The following are questions that were asked during the session:

**Q**. What tools do you use for identifying packers?

**A**. A variety of tools can be leveraged for this activity to help building Yara rules. No single tool seems to hold the capability to identify all packers. The following, however, constitute a good start:

* Python-PEid - <https://github.com/libcrack/python-peid>
* Detect-it-Easy - <http://ntinfo.biz/> or <https://github.com/horsicq/Detect-It-Easy>
* RDG Packer Detector - <http://www.rdgsoft.net/>
* PackerID - <https://github.com/sooshie/packerid>
* Guidance Software Packer Detection- <https://www2.guidancesoftware.com/appcentral/pages/product.aspx?cat=GuidanceSoftware&pid=180010024WS&fromCart=1>

Good source references:

* <https://www.blackhat.com/docs/us-15/materials/us-15-Choi-API-Deobfuscator-Resolving-Obfuscated-API-Functions-In-Modern-Packers.pdf>
* <http://ftp.cs.wisc.edu/paradyn/papers/Roundy12Packers.pdf>

**Q**. Are there good tools or ways to organize and manage yara rules?

**A**. Managing Yara rules don’t have to be a difficult process. Yara Manager (<https://github.com/kevthehermit/YaraManager>) is a web-based manager for Yara rules that allows for easy organization.

Additionally, splitting rules up by category and classification and then leveraging the include directive to bring rules in by group as you need them and when you need them will go a long way towards organization. Metadata is crucial. Providing good metadata within the rule builds in the durability to understand the reason the rule was initially created, if it was updated and how you planned on employing it.

**Q**. What do you put in metadata?

**A**. It can vary, but at a minimum, its best to place some or all of the following items in the metadata section:

* Date created (or updated)
* Hash value – one or more entries that map to samples used to derive the rule
* Reference to the reason or basis for creation.
  + Website link, one or two sentence description
* Creator’s name
* Category/Classification (to help when importing by directive)