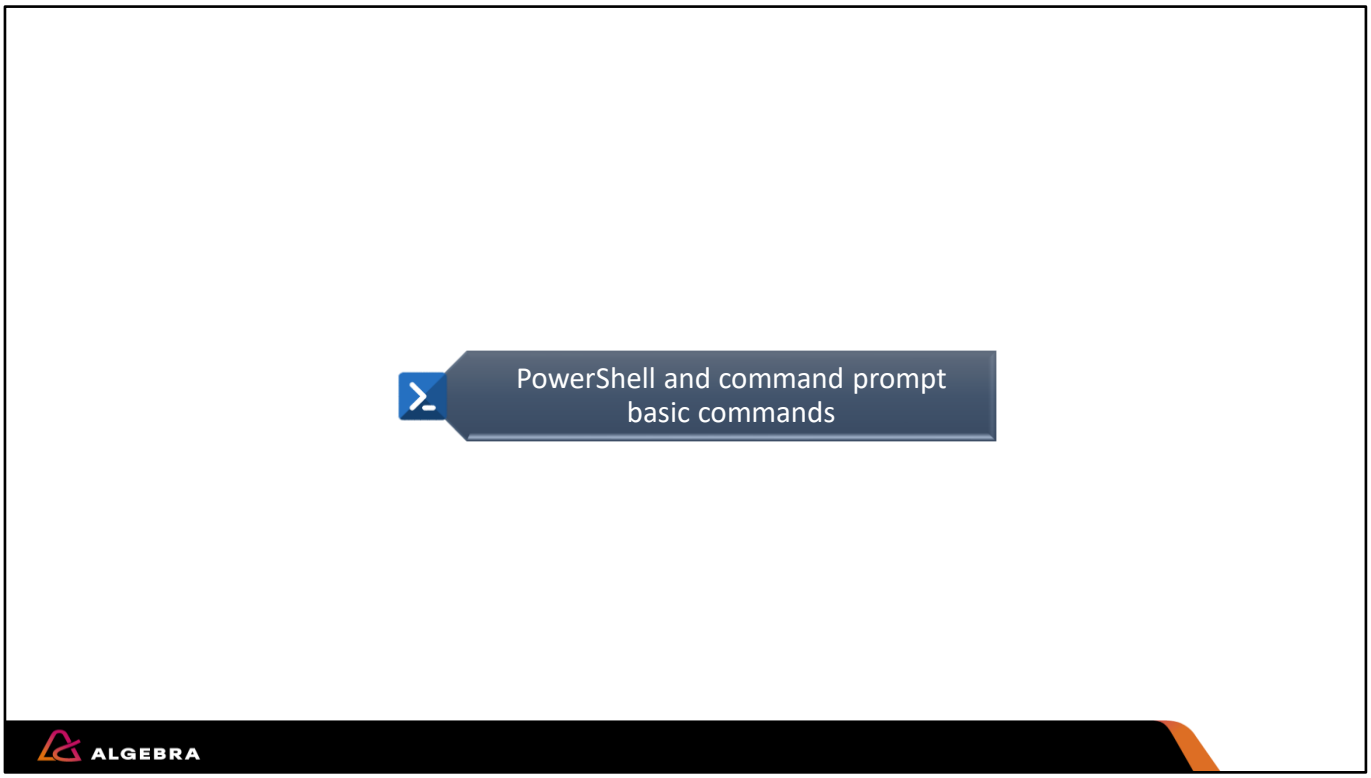
Windows basics

ROBERT PETRUNIĆ



Command prompt: Networking tools

ipconfig

- IP troubleshooting and configuration, MAC address
- Example: `ipconfig /all`

nslookup

- DNS troubleshooting and information, zone transfer
- It works in interactive or non-interactive mode
- Example (noninteractive mode): `nslookup www.petrunic.com`

arp

- ARP to IP mapping, add/remove static mappings
- Example: `arp -a`

Command prompt: Networking tools

Ping

- Network connectivity testing
- Example: `ping www.eduron.info`

Tracert

- Network connectivity testing, identifying routers between nodes
- Example: `tracert www.eduron.info`

telnet

- Port testing, had to be enabled on windoows Vista+
- Example: `telnet www.eduron.info 80`

ALGEBRA

Command prompt: Networking tools

netstat

- socket testing, listening and active, process to socket mapping
- Example: `netstat -abno` (b requires elevated prompt)

nbtstat

- Netbios testing
- Example: `nbtstat -a www.eduron.info`

route

- Route configuration and testing
- Example: `route add 10.10.10.0 MASK 255.255.255.0 192.168.168.254`

ALGEBRA

Command prompt: Managing services

sc

- Manipulating services from command line
- Example : `sc queryex`

netsh

- Manipulating many servces including ip forwarding
- It works in interactive or non-interactive mode
- Example: `netsh wlan show profiles`

Command prompt: File transfer

ftp (File Transfer Protocol)

- Transferring files
- Example: `ftp www.eduron.info`
  - Type username and password, get file1.txt

tftp (Trivial FTP)

- Transferring files without authentication
- Example: `tftp -i ftp.eduron.info GET file1.txt`

Command prompt: scripting

Wscript

- Windows GUI scripting host – running VB scripts
- Example: `wscript file1.vbs`

Cscript

- Windows command prompt scripting – running VB scripts
- Example: `cscript file1.vbs`

Command prompt: tasks and settings

tasklist

- Command line tool giving output similar to task manager
- Example: `tasklist /svc` (this will show modules running under svchost)

wmic

- Windows management instrumentation (WMI) client
- Powerful tool to enumerate windows settings

Command prompt: Net commands

Net commands:

- Net user – view, create new user, change user password
  - Example: `net user robert Pa$$w0rd /add`
- Net group (localgroup) – view, add/remove user from group
  - Example: `net localgroup administrators robert /add`
- Net file – files opened via network connection
  - Example: `net file`
- Net session – active network sessions
  - Example: `net sessions`

ALGEBRA

Command prompt: Net commands

Net commands:

- Net share – creating a share
  - Example: `net share myshare=C:\MyFiles`
- Net view – view network shares
  - Example: `net view`
- Net start – viewing or starting services
  - Example: `net start server`
- Net stop – stopping services
  - Example: `net stop server`
- Net time – checking network time
  - Example: `net time`

Command prompt: Net commands

Net commands:

- Net statistics – view server or workstation statistics
  - Example: `net statistics server`
  - Example: `net statistics workstation`
- Net helpmsg – prints net command error messages exaplined by message index
  - Example: `net helpmsg 33`

PowerShell: Execution policy

Allows or disallows the scripts execution (typing commands still works)

- Get-ExecutionPolicy
- Set-ExecutionPolicy

Options:

- **AllSigned** – only digitally signed scripts by a trusted publisher are allowed to be run (including local scripts)
- **Bypass** - nothing is blocked and there are no warnings or prompts
- **Default**
  - **Restricted** for Windows clients
  - **RemoteSigned** for Windows servers.
- **RemoteSigned** – only digitally signed scripts by a trusted publisher are allowed to be run (local scripts allowed)
- **Restricted** – allows individual commands to be run, but blocks all scripts and configuration files
- **Undefined** – no policy defined – defaults used (Restricted for Windows clients, and RemoteSigned for Servers)
- **Unrestricted** – allowed unsigned scripts (default policy for non-windows computers)

ALGEBRA

us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.1

Powershell: basic commands

Get-Help

- Ex: Get-help Get-process

Get-Member

- Ex: Get-Member -InputObject user

Get-Process

- Get-Process -Name powershell_ise

Stop-Process

- Get-Process -Name notepad | Stop-Process -WhatIf

ALGEBRA

objects?view=powershell-7.1

Powershell: basic commands

Get-Service

- Get-service –name wscsvc

Start-Service

- Start-service wscsvc -whatif

Stop-Service

- Stop-service wscsvc -whatif

ALGEBRA

objects?view=powershell-7.1

PowerShell network commands

Check IP and adapter configuration

- Get-NetIPConfiguration
- Get-NetIPInterface
- Get-NetIPAddress
- Get-NetAdapter
- Get-NetConnectionProfile

Check routing table

- Get-NetRoute

nslookup, DNS cache, DNS client configuration

- Resolve-DnsName -Name "play.h4ck3r.one"
- Clear-DnsClientCache
- Get-DNSClientCache
- Get-DNSClientServerAddress
- Get-DnsClient

ALGEBRA

troubleshooting/
https://www.channelpronetwork.com/blog/entry/powershell-ip-commands
http://blog.radunchev.com/2018/07/08/powershell-networking-commands-windows/

PowerShell network commands

Ping:

- Test-NetConnection -ComputerName 1.1.1.1

Traceroute

- Test-NetConnection play.h4ck3r.one –traceroute –verbose

Netstat

- Get-NetTCPConnection -State Established

Check TCP connections (open sockets)

- Get-NetTCPConnection

Test if the port is open

- Test-NetConnection play.h4ck3r.one -Port 9001

ALGEBRA

PowerShell network commands

Disable/Enable NET adapter

- Disable-NetAdapter -Name "Adapter Name"
- Enable-NetAdapter -Name Adapter Name"

Set DNS configuration

- Get-NetAdapter
- Set-DnsClientServerAddress -InterfaceAlias "Ethernet" -ServerAddresses "1.1.1.1","8.8.8.8"
- Get-DnsClientSorverAddress

Set IP address configuration

- New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 10.10.10.10 -PrefixLength "24" -DefaultGateway 10.10.10.1

ALGEBRA

Environment variables

$PSVersionTable

- gives you the PS version

Get-ChildItem -Path Env:

- gives you all env variables

$env:temp or Get-ChildItem -Path Env:temp

- gives you the temp folder location