Yet Another Collection Kit (YACK)

User Guide

V0.1

8 OCT 2018

CPT Martin, Connor

Contents

[BLUF 3](#_Toc526796728)

[Overview 3](#_Toc526796729)

[Controlling Script 3](#_Toc526796730)

[Collectors 4](#_Toc526796731)

[Parsers 4](#_Toc526796732)

[Project Structure 4](#_Toc526796733)

[Using YACK 5](#_Toc526796734)

[Collect Running Processes on the Local Machine 5](#_Toc526796735)

[Run Collection on Multiple Remote Machine 5](#_Toc526796736)

[Enable WinRM 6](#_Toc526796737)

[Manually 6](#_Toc526796738)

[Group Policy 6](#_Toc526796739)

[Enable Over SMB 6](#_Toc526796740)

[Enable Over WMI 6](#_Toc526796741)

[Run Parser on Collected Result 6](#_Toc526796742)

[Developing Components 6](#_Toc526796743)

# BLUF

YACK is a PS2.0 complaint windows host collection script that can do local collection or multithreaded remote collection. Remote collections, and their data transfers, are done over PSSession (WinRM.)

# Overview

YACK’s purpose it to provide a standard method of collection and data storage format for windows host triage forensics. The goal of this will be to enable a common dataset that analytics can be developed against and thus shared. The Kansa project on GitHub is where a large portion of this project is derived from. YACK can be divided into 3 parts: the controlling script, collectors and parsers.

## Controlling Script

The controlling script is what manages both the collection of data from endpoints and the parsing of said data. It can run collection against the local host or parallel across multiple remote hosts. After collection it can the parsers against the collected data. Which collectors or parsers are ran is determined by their respective configuration files.

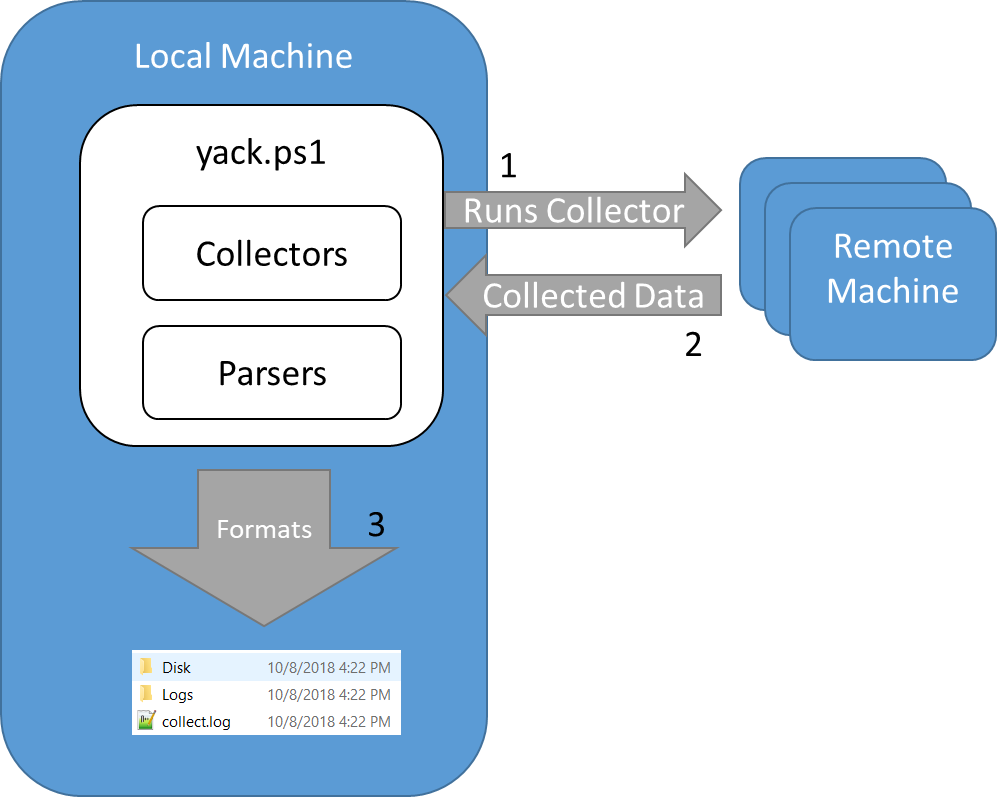


Figure 1: Overview of yack running collection.

## Collectors

Collectors are self-contained PowerShell scripts that are ran on the target machine to collect data of interest. The data is returned back to yack over the PSSession session. When data formatting is required, it is preferred that it’s formatted on the remote machine before being returned.

## Parsers

Parsers are self-contained PowerShell scripts that do additionally processing on the data that could not be done on the target systems. An example of this is if you collect a memory dump from and machine and want to parse out the running processes from memory.

## Project Structure

**Internals** - For configuring the running of the collection/script

**Config** - Folder for configs

Yack.conf - The configuration file for the controlling script

Collectors.conf - Selects which collectors are ran

Parsers.conf - Selects which parsers are ran

**Functions** - Folder for function used in yack.ps1

**yack.log** - The controlling script’s log

**Colllectors** – The directory which all collectors are stored

**\_Dependencies** - File/scripts that need to be transferred or included for the collectors to run.

**Apps** - Specific to applications

**Disk** - Information related to disk

**Exec\_Current** - Currently executing processes

**Exec\_Historical** - Evidence of execution

**Exec\_Triggers** - Items that trigger program execution

**Memory** - Memory collection or dumps

**Misc** - Anything else

**Network** - Data related to network

**System** - Data about this specific system

**User** - User behavior or activity

**Parsers –** The directory which all parsers are stored

“” - Same layout as collectors

**yack.ps1** - The controlling script

# Using YACK

The following section describes common use cases and describes how to configure yack for them.

## Collect Running Processes on the Local Machine

To collect running processes on the local machine you have to do the following:

1. Open \YACK\Internals\Config\Yack.conf
   1. CollectionMode=Local ~ this tells yack to run in local collection mode.
   2. RunCollectors=True ~ this enables collectors.
   3. RunParsers=False ~ this disables parsers.
   4. Save.
2. Open \YACK\Internals\Config\Collectors.conf
   1. Comment out all lines, if they are not already, with a # sign.
   2. Uncomment “\Exec\_Current\Get-Processes.ps1”
   3. Save
3. Open an elevated PowerShell prompt.
4. Navigate to the yack folder and run .\yack.ps1
5. Observe results inside of \YACK\Output\{timestamp]\{hostname}\Exec\_Current\Processes.csv

## Run Collection on Multiple Remote Machine

To collect running processes on a remote machine you have to do the following:

1. Enable WinRM on target machine. See Enabling WinRM.
2. Open \YACK\Internals\Config\Yack.conf
   1. CollectionMode=Remote ~ this tells yack to run in local collection mode.
   2. RunCollectors=True ~ this enables collectors.
   3. RunParsers=False ~ this disables parsers.
   4. AuthenticationMode=NTLM ~ this will prompt the user for the account
   5. Save.
3. Open \YACK\Internals\Config\Collectors.conf
   1. Comment out all lines, if they are not already, with a # sign.
   2. Uncomment “\Exec\_Current\Get-Processes.ps1”
   3. Save
4. Open \YACK\Internals\Config\Hostnames.conf
   1. Comment out all lines, if they are not already, with a # sign.
   2. Add the hostname or ip address of the machine you want to run collection on.
   3. Save

## Enable WinRM

PowerShell remoting must be enabled for YACK to work with remote collection. There are a couple different ways of doing this.

### Manually

1. Gain access to the target machine
2. Open an elevated powershell prompt
3. Run “Enable-PSRemoting -force”

### Group Policy

1. Open Group Policy Management Console
2. Navigate to Computer Policies | Administrative Templates | Windows Components | Windows Remote Management (RM) | WinRM Service.
3. Allow remote server management Server management through WinRM.

### Enable Over SMB

1. Enable win rm over smb

### Enable Over WMI

1. Enable winrm over wmi. Get-Baseline is a good example.

## Run Parser on Collected Result

Something sp,ething

# Developing YACK

Something something