SEC505: Securing Windows and PowerShell Automation

GCWN
Windows Security Administrator
giac.org/gcwn

6 Day Program | 36 CPEs | Laptop Required

You Will Be Able To

- Write PowerShell scripts for security automation
- Execute PowerShell scripts on remote systems
- Harden PowerShell itself against abuse, and enable transcription logging for your SIEM
- Use PowerShell to access the WMI service for remote command execution, searching event logs, reconnaissance, and more
- Use Group Policy and PowerShell to grant administrative privileges in a way that reduces the harm if an attack succeeds (assume breach)
- Block the lateral movement of hackers and ransomware using Windows Firewall, IPsec, DNS sinkholes, admin credential protections, and more
- Prevent exploitation using AppLocker and other Windows OS hardening techniques in a scalable way with PowerShell
- Configure PowerShell remoting to use Just Enough Admin (JEA) policies to create a Windows version of Linux sudo and setuid root
- Configure mitigations against pass-the-hash attacks, Kerberos Golden Tickets, Remote Desktop Protocol (RDP) man-in-the-middle attacks, Security Access Token abuse, and other attacks discussed in SEC504 and other SANS hacking courses
- Install and manage a full Windows Public Key Infrastructure (PKI), including smart cards, certificate auto-enrollment, Online Certificate Status Protocol (OCSP) web responders, and detection of spoofed root Certificate Authentications (CAs)
- Harden essential protocols against exploitation, such as SSL, RDP, DNS, PowerShell Remoting, and SMB

WINDOWS SECURITY AUTOMATION MEANS POWERSHELL

In the SEC505 course you will learn how to:

- Write PowerShell scripts for Windows and Active Directory security automation
- Safely run PowerShell scripts on thousands of hosts over the network
- Defend against PowerShell malware, such as ransomware
- Harden Windows Server and Windows 10 against skilled attackers

In particular, we will use PowerShell to secure Windows against many of the attacks described in the MITRE ATT&CK matrix, especially against stolen administrative credentials, ransomware, hacker lateral movement inside the LAN, and insecure Windows protocols such as RDP and SMB.

You will leave this course ready to start writing your own PowerShell scripts to help secure your Windows environment. It's easy to find Windows security checklists, but how do you automate those changes across thousands of machines? How do you safely run scripts on many remote boxes? In this course you will learn not just Windows and Active Directory security, but how to manage security using PowerShell.

DON'T JUST LEARN POWERSHELL SYNTAX, LEARN HOW TO LEVERAGE POWERSHELL AS A FORCE MULTIPLIER FOR WINDOWS SECURITY

There is another reason PowerShell has become popular: PowerShell is just plain fun! You will be surprised at how much you can accomplish with PowerShell in a short period of time, it's much more than just a scripting language, and you don't have to be a coding guru to get going.

Learning PowerShell is also useful for another kind of security: job security. Employers are looking for IT people with PowerShell skills. In response, even though SEC505 has had at least one day of PowerShell instruction for more than 10 years, today PowerShell is the centerpiece of the course. You don't have to know any PowerShell to attend this course, we will learn it together during the labs.

You can learn basic PowerShell syntax on YouTube for free, but this week goes far beyond syntax. In this course we will learn how to use PowerShell as a platform for managing security, as a "force multiplier" for the Blue Team, and as a rocket booster for your Windows IT career.

WE WILL WRITE A POWERSHELL RANSOMWARE SCRIPT AND DEFEND AGAINST IT

Unfortunately, PowerShell is being abused by hackers and malware authors. On the last day of the course, we will write our own ransomware script to see how to defend against scripts like it.

This is a fun course and a real eye-opener, even for Windows administrators with years of experience. Come have fun learning PowerShell and Windows security at the same time.

"In SEC505, real-life solutions are offered by someone who understands the roadblocks in the way. This is information I could implement tomorrow and make my network more secure."

— Mary Becken, **Egan Company**

Available Training Formats

Live Training

Live Events
sans.org/information-security-training/by-location/all

Summit Events
sans.org/cyber-security-summit

Private Training
sans.org/private-training

Online Training

OnDemand
sans.org/ondemand

Simulcast
sans.org/simulcast

Section Descriptions

SECTION 1: Learn PowerShell Scripting for Security

The first section of the course covers what you need to know to get started using PowerShell. You don't need to have any prior scripting or programming experience. We have PowerShell labs throughout the week, so today is not the only PowerShell material. We start with the essentials, then go more in-depth as the course progresses. Don't worry, you won't be left behind, the PowerShell labs walk you through every step. If you already have PowerShell experience, then there will also be intermediate topics for you.

TOPICS: Write Your Own PowerShell Scripts and Functions; Flow Control; PowerShell Core vs. Windows PowerShell; Piping Objects, Not Text; Classes, Objects, Properties, Methods; Importing Modules; Exporting Sata to CSV and XML Files

SECTION 2: You Don't Know the POWER!

How can we run PowerShell scripts on thousands of systems with just a few lines of code? This day is about remote command execution using PowerShell Remoting, the Secure Shell (SSH) service on Windows, the Task Scheduler service, and boot-up scripts assigned through Group Policy. SSH is not just for Linux. Windows now has built-in support for it as both a client and server. We can use Just Enough Admin (JEA) policies to block all remoting commands by default, then only allow the commands and arguments we want. And with JEA, PowerShell Remoting commands are magically (and safely?) elevated to administrator status, similar to setuid root on Linux.

TOPICS: PowerShell Remote Command Execution of Scripts; SSH for Windows; Smart Card Authentication for PowerShell Remoting; SSL/TLS Encryption vs. SSH; Key-based Authentication for SSH; Just Enough Admin Remoting; Running PowerShell Scripts as SYSTEM with the Task Scheduler; Group Policy-assigned Startup Scripts

Who Should Attend

- Anyone who wants to learn PowerShell automation
- "Ops" personnel in SecOps/DevOps
- Blue Team players who were terrified by SEC504
- Windows endpoint and server administrators
- Anyone implementing the CIS Critical Security Controls
- Anyone implementing the MITRE ATT&CK mitigations

Course Preview

available at: **sans.org/demo**

SECTION 3: PowerShell for WMI and Active Directory

PowerShell is deeply integrated into the Windows Management Instrumentation (WMI) service. Hackers love the WMI service too, but for the wrong reasons. The WMI service is enabled by default and accessible over the network. Active Directory (AD) is also manageable through PowerShell: we can find abandoned user accounts and disable them, enforce desired group memberships, reset passwords, and more. What about local admin accounts? Don't use Microsoft LAPS! There are better ways to protect admin passwords, including a PowerShell solution we will implement in a lab. With proper AD permissions and auditing, combined with PowerShell JEA on jump servers, we can delegate IT authority in AD much more safely.

TOPICS: PowerShell for WMI; PowerShell for Active Directory; WMI Namespaces and Classes; Remote Command Execution with WMI; WMI and AD Reconnaissance; AD Permissions and Auditing; A Better (and Free) Alternative to Microsoft LAPS

SECTION 4: Hardening Network Services with PowerShell

On day four we will use PowerShell and Group Policy to automate the hardening of many exploitable services and protocols, such as DNS, RDP, and SMB. We will see how to use PowerShell to install roles, manage services, and accomplish other security tasks for DevOps. For example, firewall rules and IPsec policies can be applied through PowerShell. The trick is to be able to apply different sets of firewall and IPsec rules to different sets of machines in a scalable, repeatable, and automated way, hence the need for PowerShell. IPsec is not just for Virtual Private Networks (VPNs)! In fact, we won't discuss VPNs at all in this section. The built-in Windows IPsec driver can authenticate users in Active Directory in order to implement share permissions for our TCP/UDP listening ports based on our users' global group memberships in Active Directory. If you move to Server Core or Server Nano, knowing PowerShell becomes a job requirement.

TOPICS: PowerShell for Server and Workstation Automation, DevOps Style; PowerShell for Firewall Rules and IPsec Policies; IPsec Is Built in and Not Just for VPNs; Automate the Hardening of Server Core, Server Nano, and Other Stand-alone Machines; PowerShell Management of Networking Settings

SECTION 5: Multi-Factor Authentication with Smart Cards and Smart Tokens

Smart cards and smart tokens, like YubiKeys, are the gold standard for multi-factor authentication (MFA). On day five, we will use PowerShell to install a certificate server that can be used to deploy smart cards and smart USB tokens. Smart cards and tokens can be used for PowerShell Remoting, signing PowerShell scripts, Remote Desktop Protocol (RDP) logons, Virtual Private Networks, and more. Everything you need to roll out a full smart card/token solution for your administrators is included with Windows. If you have a Trusted Platform Module (TPM) chip in your laptop or tablet, the TPM can also be used as a built-in smart card. TPMs also protect biometric data, encrypt BitLocker keys, and help enhance Windows 10 Credential Guard. PowerShell Remoting network traffic can be encrypted with SSL/TLS. The target server is authenticated with its certificate, just like a web server using HTTPS, but how do you install SSL/TLS certificates on thousands of workstations and servers? We will show you how to do that.

TOPICS: Smart Cards and Smart Tokens Such as YubiKeys; Smart Cards for PowerShell Remoting; SSL/TLS Encryption for PowerShell Remoting and Other Web Services; TPM Virtual Smart Cards; TPM for Biometrics and Credential Guard; Certificate Auto-enrollment; Online Certificate Status Protocol Revocation Checking

SECTION 6: Capstone: PowerShell Security, Ransomware, and DevOps

On day six we will write a PowerShell ransomware script and unleash it inside our training Virtual Machine (don't release it into the wild, you'll go to federal prison). The purpose of this ethical hacking is to discuss defenses: how can we secure PowerShell itself? PowerShell is not a single tool, so we must deploy several defenses in-depth. Most importantly, we must prevent the compromise of Domain Admin credentials. PowerShell hacking tools and ransomware with Domain Admin credentials are nearly unstoppable! As a capstone to pull together the week's material, we will also create a DevOps-style script to reconfigure nearly all the security features discussed during the week. The goal is to have an all-in-one script that can ideally reconfigure everything. Soon, we will all need to be "full stack engineers" for automation (and job security).

TOPICS: PowerShell Ransomware; PowerShell Security; PowerShell Transcription Logging of Commands; AppLocker Rules for PowerShell; Securing Domain Admin Credentials; Windows 10 Credential Guard; Privileged Access Workstations (PAWs) and Jump Servers; Capstone DevOps Script for Automation