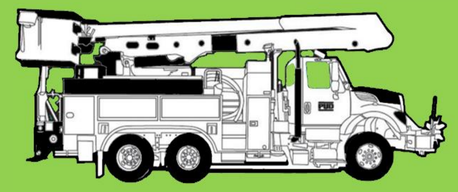
ITS Student Engineer Internship



By: Kayla Bachler Mentor: Ryan Huffman Advisor: Rob Nash, Senior Lecturer in CSS

Company Overview

As the second largest publicly owned utility in Washington, Snohomish County Public Utility District No.1 (PUD) serves over 350,000 electric and 21,000 water customers. Their service territory covers over 2,200 square miles, including all of Snohomish County and Camano Island.

My Role

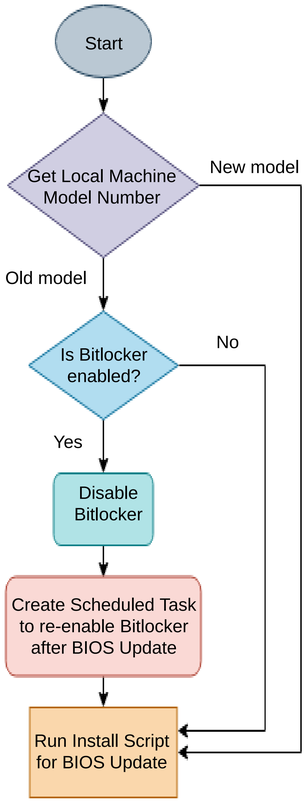
As an Information Technology Services (ITS) Student Engineer Intern, I worked with the ITS Infrastructure Support Team which specializes in maintaining the hardware and software associated with every computer in the district. A major part of the system maintenance was the mitigation of hardware-level threats (including Meltdown & Spectre) that were present during my internship.

What is Meltdown & Spectre?

Meltdown & Spectre are flaws in the design of CPUs that pose major security risks, including:

- Applications accessing unrelated memory segments, including the kernel
- Virtual Machines reading shared-host memory allocated to other VMs

BIOS Update Script: Meltdown & Spectre

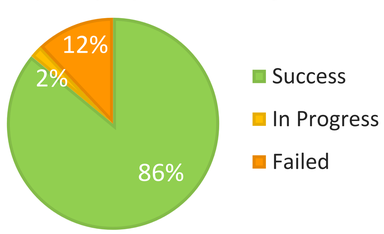



```
Push-Location $PSScriptRoot\..  # Go to directory where script exists
$logfile = ".\logs\$env:COMPUTERNAME.txt"  # Logging txt file for script outputs
$pState = gwmi -Namespace root/CIMV2/Security/MicrosoftVolumeEncryption -class Win32_EncryptableVolume
$modelpath = (Get-WmiObject Win32_Computersystem).model  # Model of local machine
$batteryErrorCode = 228377 #Stands for BATERR

# If the laptop is not connected to AC adapter return error code -1
if ((Get-WmiObject Win32_Battery).batterystatus -eq "1") {
    Exit $batteryErrorCode
}

# If the local machine is one of these models: z220, z230, 9470m or z420, step into
if ($modelpath -eq "z220" -or $modelpath -eq "z230" -or $modelpath -eq "z420" -or $modelpath -eq
"9470m"){
  "$modelpath is an older model (z220, z230, 9470m or z420)" | Out-File -FilePath $logfile -Append
   if ($pState -ne $null -and $pState -ne 0) {  # Local machine is protected/bitlocker enabled
     $pStatus = $pState.GetProtectionStatus().protectionstatus  # 0=Decrypted, 1=Encrypted, 2=Unknown
     if ($pStatus -eq 1) {  # If the PC is encrypted, disable Bitlocker
        Start-Process -FilePath -Wait ".\Scripts\DisableBitlocker.cmd" -Verb RunAs
        $Hostname = $env:COMPUTERNAME
        $taskRunAsuser = "SYSTEM"
        $service = New-Object -ComObject("Schedule.Service")
        $service.Connect($Hostname)
        $rootFolder = $service.GetFolder("\")
        $taskDefinition = $service.NewTask(0)
        $regInfo = $taskDefinition.RegistrationInfo
        $regInfo.Description = 'Enable Bitlocker after BIOS update'
        # Create Triggers for the task - Our trigger is run task at startup
        $triggers = $taskDefinition.Triggers
        $trigger = $triggers.Create(8)
        $trigger.Id = "StartUpTriggerId"
        $trigger.Enabled = $True
        # Create Actions for the task. Our action is to call the EnableBitlocker.cmd
        $command = ".\Scripts\HP\Meltdown_Spectre\Scripts\EnableBitlocker.cmd"
        $Action = $taskDefinition.Actions.Create(0)
        $Action.Path = $command
        # Create Task for Task Scheduler - Returns task creation-status as failed or succeeded
        try {
          $res = $rootFolder.RegisterTaskDefinition
                 ("EnableBitlocker", $taskDefinition, 2, $taskRunAsuser, $taskRunasUserPwd,2) | Out-String
        } catch {
          "ERROR: Failed while attempting to create Scheduled Task: "| Out-File -FilePath $logfile -Append
           $_ | Out-File -FilePath $logfile -Append
        }
      }
    }
    # Run install.cmd script for BIOS upgrade
    Start-Process -Wait -FilePath ".\Content\$modelpath\install.cmd“
} else {
  # Run install for updated BIOS - Newer models: z240, z440, 9480m or 840G3
  Start-Process -Wait -FilePath ".\Content\$modelpath\install.cmd“
}
```

Script Deployment through SCCM

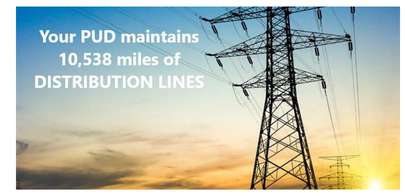


Results

After deploying our script through System Center Configuration Manager (SCCM) to 50 test computers, we achieved a high success rate of 86%. The remaining failed deployments were exclusively linked to laptops on battery saver mode, while in progress statuses were due to user interference during the script's execution.

Professional Development

- Gained experience collaborating in a professional team environment
- Increased PowerShell skill level from novice to proficient
- Implemented and documented isolated deployment tests
- Utilized SCCM for package deployment and monitoring



Summary

Contributions

- ✓ Designed a BIOS update script for Meltdown & Spectre vulnerability
- ✓ Tested and deployed the BIOS update script to all district PCs
- ✓ Wrote more than 20 additional scripts for software installation & removal
- ✓ Replaced and upgraded more than 200 PCs
- ✓ Completed annual physical inventory

Outcomes & Results

Updating the BIOS on affected systems fixes security-relevant issues. By deploying our script, we were able to mitigate vulnerabilities caused by Meltdown & Spectre while additionally improving the security procedures involving sensitive corporate data.