRIDE utilized a variety of techniques often associated with penetration tests and similar techniques. For example, the primary factor behind lateral movement and process execution within the victim environment was continuous credential capture and reuse via Mimikatz-based tools and native system commands such as »net use«, Windows scripting, and RDP connectivity. Rather than having to »break in« to achieve access from enterprise IT to the ICS network, the attacker simply harvested credentials and mimicked legitimate user activity to connect to the control system environment.

Once here, process execution and scheduling were facilitated by simple, unobfuscated scripts, service creation, and related techniques. Looking overall at the tradecraft deployed and some of the failures observed in specific ICS modules, one can hypothesize that the attack was immature in nature and that the adversary was »learning on the job« [14]. Irrespective of how »elegant« the attack may have been, it nonetheless resulted in physical process disruption executed via software, placing it in rarefied territory alongside STUXNET.

5.2 TRISIS

In mid-2017, an oil and gas processing facility in Saudi Arabia experienced a safety system fault that, after further analysis, was determined to be the result of unauthorized manipulation. Additional investigation revealed a software package built using Python designed to specifically interact with, and add additional functionality to, the precise Safety Instrumented System (SIS) deployed in the target environment. TRISIS, the rootkit installer responsible for interacting with a Schneider Electric Triconex device, was identified due to a fault caused in the victim SIS before any final attack or other manipulation could be executed [15] [16] [17].

As with CRASHOVERRIDE, significant initial attention and analysis focused on the final payload – TRISIS malware – rather than the mechanism through which the attacker was able to move into the victim environment to deploy this malware. Initially, almost no information on this phase of the event was publicly available, leaving most researchers and defenders unaware of just how this attack was executed.

Since initial reporting, multiple details emerged on the attack methodology leading up to TRISIS:

- Extensive credential harvesting via tools such as the ubiquitous Mimikatz.
- Use of publicly- or commercially-available post-exploitation frameworks such as WMImplant or CobaltStrike.
- Several lightly-customized versions of publicly-available software for remote access, enumeration, and other functionality.
- ICS network access achieved through capturing credentials for VPN links.

Aside from the introduction of various »hack tools« such as Cobalt Strike and WMImplant, the attack resembles CRASHOVERRIDE in its emphasis on credential theft as an enabling factor for subsequent activity. Furthermore, execution of the actual ICS attack portion of the event requires similar actions and prerequisites as CRASHOVERRIDE: leveraging classic IT intrusion tools to get access to the »last step« before interfacing with control system equipment, and deploying custom malware tuned to the victim environment to take care of environmental manipulation or disruption – with no requirement from the operator aside from placing the binary in the right location and executing it. Finally, from a »dwell time« perspective, discussions with incident responders at the victim organization indicated a timeline of several months to potentially over a year, allowing the attacker to enumerate and gather information on the target network – including such information as the precise manufacturer and firmware revision for the SIS device.

Overall, the attack represented initial actions leveraging relatively common or built-in system tools to enable intrusion and lateral movement, followed by the deployment of purpose-built malware designed to impact the victim environment. While CRASHOVERRIDE featured limited replay capability for other environments using equipment with the same communication protocols, TRISIS is more circumscribed in that it will only function against the precise equipment (a Schneider Electric Triconex device) with a specific firmware revision. As such, the attack is very much designed only for a particular environment and features little (if any) replay capability against other networks.

5.3 Lessons Learned from Attacks

Two clear lessons emerge from these attacks. First, each event features a distinct bifurcation aligning with operations along the kill chain, distinguishing initial access and final disruptive attack stages. Second, whereas previous ICS events – from HAVEX to the 2015 Ukraine event – »front-load« efforts with complex or at least custom toolsets, both TRISIS and CRASHOVERRIDE avoided the use of custom, actor-specific malware until the very end of the events in question.

Both observations are linked, and tie in to the concept of being no more complex or technically »artful« than necessary to achieve objectives. As these attacks were constructed and executed, adversaries relied on commonalities in IT and ICS environments (prevailing network weaknesses, single factor authentication schema, and the ubiquity of Windows systems) to facilitate intrusion using well-known, largely publicly-available capabilities.

At the final, execution stage of these events, methodology changes and purpose-built, custom software emerges to deliver an effect. What is most interesting in these cases is that unlike past events where individuals directly and manually manipulated control system equipment (such as Ukraine 2015), the actual ICS impact and alterations are instead codified into software packages designed to change breaker settings and upload new functionality in the cases of CRASHOVERRIDE and TRISIS, respectively.

This change in operation from manual to semi-automated (or at least automated in terms of ICS manipulation and exploitation logic) represents a sea-change in behavior form all prior ICS-related events excepting STUXNET. While there are many aspects of both CRASHOVERRIDE and TRISIS that were unsuccessful or immature (indeed, it is worth questioning whether these events were truly successful at all given either unintended or limited consequences of execution), both events indicate a new development in the division of labor and operator-level knowledge for executing ICS-focused intrusions.

Essentially, those individuals that are executing attacks no longer need to know much beyond standard IT intrusion tradecraft for attack pre-positioning. Once at an appropriate stage where control system equipment or related systems are accessible, specialized tools – with some measure of flexibility (CRASHOVERRIDE) or purpose-built for the victim environment (TRISIS) – can be deployed to execute all desired or necessary ICS operations. In this division of labor approach, a single developmental team or research lab housing ICS expertise can support multiple operational teams whose only remit is delivering tools to the appropriate targets.

6 Future Expectations

The above examples highlight a transition in tradecraft over the previous five years as operations become more automated in ICS capabilities while leveraging the continued »IT-ification« of multiple aspects of control system environments to facilitate initial access and lateral movement [18]. Consequently, an attack bifurcation emerges: standard Windows-centric tradecraft enables intrusions, which are concluded by tools built by subject matter experts divorced from the interactive portion of operations.

From this observation, several implications become clear. First, tooling development and application shifts from early kill chain activity (2015 Ukraine) to later kill chain stages. Second, the nature of initial tools and applications used to facilitate an intrusion increasingly resemble tactics, techniques, and procedures available to a large category of adversaries, making attribution and assessment of adversary intention difficult when identifying only early-stage events. Third, ICS-specific capabilities increasingly reside in specialist environments developing tools and software for use by others, allowing for operations to more effectively scale given the relative scarcity of ICS domain knowledge.

In attempting to divine future events from current observations, one must note that adversaries will not significantly evolve from current trends unless or until defenders create sufficient obstacles for attackers to prompt them to shift. Absent such stimulus, defenders should anticipate continued evolution of attacker tradecraft following the examples provided by CRASHOVERRIDE and TRISIS: initial attack stages using widely-available, deniable, and relatively easy to use techniques, followed by final-stage ICS operations leveraging purpose-built tools that will abstract ICS interaction away from personnel directly involved in the operation.

6.1 Implications for Offense

When viewed from an offense-focused perspective, the above trends and observations provide a readily-accessible blueprint for new (and potentially less-capable) adversaries to launch disruptive or destructive ICS operations. Essentially, many of the initial stages and prerequisites for launching such an attack – compromising the enterprise IT network then pivoting to the ICS environment to identify points of contact with control system equipment – are now equivalent to general IT network intrusions in many respects. In this fashion, the »barriers to entry« to gaining access to control system environments, once nontrivial when legitimate air gaps existed, and many systems were bespoke, nonstandard devices, have eroded due to an increasing IT-OT convergence.

To transition from merely gaining access presents a greater, but not insurmountable, difficulty. The lesson drawn from recent events is that attackers need not be control system experts, or even possess much control system knowledge, to execute an ICS-focused attack. Instead, they merely need to gain access to communicate with devices controlling industrial processes and can deploy software or related tools to undertake such operations. For state-sponsored entities, development of such capabilities can be outsourced to entities such as research laboratories, universities, or even contracted private entities to perform tool and exploit development.

For independent adversaries and pentesters, nothing quite like a »Metasploit for ICS« yet exists, but increasingly tools and capabilities trickle down from better financed entities or university researchers into commodity tools: from publicly-available proof of concept code to commercial offerings such as the Gleg Pack ICS-specific offerings [19]. While work is still required to translate what is available into specific effects packages, expertise once residing in only a handful of individuals can now be found in Python code in Github (including the TRISIS codebase) [20].

Overall, we much expect the number of entities capable of launching and completing an ICS-focused disruptive event to grow. Whereas previously such capability resided in only a few select entities, consisting of several well-financed government entities and »rockstar« researchers, now a reasonably effective division of labor model for executing attacks is available to enable less-sophisticated entities to carry out such actions.

6.2 Implications for Defense

ICS defenders once could rely on a combination of obscurity and environmental specificity to make their networks either inaccessible or conceptually impenetrable to attackers. While the post-STUXNET series of events, from HAVEX to BLACKENERGY2 to BLACKENERGY3, all provided examples of how these networks could be breached, many still assumed that the conceptual and technical barriers to actual, intentional disruptive events (as opposed to merely accessing such networks) were sufficiently high to favor defense and make attacks a largely notional, theoretical concept.

The lessons from CRASHOVERRIDE and TRISIS thus serve as a shrill wake-up call for ICS asset owners and defenders as such capabilities have proliferated while the level of difficulty to at least gain access to ICS environments has dramatically decreased. While developing ICS exploits and attack tools remains a sophisticated enterprise, the division of labor and attack bifurcation approach presents a model that can enable attackers to either finance or take advantage of ICS specialists to devise tools for their use, while possessing little ICS understanding of their own.

Against this backdrop, ICS networks have barely moved forward defensively as adversary evolution has accelerated. Defensive visibility remains limited to network traffic, and then only at certain key nodes, while adversaries have dramatically proven the utility of native system tools and protocols to prosecute an attack. Absent more extensive network visibility and, most importantly, host-based visibility, recent adversary tradecraft merely blends in with system operations at best or is invisible to defenders at worst.

From an ICS-specific standpoint, defenders rely on tools such as antivirus to detect malicious programs, when the purpose-built tools used for ICS manipulation (such as CRASHOVERRIDE and TRISIS) retain few features found in typical Windows-focused malware. As a result, such tools can pass undetected even if all binaries entering the sensitive environment are analyzed by security software. Furthermore, defenders have not yet utilized tools and data sources already at their disposal – such as process-based data from ICS historians – to gain greater visibility into environments, which can be used in conjunction with security-focused monitoring to identify ICS-specific suspicious behavior.

Overall, the implications for defense are quite simple: attacker tradecraft has advanced in ways that make attacks more likely and easier to execute, while defenders have not increased visibility and awareness of environments to meet these challenges. To address this issue, defenders must first and foremost increase visibility into both IT and ICS networks to catch the methodologies deployed by current adversaries. This includes visibility into command line execution, script framework logging (especially PowerShell), and capturing information from Windows-focused frameworks such as WMI. Absent such changes, adversaries can and will continue to hide in legitimate system activity to evade detection.

Once the visibility issue is addressed, defenders can then move on to the analysis and detection phase to counter threats. Simple IOC-based identification and blocking will fail in this respect, as adversaries will use malicious methods more so than malicious software, and the only items amenable to an IOC approach – the final ICS attack payload such as a CRASHOVERRIDE or TRISIS – will almost certainly be purpose-built for the environment in question, thus never seen before nor to be seen again. Thus, defenders must focus on ways to identify malicious behaviors, those collections of observations that define how an adversary would act to breach or otherwise disrupt a control system network. This requires the ability to take collected data across the three major sources for control system environments – network, host, and process data – and correlate observations to identify those collections of activities associated with malicious actions on objectives.

7 Conclusion

ICS-focused disruptive events entered a new phase of development and execution starting in 2016 with the discovery of CRASHOVERRIDE as the mechanism causing the 2016 Ukraine power event. Combined with TRISIS the following year, these events showed a distinct shift toward automating ICS interaction while deploying generic – but effective – tradecraft to pre-position and gain access for delivering a disruptive event. The implications for attacks are that we should anticipate more of them as ICS-specific knowledge and capability increasingly resides in dedicated teams capable of supporting multiple operations through research and tool development, while initial access methodology requires a set of skills easily attainable by multiple entities. From a defensive perspective, continued lack of visibility let alone event correlation to identify malicious behaviors will continue to place defenders at a disadvantage relative to attackers, requiring significant investment in improving network, host, and process-centric visibility and analysis. Overall, the ICS threat landscape continues to grow in terms of number of adversaries, requiring dedicated efforts to build up defensive capabilities to meet an ever-growing challenge.

8 About the Author

Joe Slowik currently hunts ICS adversaries for Dragos, pursuing threat activity groups through their malware, their communications, and any other data available. Prior to his time at Dragos, Joe ran the Incident Response team at Los Alamos National Laboratory, and served as an Information Warfare Officer in the US Navy. Throughout his career in network defense, Joe has consistently worked to »take the fight to the adversary« by applying forward-looking, active defense measures to constantly keep threat actors off balance. An important part of this strategy is understanding adversary techniques and actions: good defense requires knowing (and at times practicing) offense.

9 References

1. M. Assante and T. Conway, »An Abbreviated History of Automation & Industrial Controls Systems and Cybersecurity,« August 2014. [Online]. Available: https://ics.sans.org/media/An-Abbreviated-History-of-Automation-and-ICS-Cybersecurity.pdf. [Accessed 02 11 2018].

2. AFLCMC/HNJG, »Broad Agency Announcement (BAA ESC 12-0011): Cyberspace Warfare Operations Capabilities (CWOC) Technology Concept Demonstrations,« United States Department of the Air Force, San Antonio, 2012.

3. K. Zetter, »An Unprecedented Look at STUXNET, the World's First Digital Weapon,« Wired, 03 November 2014. [Online]. Available: https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/. [Accessed 04 November 2018].

4. Kaspersky Lab Global Research and Analysis Team, »Energetic Bear - Crouching Yeti,« Kaspersky, July 2014. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08080817/EB-YetiJuly2014-Public.pdf. [Accessed 04 November 2018].

5. US-CERT/ICS-CERT, »Ongoing Sophisticated Malware Campaign Compromising ICS (Update E),« US-CERT, 10 December 2014. [Online]. Available: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-281-01B. [Accessed 04 November 2018].

6. R. M. Lee, M. J. Assante and T. Conway, »Analysis of the Cyber Attack on the Ukrainian Power Grid,« 18 March 2016. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf. [Accessed 04 November 2018].

7. J. Slowik, »Evolution of ICS Attacks and the Prospects for Future Disruptive Events,« 25 February 2019. [Online]. Available: https://dragos.com/wp-content/uploads/Evolution-of-ICS-Attacks-and-the-Prospects-for-Future-Disruptive-Events-Joseph-Slowik-1.pdf. [Accessed 04 March 2019].

8. M. Assante, J. Slowik and B. Miller, »Defending the ICS Ahead of the Patch: WannaCry Lessons Learned,« SANS Institute, 26 May 2017. [Online]. Available: https://www.sans.org/webcasts/defending-ics-patch-wannacry-lessons-learned-105175. [Accessed 04 March 2019].

9. M. J. Assante and R. M. Lee, »The Industrial Control System Cyber Kill Chain,« October 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297. [Accessed November 05 2018].

10. A. Greenberg, »Crash Override: The Malware that Took Down a Power Grid,« Wired, 12 June 2017. [Online]. Available: https://www.wired.com/story/crash-override-malware/. [Accessed 04 November 2018].

11. Dragos Inc., »CRASHOVERRIDE: Analysis of the Threat to Electric Grid Operations,« 13 June 2018. [Online]. Available: https://dragos.com/wp-content/uploads/CrashOverride-01.pdf. [Accessed 04 November 2018].

12. A. Cherepanov, »WIN32/Industroyer A New Threat for Industrial Control Systems,« 12 June 2017. [Online]. Available: https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32_Industroyer.pdf. [Accessed 04 November 2018].

13. J. Slowik, »Anatomy of an Attack: Detecting and Defeating CRASHOVERRIDE,« 12 October 2018. [Online]. Available: https://dragos.com/wp-content/uploads/CRASHOVERRIDE2018.pdf. [Accessed 04 November 2018].

14. J. Slowik, »CRASHOVERRIDE: When ›Advanced‹ Actors Look Like Amateurs,« 03 November 2018. [Online]. Available: https://pylos.co/2018/11/03/crashoverride-when-advanced-actors-look-like-amateurs/. [Accessed 01 March 2019].

15. Dragos Inc., »TRISIS Malware: Analysis of Safety System Targeted Malware,« 13 December 2017. [Online]. Available: https://dragos.com/wp-content/uploads/TRISIS-01.pdf. [Accessed 04 November 2018].

16. B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker and C. Glyer, »Attackers Deploy New ICS Attack Framework ›TRITON‹ and Cause Operational Disruption to Critical Infrastructure,« FireEye, 14 December 2017. [Online]. Available: https://www.fireeye.com/blog/threat-

research/2017/12/attackers-deploy-new-ics-attack-framework-triton.html. [Accessed 04 November 2018].

17. C. Bing, »Trisis has the Security World Spooked, Stumped, and Searching for Answers,« CyberScoop, 16 January 2018. [Online]. Available: https://www.cyberscoop.com/trisis-ics-malware-saudi-arabia/. [Accessed 04 November 2018].

18. K. Brocklehurst, »IT-OT Convergence: Who Owns ICS Security?«, Automation.com, 20 May 2017. [Online]. Available: https://www.automation.com/automation-news/article/it-ot-convergence-and-conflict-who-owns-ics-security. [Accessed 03 March 2019].

19. Gleg, »Gleg SCADA+,« Gleg, 01 January 2018. [Online]. Available: http://gleg.net/agora_scada.shtml. [Accessed 03 March 2019].

20. MDudek-ICS, »TRISIS-TRITON-HATMAN,« Github, 27 November 2018. [Online]. Available: https://github.com/MDudek-ICS/TRISIS-TRITON-HATMAN. [Accessed 03 March 2019].