A Pinch of Salt

A Seasoned Introduction to Automating with SaltStack

About Me – Corey Forman

- In DFIR field for 20+ years
- Senior Computer Forensic Analyst – Canada Revenue Agency
- Spent 17 years in the Canadian Armed Forces, the last 7 of which as one of DND's first-ever Cyber Operators, running the Forensics team
- SANS Subject Matter Expert
- Instructor at Canadian Police College (CMPFOR)
- Contributor to SANS SIFT Workstation, and REMnux Malware Analysis toolkit
- Creator of a few forensics tools
- github.com/digitalsleuth

What is SaltStack (Salt)?

- ~~Table Salt, Sodium, NaCl~~
- Robust automation and infrastructure management platform
- Remote execution, provisioning, orchestration
- Platform and OS agnostic (Servers, IOT, Networks)
- Uses agents (master and minions)
- Based on Python 3.x, YAML (Yet Another Markup Language), and Jinja
- Open Source!

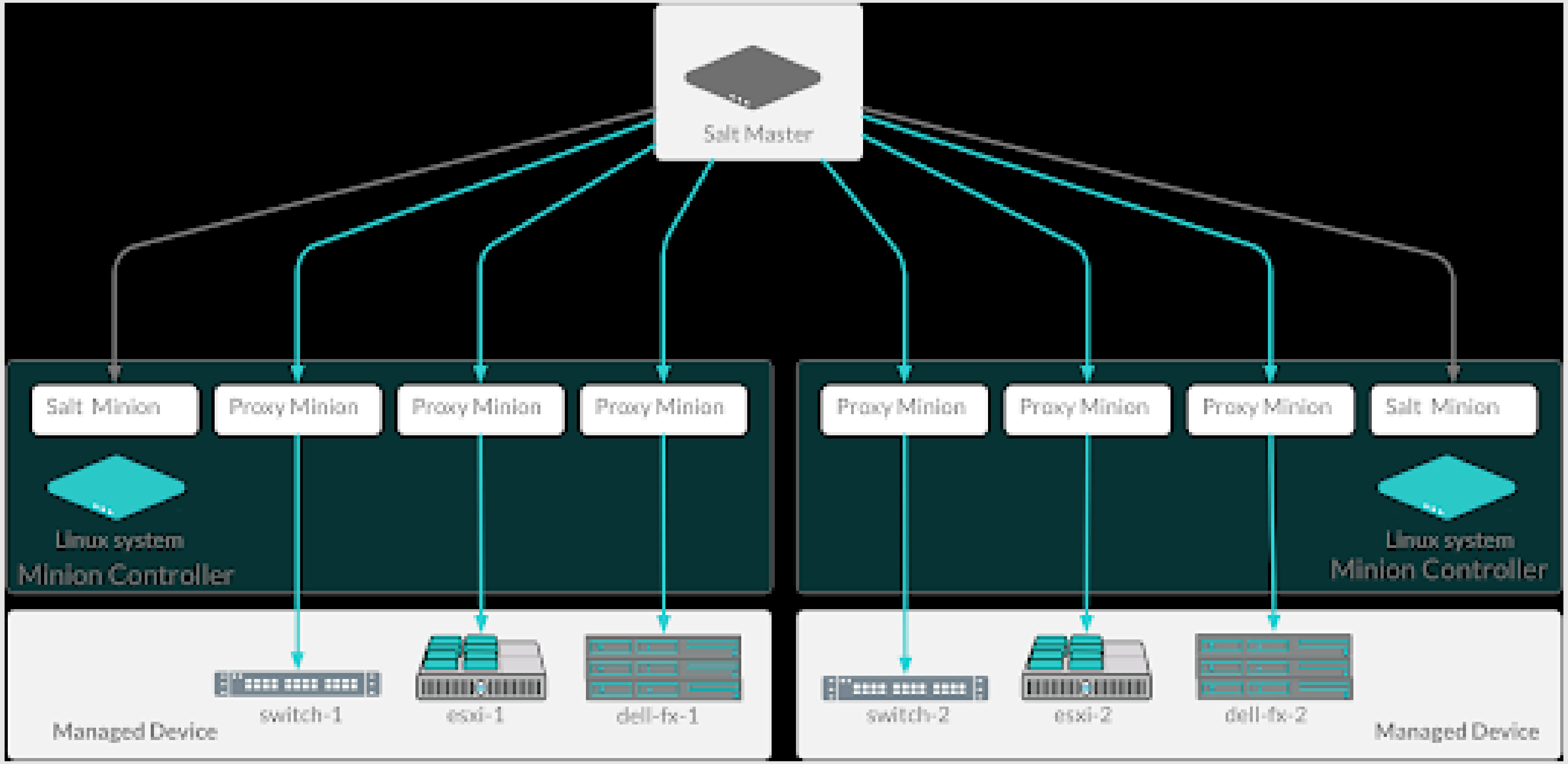
What Can It Do?

- Remote management of a single system or full domain
  - Execute applications
  - Configuration management
  - Install or remove software
- LDAP and Active Directory Integration
- Reporting and Scheduling of Tasks
- Network Device Management

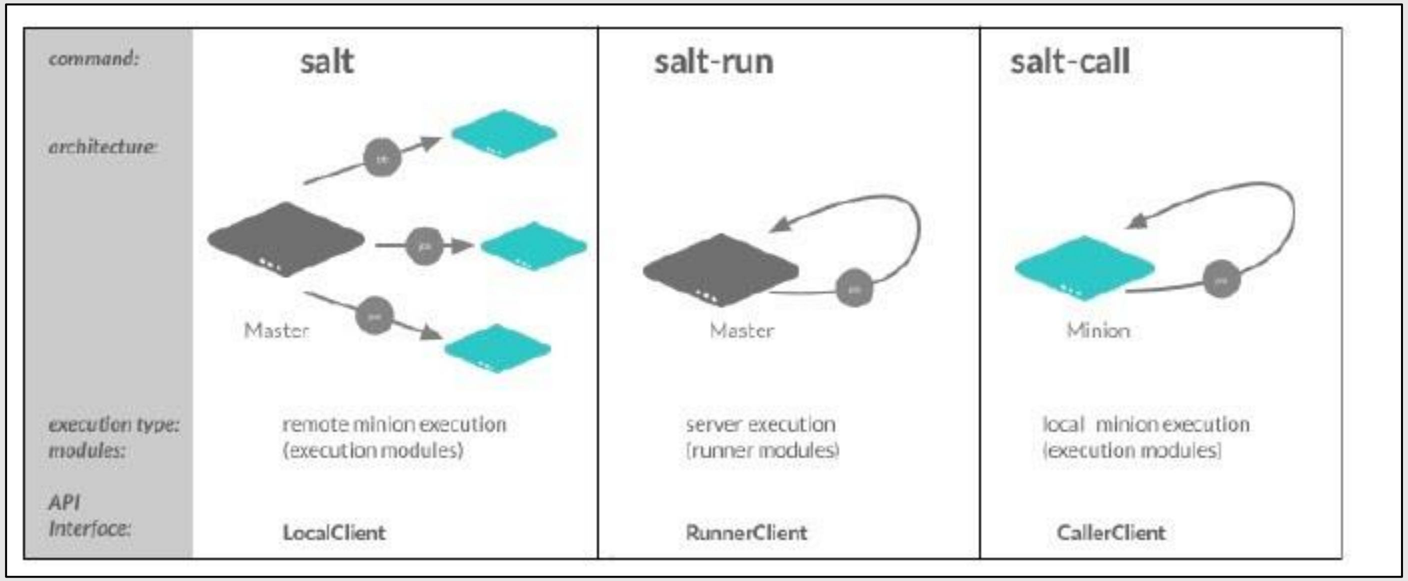
How It Works

- The “salt-master” service is run on a server of your choosing
- The “salt-minion” service is run on the devices you wish to manage, or the system can be ‘agentless’ by use of a proxy. The minion can also act as a stand-alone system, and execute commands.
- A “state” file is created using YAML and Jinja and placed in the appropriate location on the server (or stand-alone minion) for execution
- A command is run on the master (or stand-alone minion) to initiate the “state” on the target system
- Command is run

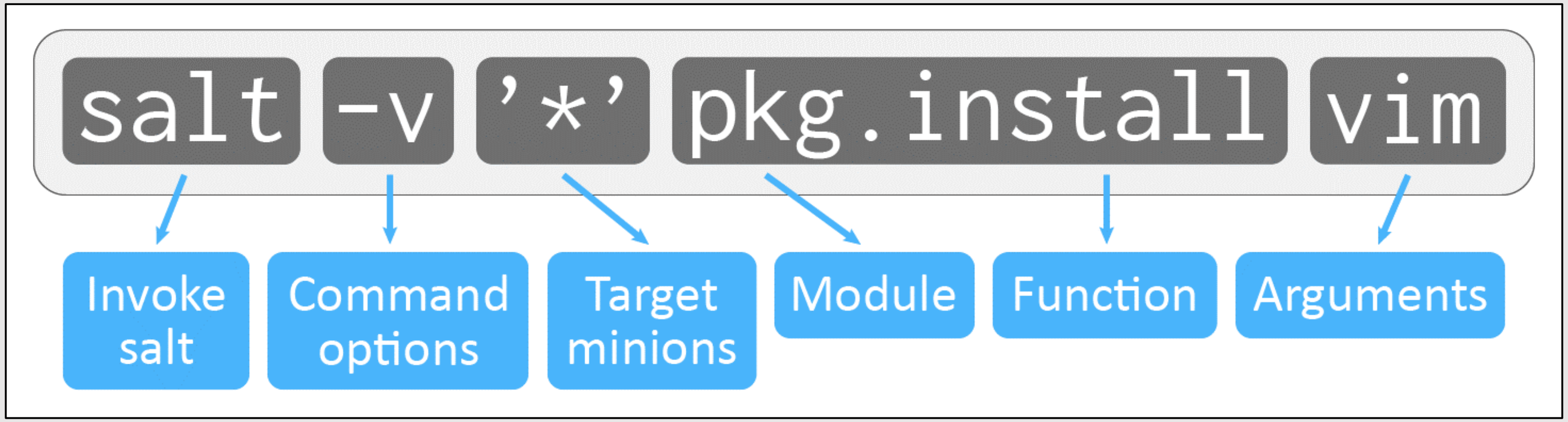
How It Works


Salt Master
Salt Minion
Proxy Minion
Proxy Minion
Proxy Minion
Proxy Minion
Proxy Minion
Proxy Minion
Salt Minion
Linux system
Minion Controller
Linux system
Minion Controller
switch-1
esxi-1
dell-fx-1
Managed Device
switch-2
esxi-2
dell-fx-2
Managed Device

How It Works

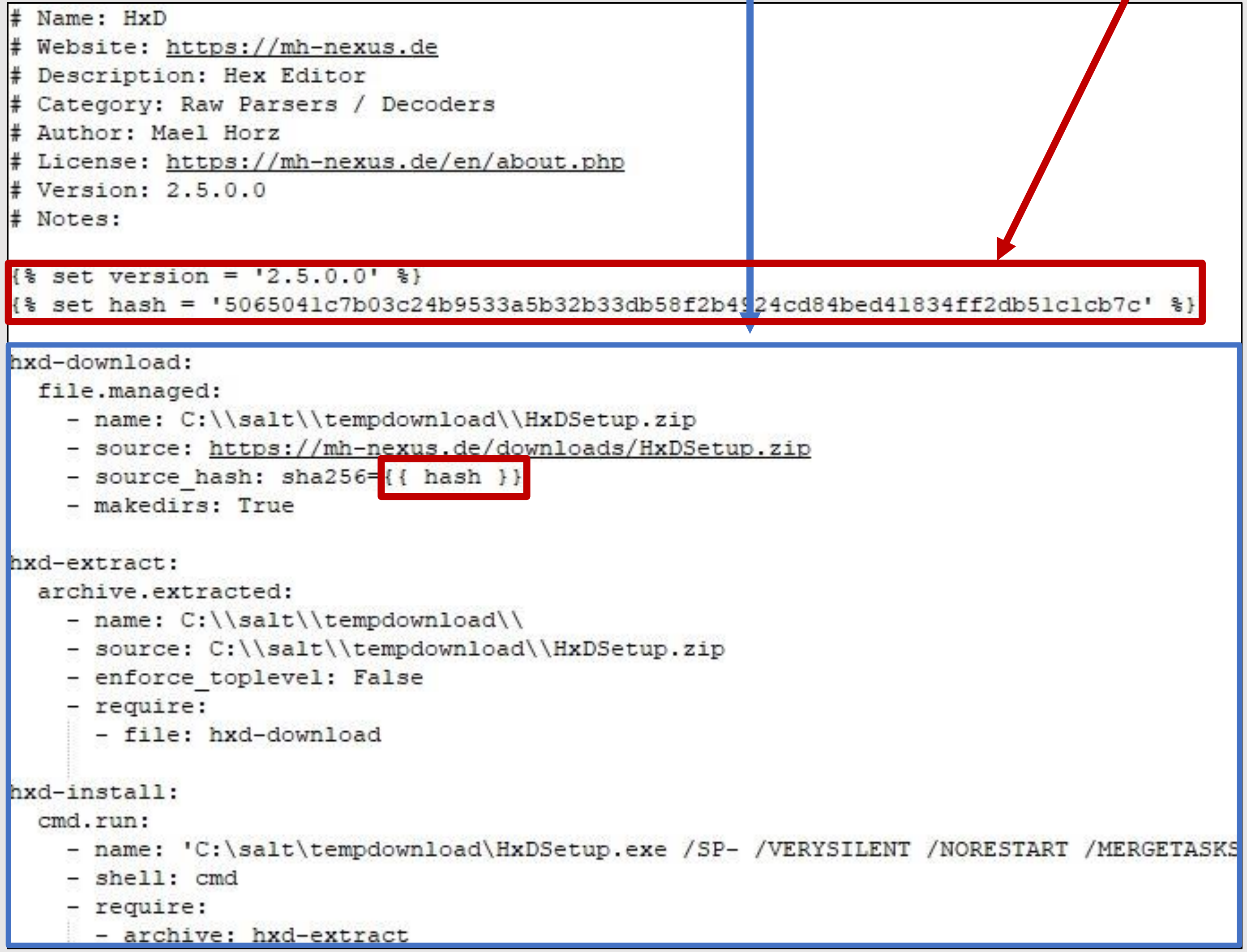


command:	salt	salt-run	salt-call
architecture:	Master	Master	Minion
execution type: *modules:*	remote minion execution (execution modules)	server execution (runner modules)	local minion execution (execution modules)
API Interface:	LocalClient	RunnerClient	CallerClient



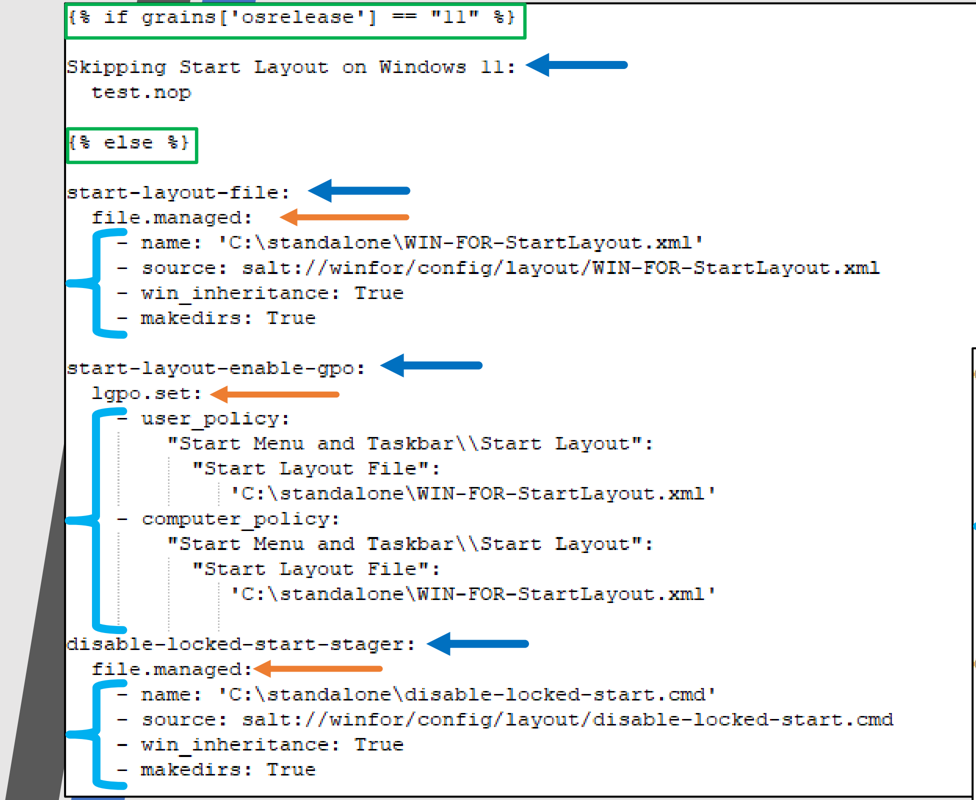
```
salt -v '*' pkg.install vim
```

How It Works – YAML and Jinja

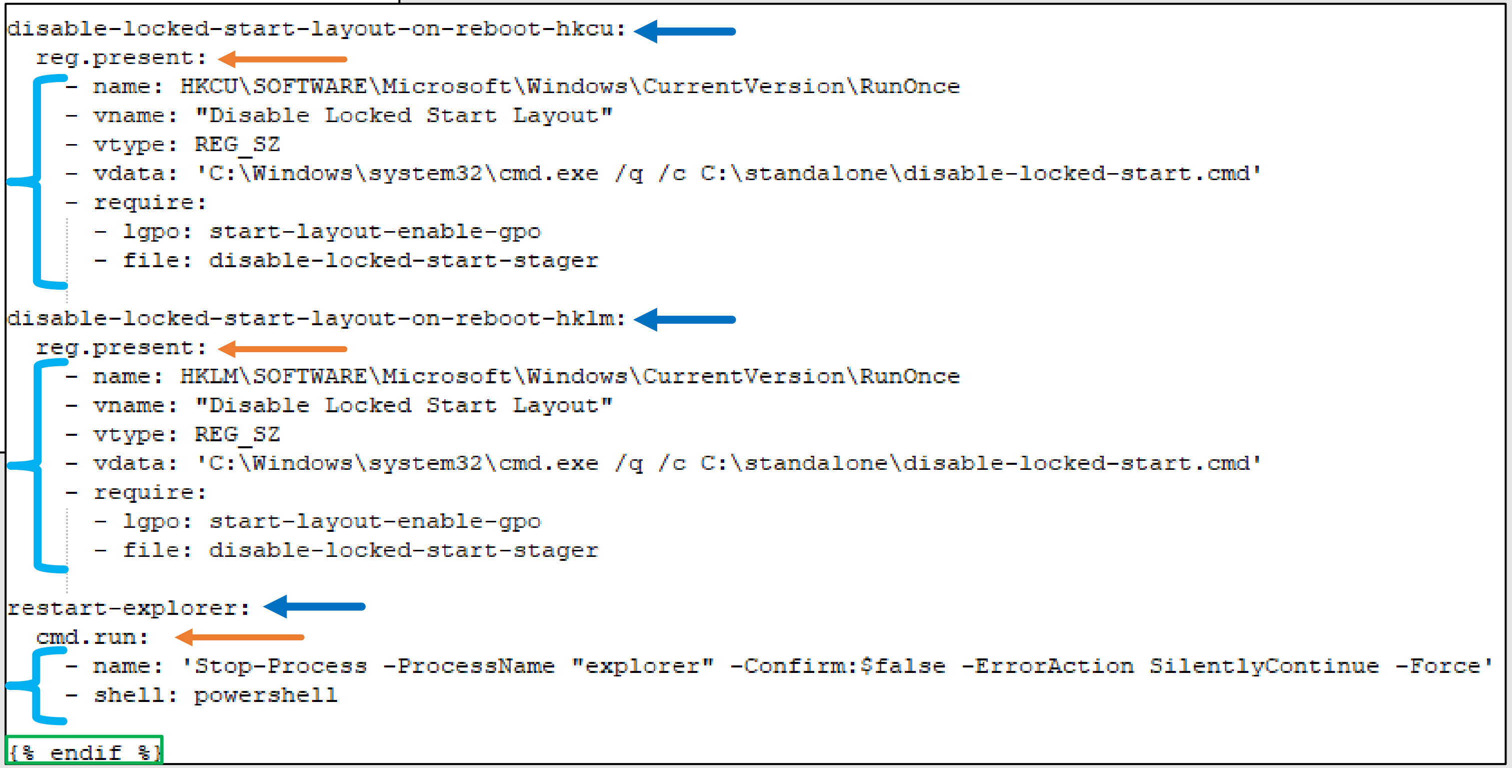


```
# Name: HxD
# Website: https://mh-nexus.de
# Description: Hex Editor
# Category: Raw Parsers / Decoders
# Author: Mael Horz
# License: https://mh-nexus.de/en/about.php
# Version: 2.5.0.0
# Notes:

{% set version = '2.5.0.0' %}
{% set hash = '5065041c7b03c24b9533a5b32b33db58f2b4924cd84bed41834ff2db51c1cb7c' %}

hxd-download:
  file.managed:
    - name: C:\\salt\\tempdownload\\HxDSetup.zip
    - source: https://mh-nexus.de/downloads/HxDSetup.zip
    - source_hash: sha256={{ hash }}
    - makedirs: True

hxd-extract:
  archive.extracted:
    - name: C:\\salt\\tempdownload\\
    - source: C:\\salt\\tempdownload\\HxDSetup.zip
    - enforce_toplevel: False
    - require:
      - file: hxd-download

hxd-install:
  cmd.run:
    - name: 'C:\salt\tempdownload\HxDSetup.exe /SP- /VERYSILENT /NORESTART /MERGETASKS
    - shell: cmd
    - require:
      - archive: hxd-extract
```

How It Works – YAML and Jinja



```
{% if grains['osrelease'] == "11" %}

Skipping Start Layout on Windows 11:
  test.nop

{% else %}

start-layout-file:
  file.managed:
    - name: 'C:\standalone\WIN-FOR-StartLayout.xml'
    - source: salt://winfor/config/layout/WIN-FOR-StartLayout.xml
    - win_inheritance: True
    - makedirs: True

start-layout-enable-gpo:
  lgpo.set:
    - user_policy:
        "Start Menu and Taskbar\\Start Layout":
          "Start Layout File":
             'C:\standalone\WIN-FOR-StartLayout.xml'
    - computer_policy:
        "Start Menu and Taskbar\\Start Layout":
          "Start Layout File":
             'C:\standalone\WIN-FOR-StartLayout.xml'

disable-locked-start-stager:
  file.managed:
    - name: 'C:\standalone\disable-locked-start.cmd'
    - source: salt://winfor/config/layout/disable-locked-start.cmd
    - win_inheritance: True
    - makedirs: True
```

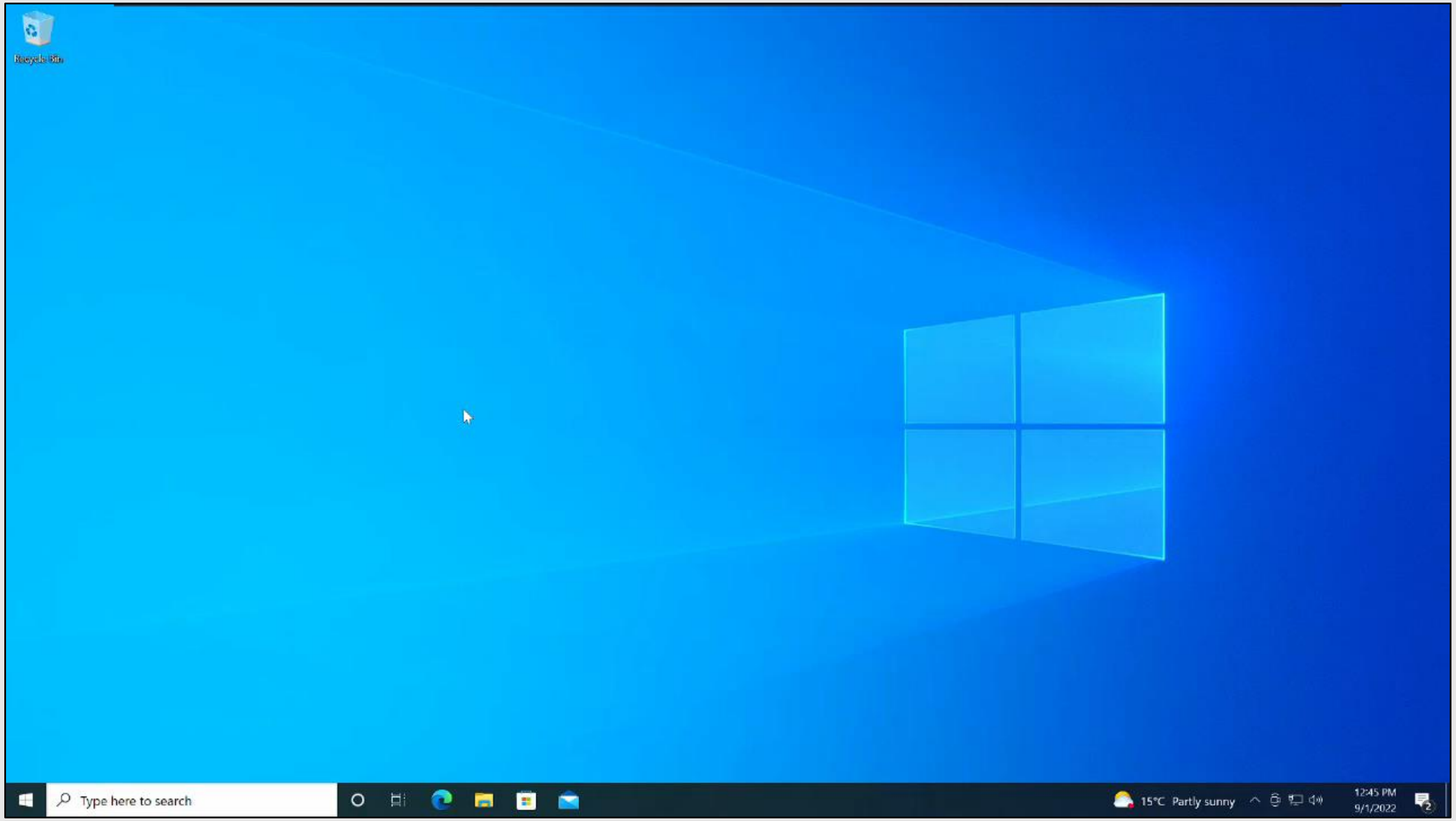


```
disable-locked-start-layout-on-reboot-hkcu:
  reg.present:
    - name: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
    - vname: "Disable Locked Start Layout"
    - vtype: REG_SZ
    - vdata: 'C:\Windows\system32\cmd.exe /q /c C:\standalone\disable-locked-start.cmd'
    - require:
      - lgpo: start-layout-enable-gpo
      - file: disable-locked-start-stager

disable-locked-start-layout-on-reboot-hklm:
  reg.present:
    - name: HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
    - vname: "Disable Locked Start Layout"
    - vtype: REG_SZ
    - vdata: 'C:\Windows\system32\cmd.exe /q /c C:\standalone\disable-locked-start.cmd'
    - require:
      - lgpo: start-layout-enable-gpo
      - file: disable-locked-start-stager

restart-explorer:
  cmd.run:
    - name: 'Stop-Process -ProcessName "explorer" -Confirm:$false -ErrorAction SilentlyContinue -Force'
    - shell: powershell

{% endif %}
```

How You Can Use It

- Automate the installation and maintenance of a forensic workstation
  - SIFT, REMnux, m.a.t. (Mobile Analysis Toolkit), WIN-FOR (Windows Forensics Installer)
- Automate the execution of commands on forensic acquisitions
  - Run a single command to start the triage of multiple pieces of evidence, including hashing, data extractions
  - Run triage applications on evidence items and copy the results to a specific location
  - Alert when an activity is complete
- Schedule the query of new version downloads for software
- Do all of this with one simple command!

Usage Example: Automating Installs - WIN-FOR



Live Example: Automating Triage

In Closing

- SaltStack is an excellent tool for automation
- Installs on nearly every operating system
- Relatively easy to learn
- Scriptable – Since it's Python-based, it has modules which can be used in your own scripts!
- The WIN-FOR (Windows Forensics) installer and the m.a.t. installer (Mobile Analysis Toolkit) can be found at https://github.com/digitalsleuth/WIN-FOR and https://github.com/digitalsleuth/mat-salt
- corey[at]digitalsleuth[dot]ca

Questions?