(T) = theory; (P) = practical demo

- About me (T)
- About presentation (T)
- Agenda (T)
- What is PowerShell and Why use it (T)
    - System.Management.Automation.dll (T)
    - Default on Windows (T)
    - Integration with all MS products (T)
    - .NET (T)
    - COM (T)
    - Registry (T)
    - APIs (T)
    - WMI (T)
    - Compiles C# on the fly (T)
    - Trusted by AV (T)
    - Etc. (T)
- Background Info (T)
    - Typical PowerShell command (P)
    - -encodedCommand param for powershell.exe (T)
    - PSRemoting (similar to ssh) (T)
    - WMI (T)
        - Architecture (T)
        - Classes (T)
        - Instances (T)
        - WQL (similar to SQL, but read-only) (T)
        - Example WMI query (P)
        - WMI events (T)
- Reverse Shell (T) + (P)
- Wrap .ps1 (powershell script extension) in Exe (P)
- VirusTotal (T) + (P)
- Listener (T) + (P)
- Initial exploit (P)
- The fun begins ! (P)
- Perform Host Recon : WMI + .NET, some info can be obtained in multiple ways (T) + (P)
    - Username (P)
    - Computername (P)
    - Domain (P)
    - Groups that current user is member of (P)
    - Users in Domain Admin group (P)
    - IP configuration (P)
    - Listening Network Connections (P)

  - Established Network Connections (P)
  - DNS Cache (P)
  - Antivirus Product Name and Status (P)
  - Domain Controller (P)
  - Operating System (P)
  - Computer System (hardware) (P)
  - BIOS (P)
  - Installed Software packages (P)
  - Domain Computers (P)
  - Domain Users (P)
  - Context privileges (P)
- Perform network mapping  (T) + (P)
  - Send ARP Request (P)
  - Ping (P)
  - TCP Port Scan (P)
- UAC (T)
- Bypass UAC (P)
  - By Registry (T)
  - Registry Size too small for our payload (T)
  - Transfer File – Write Raw Bytes to Disk (P)
  - Modify Registry To Achieve goal (P)
  - Open second listener (P)
  - Execute payload (P)
- Get context privileges from second shell (P)
- Clean Registry used for bypassing UAC (P)
- Get Persistence (T) + (P)
  - WMI Permanent Events (T) + (P)
    - Event Filter (T) + (P)
    - Consumer (T) + (P)
      - Our payload is reverse shell (P)
    - Filter to Consumer Binding (T) + (P)
- How to get domain admin (T)
- Scheduled Task (T)
  - Can be configured to run with Highest privileges of other user / group (T)
  - Triggered At Log on to work for our purpose (T)
  - Need to trigger a domain admin user to log on (T)
- Block favorite program (Code.exe) from running (T) + (P)
  - WMI Permanent Events (P)
- Create scheduled task (P)
  - Trigger (P)
  - Context under which it will run (P)
  - Action (P)
    - Connect to DC (P)

                - Create new user under Domain Admins group (P)
            - Clean WMI Permanent Event that blocks program (P)
            - Remove scheduled task (P)

- Dev/User is annoyed by favorite program not running (P)
- Calls Helpdesk/Admin to investigate (P)
- Helpdesk/Admin uses remote desktop to log on and investigate (P)
- Everything works fine for Helpdesk/Admin (P)
- Helpdesk/Admin ignores Dev/User for the rest of the day for wasting his/her time (P)
- Attacker has new user under Domain Admins group (P)
- Uses PSRemoting to execute commands on domain controller (T) + (P)
    - Create session (T) + (P)
    - Perform Host Recon (P)
    - Disables Windows Defender (P)
    - Can create Reverse shell Persistence (!)
    - Can do anything (!) - Got keys to the kingdom
- PowerShell Malware (T)
    - PowerWorm
    - Powerliks
    - PowerSniff
    - PowerWare – ransomware
    - POSHY (APT29)
    - Stuxnet – no powershell but uses WMI
- Defenses (T)
    - PowerShell v5
    - Logging
    - JEA
    - AMSI
    - AppLocker
    - Constrained Language Mode
    - Autoruns from Sysinternals (context speficic)