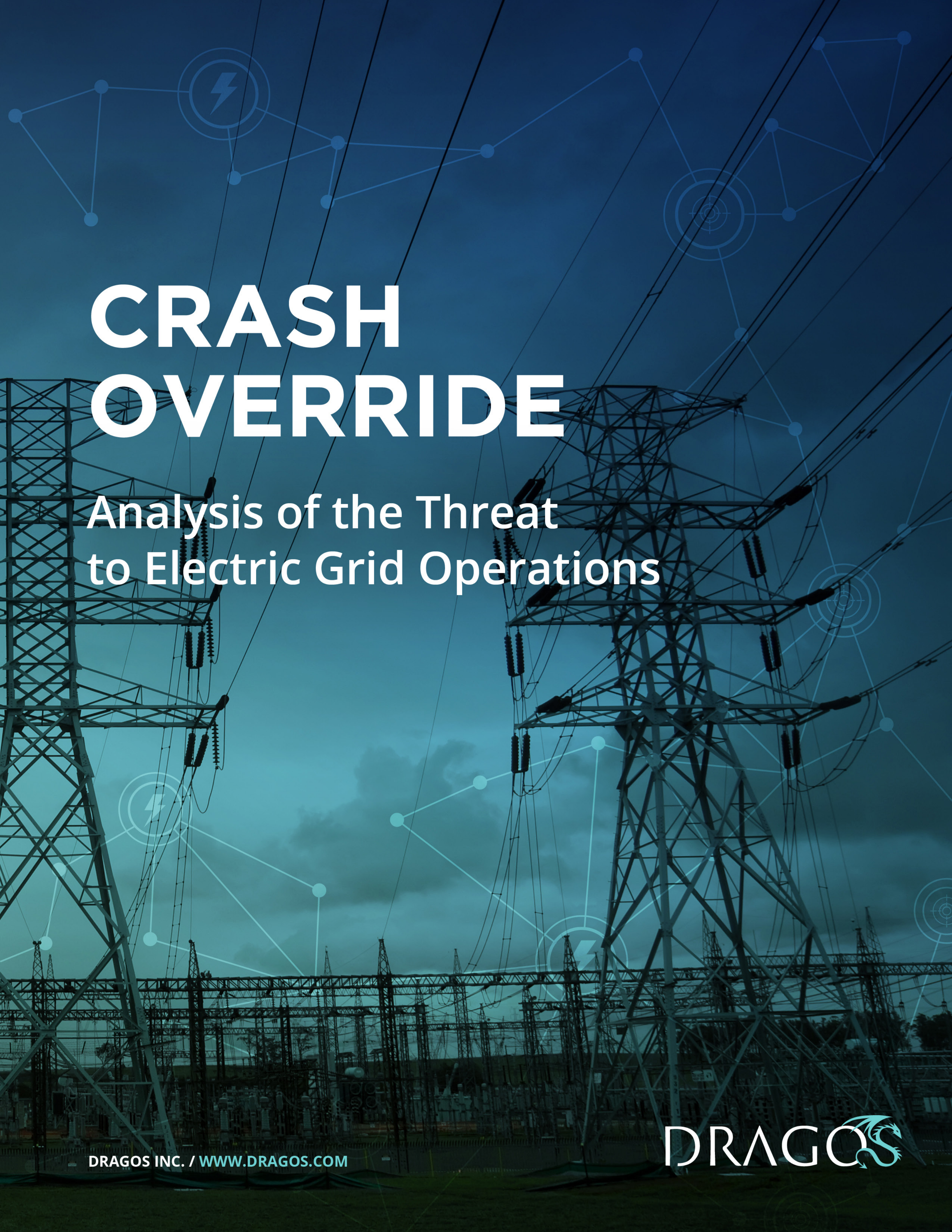
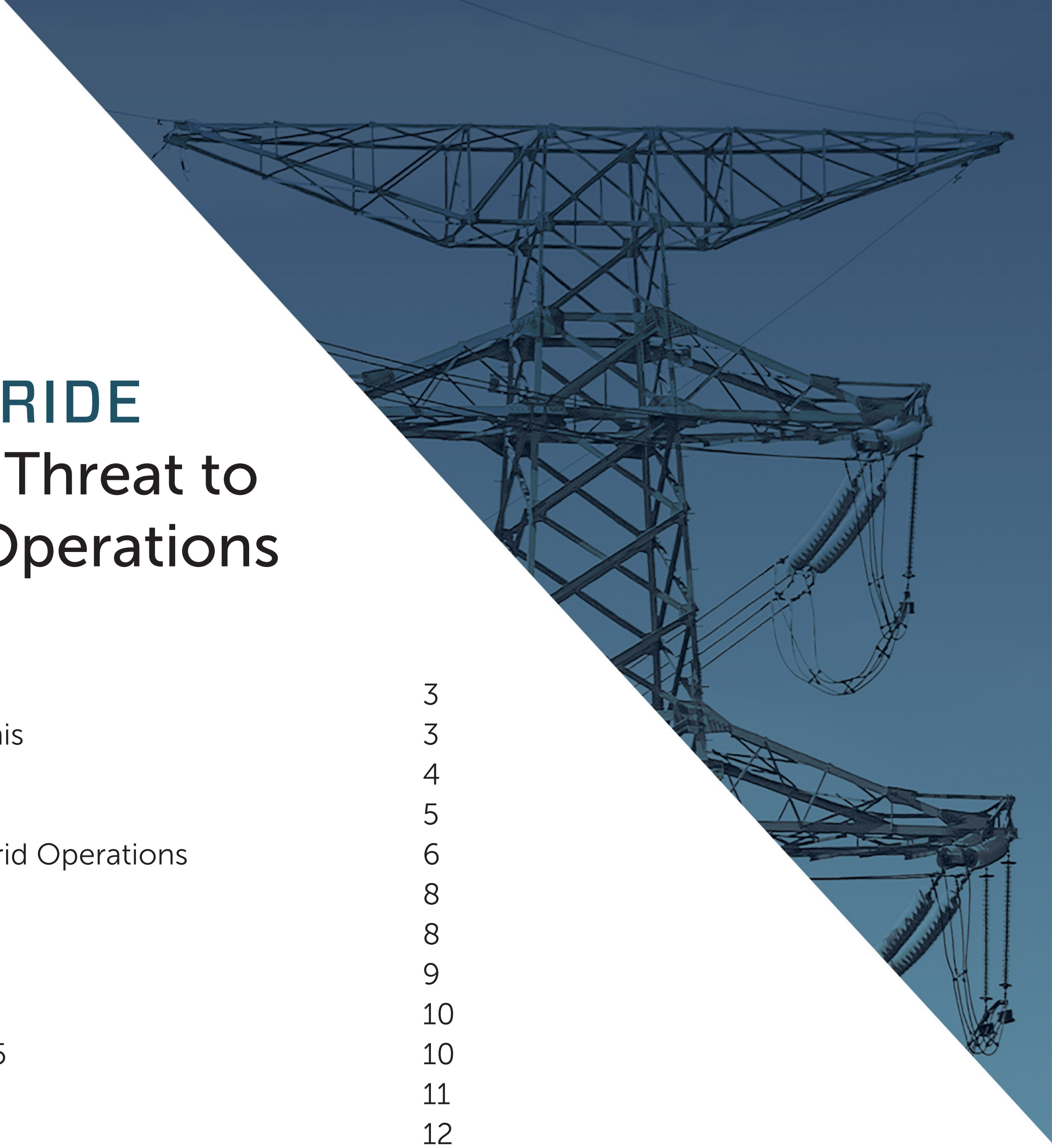


CRASH OVERRIDE

Analysis of the Threat to Electric Grid Operations

DRAGOS INC. / WWW.DRAGOS.COM



CRASHOVERRIDE

Analyzing the Threat to Electric Grid Operations

Contents

Executive Summary

Dragos, Inc. was notified by the Slovakian anti-virus firm ESET of an ICS tailored malware on June 8th, 2017. The Dragos team was able to use this notification to find samples of the malware, identify new functionality and impact scenarios, and confirm that this was the malware employed in the December 17th, 2016 cyber-attack on the Kiev, Ukraine transmission substation which resulted in electric grid operations impact. This report serves as an industry report to inform the electric sector and security community of the potential implications of this malware and the appropriate details to have a nuanced discussion.

Why Are We Publishing This

Security firms must always balance a need to inform the public against empowering adversaries with feedback on how they are being detected and analyzed. This case is no different. In fact, it is more important given that there is no simple fix as the capability described in this report takes advantage of the knowledge of electric grid systems. It is not an aspect of technical vulnerability and exploitation. It cannot just be patched or architected away although the electric grid is entirely defensible. Human defenders leveraging an active defense such as hunting and responding internally to the industrial control system (ICS) networks can ensure that security is maintained.

Key Takeaways

- The malware self-identifies as “crash” in multiple locations thus leading to the naming convention “CRASHOVERRIDE” for the malware framework.
- CRASHOVERRIDE is the first ever malware framework designed and deployed to attack electric grids.
- CRASHOVERRIDE is the fourth ever piece of ICS-tailored malware (STUXNET, BLACKENERGY 2, and HAVEX were the first three) used against targets and the second ever to be designed and deployed for disrupting physical industrial processes (STUXNET was the first).
- CRASHOVERRIDE is not unique to any particular vendor or configuration and instead leverages knowledge of grid operations and network communications to cause impact; in that way, it can be immediately re-purposed in Europe and portions of the Middle East and Asia.
- CRASHOVERRIDE is extensible and with a small amount of tailoring such as the inclusion of a DNP3 protocol stack would also be effective in the North American grid.
- CRASHOVERRIDE could be leveraged at multiple sites simultaneously, but the scenario is not cataclysmic and would result in hours, potentially a few days, of outages, not weeks or more.
- Dragos assesses with high confidence that the same malware was used in the cyber-attack to de-energize a transmission substation on December 17, 2016, resulting in outages for an unspecified number of customers.
- The functionality in the CRASHOVERRIDE framework serves no espionage purpose and the only real feature of the malware is for attacks which would lead to electric outages.
- CRASHOVERRIDE could be extended to other industries with additional protocol modules, but the adversaries have not demonstrated the knowledge of other physical industrial processes to be able to make that assessment anything other than a hypothetical at this point and protocol changes alone would be insufficient.
- Dragos, Inc. tracks the adversary group behind CRASHOVERRIDE as ELECTRUM and assesses with high confidence through confidential sources that ELECTRUM has direct ties to the Sandworm team. Our intelligence ICS WorldView customers have received a comprehensive report and this industry report will not get into sensitive technical details but instead focus on information needed for defense and impact awareness.

Background

On June 8th, 2017 the Slovak anti-virus firm ESET shared a subset of digital hashes of the malware described below and a portion of their analysis with Dragos. The Dragos team was asked to validate ESET's findings to news publications ESET had contacted about the story which would be published June 12th, 2017. Dragos would like to thank ESET for sharing the digital hashes which allowed the Dragos team to spawn its investigation. Without control of the timeline, it was Dragos' desire to publish a report alongside ESET's report to capture the nuance of electric grid operations. The report also contains new discoveries, indicators, and implications of the tradecraft. Also, because of the connection to the activity group Dragos tracks as ELECTRUM, it was our decision that an independent report was warranted. The Dragos team has been busy over the last 96 hours reproducing and verifying ESET's analysis, hunting and analyzing new samples of the malware, identifying additional infections, notifying appropriate companies, and informing our customers. Importantly, Dragos also updated ICS vendors that needed to be made aware of this capability, relevant government agencies, many national computer emergency response teams (CERTs), and key players in the electric energy community. Our many thanks to those involved.

If you are a Dragos, Inc. customer, you will have already received the more concise and technically in-depth intelligence report. It will be accompanied by follow-on reports, and the Dragos team will keep you up-to-date as things evolve. It is in Dragos' view that the following report contains significant assessments that deserve a wide audience in the electric sector. Avoiding hype and fear should always be paramount but this case-study is of immediate significance, and this is not a singular contained event. The CRASHOVERRIDE capability is purpose built to impact electric grid operations and has been created as a framework to facilitate the impact of electric grids in other countries in the future outside the attack that took place with it December 17th, 2016 in Ukraine. However, as always, the defense is doable.

Introduction to Electric Grid Operations

As with most ICS specific incidents, the most interesting components of the attack are in how the adversary has demonstrated they understand the physical industrial process. Whereas vulnerabilities, exploits, and infection vectors can drive discussions in intrusion analysis of IT security threats that is not the most important aspect of an ICS attack. To fully understand the CRASHOVERRIDE framework, its individual capabilities, and overall impact on ICS security it is important to understand certain fundamentals of electric grid operations.

Simplistically, the electric grid can be categorized into three functions: generation of electricity at power plants, transmission from the power plants across typically long distances at high voltage, and then stepped down to lower voltage to distribution networks to power customers. Along these long transmission and distribution systems are substations to transform voltage levels, serve as switching stations and feeders, and fault protection.

Many industries feed into the electric grid, and those differences require different systems and communications. As an example, while a power plant feeds energy into the electric grid there is no one-size-fits-all approach to power plants. There are power plants that cover different sources of fuel including coal-fired, nuclear generation, wind farm, solar farm, gas turbine power, hydroelectric and more. This means that the electric grid must be a robust, almost living creature, which moves and balances electricity across large regions. Electric grids use a special type of industrial control system called a supervisory control and data acquisition (SCADA) system to manage this process across large geographical areas. Transmission and distribution owners have their substations in their particular geographical footprint and control centers manage the cross-territory SCADA systems 24/7 by human operators. These control centers often regularly manage the continual demand and response of their customers, respond to faults, and plan and work with neighboring utilities.

This simplistic view of grid operations is similar around the world. There are often vendor and network protocol differences between countries but the electrical engineering, and the overall process is largely the same between nations. As an example, these systems use SCADA and leverage systems such as remote terminal units (RTUs) to control circuit breakers. As the breakers open and close, substations are energized or de-energized to balance power across the grid. Some network protocols such as IEC 104, a TCP-based protocol, and its serial protocol companion IEC 101, are often regional specific. Europe, some of Asian, and portions of the Middle East leverage these protocols to control RTUs from the SCADA human machine interfaces (HMIs).

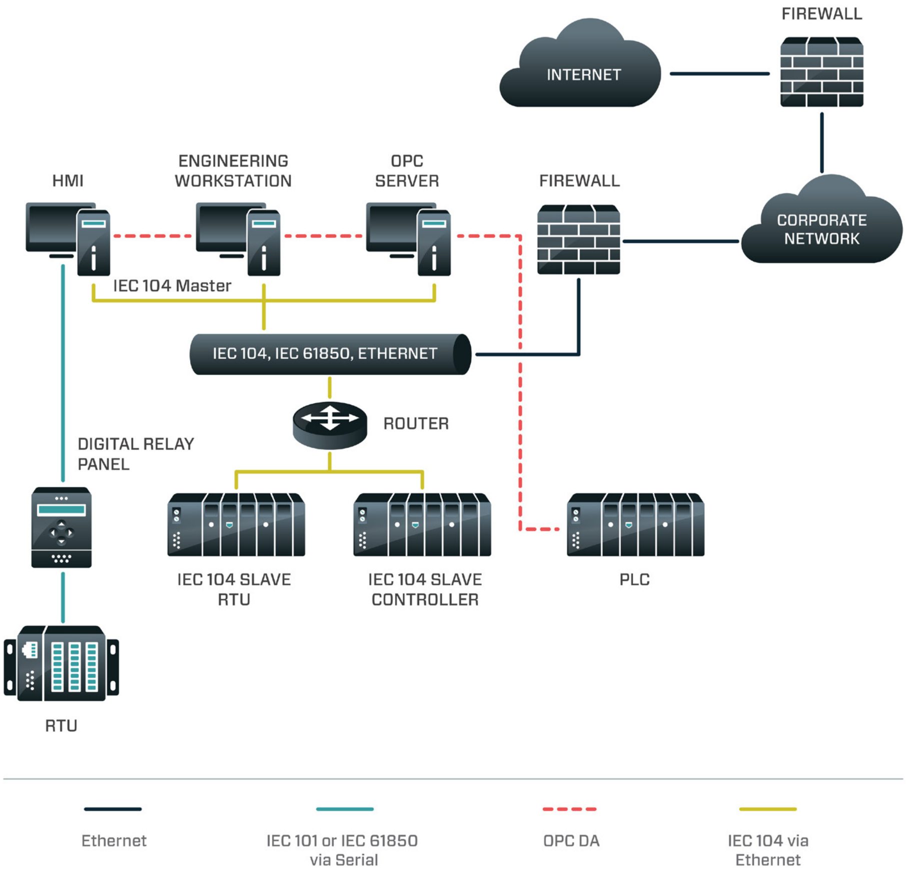


Figure 1: Simplistic Mockup of Electric Grid Operations Systems and Communications Relevant for CRASHOVERRIDE

In North America, the protocol of choice for this is the Distributed Network Protocol 3 (DNP3). The various protocols purposes are largely the same though: control physical equipment through RTUs, programmable logic controllers (PLCs), and other final control elements via HMIs as a part of the larger SCADA system. Some protocols have been adopted cross-country including IEC 61850 which is usually leveraged from an HMI to work with equipment such as digital relays and other types of intelligent electronic devices (IEDs). IEDs are purpose built microprocessor-based control devices and can often be found alongside power equipment such as circuit breakers. IEDs and RTUs operate in a master/slave capacity where the slave devices are polled and sent commands by master devices.

Substations manage the flow of power through transmission or distribution lines. Management of energizing and de-energizing of these lines ultimately control when and where the flow of power moves in and out of the substation. If you “open” a breaker you are removing the path where the electricity is flowing, or de-energizing it. If you “close” a breaker then you are energizing the line by closing the gap and allowing the power to “flow.” This concept is similar to anyone who has tripped (opened) a breaker in their house. Traditional “IT” or “IT security” staff may be confused on this terminology as it is opposite to how one would describe firewall rules where “open” means network traffic may flow and “closed” means network traffic is prohibited.

The grid is a well-designed system, and while damage can be done, it is vital to understand that in nations around the world the electric community has designed the system to be reliable and safe which has a natural byproduct of increased security. In the United States as an example, reliability is reinforced with regular training and events such as the North American grid’s GridEx where grid operators train for events from hurricanes, to terrorist incidents, to cyber-attacks and how they will respond to such outages. There is constantly a balance that must be understood when referring to grid operations: yes, the systems are vulnerable and more must be done to understand complex and multi-stage attacks, but the grid is also in a great defensible position because of the work of so many over the years.

Evolution of Tradecraft

CRASHOVERRIDE represents an evolution in tradecraft and capabilities by adversaries who wish to do harm to industrial environments. To fully appreciate the malware it is valuable to compare it to its predecessors and the Ukraine 2015 cyber attack.

STUXNET

The STUXNET malware has been written about extensively and referenced, at times, unfortunately, in comparison to most ICS related incidents and malware. It was the first confirmed example of ICS tailored malware leveraged against a target. The Windows portion of the code with its four zero-day exploits gained a lot of notoriety. However, it was the malware’s payload that was specific to ICS that was the most interesting component. The tradecraft exhibited by STUXNET was the detailed understanding of the industrial process. In IT networks, it is important for adversaries to identify vulnerabilities and exploit them to load malware and gain privileges on systems.

In ICS networks though, some of the most concerning issues are related to an adversary's ability to learn the physical process such as the engineering of the systems and their components in how they work together. STUXNET's greatest strength was leveraging functionality in Siemens equipment to interact with nuclear enrichment centrifuges through abuses of intended functionality. The purpose of the Siemens equipment was to be able to control and change the speed of the centrifuges. Stuxnet did this as well but with pre-programmed knowledge from the attackers on the speeds that would cause the centrifuge to burst from their casings. ICS tailored malware leveraging knowledge of industrial processes was now a thing. However, it was specific to Siemens equipment and unique to the Natanz facility in Iran. While tradecraft and exploits can be replicated, it was not reasonable to re-purpose the Stuxnet capability.

Dragonfly/HAVEX

The Dragonfly campaign was an espionage effort that targeted numerous industrial control system locations, estimates put it at over 2,000 sites, with a large emphasis on electric power and petrochemical asset owners. The Dragonfly campaign leveraged the HAVEX malware. There are often not many commonalities between different industrial sites. Even a single substation in one company can be almost entirely different than a substation in the same company based on vendors, implementation, integration, and the physical processes required at each site. One of the few commonalities across numerous ICS industries though is the OPC protocol. It is designed to be the universal translator for many industrial components and is readily accessible in an HMI or dedicated OPC server. The HAVEX malware leveraged legitimate functionality in the OPC protocol to map out the industrial equipment and devices on an ICS network. It was a clever use of the protocol and while the malware itself was not complex the tradecraft associated with the usage of OPC was sophisticated. However, the Dragonfly campaign was focused entirely on espionage. There was no physical disruption or destruction of the industrial process. Instead, it was the type of data you would want to leverage to design attacks in the future built for the specific targets impacted with the malware.

BLACKENERGY 2

The Sandworm team has targeted numerous industries ranging from western militaries, governments, research organizations, defense contractors, and industrial sites. It was their use of the BLACKENERGY 2 malware that caught the ICS industry's attention. This ICS tailored malware contained exploits for specific types of HMI applications including Siemens SIMATIC, GE CIMPLICITY, and Advantech WebAccess. BLACKENERGY 2 was a smart approach by the adversaries to target internet connected HMIs. Upon exploitation of the HMIs, the adversaries had access to a central location in the ICS to start to learn the industrial process and gain the graphical representation of that ICS through the HMI. The targeting of HMIs alone is often not enough to cause physical damage, but it is an ideal target for espionage and positioning in an ICS. Gaining a foothold in the network that had access to numerous components of the ICS while maintaining command and control to Internet locations, positioned it well for espionage.

Ukraine Cyber Attack 2015

The cyber-attack on three power companies in Ukraine on December 23rd, 2015 marked a revolutionary event for electric grid operators. It was the first known instance where a cyber-attack had disrupted electric grid operations. The Sandworm team was attributed to the attack and their use of the BLACKENERGY 3 malware. BLACKENERGY 3 does not contain ICS components in the way that BLACKENERGY 2 did. Instead, the adversaries leveraged the BLACKENERGY 3 malware to gain access to the corporate networks of the power companies and then pivot into the SCADA networks. While in the environment the adversaries performed their reconnaissance and eventually leveraged the grids systems against itself. They learned the operations and used the legitimate functionality of distribution management systems to disconnect substations from the grid leaving 225,000+ customers without power for upwards of 6 hours until manual operations could restore power. However, due to the wiping of Windows systems through the KillDisk malware and destruction of serial-to-Ethernet devices through malicious firmware updates, the Ukrainian grid operators were without their SCADA environment, meaning they lost the ability for automated control, for upwards of a year in some locations. The most notable aspect of the attack was the adversary's focus on learning how to leverage the systems against themselves. Malware enabled the attack, and malware delayed restoration efforts, but it was the direct interaction of the adversary leveraging the ICS against itself that resulted in the electric power disruptions, not malware.

CRASHOVERRIDE

The CRASHOVERRIDE malware impacted a single transmission level substation in Ukraine on December 17th, 2016. Many elements of the attack appear to have been more of a proof of concept than what was fully capable in the malware. The most important thing to understand though from the evolution of tradecraft is the codification and scalability in the malware towards what has been learned through past attacks. The malware took an approach to understand and codify the knowledge of the industrial process to disrupt operations as STUXNET did. It leveraged the OPC protocol to help it map the environment and select its targets similar to HAVEX. It targeted the libraries and configuration files of HMIs to understand the environment further and leveraged HMIs to connect to Internet-connected locations when possible as BLACKENERGY 2 had done. And it took the same type of approach to understanding grid operations and leveraging the systems against themselves displayed in Ukraine 2015's attack. It did all of these things with added sophistication in each category giving the adversaries a platform to conduct attacks against grid operations systems in various environments and not confined to work only on specific vendor platforms. It marks an advancement in capability by adversaries who intend to disrupt operations and poses a challenge for defenders who look to patching systems as a primary defense, using anti-malware tools to spot specific samples, and relying upon a strong perimeter or air-gapped network as a silver-bullet solution. Adversaries are getting smarter, they are growing in their ability to learn industrial processes and codify and scale that knowledge, and defenders must also adapt.

Capabilities

Capabilities Overview

The CRASHOVERRIDE malware is a modular framework consisting of an initial backdoor, a loader module, and several supporting and payload modules.

The most important items are the backdoor, which provides access to the infected system, the loader module, which enables effects on the target, and the individual payload modules. Dragos focused our analysis on the previously mentioned items as they are most relevant for defending grid operations.

Dragos analysts were able to obtain two samples of the malware related to effects on the targeted industrial control system. One sample was the IEC 104 protocol module, and the other sample was the data wiper. Both samples shared common design characteristics indicative of being part of a broader ICS attack and manipulation framework. ESET was able to uncover an additional IEC 61850 and OPC module which they have analyzed and shared with Dragos.

Below contains an overview of program execution flow and dependency.

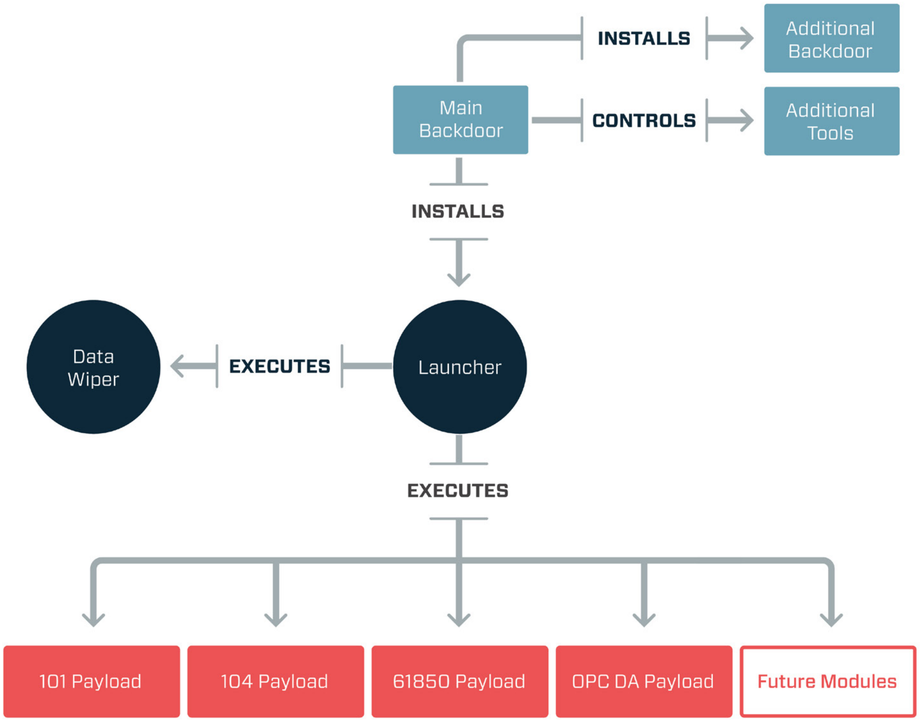


Figure 2. CRASHOVERRIDE Module Overview Including ESET's Discoveries

Module Commonalities

Dragos analysts were able to determine the compile time for both modules obtained as being within 12 minutes of each other just after 2:30 am on December 18th in an unknown time zone although timestamps for both samples were zeroed out. These times falls in the same timeframe as the Ukraine events. Both module samples exported a function named Crash that served as the main function to begin execution. The common Crash function enables the ability to “plug and play” additional modules.

Backdoor/RAT Module

Key Features

- Authenticates with a local proxy via the internal network established before the backdoor installation
- After authentication opens HTTP channel to external command and control server (C2) through internal proxy
- Receives commands via the external command and control (C2) server
- Creates a file on the local system (contents not determined)
- Overwrites an existing service to point to the backdoor so the malware persists between reboots

Details

Access to the ICS network flows through a backdoor module. Dragos obtained four samples which all featured similar functionality. On execution, the malware attempts to contact a hard-coded proxy address located within the local network. ELECTRUM must establish the internal proxy before the installation of the backdoor.

The malware expects to communicate to an internal proxy listening on TCP 3128. This port is a default port associated with the Squid proxy. The beaconing continues without pause until it establishes a connection. The backdoor then sends a series of HTTP POST requests with the victim’s Windows GUID (a unique identifier set with every Windows installation) in the HTTP body. This information authenticates the targeted machine to the command and control (C2) server. If the C2 server does not respond, the backdoor will exit.

If the authentication is successful to the internal proxy, the malware attempts to perform an HTTP CONNECT to an external C2 server via the internal proxy. Across four samples, Dragos identified three different C2 addresses which were likely part of the December 2016 attack on Ukraine:

```
95.16.88.6

93.115.27.57

5.39.218.152
```

A check of the TOR project's ExoneraTOR service indicates that 93.115.27.57 and 5.39.218.152 were both listed as active TOR nodes during the events in Ukraine while 95.16.88.6 was not.

When performing the HTTP CONNECT, the malware attempts to identify the system default user agent. If this cannot be determined or does not exist, then a hard-coded default for the malware is used:

```
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; InfoPath.1)
```

The malware can be configured to beacon out periodically afterwards via a hard-coded configuration value. The implant is designed to retrieve commands from the C2 server:

- Create a new process as logged in user
- Create a new process as specified user via CreateProcessWithLogon
- Write a file
- Copy a file
- Execute a command as logged in user
- Execute a command as specified user
- Kill the backdoor
- Stop a service
- Specify a user (log in as user) and stop a service
- Specify a user (log in as user) and start a service
- Alter an existing service to point to specified process and change to start at boot

Execution results in several artifacts left on the host. During execution, the malware checks for the presence of a mutex value. Mutexes are program objects that name resources to enable sharing with multiple program threads. In this case, CRASHOVERRIDE checks the following:

```
\Sessions\1\Windows\ApiPortection
```

The backdoor may also create and check a blank mutex name. Reviewing memory during execution and analysis of other modules in the malware indicates that \Sessions\1\Windows\ appears multiple times, indicating that a check may be performed.

The backdoor writes a file to either C:\Users\Public\ or C:\Users\<Executing User>

The contents of this file were not discovered during our analysis, and it did not appear to be vital to the malware functionality. However, this is a good indicator of the observed activity and may be leveraged to detect this specific sample through host-based indicator checking.

The service manipulation process is the only persistence mechanism for the malware. When used, the adversary can select an arbitrary system service, direct it to refer to CRASHOVERRIDE, and ensure it is loaded on system boot. If this fails, the malware, although present on disk, will not start when the machine reboots.

When evaluating the options provided to the adversary, an important piece of functionality associated with most remote access tools is absent: a command to exfiltrate data. While this functionality could be created via the command execution options, one would expect this option to be explicit given options to download and copy files on the host if the adversary intended to use the tool as an all-encompassing backdoor and espionage framework. Instead, the functionality of this tool is explicitly designed for facilitating access to the machine and executing commands on the system and cannot reasonably be confused as an espionage platform, data stealer, or another such item.

Launcher Module

Key Features

- Loads payload modules which manipulate the ICS and cause destruction via the wiper
- Starts itself as a service likely to hide better
- Loads the payload module(s) defined on the command line during execution
- Launches the payload and begins either 1 or 2 hours countdown before launching the data wiper (variant dependent)

Details

Within the attack sequence, the ICS payload modules and data wiper module must be loaded by a separate loader EXE. Dragos obtained one sample of this file called the Launcher.

The launcher takes three parameters on start:

```
    Launcher.exe <Working Directory> payload.dll configuration.ini
```

On launch, the sample analyzed starts a service named defragsvc. It then loads the module DLL via an exported function named Crash. A new thread is created at the highest priority on the executing machine. Control then passes from the launcher to the loaded module while the launcher waits two hours before executing the data wiper.

Data Wiper Module

Key Features

- Clears all registry keys associated with system services
- Overwrites all ICS configuration files across the hard drives and all mapped network drives specifically targeting ABB PCM600 configuration files in this sample
- Overwrites generic Windows files
- Renders the system unusable

Details

Once executed, the data wiper module clears registry keys, erase files, and kill processes running on the system. A unique characteristic of the wiper is that the main functionality was implemented within the Crash function.

The first task of the wiper writes zeros into all of the registry keys in:

```
SYSTEM\CurrentControlSet\Services
```

This registry tree contains initialization values for each service on the system. Removal of these values renders a system inoperable. The next wiper task targets ICS configuration files across the local hard drive and mapped network drives. The malware authors included functionality to target drives lettered C-Z.

The wiper also targets file types unique to ABB's PCM600 product used in substation automation in addition to more general Windows files. The below table outlines some of the unique file extensions used by industrial control systems.

File Extension	Usage
.pcmp	PCM600 Project (ABB)
.pcmi	PCM600 IEC File (ABB)
.pcmt	PCM600 Template IED File
.CIN	ABB MicroScada
.PL	Programmable Logic File
.paf	PLC Archive File
.SCL	Substation Configuration Language
.cid	Configured IED Description
.scd	Substation Configuration Description

Table 1. File extensions targeted by the data wiper module

IEC 104 Module

Key Features

- Reads a configuration file defining the target (likely an RTU) and action to take
- 'Kills' legitimate the master process on the victim host
- Masquerades as the new master
- Enters one of four modes:
    - Sequence mode: continuously sets RTU IOAs to open
    - Range mode: (1) Interrogates each RTU for valid IOAs; (2) toggles each IOA between open and closed state
    - Shift mode: unknown at this time
    - Persist mode: unknown at this time/not fully implemented

IEC 104 Module Contains Handling for All Protocol Transmission Types

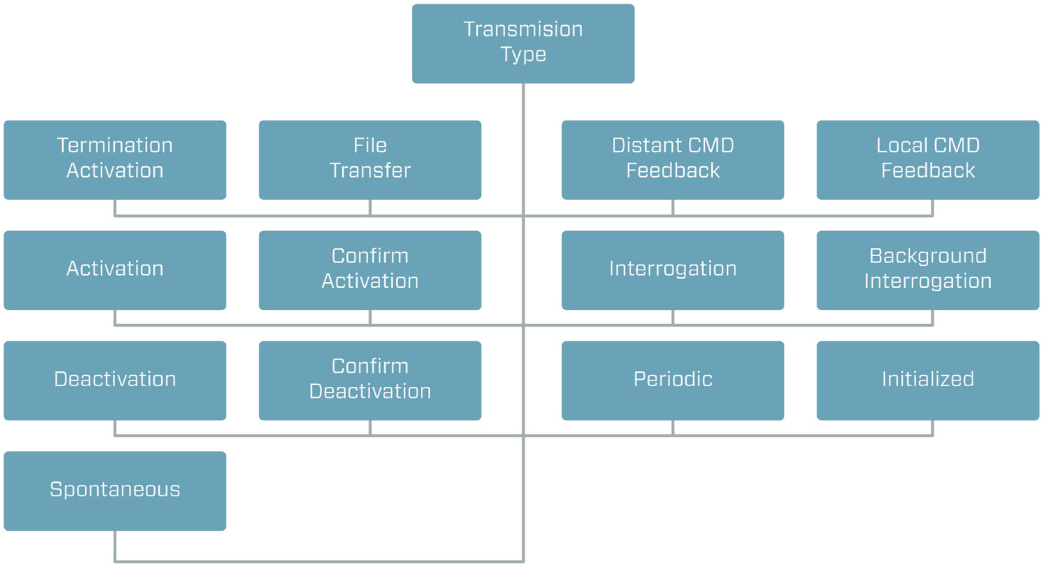


Figure 3. Protocol Transmission Types in IEC 104

IEC 104 Module Execution Flow

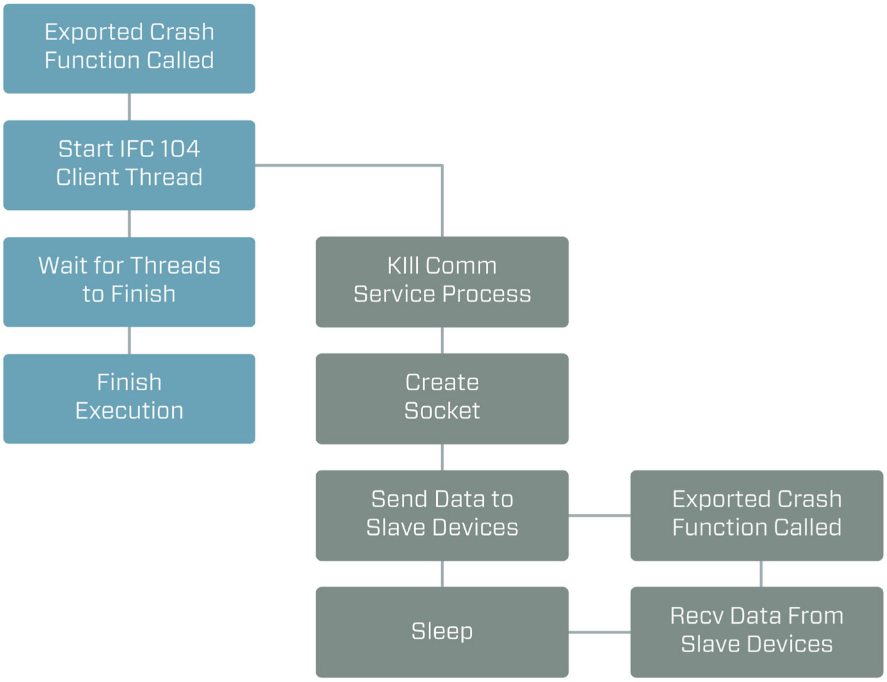

Exported Crash Function Called
Start IFC 104 Client Thread
Wait for Threads to Finish
Finish Execution
KIll Comm Service Process
Create Socket
Send Data to Slave Devices
Sleep
Exported Crash Function Called
Recv Data From Slave Devices

Figure 4: Execution Flow of IEC 104 Module in CRASHOVERRIDE

Details

The CRASHOVERRIDE IEC 104 module is a complete implementation of IEC 104 to serve in a "MASTER" role. This raw functionality creates a Swiss army knife for substation automation manipulation yet also provides tailored functionality. The functions exposed to the malware operator are confined by the options of the configuration file. This report outlines the options analyzed today but notes that extending and enhancing functionality is straight forward with the robust protocol implementation.

The design of the IEC 104 module differs from the wiper and suggests that a secondary group of developers could have been involved. Instead of the exported crash function containing the primary execution instructions, the function parses the config file then starts a thread containing the IEC 104 master. The configuration file can have multiple entries offset by [STATION], followed by 13 values:

File Extension	Usage
target_ip	NONE
target_port	NONE
logfile	NONE
adsu	NONE
stop_comm_service	1
change	1
first_action	on
silence	0
uselog	0
stop_comm_service_name	<blank>
timeout	1 second
socket_timeout	15 seconds
range	NONE

Table 2. IEC-104 module configuration file fields

The configuration file is critical to achieving an effect on the target, as target specifications for the device must be provided by the operator in the configuration file for the module to function. There are no observed automated means of enumerating the network and then impacting RTUs.

Each [STATION]entry spawns a thread for follow-on effects against ICS equipment. Once the IEC 104 master thread begins, the first action is to try to kill the communications service process which acts as the master process. Once the module stops the communications service process, a socket opens with the target IP and destination port sending data to slave devices and receiving the resulting responses.

Depending on the mode defined within the configuration file the module may:

- Set specific values
- Enumerate IOAs on the target devices
- Continuously set the IOA to open, or
- Continuously toggle the IOA between open and closed states.

This module contains no interactive capability.

RTUs and PLCs, in simplistic terms, act on input and output. Each discrete input and output is tied to a memory address. Depending on implementation these addresses are referred to as coils, registers, or for IEC 104: information object addresses (IOAs). IOAs are typed and can hold different value types, such as Boolean or Unsigned Integer values. The 104 module properly understands how to enumerate and discover IOAs to operate breakers.

IEC 101 Module

This module was unavailable to Dragos at the time of publication. ESET's analysis claims the functionality is equivalent to the IEC 104 module except with communications over serial. However, Dragos was able to confirm that the module exists.

IEC 61850 Module

This module was unavailable to Dragos at the time of publication. ESET's analysis claims once executed the module leverages a configuration file to identify targets and without a configuration file it enumerates the local network to identify potential targets. It communicates with the targets to identify whether the device controls a circuit breaker switch. For certain variables (no further information available) it will change their state while also generating an action log. However, Dragos was able to confirm that this module does exist.

OPC DA Module

This module was unavailable to Dragos at the time of publication. ESET's analysis claims the module does not require a configuration. It enumerates all OPC servers and their associated items looking for a subset related to ABB containing the string ctl. It then writes 0x01 twice into the item overwriting the proper value giving the device a primary value out of limits device status. However, Dragos was able to confirm that this module exists.

SIPROTEC DoS Module

This module was unavailable to Dragos at the time of publication. ESET's analysis claims the module sends UDP packets to port 50000 exploiting CVE-2015-5374 causing the SIPROTEC digital relay to fall into an unresponsive state. Dragos could not validate that this module exists.

Capability Conclusions

ELECTRUM's ability to adopt a development style described above has several implications: first, developers can integrate new protocols into the overall framework quickly. Second, ELECTRUM could easily leverage external development teams skilled at exploiting industrial control systems. Some adversaries would likely approach capability development through a 'two-tier' approach: a core development team skilled at writing the overall framework and a second team knowledgeable about a given control system. The platform team would take the control system modules and add logic to fit them within the platform. The IEC 104 module demonstrates this approach.

Given the execution described with secondary threads the team authoring the Crash function likely did not author the IEC 104 master portion of the code. Both development teams probably worked together to decide on a log file format for consumption by the main Crash function and executed in each of the IEC 104 module threads.

Implications of capability

This section describes legitimate CRASHOVERRIDE attack and impact scenarios. Extensions of these and potential hypothetical scenarios were deemed indeterministic and will not be addressed.

Attack Option: De-energize substation

CRASHOVERRIDE, based on prior knowledge, must have a configuration file for targeting information of one or multiple RTUs. This configuration option allows for several types of activities. One operation the configuration option allows is 'sequence.'

.

The command sequence polls the target device for the appropriate addresses. Once it is at the subset of known addresses, it can then toggle the value. The command then begins an infinite loop and continues to set addresses to this value effectively opening closed breakers. If a system operator tries to issue a close command on their HMI the sequence loop will continue to re-open the breaker. This loop maintaining open breakers will effectively de-energize the substation line(s) preventing system operators from managing the breakers and re-energize the line(s).

The effects of de-energizing a line or substation largely depends on the system dynamics, power flows, and other variables. In some circumstances, it may have no immediate impact while in others it could put customers into an outage. It is important to note that grid operations encompass failure modes and operations can normally compensate. That is, after all, why humans are 'in the loop' to monitor and maintain the system.

From a recovery standpoint, the remote staff will effectively have lost control of the breakers and will be required to send crews to the substation. If the CRASHOVERRIDE loop continues unabated, then the crews will likely sever communications as both a troubleshooting and recovery action. Severing communications puts the substation in manual operation where a physical presence is now required. This could result in a few hours of outages

Attack Option: Force an Islanding event

Dragos is currently investigating a separate and more disruptive attack option in CRASHOVERRIDE as described by ESET. As before, the attacker must have a configuration file for targeting information of one or multiple RTUs. This configuration file now uses the range command to begin a loop that toggles the status of the breaker between open and close continuously. The changing breaker status will invoke automated protective operations to isolate (commonly referred to as 'islanding') the substation. This is an intentional self-protective capability of grid operations.

In effect, this breaker strobing takes the substation offline due to the protective relay scheme's automated operations causing perturbations of some degree on the grid as scientific principles define how the behavior interacts with frequencies and phases. The variables of these effects will dictate impacts but could cause system instabilities depending on the effectiveness of the protection relays and their operations. Grid operation contingencies become more critical if multiple substations were under attack likely resulting in many small islanding events. This is assuming coordinated targeting of multiple electric sites and could result in a few days of outages.

Adding Amplification Attacks

Forcing an islanding of a substation through continual breaker manipulation is significant by itself. However, CRASHOVERRIDE has the potential to amplify this attack even more. Two separate CRASHOVERRIDE modules offer this opportunity.

Using OPC to create a Denial of Visibility

The OPC module ESET analysis suggests it can brute force values. Module OPC.exe will send out a 0x01 status which for the target systems equates to a "Primary Variable Out of Limits" misdirecting operators from understanding protective relay status.

Bit Mask	**Definition**
0x10	**More Status Available** – More status information is available via Command 48, Read Additional Status Information.
0x08	**Loop Current Fixed** – The Loop Current is being held at a fixed value and is not responding to process variations.
0x04	**Loop Current Saturated** – The Loop Current has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further.
0x02	**Non-Primary Variable Out of Limits** – A Device variable not mapped to the PV is beyond its operating limits.
0x01	**Primary Variable Out of Limits** – The PV is beyond its operating limits.

The outcome of the action infers that various systems can either perform actions on wrong information or report incorrect information to system operators. This Denial of Visibility will amplify misunderstanding and confusion while system operators troubleshoot the problem as their system view will show breakers closed when they are open.

.

Using CVE-2015-5374 to Hamper Protective Relays

A second, and more severe, amplifying attack would be to neutralize the automated protective system by creating a Denial of Service against some or all of the protective relays. This possibility exists in a tool ESET has claimed to have discovered that implements the known CVE-2015-5374 Denial of Service condition to the Siemens SIPROTEC relays. Siemens released a patch for this in July 2015 under Siemens advisory SCA-732541. At this time it is believed that CVE-2015-5374 causes a denial of service (DoS) of the complete relay functionality and not just the network communications module. Dragos has independent evidence that this module exists but it cannot be confirmed.

Hampering the protective scheme by disabling the protective relays can broaden the islanding event and, if done at scale, could trigger a larger event causing multiple substations and lines "islanding" from the electric grid. Siemens SIPROTEC was likely chosen in this attack only because that was the vendor device at the Ukraine Kiev site attacked in December 2016. This same tactic against digital relays, albeit not the same exploit, could have a similar impact on grid operations. However, there are many different types of digital relays each with different configurations. This amplifying attack would be very difficult to do at scale properly and would require a significant investment on behalf of the adversary.

Defense Recommendations

Doing the basics is always appropriate, and it significantly helps move ICS into a defensible position. However, they are not worth repeating here, and instead, more tailored approaches specific to ICS security analysts trying to defend against CRASHOVERRIDE and similar capabilities are presented below:

- Electric utility security teams should have a clear understanding of where and how IEC 104 and IEC 61850 protocols are used. North American electric utilities should include DNP3 on this list in case the malware is extended to impact U.S. systems. Look specifically for increased usage of the protocols against baselines established in the environment. Also, look for systems leveraging these protocols if they have not before and specifically try to identify systems that are generating new network flows using these protocols.
- Similarly, understand OPC implementations and identify how the protocol is being used. It is a protocol that is pervasive across numerous sectors. Also, CRASHOVERRIDE is the second, out of four, ICS tailored malware suite with OPC capabilities. OPC will appear abnormal in the CRASHOVERRIDE usage as it is being used to scan all devices on the network which would generate more traffic than usual.
- Robust backups of engineering files such as project logic, IED configuration files, and ICS application installers should be offline and tested. This will help reduce the impact of the wiper functionality.
- Prepare incident response plans for this attack and perform table top exercises bringing in appropriate stakeholders and personnel across engineering, operations, IT, and security. The scenario should include substation outages with the requirement to do manual operations while recovering the SCADA environment and gathering appropriate forensics.
- The included YARA rules and other indicators of compromise can be leveraged to search for possible infections (IOCs). The YARA rules will provide a higher confidence towards discovering an infection than the other IOCs and should be searched for against Windows OT systems especially noting HMIs. The behavioral analytics to identify the communications on the network would provide the highest capability to detect this and similar threats.

While some defenses and architecture changes may have value in other situations, the following are responses that are not appropriate for this attack:

- Transmission and distribution companies should not rely on the usage of other protocols such as DNP3 as a protection mechanism. The completeness of the CRASHOVERRIDE framework suggests there may be other undisclosed modules such as a DNP3 module. Also, adding this functionality into the existing framework would not require extensive work on the part of the adversary.
- Air gapped networks, unidirectional firewalls, anti-virus in the ICS, and other passive defenses and architecture changes are not appropriate solutions for this attack. No amount of security control will protect against a determined human adversary. Human defenders are required

Indicators

TYPE	SUBTYPE	IOC	Description	ICS Kill Chain	Impact
Host	Mutex Value	ApiPortection9d3	Mutex value checked	Stage 2: Install	Recon
Host	Mutex Value	<Blank Value>	Mutex value created	Stage 2: Install	Recon
Host	File	C:\Users\<Public OR Executing User>\imapi	File dropped and deleted after program exit	Stage 2: Install	Recon
Host	Service Name	defragsvc	Name given to service start	Stage 2: C2	Remote Access
Network	IP Address	95.16.88.6	External C2 server [DEC 2016] (likely TOR node at time of attack)	Stage 2: C2	Remote Access
Network	IP Address	93.115.27.57	External C2 server [DEC 2016] (likely TOR node at time of attack)	Stage 2: C2	Remote Access
Network	IP Address	5.39.218.152	External C2 server [DEC 2016] (likely TOR node at time of attack)	Stage 2: C2	Remote Access
Network	User Agent String	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; InfoPath.1)	Default user agent string used in C2 if unable to get system default user agent string	Stage 2: C2	Remote Access
Host	Command Line	<Drive>:\<name>.exe -ip=<IP_address> -ports=<ports>	Command line arguments used to launch custom port scanner observed with malware. Command line logging required to track.	Stage 2: Develop	Recon
Host	Registry Key	HKLM\SYSTEM\CurrentControlSet\Services\<target_service_name>\ImagePath <path to malware>	Change in Service Image Path in the system registry to point to malware allowing malware to restart on system reboot.	Stage 2: Installation	Persistence
Host	SHA1 File Hash	F6C21F8189CED6AE150F9EF2E82A3A57843B587D	Traffic to <internalIP>:3128, HTTP CONNECT to 5.39.218.152:443. Backdoor/RAT.	Phase2: C2	Remote Access
Host	SHA1 File Hash	CCCCE62996D578B984984426A024D9B250237533	Traffic to <internalIP>:3128, HTTP CONNECT to 5.39.218.152:443. Backdoor/RAT.	Phase2: C2	Remote Access
Host	SHA1 File Hash	8E39ECA1E48240C01EE570631AE8F0C9A9637187	Backdoor/RAT Proxy + HTTP CONNECT to 93.115.27.57:443.	Phase2: C2	Remote Access
Host	SHA1 File Hash	2CB8230281B86FA944D3043AE906016C8B5984D9	Backdoor/RAT Proxy + HTTP CONNECT to 95.16.88.6:443	Phase2: C2	Remote Access

Host	SHA1 File Hash	79CA89711CDAEDB16B0CCCCFDCFBD6AA7E57120A	Launcher for payload DLL. Takes input as three command line parameters – working directory, module, and config file.	Stage 2: Attack	Loss of Control
Host	SHA1 File Hash	94488F214B165512D2FC0438A581F5C9E3BD4D4C	Module for 104 effect. Exports 'Crash' which is invoked by launcher. Functionality requires config file.	Stage 2: Attack	Loss of Control
Host	SHA1 File Hash	5A5FAFBC3FEC8D36FD57B075EBF34119BA3BFF04	Wiper module, wipes list of files by extension, removes system processes, and makes registry changes to prevent system boot.	Stage 2: Attack	Destruction
Host	SHA1 File Hash	B92149F046F00BB69DE329B8457D32C24726EE00	Wiper module, wipes list of files by extension, removes system processes, and makes registry changes to prevent system boot.	Stage 2: Attack	Destruction
Host	SHA1 File Hash	B335163E6EB854DF5E08E85026B2C3518891EDA8	Custom-built port scanner.	Stage 2: Develop	Recon
Host	SHA1 File Hash	7FAC2EDDF22FF692E1B4E7F99910E5DBB51295E6	OPC Data Access protocol enumeration of servers and addresses	Stage 2: Attack	Loss of Control
Host	SHA1 File Hash	ECF6ADF20A7137A84A1B319CCAA97CB0809A8454	IEC-61850 enumeration and address manipulation	Stage 2: Attack	Loss of Control
Host	Filename	opc.exe	OPC Data Access protocol enumeration of servers and addresses	Stage 2: Attack	Loss of Control
Host	Filename	61850.exe	IEC-61850 enumeration and address manipulation	Stage 2: Attack	Loss of Control
Host	Filename	defragsvc.exe	Launcher for payload DLL	Stage 2: Attack	Loss of Control
Host	Filename	haslo.exe	Wiper module, wipes list of files by extension, removes system processes, and makes registry changes to prevent system boot.	Stage 2: Attack	Destruction
Host	Filename	avtask.exe	Backdoor/RAT	Stage 2: C2	Remote Access
Host	Filename	port.exe	Portscanner	Stage 2: Develop	Recon
Host	Filename	104.dll	IEC-104 module	Stage 2: Attack	Loss of Control
Host	Filename	haslo.dat	Wiper module	Stage 2: Attack	Destruction
Host	Filename	svchost.exe	Launcher module	Stage 2: Install	Remote Access
Host	Filename	tiersvc.exe	Backdoor/RAT	Stage 2: C2	Remote Access
OPC Server	OPC Group	Aabdul	OPC DA Module	Stage 2: Attack	Loss of Visibility

Yara Rules

Also found at https://github.com/dragosinc/CRASHOVERRIDE

```
import “pe”
import “hash”

rule dragos_crashoverride_exporting_dlls
{
      meta:
            description = “CRASHOVERRIDE v1 Suspicious Export”
            author = “Dragos Inc”
      condition:
            pe.exports(“Crash”) & pe.characteristics
}

rule dragos_crashoverride_suspcious
{
      meta:
            description = “CRASHOVERRIDE v1 Wiper”
            author = “Dragos Inc”
      strings:
            $s0 = “SYS_BASCON.COM” fullword nocase wide
            $s1 = “.pcmp” fullword nocase wide
            $s2 = “.pcmi” fullword nocase wide
            $s3 = “.pcmt” fullword nocase wide
            $s4 = “.cin” fullword nocase wide
      condition:
            pe.exports(“Crash”) and any of ($s*)
}
```

YARA Rules

```
rule dragos_crashoverride_name_search {
      meta:
            description = “CRASHOVERRIDE v1 Suspicious Strings and Export”
            author = “Dragos Inc”
      strings:
            $s0 = “101.dll” fullword nocase wide
            $s1 = “Crash101.dll” fullword nocase wide
            $s2 = “104.dll” fullword nocase wide
            $s3 = “Crash104.dll” fullword nocase wide
            $s4 = “61850.dll” fullword nocase wide
            $s5 = “Crash61850.dll” fullword nocase wide
            $s6 = “OPCClientDemo.dll” fullword nocase wide
            $s7 = “OPC” fullword nocase wide
            $s8 = “CrashOPCClientDemo.dll” fullword nocase wide
            $s9 = “D2MultiCommService.exe” fullword nocase wide
            $s10 = “CrashD2MultiCommService.exe” fullword nocase wide
            $s11 = “61850.exe” fullword nocase wide
            $s12 = “OPC.exe” fullword nocase wide
            $s13 = “haslo.exe” fullword nocase wide
            $s14 = “haslo.dat” fullword nocase wide
      condition:
            any of ($s*) and pe.exports(“Crash”)
}
```

YARA Rules

```
rule dragos_crashoverride_hashes {
    meta:
        description = “CRASHOVERRIDE Malware Hashes”
        author = “Dragos Inc”
    condition:
        filesize < 1MB and
        hash.sha1(0, filesize) == “f6c21f8189ced6ae150f9ef2e82a3a57843b587d” or
        hash.sha1(0, filesize) == “cccce62996d578b984984426a024d9b250237533” or
        hash.sha1(0, filesize) == “8e39eca1e48240c01ee570631ae8f0c9a9637187” or
        hash.sha1(0, filesize) == “2cb8230281b86fa944d3043ae906016c8b5984d9” or
        hash.sha1(0, filesize) == “79ca89711cdaedb16b0ccccfdcfbd6aa7e57120a” or
        hash.sha1(0, filesize) == “94488f214b165512d2fc0438a581f5c9e3bd4d4c” or
        hash.sha1(0, filesize) == “5a5fafbc3fec8d36fd57b075ebf34119ba3bff04” or
        hash.sha1(0, filesize) == “b92149f046f00bb69de329b8457d32c24726ee00” or
        hash.sha1(0, filesize) == “b335163e6eb854df5e08e85026b2c3518891eda8”
}
```

YARA Rules

```
rule dragos_crashoverride_moduleStrings {
      meta:
            description = “IEC-104 Interaction Module Program Strings”
            author = “Dragos Inc”
      strings:
            $s1 = “IEC-104 client: ip=%s; port=%s; ASDU=%u” nocase wide ascii
            $s2 = “ MSTR ->> SLV” nocase wide ascii
            $s3 = “ MSTR <<- SLV” nocase wide ascii
            $s4 = “Unknown APDU format !!!” nocase wide ascii
            $s5 = “iec104.log” nocase wide ascii
      condition:
            any of ($s*)
}

rule dragos_crashoverride_configReader
{
    meta:
        description = “CRASHOVERRIDE v1 Config File Parsing”
        author = “Dragos Inc”
    strings:
        $s0 = { 68 e8 ?? ?? ?? 6a 00 e8 a3 ?? ?? ?? 8b f8 83 c4 ?8 }
        $s1 = { 8a 10 3a 11 75 ?? 84 d2 74 12 }
        $s2 = { 33 c0 eb ?? 1b c0 83 c8 ?? }
        $s3 = { 85 c0 75 ?? 8d 95 ?? ?? ?? ?? 8b cf ?? ?? }
    condition:
        all of them
}
```

YARA Rules

```
rule dragos_crashoverride_weirdMutex
{
    meta:
        description = “Blank mutex creation assoicated with CRASHOVERRIDE”
        author = “Dragos Inc”
    strings:
        $s1 = { 81 ec 08 02 00 00 57 33 ff 57 57 57 ff 15 ?? ?? 40 00 a3 ?? ?? ?? 00
85 c0 }
        $s2 = { 8d 85 ?? ?? ?? ff 50 57 57 6a 2e 57 ff 15 ?? ?? ?? 00 68 ?? ?? 40 00}
    condition:
        all of them
}


rule dragos_crashoverride_serviceStomper
{
    meta:
        description = “Identify service hollowing and persistence setting”
        author = “Dragos Inc”
    strings:
        $s0 = { 33 c9 51 51 51 51 51 51 ?? ?? ?? }
        $s1 = { 6a ff 6a ff 6a ff 50 ff 15 24 ?? 40 00 ff ?? ?? ff 15 20 ?? 40 00 }
    condition:
        all of them
}
```

YARA Rules

```
rule dragos_crashoverride_wiperModuleRegistry
{
    meta:
        description = “Registry Wiper functionality assoicated with CRASHOVERRIDE”
        author = “Dragos Inc”
    strings:
        $s0 = { 8d 85 a0 ?? ?? ?? 46 50 8d 85 a0 ?? ?? ?? 68 68 0d ?? ?? 50 }
        $s1 = { 6a 02 68 78 0b ?? ?? 6a 02 50 68 b4 0d ?? ?? ff b5 98 ?? ?? ?? ff 15 
04 ?? ?? ?? }
        $s2 = { 68 00 02 00 00 8d 85 a0 ?? ?? ?? 50 56 ff b5 9c ?? ?? ?? ff 15 00 ?? 
?? ?? 85 c0 }
    condition:
        all of them
}


rule dragos_crashoverride_wiperFileManipulation
{
    meta:
        description = “File manipulation actions associated with CRASHOVERRIDE wip-
er”
        author = “Dragos Inc”
    strings:
        $s0 = { 6a 00 68 80 00 00 00 6a 03 6a 00 6a 02 8b f9 68 00 00 00 40 57 ff 15 
1c ?? ?? ?? 8b d8 }
        $s2 = { 6a 00 50 57 56 53 ff 15 4c ?? ?? ?? 56 }
    condition:
        all of them
}
```