Master of Science in Internetworking – Computing Science

MINT 709 Report

Implementation of Portable Security Analysis Tool

Submitted By – Gaganpreet Singh Sandhu

CCID – gsandhu1@ualberta.ca

Table of Contents

Abstract

Transition to application virtualization is causing IT organizations to reconsider the methods it uses for deployment of services like IT support, application usage, management, etc. It is becoming important to have a vision for the application delivery methods for the future. Today the choice of software used by various organizations depends on how protected the software code is and how security is applied on top of the software code. This is the point where we define and compare proprietary and open-source software. This report initially aims at discussing the significance of both types – proprietary and open-source, focusing primarily on security analysis tools.

The network infrastructure of any organization composes of various threats and vulnerabilities and hence the need for security analysis tools becomes handy. Provided the network can break and outages can be caused, this should not become the reason for the failure of regular security scan checks for the isolated/broken network. There is a need for a new deployment method other than the legacy client-server method (one server scanning many networks/clients) to run scans even without network connectivity. The portability of such security analysis tools becomes the need for the hour in these situations, where, the tool can work from any machine or operating system and provide the necessary results continuously. This report focuses on a brief discussion of the methods and tools required for performing security analysis and deciding what tool can be converted into a portable version.

This report then briefly discusses the tools available for creating portable applications and describes the implementation process for creating the portable version for one of the security analysis tools. Open-source and free software are targeted to create a portable version of the security analysis tool. After creating such tools, its risk management becomes equally important as its usage hence proper guidelines are required to be set up for using these tools. The report finally discusses the future of application deployment methods and how it can lessen the complexity of the current system.

Introduction

"In the context of IT security, businesses have two kinds of objectives: mission and compulsory. A mission objective is directly related to producing revenue or enhancing profits. Compulsory objectives are those that must be achieved as a matter of prudence or regulation. Endpoint or Network security is a compulsory business objective." [1]

The largest threat of organizational breach occurs at the endpoint level which also acts as an initial attack vector. [2]. A vulnerability in any system is the attack vector used by hackers. The US National Institute of Standards and Technology (NIST) defines vulnerability as "a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source."

The existence of endpoint vulnerabilities attracts hackers which results in system compromise. It is vital to detect vulnerabilities and fix them as soon as possible since business runs with the connectivity of different systems/sites/people. These vulnerabilities are required to be detected within seconds and minutes and hence the continuous assessment of business systems should be part of the compliance policy of any organization.

With the advent in the increase of major data breaches, organizations realize that the financial aspect and business continuity depends on risk management within the IT domain. Hence the organizations majorly use vulnerability management techniques in order to make sure that the systems, applications, communications between devices (including social engineering activities) are compliant with respect to the industry security standards. Infrastructure security starts at detecting vulnerabilities (or at the places which have no protection).

To be better prepared and face threats, security analysis is the first step towards better IT hygiene for any organization.

"There is one simple countermeasure for existing vulnerabilities that will protect you, should a hacker scan your machines with a [vulnerability assessment scanner] – scan your own systems first. Make sure to address any problems reported by the scanner, and then a scan by a hacker will give him no edge." [3]

Programs that provide security analysis uses tools and techniques that help to detect a past/present/future security compromise that has/could have happened by analyzing large volumes of data. Analysis tools/techniques are made from a pre-defined set of policies and parameters that the tool can poll and check from a device to be assessed. Next-generation analysis tools also can work based on Machine Learning, Deep Learning and Artificial Intelligence which makes them more complex and resource using tools.

This report is focused on network scanning/auditing based on user-defined parameters and defined set of rules.

Methods to perform security analysis

There are multiple ways in which security analysis can be done, some of the methods are mentioned below:

Port Scanning – It is the most used technique in order to find the exploit. Systems connected to a network possess availability of open ports based on which an attacker or a tester can detect - what ports are open, type of service that is available, support of anonymous services, authentication requirement (login services), system and interface information, etc. Multiple port scanning methods are available in order to detect vulnerabilities of a running system. The whole idea behind these various methods includes sending multiple types of packets (initialization, reset or finalization) in order to get a response from unknown/known endpoints to collect data and detect various impactful areas that can be exploited. Some of the available methods include – TCP NULL, TCP ACK, ICMP Ping Sweep, TCP Reverse Indent, TCP Half Open, TCP Fin, ARP scans, etc.

Every machine is open to port scanning vulnerability assessment. If an exploit is detected on the endpoint, the hacker can use the exploit to open more ports on the machine and make the system or network compromised.

Wireless Network Scanning – Primary focus of wireless scanning is to detect anomalies in misconfiguration of the wireless network devices which are used to either provide authentication services, connection services, access services or the type of encryption that the device provides for communication with end devices who try to connect to the network via Wi-Fi. Enterprise wireless networks can get compromised since the Guest network is also connected to the organizational network. This type of scanning can also help in detecting the rogue access points which are used to get access to different devices that try to initiate a wireless connection with it. This type of scan can be implemented by just creating a list of available wireless access points with their MAC addresses and then scanning the area for rogue access points that don't comply with the list and then remediating the network by applying restrictive policies in the organizational systems.

Application Scanning – The primary motive of this type of scanning is to scan the source code of the application and inserting/injecting various code values/lines in order to parse the results and observe the application behavior and find the exploit in the code. White box and black box are the common methods to perform the scan. Whereas White Box is an exhaustive method in terms of resource utilization since it uses source code to determine all possible paths the code can result to detect the unwanted outcome, the Black Box method uses random requests to the application in order to check if the results were successful. This is also called Penetration testing. Malformed data is sent in order to check for response errors. Scanner also has a known list of vulnerabilities (also called signatures) of web applications, server configuration which is used to execute many available patterns in order to find the vulnerability. Various related vulnerabilities include – SQL Injection (SQLi), Remote Code Execution, Cross-site Scripting (XSS), Vulnerable JavaScript Libraries, Overflow vulnerability, DoS, etc.

Endpoint-Protection-Tool – Various virus scanning tools are hosted on local systems. The vulnerability detection is based solely on signatures which it compares within the executable code of an application or operating system. These kinds of vulnerability scanning techniques can also

work based on heuristics in which behaviors of the code is analyzed. This also takes more time and is not as accurate as signature-based detection.

Network Scanning – Usually organizations implement network security at perimeter level but due to recent advancements in the way network behaves due to variable data flow between networks or sites due to access to critical data at various locations, it can be implemented inside the network as well in order to expand the threat detection capability and performance. One type of network scanning is performed basically by capturing the traversing data and creating cache for the data flow to detect anomalies and to provide network visibility in terms of vulnerability isolation, capacity planning and more. Network scanning can also be performed by implementing a firewall solution between two networks or sites. A firewall can scan the traffic based on the Five-Tuple parameters – Source and Destination IP Address and Ports, and the protocol used. This can help in various ways to identify any unusual behavior in the communication between two endpoints. Next-Generation firewalls can also scan the data traversing via the firewall in order to scan for viruses and application threats that are signature-based or zero-day attacks. Upon scanning, relevant policies or actions can be taken to remedy the issue.

Other Methods – These includes various deployments like cloud solutions that provide scanning services to the communication between an organization's internal resources and external network. This type of scan can be deployed by introducing a cloud proxy solution where all the requests to external network within an organization are redirected to a cloud proxy solution which scans for the requested URL's and scans them for access restriction or web reputation score in order to make a decision whether to allow or block the requests. This helps to make compliance policies if the organization has various sites from which it runs the business.

There can be **more** methods for finding the vulnerabilities and preventing them from changing into exploits. The discussion of all those methods that are available is restricted to the above-mentioned points for this report.

Tools Used for Security Analysis/Auditing

There are many tools available for security analysis and security auditing. Some of the majorly used tools are briefly described below.

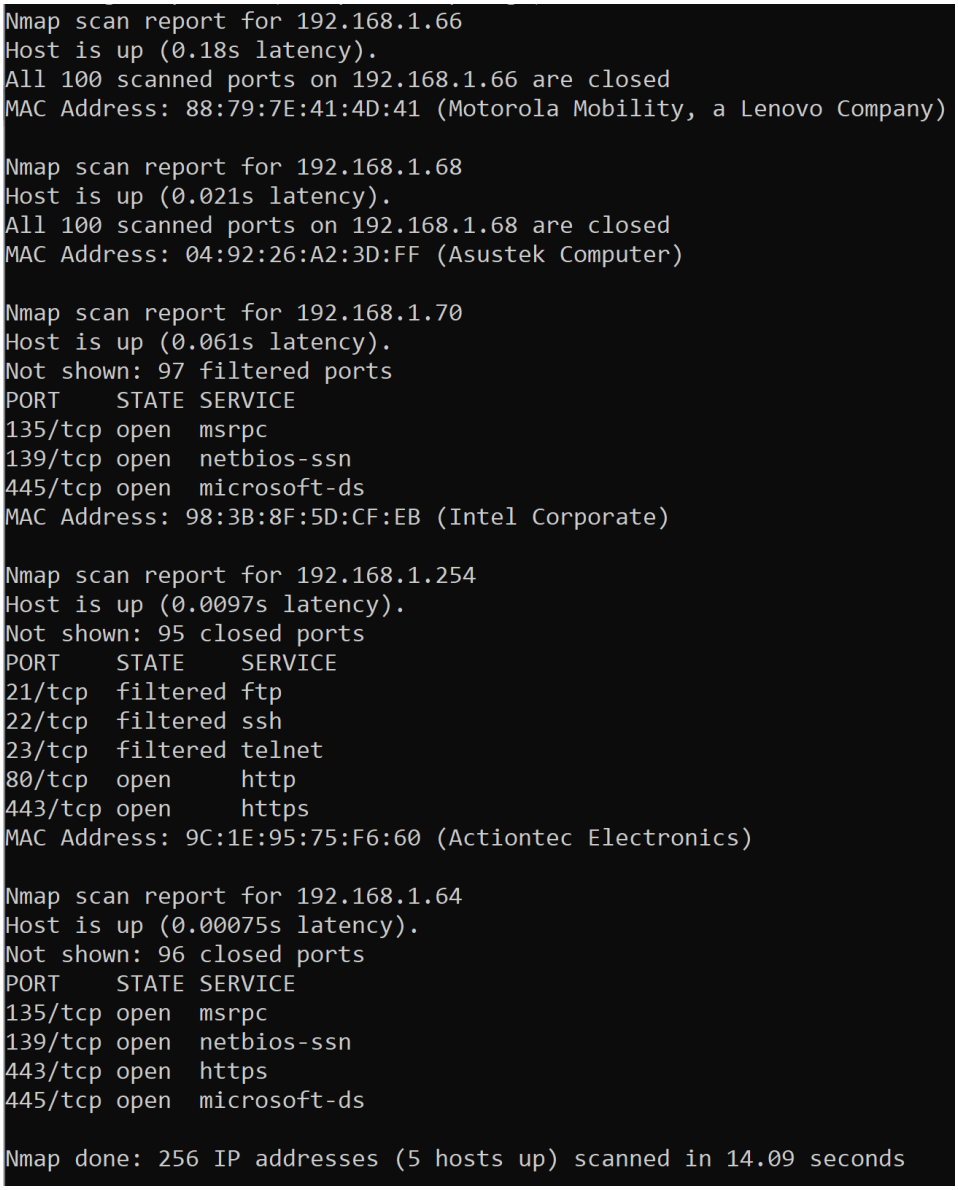
NMAP

Nmap ("Network Mapper") is an open-source tool for network exploration and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. [4]

Nmap is flexible?

- Nmap can run on most of the operating systems including Windows, Mac OS, Berkeley Software Distribution (BSD), Solaris, AIX, AmigaOS.
- As compared to other free or open-source products providing the same functions NMAP documentation has been maintained by many developers, man pages, and white papers.
- It is easy to acquire, install/deploy and to work on.
- The scripting engine and different NSE scripts allow users to develop and implement multiple types of network discovery methods.

A typical Nmap scan looks like this:



The scan was performed with the options: nmap -T4 -F 192.168.1.0/24

This type of scan is also known as a quick scan in nmap terms.

Metasploit Framework

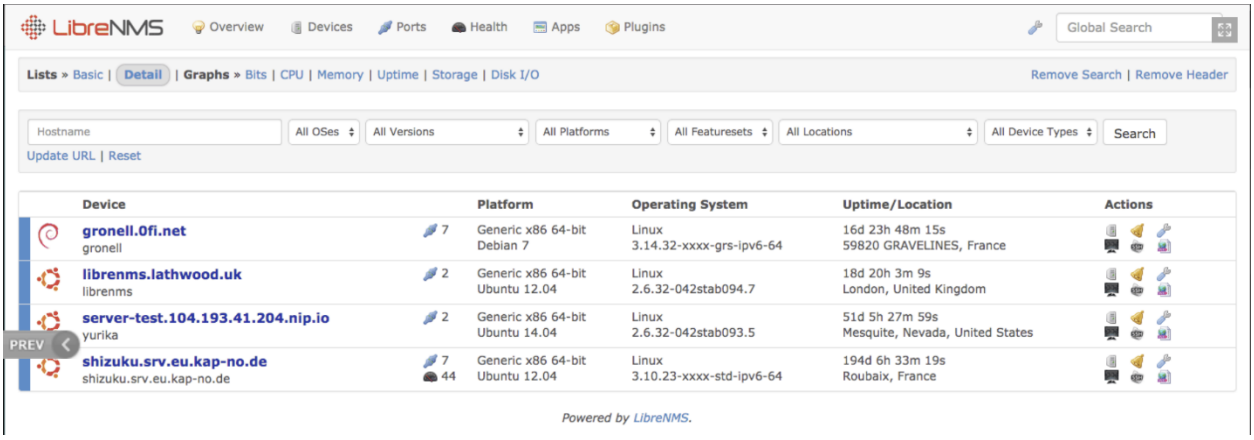
It is a penetration testing solution that has the ability to simulate real-world attacks and posses one of the largest exploit databases. Metasploit can perform multiple tasks including information gathering, enumeration, access/privilege escalation, and tracing, which means it works as both vulnerability detection as well as penetration testing tool. This open-source framework is based on Ruby. It includes more than 1677 exploits along with 500 varied payloads acting as evasive solutions to escalate the privileges, perform commands/scripts and control the target device. Even though it is available for Ubuntu Linux, Windows and RHE Linux Servers, the recommended OS on which it may run is Ubuntu Linux. Kali Linux comes with Metasploit's pro version in the bundle. The installation process requires you to disable antivirus and firewalls on the machine on which Metasploit needs to be installed. The general process includes checking for any vulnerabilities> developing/configuring an exploit> configuring/developing a code of execution with the target> deploying the exploit.

Libre NMS

Libre NMS is a network monitoring solution that provides network discovery services by using protocols such as CDP, FDP, LLDP, OSPF, BGP, SNMP, and ARP. [5] It helps in infrastructure operations with respect to alerting systems based on network/device down incidents that need immediate attention. It also has the capability to generate reports for network status continuously to determine high/low traffic regions, port status, device inventory, etc. Event logs and System logs can also be integrated with this tool for better forensics. Based on the searched network components and their MAC address this tool can categorize them based on different vendors and operating systems. Apart from these functions it can also monitor routing protocol status, health status and can configure thresholds based on multiple parameters for generating alerts. This tool is available only for RHEL CentOS 7 and Ubuntu OS. It is an open-source project with source code available on https://github.com/librenms/librenms/blob/master/doc//Installation/index.md.

- "Monitoring tools have a lot of significance in maintaining and monitoring existing nodes present in the network. Many tools are sold at a high price. Open-source projects allow small size organizations to generate routine network statistics report and check for anomalies in the network."

Below is the screenshot acquired from https://www.librenms.org/#prettyPhoto[pp_gal]/0/ which shows how node statistics can be represented in the tool.



LibreNMS GUI

Based on the SNMP queries, the above picture shows how the devices are displayed based on DNS name resolution, OS detection, and the uptime of the devices. Depending on the SNMP control strings we can also control what information can be polled and when the poll happens.

Wireshark

Wireshark is a network & protocol analyzer that allows you to understand the data that is captured from a network. The packet capture can be taken from any tool or technology eg – tcpdump, but Wireshark allows the captured data to be represented in a structured sequence. The data is represented as sections of layers in TCP/IP model with each layer data is presented in the context of different protocol options like IP in layer 3, HDLC/Ethernet at layer 2, HTTPS at layer 7, etc. When using Wireshark to capture data it places the interface into promiscuous mode so that the data can be captured and replicated into GUI. It is a complex tool with respect to the functions that it can perform in analyzing the data. It can be used to:

- Observe the protocol behavior with respect to a type of stream or communication between two endpoints.
- To check what are the specific timing, flags, sequence numbers used in the data packet. This information can be used to check for the latency in the whole network.
- Finding issues by checking the type of messages being shared between two endpoints.
- Capture data from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI. [4]

There are many more use cases for which Wireshark is beneficial. The tool is open source and is available for Windows, Linux, macOS, Solaris, FreeBSD, NetBSD.

Other Tools

There are many other tools available for security analysis and auditing including Splunk, Solar Winds, HP BMS, Nessus, Arachni, Aircrack ng, etc. The scope of this report is limited to the introduction of such tools and not all.

Introduction to Open Source and Proprietary Softwares/Applications

The idea behind open source and proprietary applications is not the same. Though the methodology to create any software can be the same the method of distribution is the polar opposite. Open-source application can be either managed by a community of developers who work together to create a program or they can be commercial open-source software, the copyright of which is controlled by organizations. The organization maintains the ownership of the product and only accepts any contributions if the contributor accepts to transfer the copyrights to the owner of the product. This also acts as a business model for any organization where an open-source product is provided for free for limited services only. If customers find it useful they can purchase the full version of the product. The support for the open-source products is also free till the time the ownership and contribution is from a community. The product is made from open standard protocols having fewer bugs. Due to the fact that it is supported by the community, free support is often considered not the fastest and hence it has a reduced business advantage for its distribution in various industries.

Proprietary tools are created with an idea for creating a software that serves a specific feature to attain the business advantage of multiple features. Usually, the product releases are stable. The products are designed for a long run that can support future OS versions and at the same time is backward compatible. The support for the proprietary products is deemed to be better as compared to open source tools. Due to the limited scope of the product, specific tech support can be easily assigned to solve the issue with the product. Though there are many advantages for commercial products, customers are dependant on the organizations for the pricing structure and opacity of the code that they use in their operating system.

Portable Applications vs Regular Applications

Regular applications are the files packaged without dependencies to be installed in a system. The installation process might require the installer to place the application in a specific directory usually under ‘Program Files’. The uninstaller is also placed in the memory. The application installer can also change some registry entries and modify system configuration. For example – Vmware creates new logical interfaces in the system configuration. In some cases, the application also starts controlling the system with respect to system calls, role-based access privileges, installation, and deletion of other software or programs. Antivirus is an example of such software. It is beneficial because of the processing speed as compared to portable apps which are slower. This type of installation can take advantage of many windows features such that multi-user distinction. The installed program can be available to and can make changes to specific user settings only via the AppData directory for a particular user and not affect others making sure that the environment is isolated when required.

Benefits

- Natively installed applications are fully integrated with the system and have the ability to communicate with other operating system features and applications.

There are some disadvantages as well for using this kind of applications:

General Trade-Offs

- They require more development time and money.
- Each platform might require a separate installer package to be installed.

Portable Applications – (also called Virtualized Applications) does not use any installer and comes with all the dependencies packaged into a single standalone executable fine, hence the physical installation is not required. This is also called application virtualization. This methodology isolates applications by creating a virtual file system and virtual registry providing resources pertaining to the original operating system during runtime. All files that are required to run the portable program are present in a single file/folder. Typically it is a compressed file, and to run it it just needs to be extracted or the executable runs the application (the significance of using portable applications). Since the end-user only sees the function and not the resources that are used to run the portable application, the portable application is isolated on the target platform. Application virtualization methodology uses a virtualization layer that acts as an alternative to the runtime environment provided by the operating system. The virtualization layer may intercept the function calls to the operating system. Because of its isolation, the application thinks that it is the only code running on the operating system. This prevents changes to the operating system components. While running via a USB, these applications do not use memory on the system and do not leave any footprint of using it. Portable applications comes with a lot of downsides as well.

Benefits of using Portable Applications

- Applications or software can be easily delivered across multiple systems.
- There is no more installation dependency.
- Application conflicts are minimalized.
- Stabilization of user-profiles because of lesser dependencies for running the application since it creates a dynamic user environment while executing.
- Even though portability helps in maintaining isolation, they operate the same way as native applications providing integration with operating system shell, IPC, etc.

General Trade-Offs

- Windows User Account Control configuration will not work with portable applications due to packaged dependencies. [6]
- Multiple users need to have the same configuration setting to run the application in a portable environment.
- Not all applications can be made portable. Limited functionality can be a downside of such applications in some cases.

- The portable application cannot write to an OS file system or registry entries.
- New updated versions have to be made portable every time once the installer packager is released.
- Portable applications cannot access any additional libraries from the operating system if they want to use advanced features.
- The infrastructure security team in some organizations do not allow to use of portable applications.
- Application performance is degraded.
- API (required for communication between the application and operating system files and registries) has to be created again for the virtualized environment.

This report is focused on the security auditing/scanner tool NMAP (Network Mapper) in a portable version.

Motivation For Creating Portable Package For Security Analysis

Different machines have different capabilities and might have different libraries required to run any specific application or program. Working on an application on one machine doesn't mean that the same application can run on any other machine provided there is some dependency mismatch which causes run-time errors. A software might require libraries that have dependencies on some other libraries which are required to be installed on the machine if the software needs to be run. The same is the case with tools and software created for security analysis.

Within an IT infrastructure domain, there are multiple domains that have their own functions or operations. These can be security, networking, database/storage, servers, application, monitoring, normal users, scientists, finance, HR. Different domains of work within an organization requires different role-based access and different software to run. Software can update a number of files, directories, and libraries making the use of any other software difficult. An example can be an Anti-virus program that can block the installation of another program because of a false positive. Hence running of any software can depend on any other installed software or present libraries, access to those libraries, directories, etc. This can be a common problem in an IT space.

To overcome this dependency nightmare there is a need for a standalone executable that has all the dependencies like scripts, libraries, files packaged together. Such files can be run from anywhere, without access to libraries and act as self-contained software that has all the dependencies linked together withing the executable. Portable packages do not change anything on the machine it is run on. All the configuration settings and files are carried with the package itself.

The main objective behind this report is to trace all the libraries that Nmap needs access to, to run it seamlessly without any dependency and without any installation requirement.

Why Nmap for portability?

Creating a portable software comes with a lot of trade-offs. One of them is a memory. If a portable software needs to be run via a USB, the amount of data and the level of details of the data it can capture is limited. Depending on the functions of different software it requires access to memory more significantly as compared to Nmap. Some of the software are mentioned below:-

Firewall – A firewall needs to have access to big amount of memory because of its related functionality which can include – maintaining a record of state os a traffic flow which requires it to maintain connection details for each and every connection in memory (Stateful firewalls), maintain threat signatures based on various protocols, use data for sandboxing, storing backup files and tech support files, storing logs for traffic analysis, etc. The functions are not limited to the operations discussed above. All these operations require a lot of free memory space.

Antivirus Protection Tools – This kind of tool needs to maintain a huge database of virus signatures offline in order to provide effective functionality. Antivirus tools scan the data in the machine and might require it to sandbox it to analyze the issue which again requires fast memory requirements, otherwise, it can be a very cumbersome process.

Monitoring Tools – As with firewalls, monitoring tools also require to store a lot of monitored data and logs for many devices that send SNMP Traps to the monitoring system manager. The data needs to be stored for at least 90 days for the forensic requirement.

Kali Linux – Kali Linux is an operating system that can be used to run many security analysis tools, hence it requires a lot of memory requirements. Running a full operating system makes it difficult for any field engineer to run it due to the complex operations. Hence there is a need to have a standalone tool to perform the required functions and provide necessary analytics. The same goes for trusted n-node security and network security tool kit.

Nmap is a small tool that requires very little memory for installation. It's completely open-source and the source code can be found at https://github.com/nmap/nmap. It is also available for almost every operating system. Since the tool is free it can be used by small to medium-sized organizations as well.

Nmap is easy to use and does not require a complex understanding of the tool. Nmap can be used in various scenarios for security auditing within an IT infrastructure environment.

- If a site loses connectivity over WAN link to another office location and network scanning/auditing has been stopped, portable Nmap can be used on the existing machines over that site to discover the devices and a report can be sent continuously to the main site to maintain continuity.
- Even for a portable version, Nmap working requires easy commands and report generation. It is not complex for general tasks.
- It is not attackable since the portable version is not loaded from the machine memory. It only uses kernel and processing to function properly.
- If the server hosting the monitoring/scanning tool is down, a portable version of Nmap can be used to continue the scanning and for reports.

Tools for Creating Standalone Nmap Package

Windows OS

Cameyo tool is used to create a portable version of Nmap. It itself is a portable application. Cameyo is available for Windows OS and is used to capture or record an installation. Initially, Cameyo takes a snapshot of the system. After taking a snapshot it waits for any additional installation to be completed and once the new software is installed it takes another snapshot. During the installation, the tool captures the changes to the system, files, folders, registry, scripts, etc. The time required to take a snapshot depends on how powerful the system is in terms of memory management and processing power. Once the new snapshot is taken it replicates all the changes to its own virtual file system copying all the necessary files, binaries, libraries, etc. into a folder. It can be compared to Linux chroot.

Cameyo offers agent-less mode implementation (Target system does not need to have Cameyo agent). File readings are copied to a virtual package. It changes the process loader with a dummy executable which acts as an alternative to OS process loading. This mode is also called an agent-less RAM mode. Cameyo works in one more mode knows as Disk mode in which the virtual file is extracted on the disk and the application loader redirects the file input-output operations to it. We are using Cameyo in DISK mode where it is virtualizing the Nmap application environment by substituting process loading and file reading and mapping and the operations work by extracting the files necessary to launch the application.

A virtual package created by Cameyo is a self-contained executable having all the virtual modules, application registries, and files.

Important files created by the Cameyo packager. [7]

- VirtApp.ini: This file contains the application properties from launching the application to performing any operation.
- Virtual engine (AppVirtDll.dll, AppVirtDll64.dll): These dynamic link libraries virtualize the application process and child processes by substituting the virtual environment similar to the host environment.
- Loader (Loader.exe): the virtual application package is concatenated to the tail of this executable file
- The virtual filesystem database (VirtFiles.db): This database holds the state of each of the application s virtual files.
- VirtReg.dat: registry entries for the application are captured in this file.
- SandboxCfg.db: It is used to differentiate between different processes and their related registry entries.

All DLL files that are packaged are kept in the location \PROG\%Program Files (x86)%\Nmap.

Some files are packaged in .pyo python format including - ScriptArgsParser.pyo, ScriptInterface.pyo

It uses variablized operating system directory path naming. For example:

The path of the operating system:

C:\Program Files (x86)\Nmap\py2exe\lib

In a virtualized form will be represented as:

C:\%Program Files (x86)%\Nmap\py2exe\lib

By this conversion feature for the path system provided by Cameyo, the agreement of the path system is accepted and works on other systems as well regardless of customized user path definitions.

Linux OS

CDE (Code, Data and Environment) is an open-source utility that is used to create a portable package for Nmap in Linux based operating system to run applications from x86-Linux to other x86-Linux machines. The source code can be found at - git://github.com/pgbovine/CDE.git

It uses ptrace (process trace) system call to track and collect the code, files, libraries and other variables required to run the application. CDE uses OKAPI (oh-copy) utility to copy the files, directory, and symlinks. OKAPI copies the files or directories making sure that the file contents or subdirectories are not changed. This is important because Nmap introspects and uses that searched path to fetch standard libraries. CDE interface is cd: the CDE code simply copies the directory with the path and utilizes the same directory/file structure as in localhost which is traced while running any application. OKAPI rewrites the contents of symlinks (all relative paths) to transform the absolute path (application dependencies file path) flow into a relative path within the destination directory.

For example:

If these packages are read/fetched from the root during nmap operation:

- /usr/lib/x86_64-linux-gnu/
- /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25

Then these will be packaged into a replica of host system directory but in cde-package directory

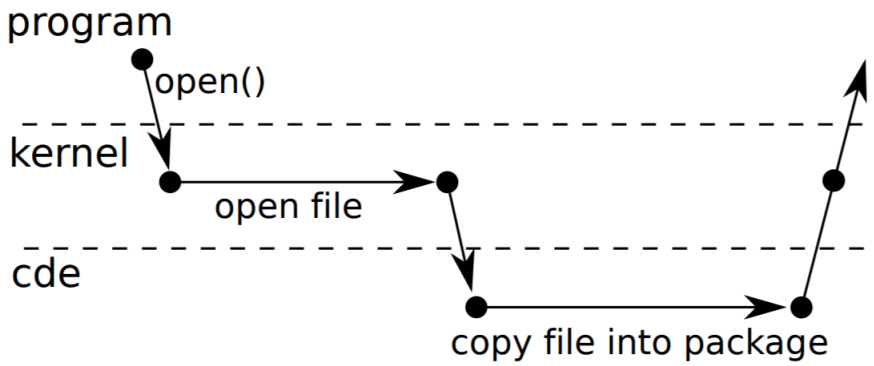
- cde-package/cde-root/usr/lib/x86_64-linux-gnu/
- cde-package/cde-root/usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25

The complete path for the exact environment variables are copied.

-rwxr-xr-x > okapi installation assigned permissions.

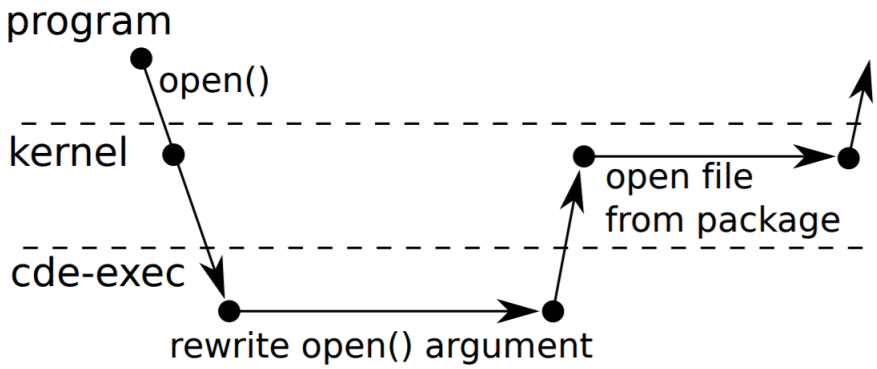
CDE Operation

The below figure shows the Timeline of flow between the target program, kernel, and cde process during an open system call.



Source: http://www.pgbovine.net/publications/CDE-create-portable-Linux-packages_LISA-2011.pdf

The below-mentioned figure shows the timeline of flow between the target program, kernel, and cde-exec during an open system call.



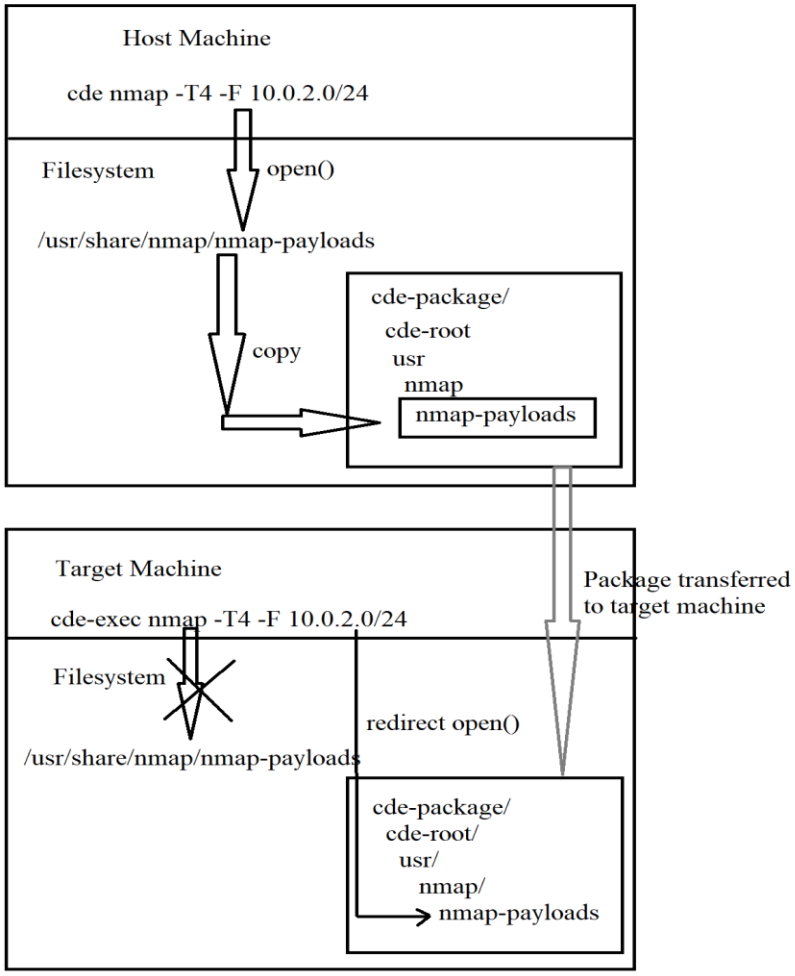
Source: http://www.pgbovine.net/publications/CDE-create-portable-Linux-packages_LISA-2011.pdf

CDE Operation with Nmap

The below figure explains one of the Nmap functions with respect to the OKAPI utility used by CDE to copy the environment variables.

Here while running Nmap with CDE, Nmap accesses one of its payloads/data. CDE with the help of ptrace and OKAPI traces the paths to the relative files and OKAPI helps to store the data in a replica operation system directories in cde-package/.

After the CDE packaging operation is done, the package can be transferred to a target machine which does not have Nmap installed. The package can be extracted there and can be run with the help of ‘cde-exec’ command. While running Nmap, dependencies(scripts, files, libraries) are run from within the virtual/replica file system directory structure created in the initial process with the help of OKAPI. In this way, the Nmap application does not check the local host file system for the files required to run for Nmap.



Since only the files that were accessed during the execution of the program are captured, it can only provide the functions on the executed program path. CDE memory also creates paths to file

system which has no files since some programs check for such paths/libraries while they are being launched to do initial check. This way the virtualized package does not go into run-time errors.

Even though CDE works like chroot it does not require root access to run because of the availability of the dependencies it needs to run.

Limitations:

- if the arguments used in Nmap operations require different dependency (file, library, script) the cde-package will not work since it did not capture the exact file system required to run that command.
- Apart from CDE, there are some limitation of ptrace as well - ptrace can cause subtle differences in the semantics of traced processes, most notably that a process being monitored by ptrace cannot itself ptrace another process [8]

Other Tools

VMware ThinApp

Its an application virtualization tool that enables packaging and managing an application within one executable file. As with Cameyo ThinApp also capture the installation of the applications. Hence it creates all the dependencies (files and registry settings) in an isolated environment. The abstracted operating system resources are presented to the application as virtualized resources. This virtual environment presented may contain all the dependencies like – environment variables, ActiveX controls, registry keys, files, etc. ThinApp uses a build process to link the virtualized environment with the compressed embedded file system and registry into a single executable.[33]

Since it runs in user mode where all the applications run, it cannot change the kernel settings or the operating system files. ThinApp allows you to change the settings of the output executable. The capture od the installation results in a text file (Project.ini). To modify the virtualized registry, open the registry text file, make changes to the file, and recompile the code again to reflect the changes. Changing the configuration of the portable executable is as simple as that with VMware Thinapp. These features come at a very high price in regards to licensing.

Docker Containers

Containers are executable instances of an image. It is well isolated from other containers and the host operating system as well. Due to this, they have an extremely small footprint on the host OS. They contain all the bins and libraries needed to run the application. Containers are run on top of docker engines that use Linux kernel features like namespaces and control groups. Docker helps automate the application deployment on the container. Apart from this, containers can be used to create portable applications as well.

Turbo.net

Turbo.net is also an application packager with some additional features. With Turbo.net functioning, we can isolate the application dependencies for TCP/UDP named object calls as well. The applications' isolated network stack can work side by side the host operating system network stack. It has the capability to run multiple applications with isolated network stack on the same host operating system side by side.

The discussion of the tools that provide features that help in creating a standalone package is limited to the above-mentioned tools. There can be many more tools that can help in virtualizing any application and run them in an isolated environment.

This report focuses on the available open-source and free tools that can be used to create the standalone package provided the functionality of the laid out policies and parameters runs w.r.t the configuration that has been done.

Using Cameyo For Creating Nmap Package on Windows OS

Before starting the installation capture process make sure that the below-mentioned binaries or libraries related to these binaries are not installed on the system.

nmap-7.80-setup.exe [4]

npcap-0.9987.exe [4]

The following operating system specification has been used to capture the installation process.

OS Name – Microsoft Windows 10 Home

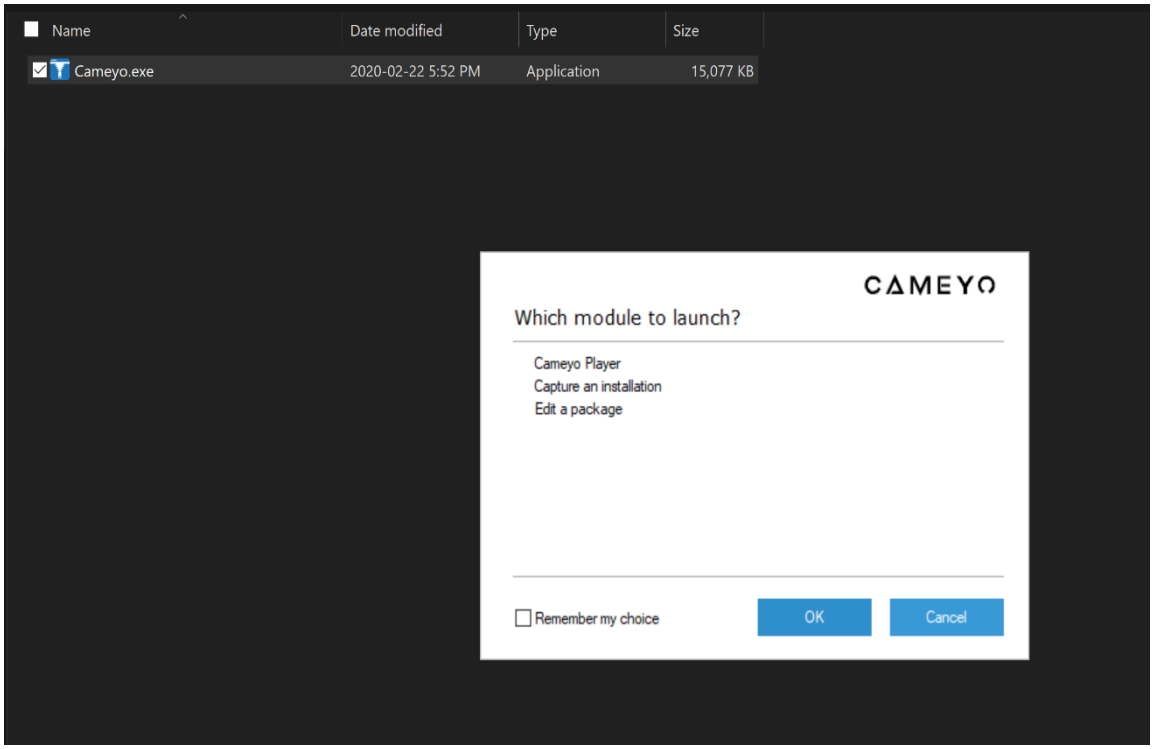
Processor – Intel® Core™ i7-1065G7 with 4 cores

RAM – 16GB DDR4x

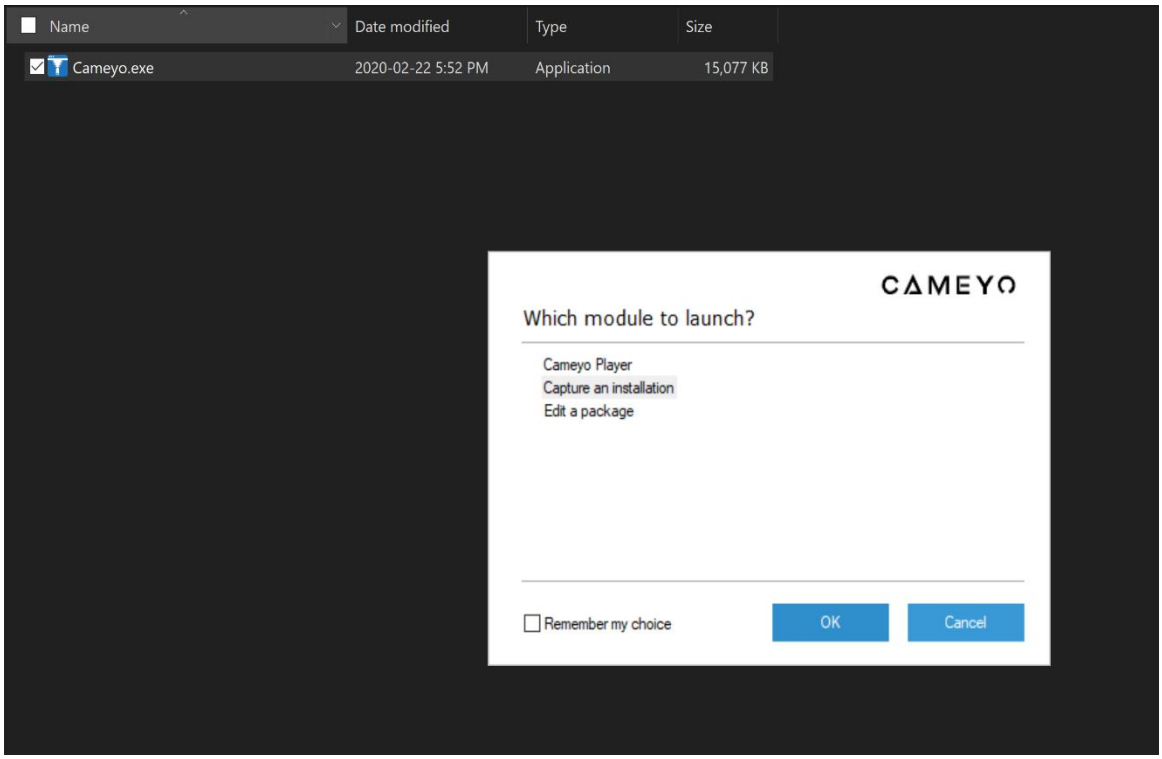
Higher processing power is recommended so that Cameyo can process taking the snapshot in less time.

It is not mandatory to perform these steps with the above-mentioned configurations.

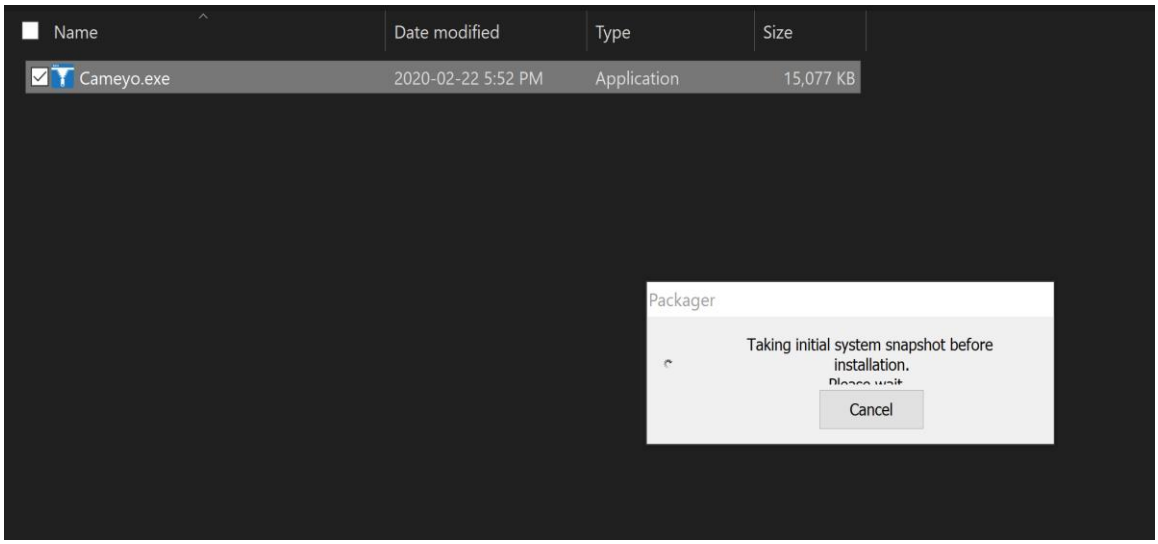
1. Open the packager file to initiate pre-configuration requirements. The Cameyo packager is itself a portable standalone file and runs all its dependencies w.r.t a single executable file as shown in the picture below.



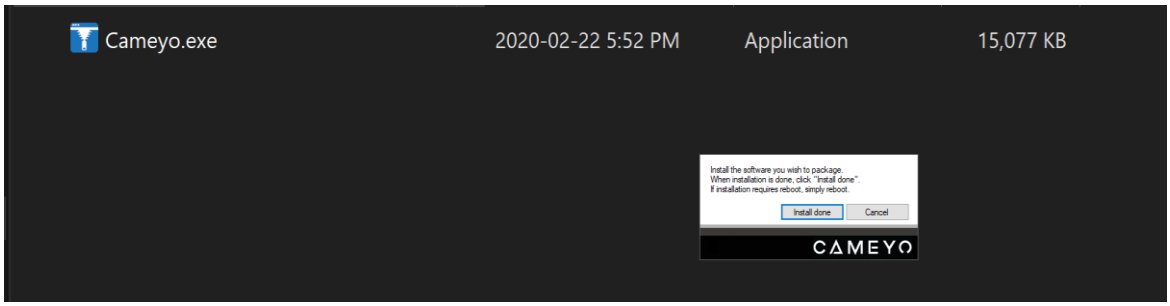
2. To package the application select the “Capture an installation” option and click OK.



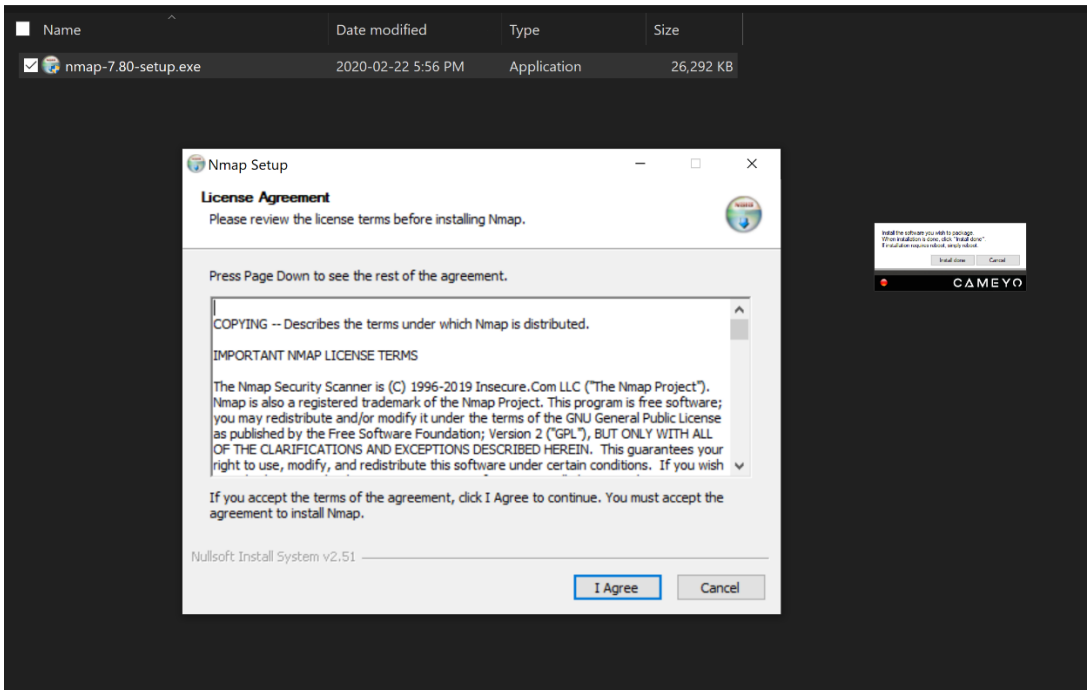
3. Cameyo will start taking a pre-installation snapshot. This is done to compare the current configuration of the system and installed applications with the changed configuration after installation of the Nmap is complete.



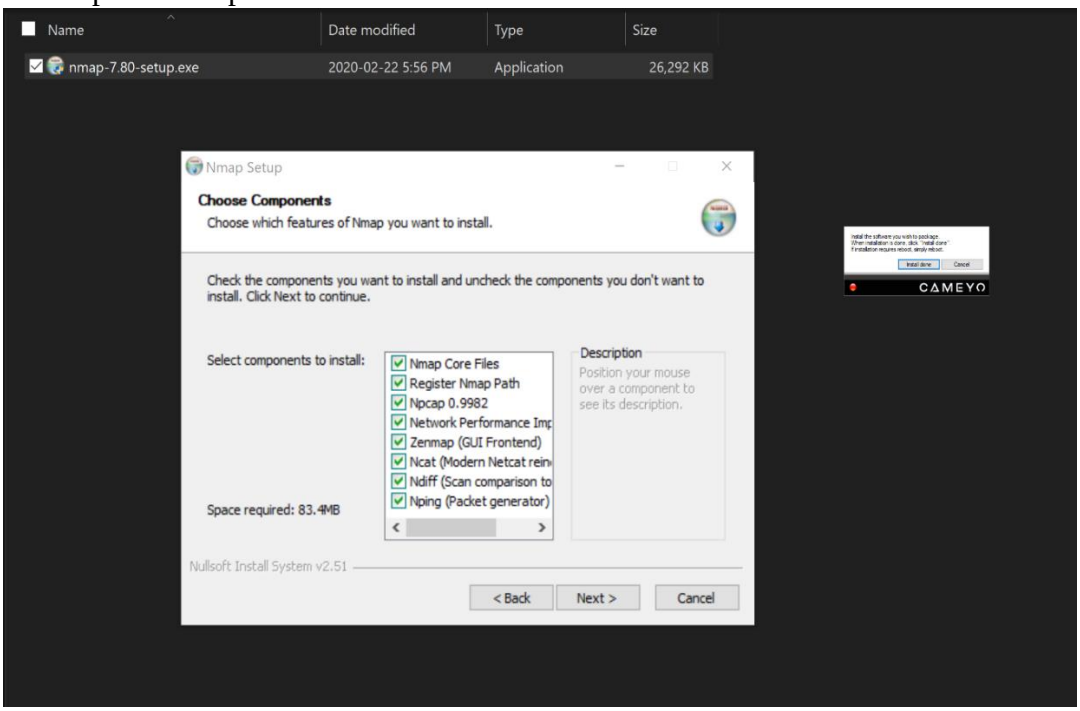
4. After the initial snapshot is taken, Cameyo will start recording all of the files and registry entries required by the NMAP application during its installation process.



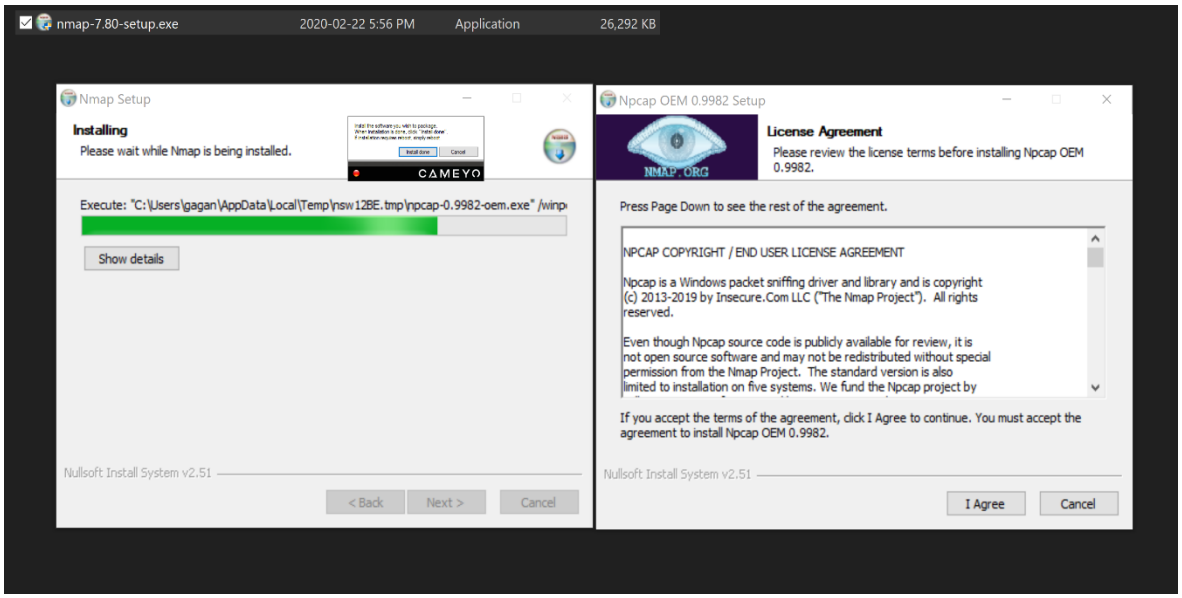
5. Start the Nmap installation process via its binaries.



6. Make sure to select the necessary installation components which include Npcap and Zenmap. All components are chosen for this installation.

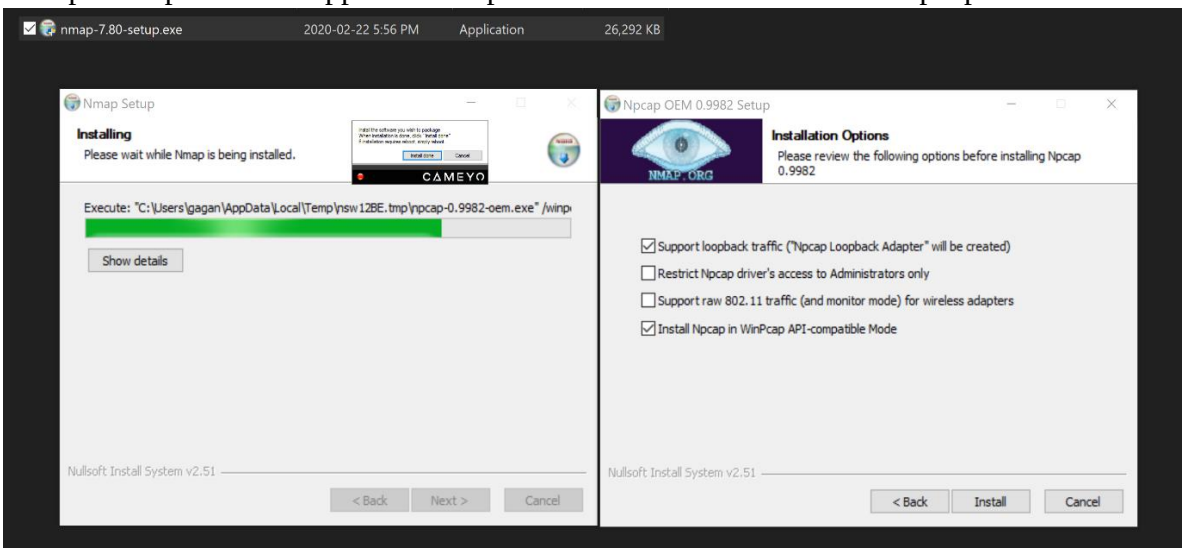


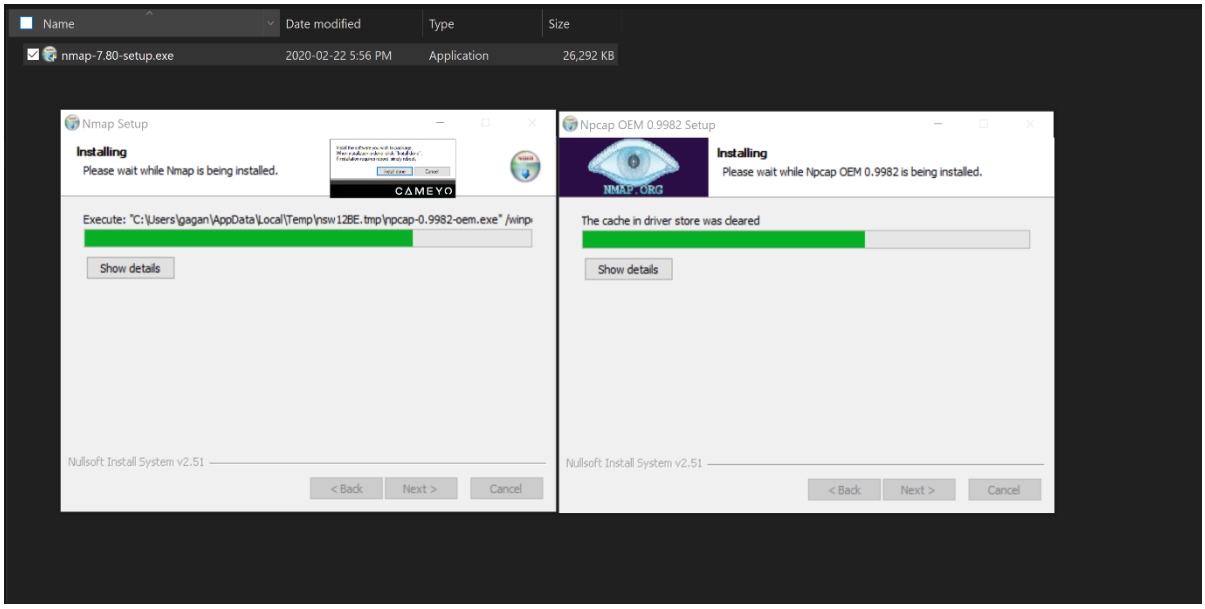
Zenmap component is required as Cameyo needs a shortcut executable file to interpret and run the Nmap application based on the binary created during the installation process.



The Npcap installation is performed with respect to its compatibility with Windows 7, 8 and 10.

7. Accept the options for support for loopback traffic and installation of Npcap in Win

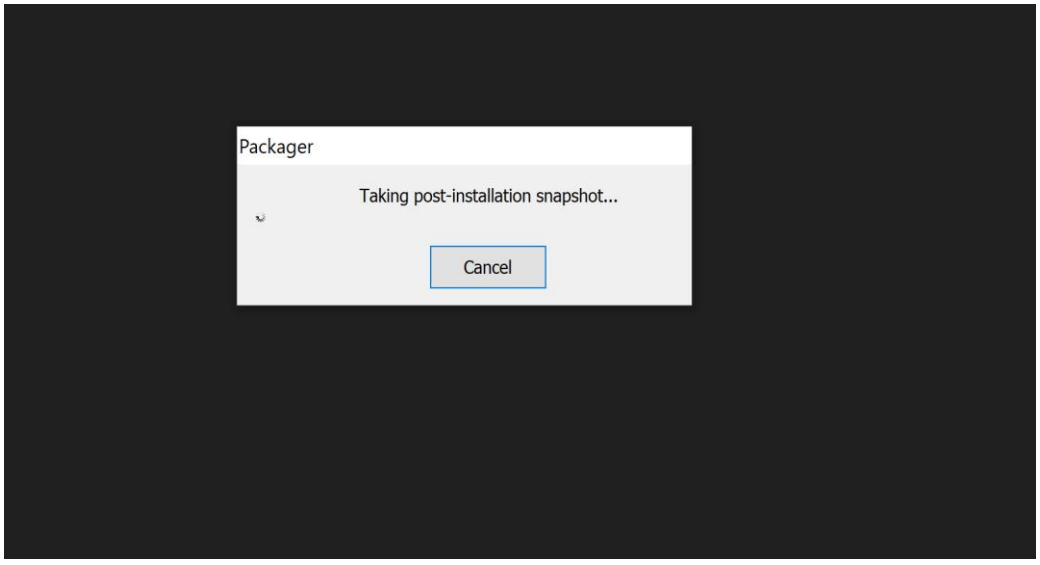


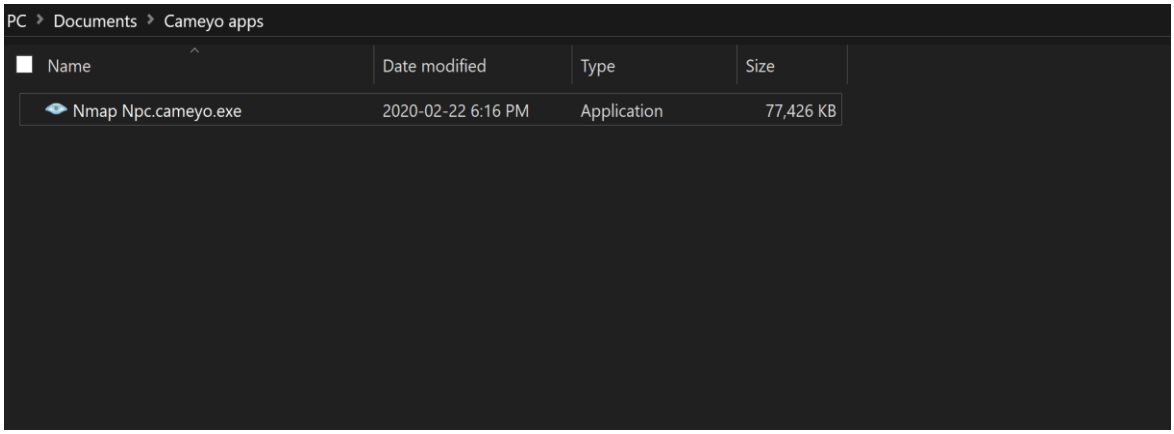


After the installation is complete, make sure to create shortcuts for Zenmap. This helps Cameyo understand the primary application requirement for opening Nmap in a portable format i.e. Zenmap GUI.

Make sure to not install any other application/software after this and only proceed to select the Installation Done option on Cameyo GUI. This action will command Cameyo to start a post-installation snapshot to check for the changes in the modification/creation of new files so that all the files can be compiled together into a standalone package.

8. After taking post snapshot Cameyo will create a standalone file that you can run from anywhere without needing its installation. This package will include all the files, libraries, scripts, and other dependencies to run Nmap.





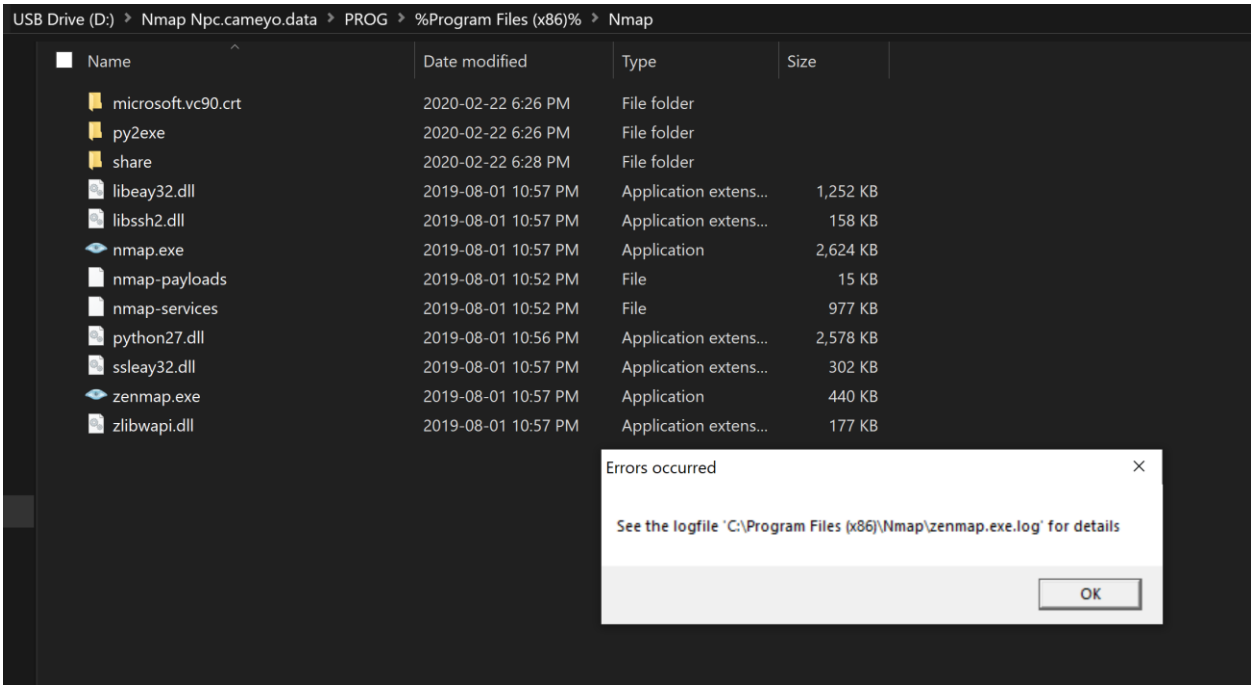
Usage and Trade-Offs of Standalone Executable on Windows

This standalone executable can be run on any other machine either via on the memory or via a USB. The executable contains all the dependencies necessary to run Nmap without installation. The Nmap is accessible only via its GUI interface i.e. Zenmap.

The executable has been tested on Windows 7, 8 and 10. The performance of the application gets slower if it is run through an external device than compared to running it from its memory.

The standalone executable though having many advantages, it comes with trade-offs with the way application functions.

- Application requires a lot of time to execute from an external memory bus. This is because, during the launch of the application for the first time, the directory structure needs some time to initialize.
- The application gets slower. Slower performance results in more time required for large network scans.
- Some of the error or log files are not created – hence some error details can be found if occurred.
  Below is the screenshot showing the same in Nmap implementation and generation of errors.



- You have to perform fresh installation and create a standalone package from the start if any of the files are lost or replaced.
- For advanced features, there might be limited functionality available. If any new NSE script has additional library dependency then the application won't give necessary results.

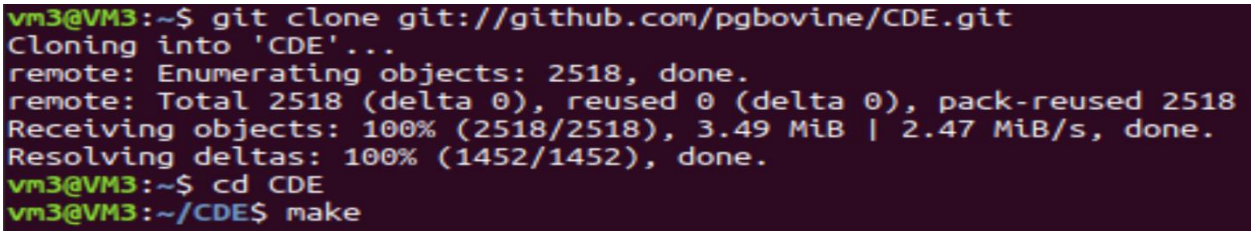
Using CDE For Creating Nmap Package on Linux

Before the installation and using CDE, make sure that below-mentioned components are present on the Linux machine:

- Nmap – Nmap can be installed on Ubuntu bu using the command – "Sudo apt install nmap". This will install three packages - libblas3, liblinear3, and nmap
- Git – It is used for version/revision control (VCS) of source code for software installation and development. The local directory can act as git repository. It is basically used for source code management. [9]
  It can be installed by using the shell with command – "sudo apt install git"
- Make – This utility automatically determines the pieces of programs that need to be recompiled and uses commands to recompile them. [10]
  The following shell command is used to install this utility - "sudo apt install make"
- For running CDE there is a need for C compiler. If it is not installed, the CDE installation stops with the error: "configure: error: no acceptable C compiler found in $PATH"
  Installing the compiler can be done by using the utility – build-essential packages. This package includes all the libraries and compilers used to compile a Debian Package.

It can be installed by using this command in shell in the root directory: “sudo apt-get install build-essential”

1. Use the following commands to fetch the packages from the GitHub repository.
   vm3@VM3:~$ git clone git://github.com/pgbovine/CDE.git



2. This will create a directory named CDE in the home directory. Move to the directory CDE and issue the command “make”. Use the below commands in the same order to do that:

   cd CDE
   make
   (This will compile the CDE code.)

3. Run any kind of Nmap script beginning with ./cde. Here quick scan command has been used for the network in which the Linux machine resides. Nmap will run normally while using CDE with it in the shell or command line. This will create a self-contained standalone executable for the command Nmap. CDE will use process tracing in order to monitor the files, scripts, and libraries that are being used nu Nmap.
   Note- You can create your own package defined by the scripts as per the organizational scanning requirements.

   If there is a need to perform an intense scan then it is recommended to create a package with that set of commands. This makes sure that all the dependencies required to run that scan are packaged into a directory. (This is also a trade-off while creating packages with CDE)

```
vm3@VM3:~$ cd CDE
vm3@VM3:~/CDE$ ./cde nmap -T4 -F 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2020-02-23 03:01 MST
Nmap scan report for _gateway (10.0.2.2)
Host is up (0.0026s latency).
Not shown: 96 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
6646/tcp open  unknown

Nmap scan report for 10.0.2.3
Host is up (0.0022s latency).
Not shown: 96 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
6646/tcp open  unknown

Nmap scan report for 10.0.2.4
Host is up (0.0030s latency).
Not shown: 96 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
6646/tcp open  unknown

Nmap scan report for VM3 (10.0.2.15)
Host is up (0.00069s latency).
All 100 scanned ports on VM3 (10.0.2.15) are closed
```

4. This will create a sub-directory named cde-package inside the CDE directory.

```
vm3@VM3:~/CDE$ ls
cde          cde-package  mike-lin-patch.txt   readelf-mini  strace-4.5.20.tar.bz2  tests
cde-exec     first-draft  nasty-bash-hack.txt  README        strace-4.6
cde.options  Makefile     okapi                scripts       strace-4.6.tar.bz2
vm3@VM3:~/CDE$ find cde-package/ | less
```

5. The package that has been created needs to be archived into a single file. To do that tar is used with cvf options. The description is mentioned below: [11]
   Tar – archiving utility
   C – creating the archive
   V – request for verbose operation
   F – specifies the name of the archive on which tar operation can be done.
   Nmap-package.tar has been used as a name for -f option.

```
vm3@VM3:~/CDE$ tar -cvf nmap-package.tar cde-package/
cde-package/
cde-package/cde.options
cde-package/cde.full-environment
cde-package/nmap.cde
cde-package/cde-exec
cde-package/cde.log
cde-package/cde.uname
cde-package/cde-root/
cde-package/cde-root/lib64/
cde-package/cde-root/lib64/ld-linux-x86-64.so.2
cde-package/cde-root/etc/
```

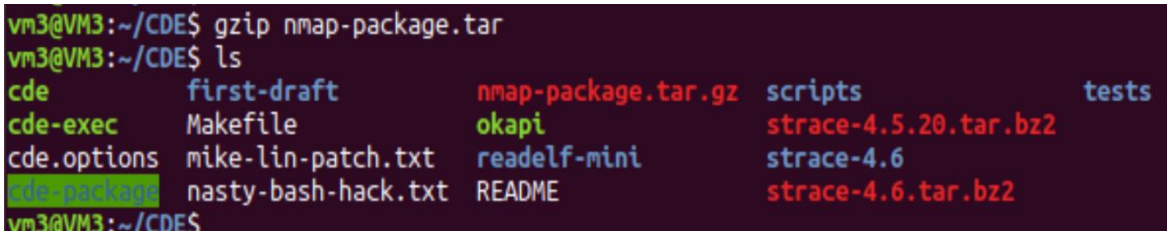
The complete list of archived files that will be used to run Nmap on a remote machine without installation is shown below. These files contain the snapshot of all the scripts, files, and libraries that Nmap used to run its operation for a quick scan.

cde-package/
cde-package/cde.options
cde-package/cde.full-environment
cde-package/nmap.cde
cde-package/cde-exec
cde-package/cde.log
cde-package/cde.uname
cde-package/cde-root
cde-package/cde-root/lib64
cde-package/cde-root/lib64/ld-linux-x86-64.so.2
cde-package/cde-root/etc
cde-package/cde-root/etc/nsswitch.conf
cde-package/cde-root/etc/alternatives
cde-package/cde-root/etc/alternatives/libblas.so.3-x86_64-linux-gnu
cde-package/cde-root/etc/localtime
cde-package/cde-root/etc/ld.so.cache
cde-package/cde-root/etc/hosts
cde-package/cde-root/usr
cde-package/cde-root/usr/share
cde-package/cde-root/usr/share/nmap
cde-package/cde-root/usr/share/nmap/nmap.xsl
cde-package/cde-root/usr/share/nmap/nmap-payloads
cde-package/cde-root/usr/share/nmap/nmap-services
cde-package/cde-root/usr/share/zoneinfo
cde-package/cde-root/usr/share/zoneinfo/America
cde-package/cde-root/usr/share/zoneinfo/America/Edmonton

cde-package/cde-root/usr/bin
cde-package/cde-root/usr/bin/nmap
cde-package/cde-root/usr/lib
cde-package/cde-root/usr/lib/x86_64-linux-gnu
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.25
cde-package/cde-root/usr/lib/x86_64-linux-gnu/liblua5.3.so.0.0.0
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libstdc++.so.6
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libssl.so.1.1
cde-package/cde-root/usr/lib/x86_64-linux-gnu/liblinear.so.3
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libpcap.so.0.8
cde-package/cde-root/usr/lib/x86_64-linux-gnu/liblua5.3.so.0
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libpcap.so.1.8.1
cde-package/cde-root/usr/lib/x86_64-linux-gnu/blas
cde-package/cde-root/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
cde-package/cde-root/usr/lib/x86_64-linux-gnu/blas/libblas.so.3
cde-package/cde-root/usr/lib/x86_64-linux-gnu/liblinear.so.3.2.
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1
cde-package/cde-root/usr/lib/x86_64-linux-gnu/libblas.so.3
cde-package/cde-root/home
cde-package/cde-root/home/vm3
cde-package/cde-root/home/vm3/CDE
cde-package/cde-root/home/vm3/CDE/nmap.cde
cde-package/cde-root/lib
cde-package/cde-root/lib/x86_64-linux-gnu
cde-package/cde-root/lib/x86_64-linux-gnu/ld-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_files-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libpthread-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libpcre.so.3
cde-package/cde-root/lib/x86_64-linux-gnu/libnsl.so.1
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_nis.so.2
cde-package/cde-root/lib/x86_64-linux-gnu/libpthread.so.0
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_compat-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libgcc_s.so.1
cde-package/cde-root/lib/x86_64-linux-gnu/libm.so.6
cde-package/cde-root/lib/x86_64-linux-gnu/libc-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_compat.so.2
cde-package/cde-root/lib/x86_64-linux-gnu/libc.so.6
cde-package/cde-root/lib/x86_64-linux-gnu/libm-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libdl-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libdl.so.2
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_nis-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libpcre.so.3.13.3
cde-package/cde-root/lib/x86_64-linux-gnu/libz.so.1.2.11

cde-package/cde-root/lib/x86_64-linux-gnu/libnsl-2.27.so
cde-package/cde-root/lib/x86_64-linux-gnu/libz.so.1
cde-package/cde-root/lib/x86_64-linux-gnu/libnss_files.so.2

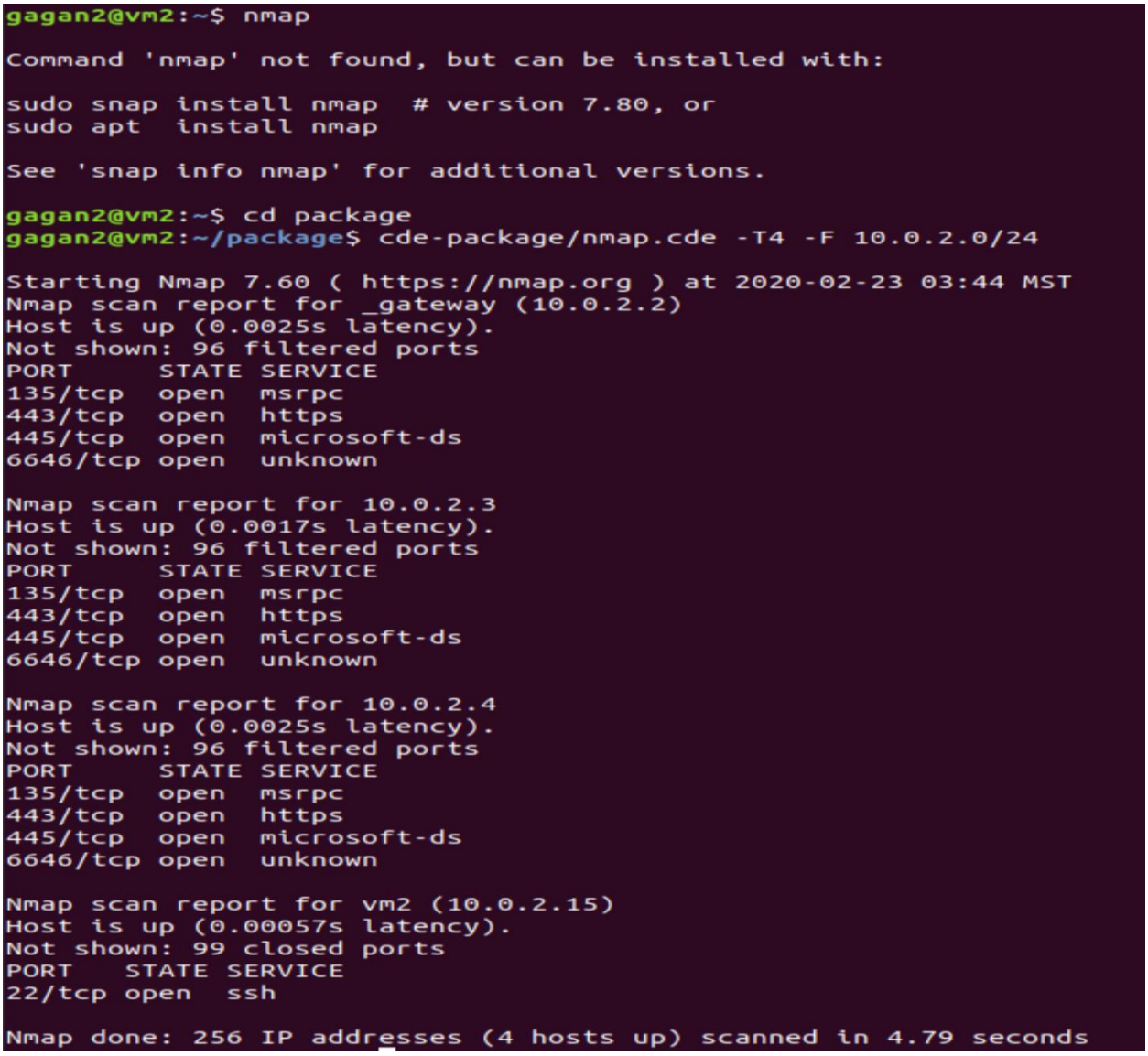
6. Gzip will be used to reduce the file size. gzip utility will automatically keep the name of the file the same.



Usage and Trade-Offs of CDE Package on Linux (Ubuntu)
Once the package has been created you can transfer that file via TFTP, SCP, SFTP, or by any other way into another machine. To check if Nmap is installed on that machine just enter the command "nmap" and hit enter. If Nmap is not installed on the machine it will ask you to install it by using the commands mentioned in the screenshot.

Once verified that Nmap is not installed you can move to the directory where Nmap package is stored by using cd utility and unzip the file by using tar utility with -zxvf options. The sample command is shown below:
tar -zxvf nmap-package.tar.gz
This will unzip all the files (same as step 5 in the last section)

To run the portable package just use the file nmap.cde created during the process mentioned in the previous section.
cde-package/nmap.cde -T4 -F 10.0.2.0/24



Other Use Cases

Audacity

Audacity is a free and open-source software used for audio editing. Its portable version is created by using a portable application packager – Enigma Virtual Box. Enigma virtual box uses a different method to package an application binaries, directories, files into a single standalone executable. Software needs to be installed in advance so that Enigma can retrieve the executable file along with the folder that contains the executable files to collect all the directories and subdirectories associated with its installation. Enigma Virtual box runs the Audacity executable along with its special loader whose function is to run along with the main code of the executable/application and

intercept all the system calls (file read/write) made by the executable and collect and virtualize the operations into a file on memory. [30]

Philips Hue

Philips Hue is a tool to connect light bulbs to the internet in order to create a usage data and use the data for saving power, security, and automation. It is created with the help of google containers which also works by isolating the environment in which the application can run along with its packaged libraries and other dependencies. Usage of this tool can be often compared to a lightweight portable application. [31]

Use cases with Cameyo

There are many packages that have been created with the help of Cameyo such as Youtube Downloader [23], WinSnap [24], Opera [25] (The steps remain the same).

Use cases with CDE

Reporting Bugs: CDE can be used to report various issues with an application. CDE package can be created while the bug occurs and the package can be sent to the developer team to evaluate the issue and provide a fix.

Google Earth application package has been created with the use of CDE [26].

Packages are posted on GitHub for multiple use cases for example - downloading WFD related data by using CDE packaged variables. [27]

More use cases can be found on http://www.pgbovine.net/cde/manual/use-cases.html.

Other Methods

Apart from using the tools for creating a standalone portable application package, other methods are also used to isolate the environment in which the application runs. For example, a portable version of Firefox is created by extracting the executable file into a USB drive and creating a file with the name start.bat in the Core directory (directory created upon extracting the Mozilla executable file). You can add the following code to the bat file[32]:

```
@echooff
startC:\Firefox\core\firefox.exe-Profile“Profile/”
exit
```

Then you can just run the bat file to execute firefox.

Risks of using Portable Application

There are multiple instances where employees use IT resources – software or devices that are not permitted by the IT department, also known as Shadow IT practices. Portable applications come under shadow IT resources Devices containing software packages and USB itself are sources of major breaches within an organization or for a personal device. TechAdvisory.org reports that 25 percent of malware (malicious programs) is spread today through USB devices. Malware can be run through the autorun feature present in the PC. Once the malware has been injected in a device, it can further spread to other devices.

Some of the potential risks associated with using portable devices is because of their small size they can be carried anywhere and can be accessed or stolen by anyone. Hackers can inject viruses into devices like USB and when you use it the machine gets affected. Portable applications can get access to organizational data and becomes a reason for creating vulnerabilities. Since portable applications do not leave any footprint on the device on which they are run, the auditing applications often are not successful in detecting these applications or software.

Risk Management

- Keep the storage device containing portable software in a safe location which had limited physical access and to trustworthy only.
- Use anti-virus software to scan the USB stick or any new portable software before using it.
- Disable autorun feature on the PC.
- Use strong encryption while storing data on a USB.
- Protect the access of a portable device or software with the help of passwords.
- Proper implementation of tools that can detect and monitor every state of an operating system either by polling/sending traps or data stored inside the device itself can be handy. Services that these tools offer might not be limited to the detection of file creation, execution, time, and the functions that the executable performed. Tools such as these can help in forensics.
- There should be proper implementation of infrastructure security where only limited access to different zones of the network is allowed. Different zones might include – LAN, WAN, DMZ, Internet, Production, Mission Critical, etc.

Ethics for using Portable Applications

- Only those software should be made portable which is approved by the IT department.
- Application Packages should not be created with the malicious code that can affect some other program running on the target machine.
- Portable application should be tested in an isolated environment before using it on production servers.

- Portable application should not be used for any other purpose other than what is required and tested.
- Practice response plan if the usage of the software goes wrong. Implement use cases for the same.

Future of Portability – Applications & Operating Systems

With the development of virtualization of applications, and then operating systems we are moving towards an era where the operating system resources are very well utilized and the cost of the operations is becoming less day by day. But, with the creation of all these technologies make the working environment complex with multiple types of operating systems, application resource usage requirements, monitoring of those requirements at the same time causes operations to become more complex and hence, as a result, we need more tools that can run in conjunction with an application or portable application running on virtualized systems to make the best use of them.

Portability should not be only available for application packages, it can be used to make the operating system portable as well. If the operating system is portable itself, it can just use the resources like a processor to run the programs saved in the OS directories present either in an external drive or on the cloud. Another perspective can be related to the building of applications that run on one operating system which would make everything easy starting from deployment to troubleshooting issues related to the combination of the two. This leads to using only one operating system everywhere which can result in better and quick solutions to multiple problems related to code. All portable applications can be designed made compatible in such a way that it runs on a single operating system where the data related to a particular user can be saved within the application profile itself and not the operating system. If the operating system is portable, the user settings can be saved in the directories of the portable operating system. All these discussion points will lead to multiple benefits:

- With portable programs and operating systems come isolation, and hence the mixup of multiple services doesn't happen resulting in better working of applications in their own space.
- Applications or Operating systems can be stored in external memory drives like External HDD, USB, CD/DVD, etc. and can be used when required. They become easy to carry.
- Even if an unnecessary portable application is present on the system, it won't run unless it is manually initiated by running its standalone executable. Eg. In regular cases, system-wide application services or their upgrades are automatic i.e. the application checks for additional data continuously making connection to the internet and utilizing resources in the background even if you're not using it, the portable applications do not run these services unless executed and those same resources can be used for other important application or services that one intends to run.
- Apart from operating system categories such as forensics and hacking/security portable operating systems can also be used for desktop replacements.

Future for portability w.r.t Security Appliances

Today the security appliances are delivered as hardware models on business sites that require it eg. Firewall, Proxy Server, Intrusion Prevention Systems, etc. It is a difficult task to recover from a hardware failure – since the operating system and hardware is integrated, the RMA – return merchandise authorization takes time for the delivery and till that time the operations are affected.

Portable versions of firewall and other services mentioned above can help in reducing the timeline for business continuity. If portable versions of such products are available, organizations can deploy these products on any operating system having the right physical resources and capable of handling the tasks that the specific product offers. This kind of virtualization on top of another operating system is already achieved with tools such as VMware and Oracle Virtual Box, but the applications are limited to run an operating system on top of a different operating system. The technology has not expanded to the point where security organizations can provide portable versions of their products having the ability to run on any operating system. More open-source projects with a involvement of larger community for developing such products is the future.

References

[1] P. Foreman, Vulnerability Management, 2nd ed., 2019.

[2] K. Murray, *Top Threats To Endpoints And How To Stay Protected,* 2019.

[3] B. Hatch, *Linux Exposed.*

[4] "NMAP," [Online]. Available: nmap.org.

[5] "LibreNMS," [Online]. Available: https://www.librenms.org/.

[6] "How User Account Control works," [Online]. Available: https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works.

[7] "Application Virtualization Technologies," [Online]. Available: https://docplayer.net/8114373-Application-virtualization-technologies-whitepaper.html.

[8] P. J. G. a. D. Engler, "CDE: Using System Call Interposition to Automatically Create Portable Software Packages," [Online]. Available: http://www.pgbovine.net/publications/CDE-create-portable-Linux-packages-short-paper_USENIX-2011.pdf.

[9] J. Mays, "How To Install Git on Ubuntu 14.04," [Online]. Available: https://www.liquidweb.com/kb/how-to-install-git-on-ubuntu-14-04/.

[10] R. S. a. R. McGrath, "make," [Online]. Available: http://man7.org/linux/man-pages/man1/make.1.html.

[11] M. PAGE, " GNU TAR Manual," [Online]. Available: http://man7.org/linux/man-pages/man1/tar.1.html.

[12] P. J. Guo, "CDE: Run Any Linux Application On-Demand Without Installation," [Online]. Available: http://www.pgbovine.net/publications/CDE-create-portable-Linux-packages_LISA-2011.pdf.

[13] r. &. c. l. michaels, "Native mobile apps: The wrong choice for business?," [Online]. Available: https://www.mrc-productivity.com/research/whitepapers/NativeAppsWrongChoice.pdf.

[14] P. Walters, "The Risks of Using Portable Devices," [Online]. Available: https://www.us-cert.gov/sites/default/files/publications/RisksOfPortableDevices.pdf.

[15] J. F. Gantz. [Online]. Available: https://download.microsoft.com/documents/en-us/sam/IDC%20Global%20Counterfeit%202013.pdf.

[16] Avecto, [Online]. Available: https://www.infosecurityeurope.com/__novadocuments/27597.

[17] Optimus Information, [Online]. Available: https://www.infosecurityeurope.com/__novadocuments/27597.

[18] Various, "Cameyo," [Online]. Available: https://en.wikipedia.org/wiki/Cameyo.

[19] P. Shannon Vallor. [Online]. Available: https://www.scu.edu/media/ethics-center/technology-ethics/IntroToCybersecurityEthics.pdf.

[20] Ashraf, "Virtualize applications with Cameyo," [Online]. Available: https://dottech.org/17969/virtualize-applications-with-cameyo/.

[21] "Cameyo - Application virtualization for Windows," [Online]. Available: https://www.techsupportalert.com/content/cameyo-application-virtualization-windows.htm.

[22] "Cameyo 3.0.1318," [Online]. Available: https://www.neowin.net/news/cameyo-301318-preview.

[23] "Portable Version," [Online]. Available: https://www.guidingtech.com/5493/portable-apps-virtualization-cameyo/.

[24] "Create Portable Applications," [Online]. Available: https://www.howtogeek.com/186132/how-to-create-portable-versions-of-applications-in-windows-8.1-using-cameyo/.

[25] "Cameyo Application Packager," [Online]. Available: https://lifehacker.com/cameyo-creates-a-portable-version-of-just-about-any-pro-5635017.

[26] P. Guo, "Escape Dependency Hell," [Online]. Available: https://www.linux.com/tutorials/escape-dependency-hell-automatically-create-portable-linux-software-using-cde/.

[27] robbriers, "cde package submission," [Online]. Available: https://github.com/ropensci/software-review/issues/284.

[28] "CDE Use Cases," [Online]. Available: http://www.pgbovine.net/cde/manual/use-cases.html.

[29] J. v. L. R. M. Ruben Spruijt, "Application Virtualization Smackdown," [Online]. Available: https://www.software2.com/downloads/16/Whitepaper_Application_Virtualization_smackdown.pdf.

[30] S. Butler, "Create a Portable Version of Any Application in Windows," [Online]. Available: https://helpdeskgeek.com/how-to/create-a-portable-version-of-any-application-in-windows/.

[31] "CASE STUDY - P H I L I P S H U E : L I G H T I N G T H E W A Y," Philips Lighting, [Online]. Available: https://static.cbsileads.com/direct/whitepapers/GuidetoOpenAppDev_eBook_Compressed_110320_17.pdf.

[32] Trisha, "Creating Portable Version of Firefox," [Online]. Available: https://www.trishtech.com/2013/09/how-to-create-portable-version-of-firefox/.

[33] VMware, "Introduction to VMware ThinApp," [Online]. Available: https://www.vmware.com/pdf/thinapp_intro.pdf.

[34] B. M. K. a. B. Santhi, "Study on Features, Statistics, and Security Measures," [Online]. Available: http://www.indjst.org/index.php/indjst/article/view/33920/27892.