Red Hat Package Manager for Payload Delivery

Presenter: Minakshi Pushpendra Chavan
(MIT21-A-03-PG-MTCST-34550 Roll No: 04)
Guide: Prof Dr. B. S. Sonawane

Agenda

1. Introduction and Abstract Information
2. Basics
    a. Package Management
    b. Update Methodologies
    c. Containerization
3. Delivery Mechanism
    a. RPM
    b. yum
4. Development
    a. Sources
    b. SPEC File
    c. Package Signing and Delivery
5. Conclusion
6. Questions & Answers

Introduction & Abstract

- Data Payload Delivery in Linux and Windows has always been a Challenge since ages, but over the decades, there have been many positive changes in the methodology which not only delivers the expected data to the end systems but also helps keep version control over the delivered payload.

- I have proposed dummy data delivery through secured and signed rpm build in this minor project whereas this work would build a solid framework for my major project.

Introduction & Abstract...

Throughout this presentation, we are going to discuss the following advantages of rpm building and data delivery through it.

- **Data Integrity**
  The data shared through rpm package can't be tampered with and once tampered, will be detected easily..
- **Easy Version Control**
  Easy Version Control over the delivered payload can be managed.
- **Secured**
  The package will be signed and hence trustworthy, unsigned or tampered data can be rejected.

Basics - Package Management

- Deployment Data delivery methodology is often called as Package Management and there are multiple ways data can be delivered.
- In Windows, .Net Framework can be used to deliver data using .msi <Microsoft Installer> files or even with .exe <Executable files>
- Whereas in Linux (especially Fedora based variants like Red Hat Enterprise Linux, CentOS, Fedora, Scientific Linux, Oracle Linux etc), we have methods like rpm, source code compilation and tarball delivery of mechanism.
- In current Project, we are focusing on rpm method to deliver the data.

Basics - Update Methodologies

- In Linux, configuration files can be updated using source package compilation or by replacement or the best and recommended method is to use rpm update.
- In latest releases, it is recommended to use the containerization engine to get the updated images instead of installing them into the system.
- RPM, in Legacy system, is the best way to keep the version control and update mechanism in place.
- YUM can be used as Text User Interface tool to update the system.

Basics - Containerization

- Containerization is a method of virtualization to run applications in an isolated namespaces in Linux.
- This is the latest and advanced way to utilize the system to the fullest.
- Containerization though is the latest and most sophisticated way today, needs manual handling of the workload management, hence to make it easier at the larger scale, Google introduced project kubernetes few years back.
- Containerization isn't discussed in detail during this presentation as it's out of scope at this moment in implementation.

Mark Ewing & Eric Troan are the initial authors of rpm method development

Fact

Mark Ewing is the Originator and creator of brand Red Hat. RPM hence is named as Red Hat Package Manager.

Red Hat was founded in 1993 by Mark Ewing and Bob Young!

Delivery Mechanisms - RPM

- RPM Package is nothing but a compressed archive of the details below.
  - Binaries
  - Libraries
  - Documentation - manual pages
  - Configuration Files
  - Package Information - like dependencies and package metadata like ChangeLog, Author's Name, Vendor, Signature, Build Host, Architecture, Package version and many more.

Delivery Mechanisms - RPM

- ➔ It's easy to install a single RPM if it has no dependencies.
- ➔ But if the package metadata is written in such a way that it has many dependencies and that too on different updated versions of other rpm packages, then more sophisticated tool called as YUM <Yellow Dog Updater Modifier> is used.
- ➔ In latest versions of Red Hat Enterprise Linux i.e. RHEL8, yum is replaced with dnf, still yum is present to keep backward compatibility.
- ➔ Let's discuss more about yum in next slide.

Delivery Mechanisms - YUM

- YUM not only resolves the dependencies automatically but very intelligently reads the package metadata to make Administrators life easier while handling packages.
- YUM works with the repositories which are stored generally in /etc/yum.repos.d directories.
- There are many public and free/paid repositories which provide safe and secure rpms. E.g. rpmforge, EPEL, vlc etc.
- YUM depends on RPM to function. Without RPM, yum won't work.
- One can install packages using yum by executing commands like "yum install httpd -y"

RPM Development - Stages

Stage 1 = Source files

a. Collect the source files in a tarball so that those can be packaged in the RPM.
b. The sources can be directly fetched from the Development team.

Stage 2 = SPEC File

c. Spec file is in fact called as a specification file, which contains the package metadata declaration as well as way to package the sources.
d. Spec also contains the ChangeLog i.e. a list of all changes in a specified format containing names of all authors along with the changes they made to the package and sources.

RPM Development - Stages

Stage 3 = RPM building

a. An rpm will be built using the build environment
b. This will not only build binary package but also a source package.
c. The RPM will be ready to go through the final crucial step of signing by the vendor.

Stage 4 = RPM Signing

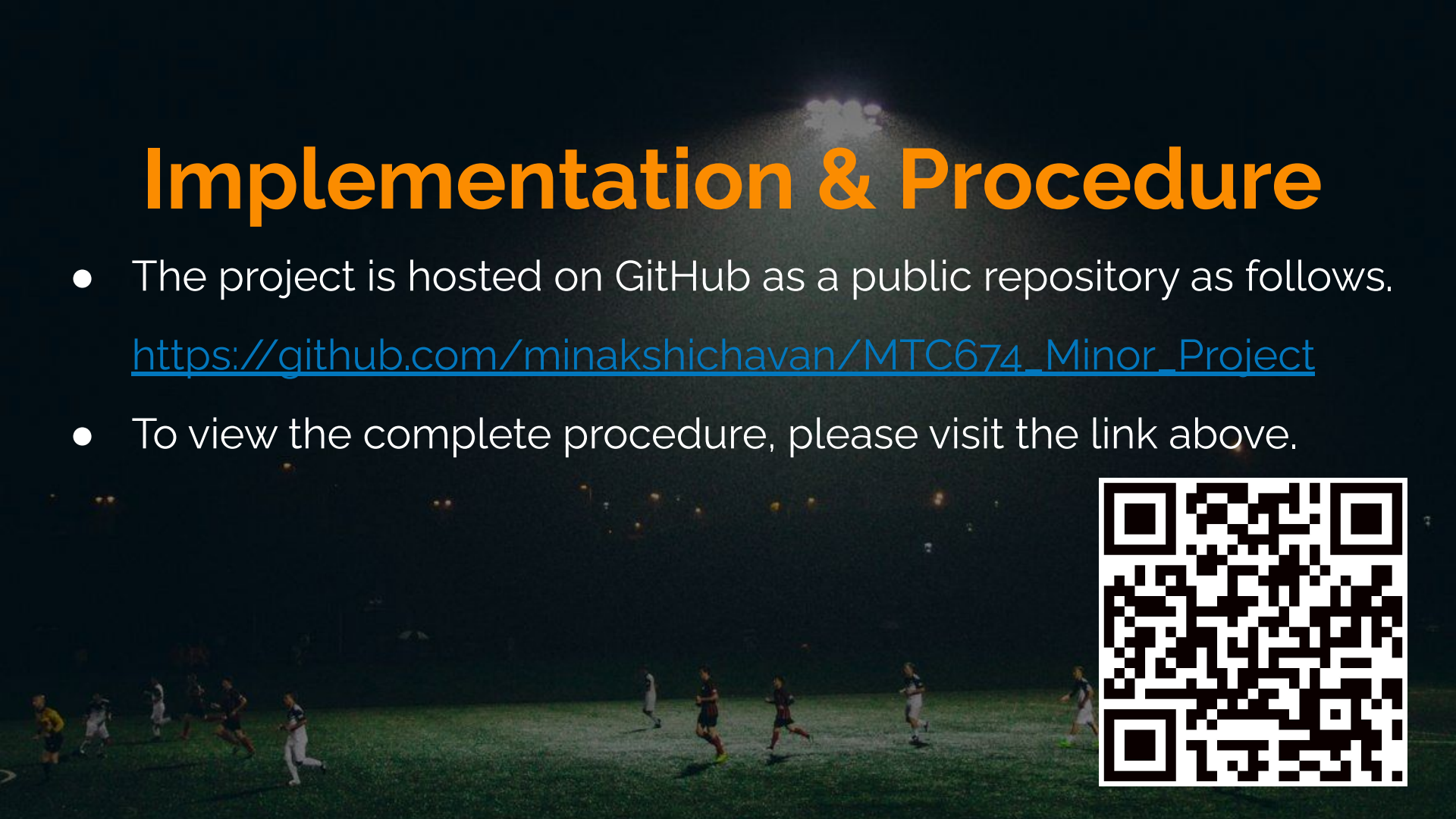
a. An rpm package can be signed by a private key generated by the vendor.
b. The package will only be installed (by default) if the public key by the vendor is marked as trusted by the clients.
c. This ensures the trust between Vendor and the clients as it ensures security and scrutiny.

Implementation & Procedure

- The project is hosted on GitHub as a public repository as follows.

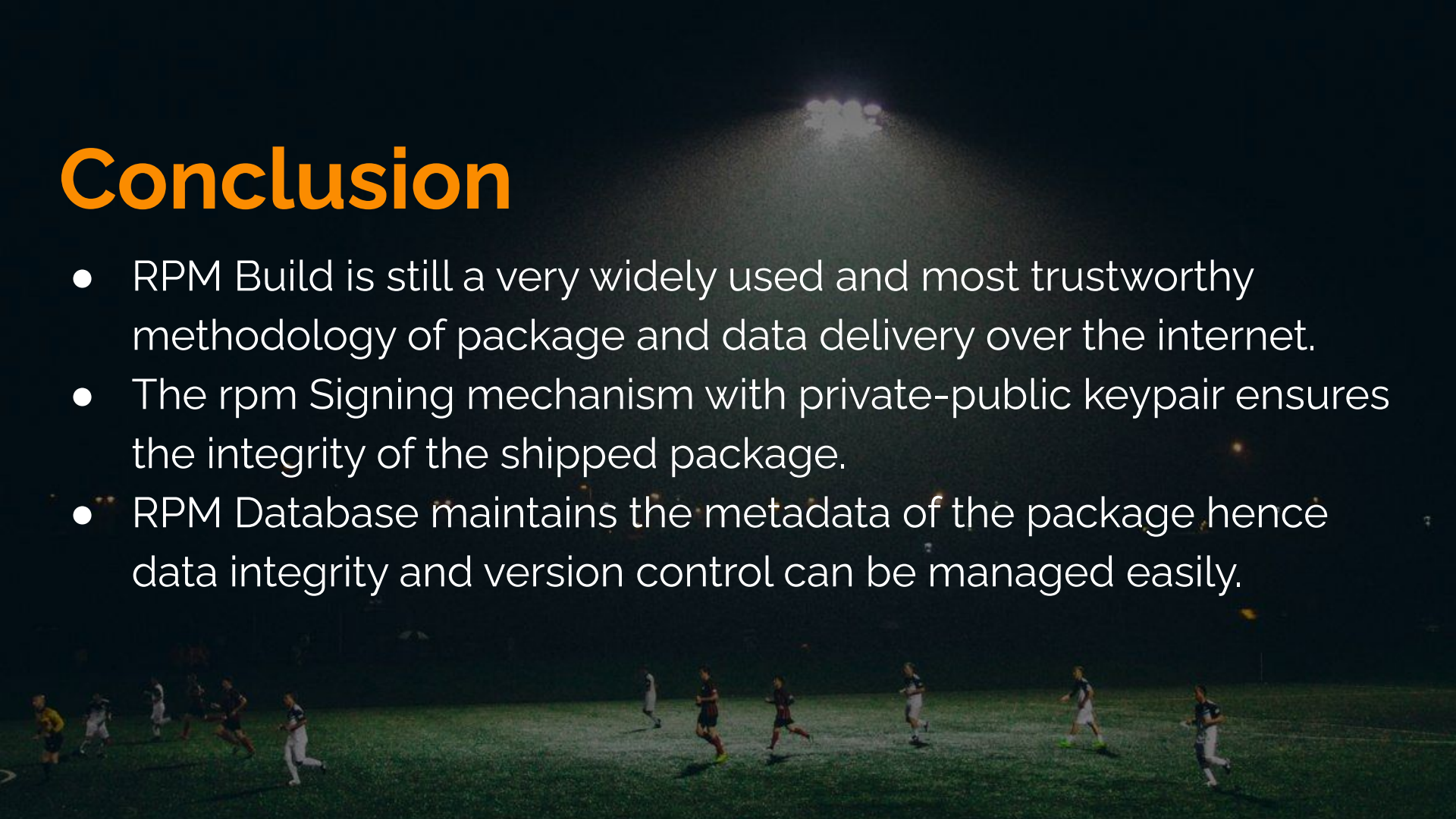
  https://github.com/minakshichavan/MTC674_Minor_Project

- To view the complete procedure, please visit the link above.



Conclusion

- RPM Build is still a very widely used and most trustworthy methodology of package and data delivery over the internet.
- The rpm Signing mechanism with private-public keypair ensures the integrity of the shipped package.
- RPM Database maintains the metadata of the package hence data integrity and version control can be managed easily.



Brainstorming time!