Package Management

Whether working as a system administrator, maintaining our own Linux machines at home, or building/upgrading/maintaining our penetration testing distribution of choice, it is crucial to have a firm grasp on the available Linux package managers and the various ways to utilize them to install, update, or remove packages. Packages are archives that contain binaries of software, configuration files, information about dependencies and keep track of updates and upgrades. The features that most package management systems provide are:

- `Package downloading`
- `Dependency resolution`
- `A standard binary package format`
- `Common installation and configuration locations`
- `Additional system-related configuration and functionality`
- `Quality control`

We can use many different package management systems that cover different types of files like ".deb", ".rpm", and others. The package management requirement is that the software to be installed is available as a corresponding package. Typically this is created, offered, and maintained centrally under Linux distributions. In this way, the software is integrated directly into the system, and its various directories are distributed throughout the system. The package management software retrieves the necessary changes for system installation from the package itself and then implements these changes to install the package successfully. If the package management software recognizes that additional packages are required for the proper functioning of the package that has not yet been installed, a dependency is included and either warns the administrator or tries to reload the missing software from a repository, for example, and install it in advance.

If an installed software has been deleted, the package management system then retakes the package's information, modifies it based on its configuration, and deletes files. There are different package management programs that we can use for this. Here is a list of examples of such programs:

Command	Description
`dpkg`	The `dpkg` is a tool to install, build, remove, and manage Debian packages. The primary and more user-friendly front-end for `dpkg` is aptitude.
`apt`	Apt provides a high-level command-line interface for the package management system.
`aptitude`	Aptitude is an alternative to apt and is a high-level interface to the package manager.
`snap`	Install, configure, refresh, and remove snap packages. Snaps enable the secure distribution of the latest apps and utilities for the cloud, servers, desktops, and the internet of things.
`gem`	Gem is the front-end to RubyGems, the standard package manager for Ruby.
`pip`	Pip is a Python package installer recommended for installing Python packages that are not available in the Debian archive. It can work with version control repositories (currently only Git, Mercurial, and Bazaar repositories), logs output extensively, and prevents partial installs by downloading all requirements before starting installation.
`git`	Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

It is highly recommended to set up our virtual machine (VM) locally to experiment with it. Let us experiment a bit in our local VM and extend it with a few additional packages. First, let us install `git` by using `apt`.

Advanced Package Manager (APT)

Debian-based Linux distributions use the `APT` package manager. A package is an archive file containing multiple ".deb" files. The `dpkg` utility is used to install programs from the associated ".deb" file. `APT` makes updating and installing programs easier because many programs have dependencies. When installing a program from a standalone ".deb" file, we may run into dependency issues and need to download and install one or multiple additional packages. `APT` makes this easier and more efficient by packaging together all of the dependencies needed to install a program.

Each Linux distribution uses software repositories that are updated often. When we update a program or install a new one, the system queries these repositories for the desired package. Repositories can be labeled as stable, testing, or unstable. Most Linux distributions utilize the most stable or "main" repository. This can be checked by viewing the contents of the `/etc/apt/sources.list` file. The repository list for Parrot OS is at `/etc/apt/sources.list.d/parrot.list`.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ cat /etc/apt/sources.list.d/parrot.list

# parrot repository
# this file was automatically generated by parrot-mirror-selector
deb http://htb.deb.parrot.sh/parrot/ rolling main contrib non-free
#deb-src https://deb.parrot.sh/parrot/ rolling main contrib non-free
deb http://htb.deb.parrot.sh/parrot/ rolling-security main contrib non-free
#deb-src https://deb.parrot.sh/parrot/ rolling-security main contrib non-free
```

APT uses a database called the APT cache. This is used to provide information about packages installed on our system offline. We can search the APT cache, for example, to find all `Impacket` related packages.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ apt-cache search impacket

impacket-scripts - Links to useful impacket scripts examples
polenum - Extracts the password policy from a Windows system
python-pcapy - Python interface to the libpcap packet capture library (Python 2)
python3-impacket - Python3 module to easily build and dissect network protocols
python3-pcapy - Python interface to the libpcap packet capture library (Python 3)
```

We can then view additional information about a package.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ apt-cache show impacket-scripts

Package: impacket-scripts
Version: 1.4
Architecture: all
Maintainer: Kali Developers <devel@kali.org>
Installed-Size: 13
Depends: python3-impacket (>= 0.9.20), python3-ldap3 (>= 2.5.0), python3-ldapdomaindump
Breaks: python-impacket (<< 0.9.18)
Replaces: python-impacket (<< 0.9.18)
Priority: optional
Section: misc
Filename: pool/main/i/impacket-scripts/impacket-scripts_1.4_all.deb
Size: 2080
<SNIP>
```

We can also list all installed packages.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ apt list --installed

Listing... Done
accountsservice/rolling,now 0.6.55-2 amd64 [installed,automatic]
adapta-gtk-theme/rolling,now 3.95.0.11-1 all [installed]
adduser/rolling,now 3.118 all [installed]
adwaita-icon-theme/rolling,now 3.36.1-2 all [installed,automatic]
aircrack-ng/rolling,now 1:1.6-4 amd64 [installed,automatic]
<SNIP>
```

If we are missing some packages, we can search for it and install it using the following command.

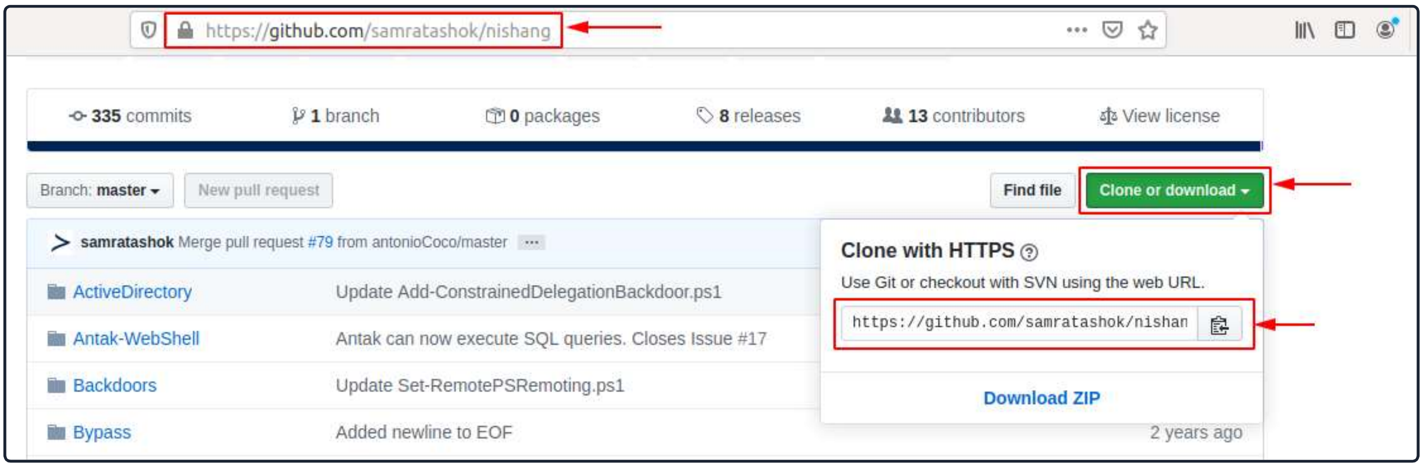
Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ sudo apt install impacket-scripts -y

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  impacket-scripts
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,080 B of archives.
After this operation, 13.3 kB of additional disk space will be used.
Get:1 https://euro2-emea-mirror.parrot.sh/mirrors/parrot rolling/main amd64 impacket-scripts all 1.4 [2,080 B]
Fetched 2,080 B in 0s (15.2 kB/s)
Selecting previously unselected package impacket-scripts.
(Reading database ... 378459 files and directories currently installed.)
Preparing to unpack .../impacket-scripts_1.4_all.deb ...
Unpacking impacket-scripts (1.4) ...
Setting up impacket-scripts (1.4) ...
Scanning application launchers
Removing duplicate launchers from Debian
Launchers are updated
```

Git

Now that we have git installed, we can use it to download useful tools from Github. One such project is called 'Nishang'. We will deal with and work with the project itself later. First, we need to navigate to the project's repository and copy the Github link before using git to download it.



https://github.com/samratashok/nishang
335 commits
1 branch
0 packages
8 releases
13 contributors
View license
Branch: master
New pull request
Find file
Clone or download
samratashok Merge pull request #79 from antonioCoco/master
Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/samratashok/nishan
Download ZIP

ActiveDirectory	Update Add-ConstrainedDelegationBackdoor.ps1	
Antak-WebShell	Antak can now execute SQL queries. Closes Issue #17	
Backdoors	Update Set-RemotePSRemoting.ps1	
Bypass	Added newline to EOF	2 years ago

Nevertheless, before we download the project and its scripts and lists, we should create a particular folder.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ mkdir ~/nishang/ && git clone https://github.com/samratashok/nishang.git ~/nishang

Cloning into '/opt/nishang/'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 1691 (delta 4), reused 6 (delta 2), pack-reused 1676
Receiving objects: 100% (1691/1691), 7.84 MiB | 4.86 MiB/s, done.
Resolving deltas: 100% (1055/1055), done.
```

DPKG

We can also download the programs and tools from the repositories separately. In this example, we download 'strace' for Ubuntu 18.04 LTS.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ wget http://archive.ubuntu.com/ubuntu/pool/main/s/strace/strace_4.21-1ubuntu1_amd64.deb

--2020-05-15 03:27:17--  http://archive.ubuntu.com/ubuntu/pool/main/s/strace/strace_4.21-1ubuntu1_amd64.deb
Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.88.142, 91.189.88.152, 2001:67c:1562::18, ...
Connecting to archive.ubuntu.com (archive.ubuntu.com)|91.189.88.142|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 333388 (326K) [application/x-debian-package]
Saving to: 'strace_4.21-1ubuntu1_amd64.deb'

strace_4.21-1ubuntu1_amd64.deb       100%[===================================================================>] 325,

2020-05-15 03:27:18 (2,69 MB/s) - 'strace_4.21-1ubuntu1_amd64.deb' saved [333388/333388]
```

Furthermore, now we can use both `apt` and `dpkg` to install the package. Since we have already worked with `apt`, we will turn to `dpkg` in the next example.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ sudo dpkg -i strace_4.21-1ubuntu1_amd64.deb

(Reading database ... 154680 files and directories currently installed.)
Preparing to unpack strace_4.21-1ubuntu1_amd64.deb ...
Unpacking strace (4.21-1ubuntu1) over (4.21-1ubuntu1) ...
Setting up strace (4.21-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

With this, we have already installed the tool and can test if it works properly.

Advanced Package Manager (APT)

```
amit8986@htb[/htb]$ strace -h

usage: strace [-CdffhiqrtttTvVwxxy] [-I n] [-e expr]...
              [-a column] [-o file] [-s strsize] [-P path]...
              -p pid... / [-D] [-E var=val]... [-u username] PROG [ARGS]
   or: strace -c[dfw] [-I n] [-e expr]... [-O overhead] [-S sortby]
              -p pid... / [-D] [-E var=val]... [-u username] PROG [ARGS]

Output format:
  -a column      alignment COLUMN for printing syscall results (default 40)
  -i             print instruction pointer at time of syscall
```

Optional Exercise:

Search for "**evil-winrm**" tool on Github and install it on our interactive instances. Try all the different installation methods.

Start Instance

1 / 1 spawns left

Waiting to start...

Previous Next Mark Complete & Next

Cheat Sheet

Table of Contents

My Workstation

O F F L I N E

Start Instance

1 / 1 spawns left