SEC586: **Blue Team Operations: Defensive PowerShell**

6 Day Program	36 CPEs	Laptop Required

Who Should Attend

- Security perations Center analysts
- System engineers
- System administrators
- Technical security managers
- Cyber threat investigators
- Computer network defense analysts

You Will Be Able To

- Write scripts and ad hoc PowerShell as needed to solve cybersecurity use cases
- Read and expand existing tooling
- Harden systems using PowerShell
- Test for visibility gaps and misconfigurations in an automated fashion
- Integrate disparate systems to enable orchestration across various platforms
- Build advanced detections using PowerShell as the underlying platform
- Automate response initiatives before an incident occurs, enabling rapid response

This Course Will Prepare You To

- Automate many common tasks to focus efforts on additional areas for improvement
- Leverage a native, cross-platform technology to maximize protection
- Enhance protection, detection, and response capabilities using PowerShell
- Reduce time to detection and time to response when incidents do occur

Prerequisites

- Basic understanding of programming concepts
- Basic understanding of information security principles

Effective Blue Teams work to harden infrastructure, minimize time to detection, and enable real-time response to keep pace with modern adversaries. Automation is a key component to facilitate these capabilities, and PowerShell can be the glue that holds together and enables the orchestration of this process across disparate systems and platforms to effectively act as a force multiplier for Blue Teams. This course will enable Information Security professionals to leverage PowerShell to build tooling that hardens systems, hunts for threats, and responds to attacks immediately upon discovery.

PowerShell is uniquely positioned for this task of enabling Blue Teams. It acts as an automation toolset that functions across platforms and it is built on top of the .NET framework for nearly limitless extensibility. SEC586 maximizes the use of PowerShell in an approach based specifically on Blue Team use cases.

Students will learn:

- PowerShell scripting fundamentals from the ground up in terms of PowerShell's capabilities as a defensive toolset
- Ways to maximize performance of code across dozens, hundreds, or thousands of systems
- Modern hardening techniques using Infrastructure-as-Code principles
- How to integrate disparate systems for multi-platform orchestration
- PowerShell-based detection techniques ranging from Event Tracing for Windows to baseline deviation and deception
- Response techniques leveraging PowerShell-based automation

This course is meant to be accessible to beginners who are new to the PowerShell scripting language as well as to seasoned veterans looking to round out their skillset. Language fundamentals are covered in-depth, with hands-on labs to enable beginning students to become comfortable with the platform. For skilled PowerShell users who already know the basics, the material is meant to solidify knowledge of the underlying mechanics while providing additional challenges to further this understanding.

The PowerPlay platform built into the lab environment enables practical, hands-on drilling of concepts to ensure understanding, promote creativity, and provide a challenging environment for anyone to build on their existing skillset. PowerPlay consists of challenges and questions mapping back to and extending the course material.

Between the course material and the PowerPlay bonus environment, SEC586 students will leave the course well equipped with the skills to automate everyday cyber defense tasks. You will return to work ready to implement a new set of skills to harden your systems and accelerate your capabilities to more immediately detect and respond to threats.

Section Descriptions

SECTION 1: PowerShell Fundamentals

Even for seasoned PowerShell users, a deep and robust understanding of the language fundamentals can be incredibly powerful for writing more efficient, readable, and usable code. Section 1 of the course focuses on building a solid foundation upon which more complex use cases can then be constructed. With a focus on Blue Team specific functions, we ll frame the discussion around the PowerShell basics in terms that will be immediately useful for students. For example, common data structures are discussed as a fundamental aspect of PowerShell and immediately applied as Blue Team triage and analysis tactics. This base is built from the ground up and accessible to students with no prior scripting experience, but with enough nuance to shed light on the "why does it work this way" question for more seasoned PowerShell users. For professionals already familiar with the basic concepts, PowerPlay offers an interactive, out-of-band challenge system for students to drill various concepts and techniques related to the course material.

TOPICS: Getting to Know PowerShell; Blue Team Use Cases; Language Basics; PowerShell Environment; Debugging; Source Control

SECTION 2: Best Practices for Blue Teams

PowerShell-based automation provides a unique, cross-platform mechanism for improving Blue Teams ' speed of execution. This course section begins with a discussion on best practices to ensure code is highly functional, readable, and supportable. Students will leave with a deep understanding of how PowerShell works under the hood, but also with a sense of how to build tools that can be supported by team members less familiar with PowerShell. This section transitions into taking the fundamentals and executing them at scale. Next, a performance section addresses important aspects of PowerShell. The section continues into building integration with other systems. With modern API-driven orchestration, PowerShell can glue together multiple systems for better troubleshooting, investigation, detection, and response. This understanding can unlock functionality that would not otherwise be possible between disparate systems. Finally, protection, analysis, triage, and response techniques driven by PowerShell are enabled by Interactive Notebooks where analysts can combine documentation and executable code.

TOPICS: Best Practices; Remote Management; PowerShell Performance; Integrations; Interactive Notebooks

Who Should Attend

- Security Operations Center analysts
- System engineers
- System administrators
- Technical security managers
- Cyber threat investigators
- Certified network defender analysts

SECTION 3: Weaponizing PowerShell

Now that we have a strong understanding of the fundamentals, this course section focuses on ways to weaponize PowerShell both from an offensive and defensive perspective. The section begins with a focus on offensive PowerShell use cases. Threat actors have long used PowerShell as an attack platform, delivering fileless malware and living off the land using built-in capabilities. The next section turns this discussion around and focuses on the Blue Team aspects of controlling PowerShell execution. The section then dives deep into log analysis and data parsing and discovery. The goal is to maximize the utility of native features of operating systems and applications while fully understanding how to find important data. If Blue Teams can identify sensitive data in unexpected locations, those data can be handled or protected properly. The section concludes with a discussion of PowerShell as a platform to enable Blue Teams to work within DevOps development practices. As modern development teams transition practices, Blue Teams must adapt. Automation plays an important role in this process, as Blue Teams fight to scale capabilities to match modern development frameworks. PowerShell provides this automation platform and can be the catalyst to enable continuous assurance of critical business services.

TOPICS: Offensive PowerShell; Controlling PowerShell; Log Analysis; Text Parsing; DevOps

SECTION 4: Know and Protect Thyself

This course section focuses on better understanding one 's own environment, maximizing visibility and testing defensive capabilities using PowerShell. The section begins with in-depth discussions on hardening infrastructure and maximizing visibility and detection capabilities. For basics such as ensuring that proper access controls exist, the theory is simple. But using traditional techniques, scaling in practice is difficult. With an automation platform like PowerShell, hardening and auditing practices can be scaled with ease, providing consistent assurance. Next, Desired State Configuration, PowerShell s configuration-as-code utility, can be used to consistently define and configure infrastructure using PowerShell to help ensure system integrity. Additional hardening techniques are discussed based on maximizing native security functionality. The section then turns to improving understanding of visibility and detection capabilities in a repeatable format via automated testing techniques that provide for a reliable and repeatable means of measuring capabilities. The focus here is to use PowerShell as a testing utility to identify visibility and detection gaps both in preventive and detective controls, but also in operational processes. Finally, a common challenge faced by Blue Teams is the overwhelming amount of data generated by endpoints and security tooling. While large volume is meant to facilitate proper detection, it can be interpreted as noise and actually harm an organization s ability to detect threats. We ll discuss analysis techniques that use PowerShell to filter through some of this noise and provide the ability to make better decisions based on available data.

TOPICS: System Hardening; Desired State Configuration; Know Thyself; Analyzing Large Data Sets

SECTION 5: Detect and Respond

With hardening and protection mechanisms now having been covered, this course section focuses entirely on detection and response strategies enabled by PowerShell automation. Advanced detection techniques such as Event Tracing for Windows and deception on endpoints and the network are implemented to provide deep visibility and weaponize existing infrastructure against threat actors. These techniques can be automated at scale to turn a "normal" enterprise network into a mine field, providing deep visibility to Blue Teams while forcing an attacker to work even more slowly and methodically to evade detection. Baselining is layered on top of these techniques to provide an ability to understand normal operating circumstances and identify outliers from that dataset. Baseline deviation detection and file integrity monitoring techniques are implemented in a way that is supportable at scale and, of course, automated using PowerShell. The course section concludes by covering response techniques meant to maximize visibility and help an operations team better understand if anomalous conditions warrant further containment and investigation. Once malicious intent is identified, response techniques focused on containment can be automated to mitigate additional harm. Layering these response techniques inside of automation playbooks can ensure proper response, containing threats but also enabling teams to quickly identify false positives and avoid unnecessary end-user friction and business impact.

TOPICS: Event Tracing for Windows; Baselining; Automating Deception; Short-term Response – Visibility; Short-term Response – Containment

SECTION 6: Capstone: Defend the Flag

The final section of SEC586 focuses entirely on hands-on application of the skills built throughout the week. Working in teams, each group must solve challenges ranging from log analysis to containment tactics. Several different challenges with increasing levels of difficulty will require groups to work together, mastering PowerShell from the perspective of Blue Team workloads, and providing a safe environment to work with PowerShell while under pressure. Challenges will ensure a deep understanding of the concepts covered throughout SEC586 while offering a fun and competitive platform to test and further build these skills.