Alex Shah

EN.695.741.81.SP25 Information Assurance Analysis

Mod 13 Assignment

April 27, 2025

Stuxnet Worm

Stuxnet was discovered in 2010 as a highly targeted worm designed by a nation state to disrupt specific industrial control systems, likely uranium processing in Iran. It spread initially through USB drives which exploited several vulnerabilities like a zero day flaw in Windows shortcut files in order to infect engineering PCs through social engineering tactics. Once inside the network, other exploits like a vulnerability in a Windows printer spooler service allowed the worm to spread to networked shares and find other devices to infect, enabling lateral movement. The initial USB spread was purposely slowed to spread to only a handful of devices per USB device, helping to avoid detection by unnecessary spread or spreading fast enough to be detectable. The worm was harmless to most infected devices because it had a specific design goal, and its payload remained dormant until it encountered the programmable controllers it was designed to manipulate. In the affected industrial control settings, the worm reconfigured PLCs used in uranium processing centrifuges. The misconfigured settings caused damage to centrifuges or inefficiently processed uranium in the targeted applications. (Mueller & Yadegari, 2010). By using multiple zero day exploits and legitimate digital certificates, the Stuxnet worm was able to penetrate hundreds of thousands of devices and move between networks unnoticed. The rootkit portion of the worm enabled it to start close to boot using real signed certificates from Realtek and others which concealed its operations. It used this capability to inject its malicious code into processes from kernel mode early in the boot sequence. In addition, the modifications the payload made to the PLCs were obfuscated when the code was read directly by infecting project files and

injecting into DLL files used when interacting with the PLC. This made discovering the methods and effects of the attack much more difficult as the worm concealed its activities.

The delivery step of the ICS kill chain circumvented the air gapped industrial systems by leveraging social engineering to deliver USB drives that penetrated the security perimeter. This was a crucial step in breaching the security of the systems. The initial infection vector was not enough to make this such an advanced persistent threat, as Stuxnet used several zero day exploits to spread in the exploitation phase. Several clever tactics were used to avoid detection as it installed and modified itself and other processes via rootkits and command and control servers. Stuxnet used C2 during the intrusion phase to receive updates and suggests that remote monitoring and exploitation were possible. Stuxnet could also update itself from other infected hosts in a peer to peer network. This could avoid Internet access which could reveal the infection. Finally in the attack stage, Weaponization, the malicious actions by manipulating PLCs were done subtly to not raise suspicions of an attack. This design allowed the worm to stay embedded in the systems and remain undetected while countless centrifuges were destroyed and the uranium enrichment program could have been hampered for years.

In order for an attack this sophisticated and well planned to be detected or prevented, several key categories of security could be enhanced. While some of the exploits were zero day and had no patch, there were multiple exploits used in the spread and infection steps of the initial attack setup. Some of these vulnerabilities could have been closed which could have slowed the progression of the worm. And the worm was able to bridge the air gapped OT system by USB devices. There should be policies to restrict USB use and stricter training for staff to prevent the critical systems from being breached using the lessons learned from Stuxnet. In addition, the worm was able to subtly increase or decrease the RPM of the centrifuge to cause issues running too quickly or slowly. Loading the new configuration could have been detected or compared to a known good configuration to detect changes. These small differences might have been caught with better monitoring and logging of sensor readouts,

or carefully reviewing the tachometer under operation to double check observations matched expected results.

BlackEnergy

In 2015 the BlackEnergy malware disrupted Ukraine's power grid through a targeted spear phishing campaign. Multiple staff at power companies in Ukraine received email with a malicious attachment. When opened with macros enabled, the malware was able to infect the machine and allow attackers to gain access to further systems. The malware itself did not spread from the initial machine, but used the stolen credentials to move laterally including the ICS network. Attackers were then able to use remote admin software to control devices and ensure they were able to interact with the grid controls. This attack used remote access to issue commands, where attackers manually disabled substations. The attackers also used a drive wiping utility to target log files and erase the MBR portion of drives and rewriting firmware for serial to Ethernet devices which aided in covering their activities as well as slowing down the power companies trying to get services back online. (MITRE, n.d.).

The Delivery, Exploit, and Install portions of the ICS kill chain were important steps to this attack's success. The BlackEnergy 3 malware was able to infect multiple machines at multiple companies by leveraging social engineering techniques to get staff to enable macros in a Microsoft Office document. While the reconnaissance stage of the kill chain was likely a crucial step in identifying vulnerable operations and individuals, little is known about the origin of the attack. The selected power companies did have some factors in common like a high reliance on automation.

The malware infected these companies through the targeted emails, and connected to a remote server to retrieve and install the malware, which allowed attackers access to internal systems and to develop tailored payloads for each environment before setting off the attack. This portion of the kill chain was also especially important. During Stage 2 where the attackers were inside the network, they developed subtle differences in their plans to attack the specific hardware and capabilities of each of

the companies affected. For example, the differences in power distribution management systems, and taking down the uninterruptible power supply at one company's data center when their attack launched. The attackers also developed malicious firmware for serial to ethernet devices in order to execute drive wiping and prevent operator commands from being able to restore power grid function.

In order to prevent this type of attack from happening again, training against phishing and social engineering attacks would hopefully prevent a similar initial infection route from occurring. If the phishing attack were still successful however, IPS and antivirus software could help alert about a compromised machine, and multi factor authentication could prevent the use of stolen credentials to move laterally or impersonate a legitimate user. In addition, each trusted communication paths can be exploited by an attacker, so they should be evaluated for risk and better monitored for intrusion and data exfiltration. The defense methods in place like firewalls did not prevent remote access tools used in the later stages of the attack, as well as additional safeguards that could be put in place to further segment the ICS network from the IT systems. Credential theft enabled lateral movement and theft of Windows domain credentials allowed further penetration into the network by impersonating legitimate users. This access failed to raise alarms and demonstrates how insufficient access control can enable failures in the IT/OT boundary. The attackers also used drive wiping and log erasing to hide their tracks and other mechanisms to prevent defenders from figuring out the impact and with restoration efforts. Centralizing logs and increasing monitoring, as well as keeping backups and spares ready in case of a shutdown or attack can aid defenders in identifying forensic evidence and restoring functionality.

CRASHOVERRIDE/INDUSTOYER & Ukraine 2016

CRASHOVERRIDE was another malware attack on Ukraine's energy grid during the winter months of 2016. The initial infection was likely similar to the BlackEnergy attack, where spear phishing and credential gathering allowed attackers to enter the network, but little is known. From there

the attackers conducted reconnaissance and scanned devices to create a plan of attack. The attackers were able to remotely discover systems, monitor, and study the environment which enabled them to interface with SCADA devices and use ICS protocols directly during the attack such as IEC-101, IEC-104, IEC-61850 and OPC-DA13. (Slowik, 2019). The attackers were also able to build modular protocol aware malware to automate substation breaker operations, representing a step up in sophistication compared to the previous BlackEnergy attack where the attack commands were issued manually. CRASHOVERRIDE was also considered a failure because the intended scope of the attack was not entirely achieved. The more advanced attack did not persist as long or affect as many people as it could have.

The most critical parts of the ICS kill chain in the attack included the time the attackers had within the network to gather intelligence and develop a software approach that could automate attacks based on the specific ICS protocol stacks. The development stage also included carefully crafted wiper program that leveraged knowledge of the grid and vendors to end processes, as well as delete logs and files defenders needed in an attempt to make recovery and forensics more difficult. And while the damage could have been more severe due to the ineffective relay triggering, the protocol module to open the substation breakers was a significant portion of the action on objective.

It was likely that the CRASHOVERRIDE attackers had infiltrated the network well in advance of the attack and had conducted scans and masqueraded as legitimate users for some time. With more careful monitoring and logging efforts, the attackers might have been detected before the attack could get off the ground. The specific protocol interactions might be nonstandard for IDS tools, but in an ICS environment tools like Snort might be used to analyze traffic and find anomalies in ICS protocol interactions. The BlackEnergy malware had recently revealed the importance to MFA and locking down credentials but it seems the lessons had not been put into practice by the time the CRASHOVERRIDE attackers had infiltrated their target network. With better access control, limitations on remote capabilities, and auditing credentials and user commands, the attackers might

have been stopped or slowed down long enough to detect their activity in the planning and testing stages. Again, the BlackEnergy case that happened the year before this attack could have been a reminder to the power provider to maintain spares and backups as the attackers wiped files and attempted to thwart restoration and forensics.

While the BlackEnergy attack in 2015 was a separate incident, in 2016 Ukraine's energy grid was attacked using the CRASHOVERRIDE malware to manipulate ICS systems directly. This energy outage affected several thousand people over the course of about an hour. The CRASHOVERRIDE/INDUSTROYER malware was used in the Ukrainian Power Grid attack in 2016, so they are not two separate cases, but one.

Oldsmar Water Treatment Facility

In Florida, an unknown attacker was able to remotely connect to a water treatment facility and control the interface in order to change chemical settings. This attack was extremely simple and straightforward compared to well developed ICS attacks like Stuxnet and the multiple attempts at taking down Ukraine's energy grid. In this instance, an opportunistic hacker was able to log in to a TeamViewer remote desktop account at a water treatment plant using credentials found in a password leak. They used valid credentials to access the human machine interface, the desktop application controlling the water treatment activities, and with the mouse adjusted parameters such as chemicals being added to the water. They attempted to add a lethal amount of lye to the water supply, which was only noticed by an employee who saw mouse movement and realized something was wrong after the attacker tried to change a dangerous parameter. (U.S. Department of Energy, 2021b). There was no malware in this incident, and the one compromised account led to the attacker remotely connecting to one machine. Therefore there was no spread during this attack.

The attacker leveraged a compromised account, making the intrusion stage of the ICS kill chain the most important part of this attack. This was most likely opportunistic and meant that there was little

reconnaissance or development necessary. The attacker simply logged in, determined what they had access to in a brief targeting stage, and then they pivoted to action where they made changes to the parameters that could have the most impact to the target. The attacker did not appear to perform any extensive reconnaissance before acting. It raises real concerns that an opportunistic hacker could cause material harm with limited skills or familiarity with the targeted systems.

The defensive steps needed to counter this attack are also straightforward, but many small operations leave their defenses down for convenience or due to complacency. The remote desktop account should not have been allowed to remain logged in to a critical work station. But if remote desktop was required for monitoring, it should not be allowed to remain logged in or to accept connections without alerting or logging. And it certainly should not have been capable of changing critical parameters. The HMI system also did not require confirmation or a role based control to prevent the unauthorized changes. The attacker's outside IP address could have been caught with an intrusion system to determine that an outside actor was able to log in. Due to a data leak, the credentials were available to the attacker and others. Compromised accounts are often disclosed or able to be found in account compromise databases. While it may not have been disclosed in time for the water treatment plant to have known about it, checking accounts for compromise and acting quickly when data leaks are disclosed will shut the window on the ability for hackers to use valid credentials. MFA should also be used so that even in the event the valid credentials are found by an attacker, the additional security layer should prevent access. In this particular case, the attack was noticed by an employee physically observing the changes to the parameters. In addition to alerts for remote login activity, alerts and confirmation/delay for changes to the parameters could also prevent disastrous changes even by mistake to the configuration. For example there might be an acceptable window for the amount of lye, and anything outside that window should require an override or confirmation in some way. Other systems could also provide alerting when parameters exceed safe limits, such as a sensor that alerts when a large amount of chemicals being added to the water exceed a threshold.

Colonial Pipeline

The Colonial pipeline attack was an extortion attempt on US oil pipeline infrastructure in 2021. The pipeline company Colonial was targeted by the Darkside group or a user of their capabilities through ransomware as a service. The initial attack vector was a VPN that was misconfigured to not use MFA. Stolen credentials let attackers access Colonial's network where they could then move laterally and identify and access critical systems and shares. Darkside targeted the corporate data in a ransomware attack, attempting to extort the company with the data they exfiltrated prior to encrypting it on the company devices. This type of ransomware, a double extortion model, puts further pressure on targets who might have backups, but in this case the data was less critical than the potential threat to the pipeline and operations. Darkside was able to deploy the ransomware to business and operations servers and although pipeline operations were not directly targeted, the encrypted systems and risk involved caused Colonial to shut down the pipeline preemptively. (U.S. Department of Energy, 2022).

The initial access to the IT systems enabled the attack to propagate to critical systems, exfiltrate data, and cause enough of a concern that operations had to be suspended. In the privilege escalation and lateral movement after the initial infection, attackers were able to use Active Directory which provided valuable data to exfiltrate. C2 during the intrusion phase gathered hundreds of gigabytes of data such as accounting records and research. While the attackers actions targeted the IT systems, the ransomware had a significant risk and effect on the operations of the pipeline. Business and logistics functions were disrupted which led to the shutdown of the pipeline, which shows the cascading effects of IT systems on OT.

While the attack did not directly target ICS/OT systems, the risk to critical infrastructure and the lack of separation between IT/OT systems at the pipeline facility led to US federal policy considerations to prevent future attacks and impact. The VPN account should have had monitoring to reveal the external access and logins. The misconfigured VPN did not use MFA, which should be enabled as an extra layer of security for all possible accounts. The data exfiltration was in the hundreds

of gigabytes and could have been observed through monitoring or anomaly detection. Better defenses and separation between IT/OT environments could also lessen the impact of IT related attacks affecting business procedures and the OT devices. Zero trust architecture, access control systems, and stronger authentication should be employed in critical infrastructure. IT and OT networks should also be segmented or air gapped to prevent infection crossing the boundary. Intrusion detection such as IPS and monitoring with centralized logs could have revealed some of the activities and movement on the network.

Sources

Dragos Inc. (2017). *CRASHOVERRIDE: Analysis of the threat to electric grid operations*. Retrieved from https://www.dragos.com/resources/whitepaper/crashoverride-analyzing-the-malware-that-attacks-power-grids/

MITRE. (n.d.). *CRASHOVERRIDE (C0028)*. MITRE ATT&CK. Retrieved from https://attack.mitre.org/campaigns/C0028/

Mueller, P., & Yadegari, B. (2010). *The Stuxnet worm: A case study of ICS attack methods*.

Slowik, J. (2019). *CRASHOVERRIDE: Reassessing the 2016 Ukraine electric power event as a protection-focused attack*. Dragos Inc. https://www.dragos.com/resources/whitepaper/crashoverride-reassessing-the-2016-ukraine-electric-power-event-as-a-protection-focused-attack/

U.S. Department of Energy. (2021, November 18). *CyOTE case study: CRASHOVERRIDE/INDUSTROYER*. Cybersecurity for the Operational Technology Environment (CyOTE).

U.S. Department of Energy. (2022, February 8). *CyOTE case study: DarkSide*. Cybersecurity for the Operational Technology Environment (CyOTE).

U.S. Department of Energy. (2021, September 23). *CyOTE case study: Oldsmar water treatment facility.* Cybersecurity for the Operational Technology Environment (CyOTE).