T1562.004 - Impair Defenses: Disable or Modify System Firewall

Description from ATT&CK

> Adversaries may disable or modify system firewalls in order to bypass controls limiting network usage. Changes could be disabling the entire mechanism as well as adding, deleting, or modifying particular rules. This can be done numerous ways depending on the operating system, including via command-line, editing Windows Registry keys, and Windows Control Panel.
> Modifying or disabling a system firewall may enable adversary C2 communications, lateral movement, and/or data exfiltration that would otherwise not be allowed. For example, adversaries may add a new firewall rule for a well-known protocol (such as RDP) using a non-traditional and potentially less securitized port (i.e. Non-Standard Port).(Citation: change_rdp_port_conti)
>
> Adversaries may also modify host networking settings that indirectly manipulate system firewalls, such as interface bandwidth or network connection request thresholds.(Citation: Huntress BlackCat) Settings related to enabling abuse of various Remote Services may also indirectly modify firewall rules.

Atomic Tests

Atomic Test #1 - Disable Microsoft Defender Firewall

Disables the Microsoft Defender Firewall for the current profile. Caution if you access remotely the host where the test runs! Especially with the cleanup command which will re-enable firewall for the current profile...

Supported Platforms: Windows

auto_generated_guid: 88d05800-a5e4-407e-9b53-ece4174f197f

Attack Commands: Run with `command_prompt`!

```
netsh advfirewall set currentprofile state off
```

Cleanup Commands:

```
netsh advfirewall set currentprofile state on >nul 2>&1
```

Atomic Test #2 - Disable Microsoft Defender Firewall via Registry

Disables the Microsoft Defender Firewall for the public profile via registry Caution if you access remotely the host where the test runs! Especially with the cleanup command which will re-enable firewall for the current profile...

Supported Platforms: Windows

auto_generated_guid: afedc8c4-038c-4d82-b3e5-623a95f8a612

Attack Commands: Run with `command_prompt`!

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Paramet
```

Cleanup Commands:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Paramet
```

Atomic Test #3 - Allow SMB and RDP on Microsoft Defender Firewall

Allow all SMB and RDP rules on the Microsoft Defender Firewall for all profiles. Caution if you access remotely the host where the test runs! Especially with the cleanup command which will reset the firewall and risk disabling those services...

Supported Platforms: Windows

auto_generated_guid: d9841bf8-f161-4c73-81e9-fd773a5ff8c1

Attack Commands: Run with `command_prompt`!

```
netsh advfirewall firewall set rule group="remote desktop" new enable=Yes
netsh advfirewall firewall set rule group="file and printer sharing" new enable=Ye
```

Cleanup Commands:

```
netsh advfirewall reset >nul 2>&1
```

Atomic Test #4 - Opening ports for proxy - HARDRAIN

This test creates a listening interface on a victim device. This tactic was used by HARDRAIN for proxying.

reference: https://www.us-cert.gov/sites/default/files/publications/MAR-10135536-F.pdf

Supported Platforms: Windows

auto_generated_guid: 15e57006-79dd-46df-9bf9-31bc24fb5a80

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
netsh advfirewall firewall add rule name="atomic testing" action=allow dir=in prot
```

Cleanup Commands:

```
netsh advfirewall firewall delete rule name="atomic testing" protocol=TCP localpor
```

Atomic Test #5 - Open a local port through Windows Firewall to any profile

This test will attempt to open a local port defined by input arguments to any profile

Supported Platforms: Windows

auto_generated_guid: 9636dd6e-7599-40d2-8eee-ac16434f35ed

Inputs:

Name	Description	Type	Default Value
local_port	This is the local port you wish to test opening	integer	3389

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
netsh advfirewall firewall add rule name="Open Port to Any" dir=in protocol=tcp lo
```

Cleanup Commands:

```
netsh advfirewall firewall delete rule name="Open Port to Any" | Out-Null
```

Atomic Test #6 - Allow Executable Through Firewall Located in Non-Standard Location

This test will attempt to allow an executable through the system firewall located in the Users directory

Supported Platforms: Windows

auto_generated_guid: 6f5822d2-d38d-4f48-9bfc-916607ff6b8c

Inputs:

Name	Description	Type	Default Value
exe_file_path	path to exe file	path	PathToAtomicsFolder\T1562.004\bin\AtomicTest.exe

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
Copy-Item "#{exe_file_path}" -Destination "C:\Users\$env:UserName" -Force
netsh advfirewall firewall add rule name="Atomic Test" dir=in action=allow program
```

Cleanup Commands:

```
netsh advfirewall firewall delete rule name="Atomic Test" | Out-Null
Remove-Item C:\Users\$env:UserName\AtomicTest.exe -ErrorAction Ignore
```

Atomic Test #7 - Stop/Start UFW firewall

Stop the Uncomplicated Firewall (UFW) if installed.

Supported Platforms: Linux

auto_generated_guid: fe135572-edcd-49a2-afe6-1d39521c5a9a

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
ufw disable
```

Cleanup Commands:

```
ufw enable
ufw status verbose
```

Dependencies: Run with `sh`!

Description: Check if ufw is installed on the machine.

Check Prereq Commands:

```
if [ ! -x "$(command -v ufw)" ]; then echo -e "\n***** ufw NOT installed *****\n";
if echo "$(ufw status)" |grep -q "inactive"; then echo -e "\n***** ufw inactive **
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #8 - Stop/Start Packet Filter

Stop the Packet Filter if installed.

Supported Platforms: Linux

auto_generated_guid: 0ca82ed1-0a94-4774-9a9a-a2c83a8022b7

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
service pf stop
service pf disable
```

Cleanup Commands:

```
service pf enable
service pf start
service pf status
```

Dependencies: Run with `sh`!

Description: Check if pfctl is installed on the machine.

Check Prereq Commands:

```
if [ ! -x "$(command -v pfctl)" ]; then echo -e "\n***** PF NOT installed *****\n"
if [ "$(kldstat -n pf)" = "" ]; then echo -e "\n***** PF inactive *****\n"; exit 1
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #9 - Stop/Start UFW firewall systemctl

Stop the Uncomplicated Firewall (UFW) if installed, using systemctl.

Supported Platforms: Linux

auto_generated_guid: 9fd99609-1854-4f3c-b47b-97d9a5972bd1

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
systemctl stop ufw
```

Cleanup Commands:

```
systemctl start ufw
systemctl status ufw
```

Dependencies: Run with `sh`!

Description: Check if systemctl and ufw is installed on the machine.

Check Prereq Commands:

```
if [ ! -x "$(command -v systemctl)" ]; then echo -e "\n***** systemctl NOT installe
if [ ! -x "$(command -v ufw)" ]; then echo -e "\n***** ufw NOT installed *****\n";
if echo "$(ufw status)" |grep -q "inactive"; then echo -e "\n***** ufw inactive ***
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #10 - Turn off UFW logging

Turn off the Uncomplicated Firewall (UFW) logging.

Supported Platforms: Linux

auto_generated_guid: 8a95b832-2c2a-494d-9cb0-dc9dd97c8bad

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
ufw logging off
```

Cleanup Commands:

```
ufw logging low
ufw status verbose
```

Dependencies: Run with `sh`!

Description: Check if ufw is installed on the machine and enabled.

Check Prereq Commands:

```
if [ ! -x "$(command -v ufw)" ]; then echo -e "\n***** ufw NOT installed *****\n";
if echo "$(ufw status)" |grep -q "inactive"; then echo -e "\n***** ufw inactive **
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #11 - Add and delete UFW firewall rules

Add and delete a rule on the Uncomplicated Firewall (UFW) if installed and enabled.

Supported Platforms: Linux

auto_generated_guid: b2563a4e-c4b8-429c-8d47-d5bcb227ba7a

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
ufw prepend deny from 1.2.3.4
ufw status numbered
```

Cleanup Commands:

```
{ echo y; echo response; } | ufw delete 1
ufw status numbered
```

Dependencies: Run with `sh`!

Description: Check if ufw is installed on the machine and enabled.

Check Prereq Commands:

```
if [ ! -x "$(command -v ufw)" ]; then echo -e "\n***** ufw NOT installed *****\n";
if echo "$(ufw status)" |grep -q "inactive"; then echo -e "\n***** ufw inactive **
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #12 - Add and delete Packet Filter rules

Add and delete a rule on the Packet Filter (PF) if installed and enabled.

Supported Platforms: Linux

auto_generated_guid: 8b23cae1-66c1-41c5-b79d-e095b6098b5b

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
echo "block in proto tcp from 1.2.3.4 to any" | pfctl -a pf-rules -f -
pfctl -a pf-rules -s rules
```

Cleanup Commands:

```
pfctl -a pf-rules -F rules
sed -i "" '/anchor pf-rules/d'
```

```
pfctl -f /etc/pf.conf
```

Dependencies: Run with `sh`!

Description: Check if pf is installed on the machine and enabled.

Check Prereq Commands:

```
if [ ! -x "$(command -v pfctl)" ]; then echo -e "\n***** PF NOT installed *****\n"
if [ "$(kldstat -n pf)" = "" ]; then echo -e "\n***** PF inactive *****\n"; exit 1
```

Get Prereq Commands:

```
echo "anchor pf-rules >> /etc/pf.conf"
pfctl -f /etc/pf.conf
```

Atomic Test #13 - Edit UFW firewall user.rules file

Edit the Uncomplicated Firewall (UFW) rules file /etc/ufw/user.rules.

Supported Platforms: Linux

auto_generated_guid: beaf815a-c883-4194-97e9-fdbbb2bbdd7c

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
echo "# THIS IS A COMMENT" >> /etc/ufw/user.rules
grep "# THIS IS A COMMENT" /etc/ufw/user.rules
```

Cleanup Commands:

```
sed -i 's/# THIS IS A COMMENT//g' /etc/ufw/user.rules
```

Dependencies: Run with `sh`!

Description: Check if /etc/ufw/user.rules exists.

Check Prereq Commands:

```
if [ ! -f "/etc/ufw/user.rules" ]; then echo -e "\n***** ufw NOT installed *****\n
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #14 - Edit UFW firewall ufw.conf file

Edit the Uncomplicated Firewall (UFW) configuration file /etc/ufw/ufw.conf which controls if the firewall starts on boot and its logging level.

Supported Platforms: Linux

auto_generated_guid: c1d8c4eb-88da-4927-ae97-c7c25893803b

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
echo "# THIS IS A COMMENT" >> /etc/ufw/ufw.conf
grep "# THIS IS A COMMENT" /etc/ufw/ufw.conf
```

Cleanup Commands:

```
sed -i 's/# THIS IS A COMMENT//g' /etc/ufw/ufw.conf
cat /etc/ufw/ufw.conf
```

Dependencies: Run with `sh`!

Description: Check if /etc/ufw/ufw.conf exists.

Check Prereq Commands:

```
if [ ! -f "/etc/ufw/ufw.conf" ]; then echo -e "\n***** ufw NOT installed *****\n";
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #15 - Edit UFW firewall sysctl.conf file

Edit the Uncomplicated Firewall (UFW) configuration file for setting network variables /etc/ufw/sysctl.conf.

Supported Platforms: Linux

auto_generated_guid: c4ae0701-88d3-4cd8-8bce-4801ed9f97e4

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
echo "# THIS IS A COMMENT" >> /etc/ufw/sysctl.conf
grep "# THIS IS A COMMENT" /etc/ufw/sysctl.conf
```

Cleanup Commands:

```
sed -i 's/# THIS IS A COMMENT//g' /etc/ufw/sysctl.conf
cat /etc/ufw/sysctl.conf
```

Dependencies: Run with `sh`!

Description: Check if /etc/ufw/sysctl.conf exists.

Check Prereq Commands:

```
if [ ! -f "/etc/ufw/sysctl.conf" ]; then echo -e "\n***** ufw NOT installed *****\
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #16 - Edit UFW firewall main configuration file

Edit the Uncomplicated Firewall (UFW) main configuration file for setting default policies /etc/default/ufw.

Supported Platforms: Linux

auto_generated_guid: 7b697ece-8270-46b5-bbc7-6b9e27081831

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
echo "# THIS IS A COMMENT" >> /etc/default/ufw
grep "# THIS IS A COMMENT" /etc/default/ufw
```

Cleanup Commands:

```
sed -i 's/# THIS IS A COMMENT//g' /etc/default/ufw
```

Dependencies: Run with `sh`!

Description: Check if /etc/default/ufw exists.

Check Prereq Commands:

```
if [ ! -f "/etc/default/ufw" ]; then echo -e "\n***** ufw NOT installed *****\n"; (
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #17 - Tail the UFW firewall log file

Print the last 10 lines of the Uncomplicated Firewall (UFW) log file /var/log/ufw.log.

Supported Platforms: Linux

auto_generated_guid: 419cca0c-fa52-4572-b0d7-bc7c6f388a27

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
tail /var/log/ufw.log
```

Cleanup Commands:

```

```

Dependencies: Run with `sh`!

Description: Check if /var/log/ufw.log exists.

Check Prereq Commands:

```
if [ ! -f "/var/log/ufw.log" ]; then echo -e "\n***** ufw NOT logging *****\n"; ex
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #18 - Disable iptables

Some Linux systems may not activate ufw, but use iptables for firewall rules instead. (ufw works on top of iptables.) Attackers cannot directly disable iptables, as it is not implemented as a service like ufw. But they can flush all iptables rules, which in fact "disable" iptables.

Supported Platforms: Linux

auto_generated_guid: 7784c64e-ed0b-4b65-bf63-c86db229fd56

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
iptables-save > /tmp/iptables.rules
iptables -F
```

Cleanup Commands:

```
iptables-restore < /tmp/iptables.rules
```

Dependencies: Run with `sh`!

Description: Check if iptables is installed on the machine.

Check Prereq Commands:

```
if [ ! -x "$(command -v iptables)" ]; then echo -e "\n***** iptables NOT installed
```

Get Prereq Commands:

```
echo ""
```

Atomic Test #19 - Modify/delete iptables firewall rules

Instead of completely "disabling" iptables, adversaries may choose to delete a certain rule, which, for example, blocks data exfiltration via ftp. By doing so, they may cause less noise to avoid detection.

Supported Platforms: Linux

auto_generated_guid: 899a7fb5-d197-4951-8614-f19ac4a73ad4

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
iptables -D OUTPUT -p tcp --dport 21 -j DROP
```

Cleanup Commands:

```
iptables-restore < /tmp/iptables.rules
```

Dependencies: Run with `sh`!

Description: Check if iptables is installed on the machine.

Check Prereq Commands:

```
if [ ! -x "$(command -v iptables)" ]; then echo -e "\n***** iptables NOT installed
if ! echo "$(iptables -L)" | grep -q "DROP .*dpt:ftp"; then echo -e "\n***** this
```

Get Prereq Commands:

```
iptables-save > /tmp/iptables.rules
if echo "$(iptables -L)" | grep -q "DROP .*dpt:ftp"; then echo "Rule found"; else
```

Atomic Test #20 - LockBit Black - Unusual Windows firewall registry modification -cmd

An adversary tries to modify the windows firewall registry

Supported Platforms: Windows

auto_generated_guid: a4651931-ebbb-4cde-9363-ddf3d66214cb

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\DomainProfile" /v Enable
reg add "HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile" /v Enab
```

Cleanup Commands:

```
reg delete "HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\DomainProfile" /v Enal
reg delete "HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile" /v E
```

Atomic Test #21 - LockBit Black - Unusual Windows firewall registry modification -Powershell

An adversary tries to modify the windows firewall registry.

Supported Platforms: Windows

auto_generated_guid: 80b453d1-eec5-4144-bf08-613a6c3ffe12

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
New-ItemProperty "HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\DomainProfile"
New-ItemProperty "HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfil
```

Cleanup Commands:

```
Remove-ItemProperty "HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\DomainProfi
Remove-ItemProperty "HKLM:\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardPro
```

Atomic Test #22 - Blackbit - Disable Windows Firewall using netsh firewall

An adversary tries to modify the windows firewall configuration using the deprecated netsh firewall command (command still works).

Supported Platforms: Windows

auto_generated_guid: 91f348e6-3760-4997-a93b-2ceee7f254ee

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
netsh firewall set opmode mode=disable
```

Cleanup Commands:

```
netsh firewall set opmode mode=enable >nul 2>&1
```

Atomic Test #23 - ESXi - Disable Firewall via Esxcli

Adversaries may disable the ESXI firewall via ESXCLI

Supported Platforms: Windows

auto_generated_guid: bac8a340-be64-4491-a0cc-0985cb227f5a

Inputs:

Name	Description	Type	Default Value
vm_host	Specify the host name of the ESXi Server	string	atomic.local
plink_file	Path to Putty	path	PathToAtomicsFolder\..\ExternalPayloads\plink.exe

username	username used to log into ESXi	string	root
password	password used to log into ESXI	string	n/a

Attack Commands: Run with `command_prompt`!

```
#{plink_file} -ssh #{vm_host} -l #{username} -pw #{password} -m PathToAtomicsFolde
```

Cleanup Commands:

Dependencies: Run with `powershell`!

Description: The plink executable must be found in the ExternalPayloads folder.

Check Prereq Commands:

```
if (Test-Path "#{plink_file}") {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -ErrorAction I
Invoke-WebRequest "https://the.earth.li/~sgtatham/putty/latest/w64/plink.exe" -Outl
```

Atomic Test #24 - Set a firewall rule using New-NetFirewallRule

This test will attempt to create a new inbound/outbound firewall rule using the New-NetFirewallRule commandlet.

Supported Platforms: Windows

auto_generated_guid: 94be7646-25f6-467e-af23-585fb13000c8

Inputs:

Name	Description	Type	Default Value
direction	Direction can be Inbound or Outbound	string	Inbound
local_port	This is the local port you wish to test opening	integer	21
protocol	This is the protocol	string	TCP
action	This is the action	string	allow

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
New-NetFirewallRule -DisplayName "New rule" -Direction "#{direction}" -LocalPort "
```

Cleanup Commands:

```
Remove-NetFirewallRule -DisplayName "New rule"
```