Harness: Powershell Weaponization Made Easy (or at least easier)

Rich Kelley

@RGKelley5

What's this all about anyway?

- Audience:
  - Penetration testers
  - Red Teams
  - Powershell activists
  - Python enthusists

- Bottom line
  - Powershell weaponization can be somewhat cumbersome
  - Hopefully I've made that a little easier with the Harness tool set

Who is this guy?

- Computer science background
- Prior US Air Force Communications Officer
- Network engineer, software developer, penetration tester
- Currently focused on application pen testing
- Mostly I enjoy writing obscure utilities
  - pyhashcat
  - Keyboard walk generators

Why should I care?

- "…Microsoft's Post-Exploitation Language" - @obsuresec

- Defenders should be more aware of the damage attackers can do with Powershell alone

- We need more research into incident response related to malicious Powershell use
  - *DEF CON 22 - Ryan Kazanciyan and Matt Hastings, Investigating PowerShell Attacks*

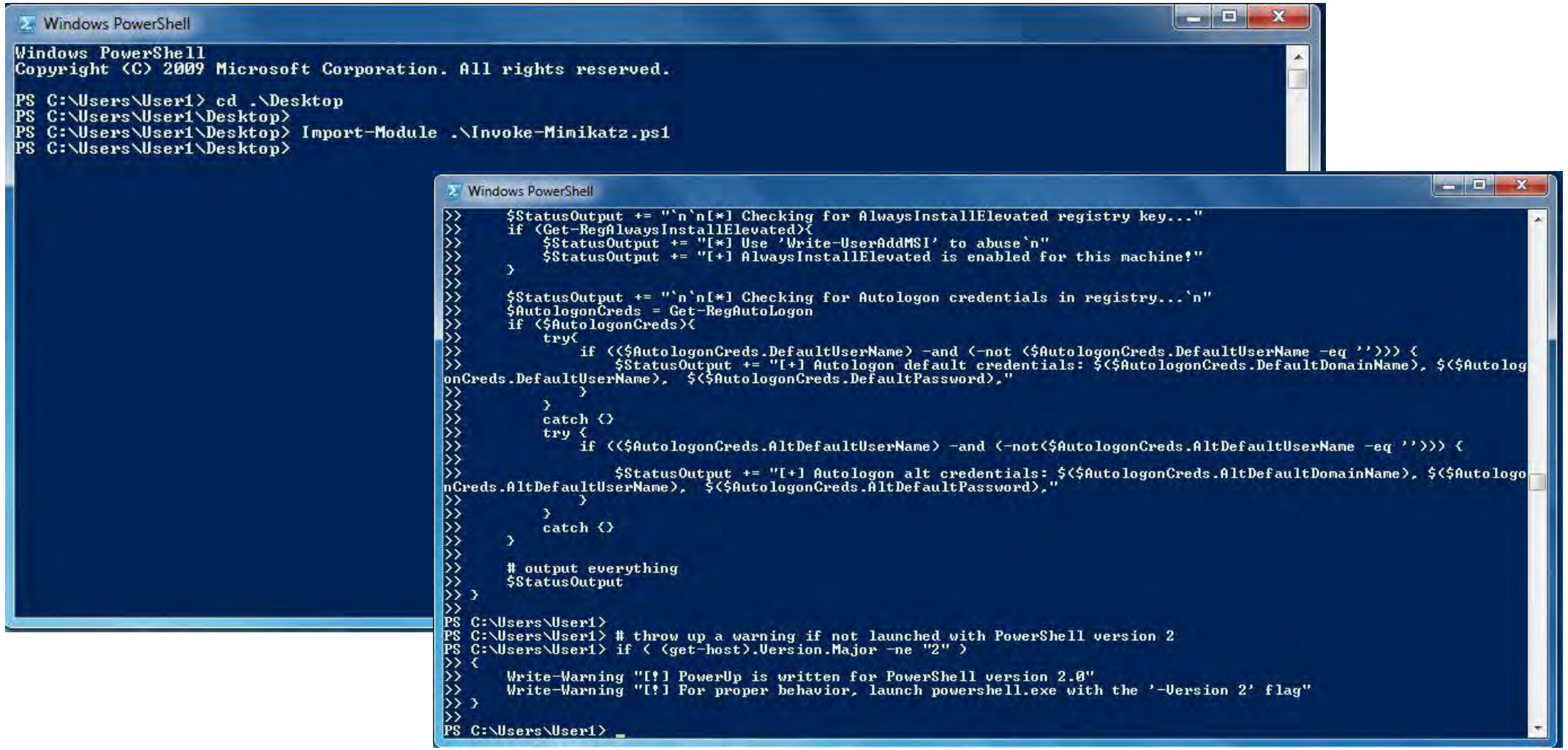
Powershell weaponization problem?

"How do you get your [Powershell] scripts running on your target machines, and effectively get your results back?" - @harmj0y

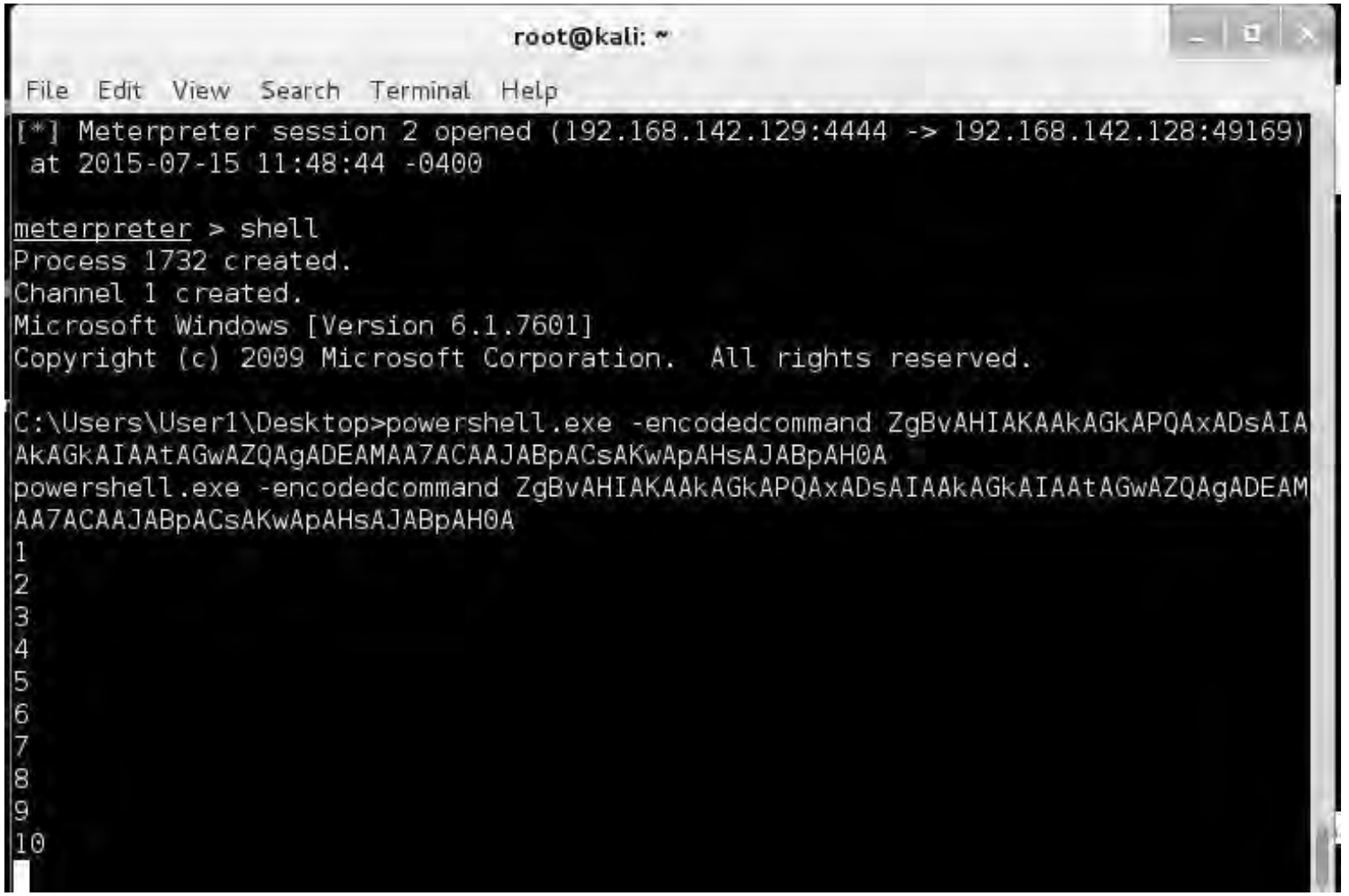
Hasn’t this problem been solved?

- Yep, but I’m a developer. Why use someone else’s solution when I can write my own (I’m kidding…sort of)
- Previous solutions were not as seamless as I wanted
  - Step 1: Gain access
  - Step 2: ?????
  - Step 3: Use powershell
  - Step 4: Pwn all things!
- A couple of very cool new solutions have recently been released

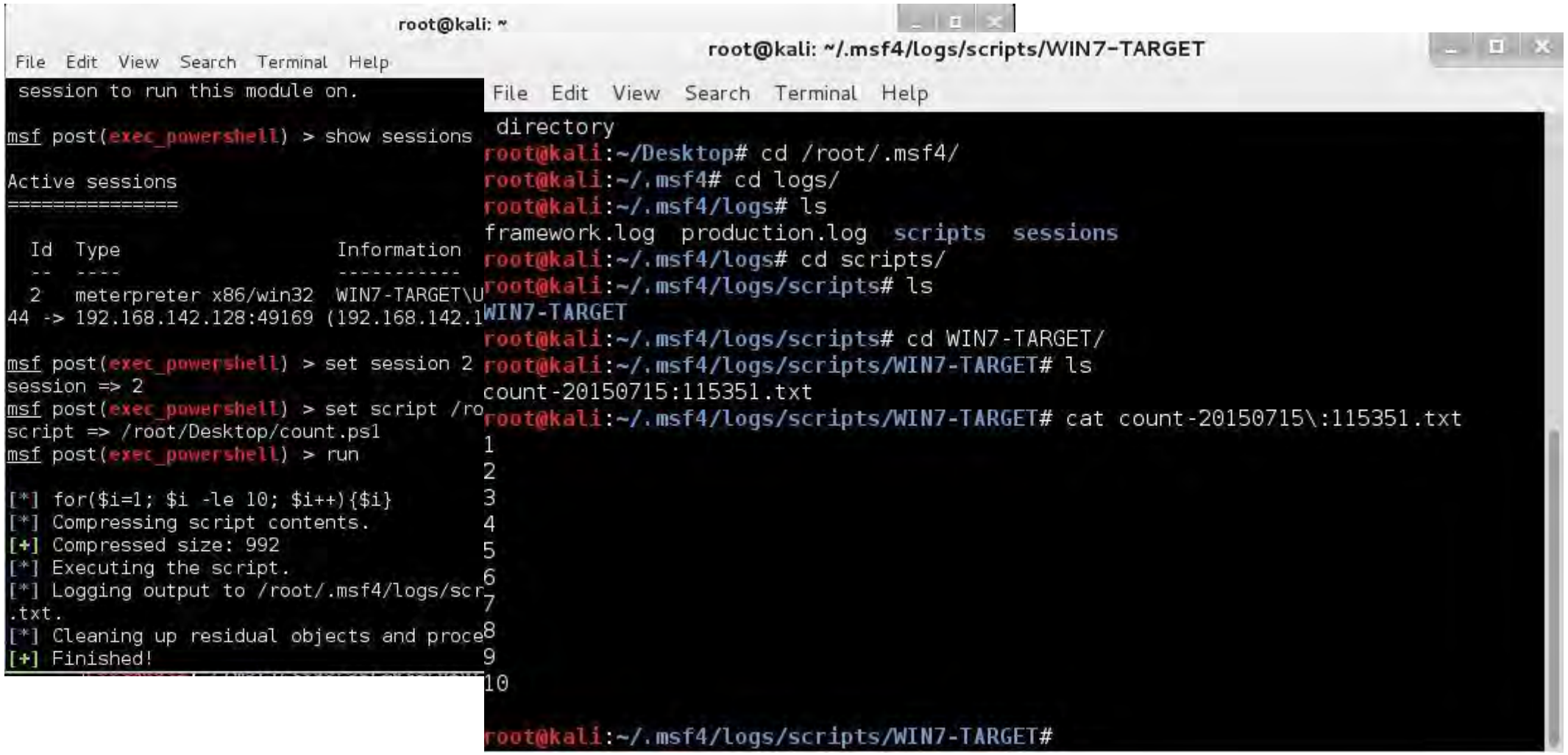
RDP – Copy/Paste or Import-Module



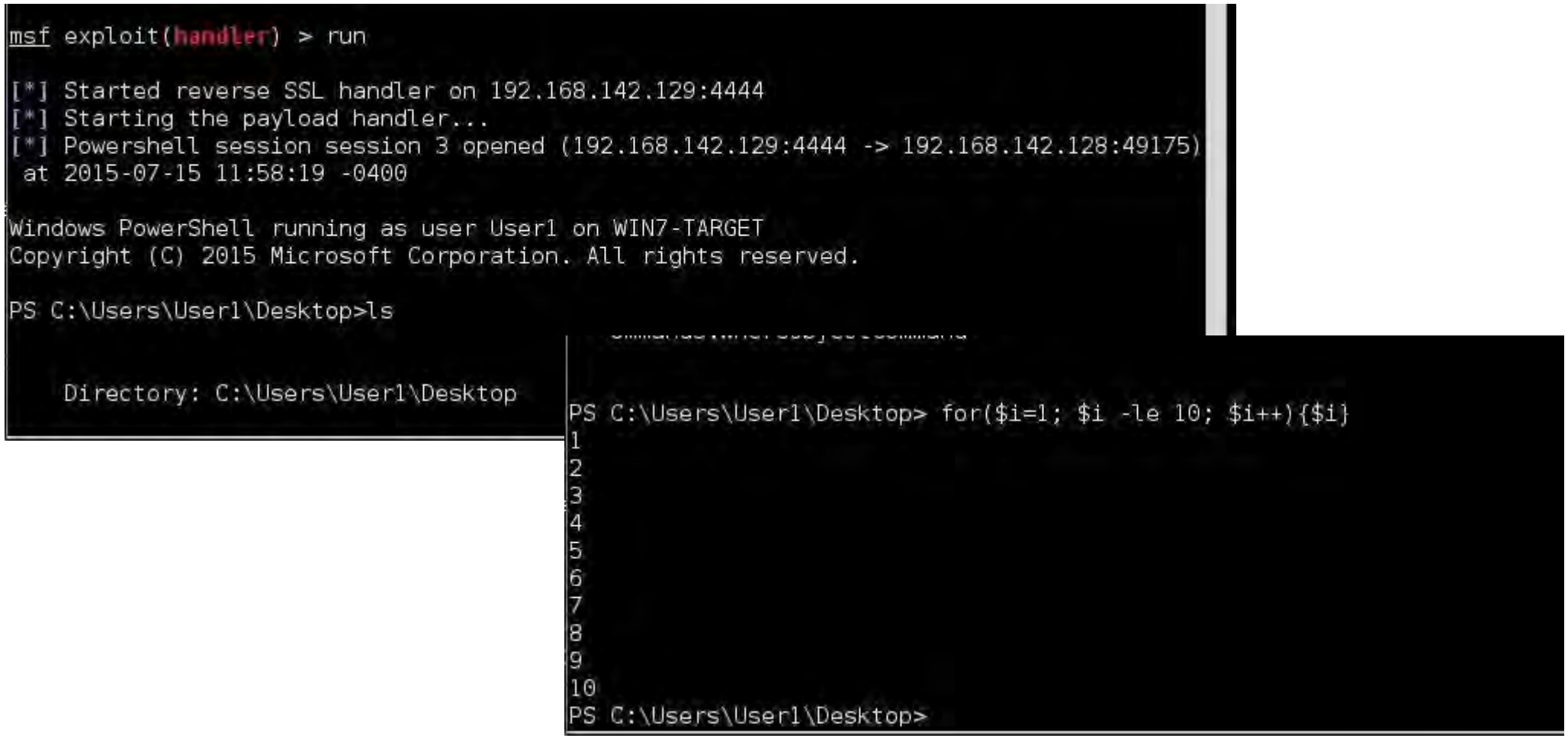
Remote shell – call powershell.exe



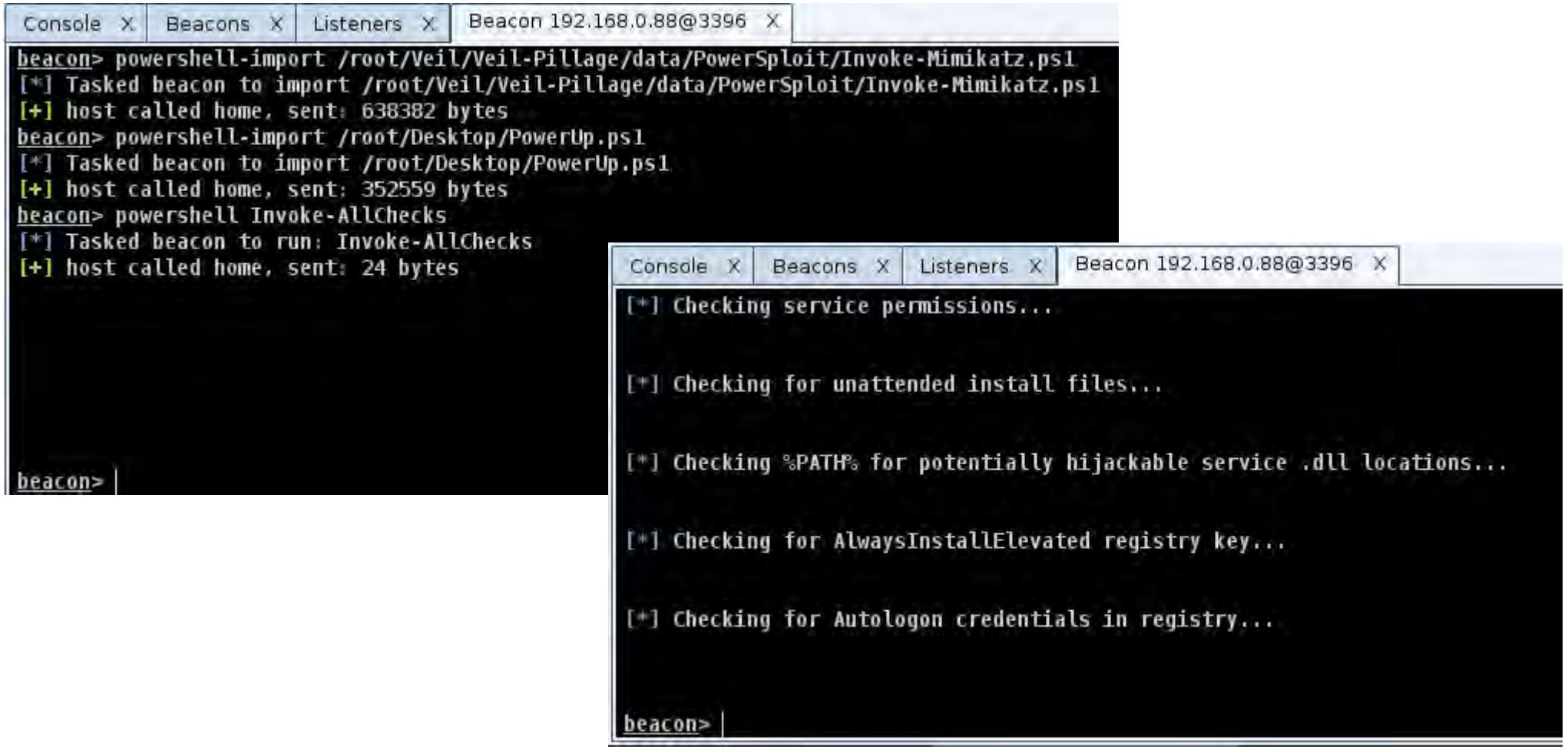
Metasploit – exec_powershell



Metasploit – Interactive PS Payloads



Cobalt Strike – Beacon



My Development Requirements

1. Fully interactive remote Powershell console with the same capabilities as the native Powershell.exe

2. Ability to seamlessly import modules across the wire

Demo Time!

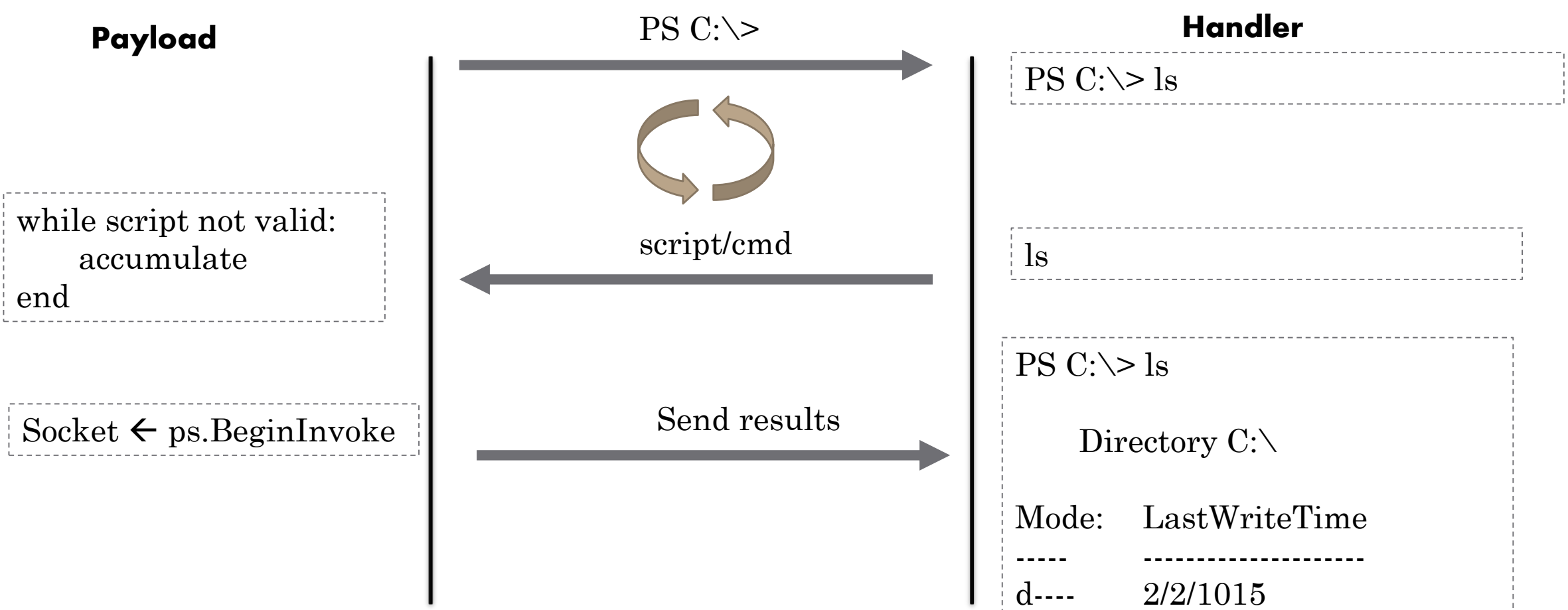
Under the hood

- Payload Requirements
  - .NET 3.0+
  - Powershell 2.0
  - System.Management.Automation Assembly
- Tested on:
  - Windows 7
  - Window 8
  - Windows 8.1
  - Windows Server 2008 R2
  - Windows Server 2012

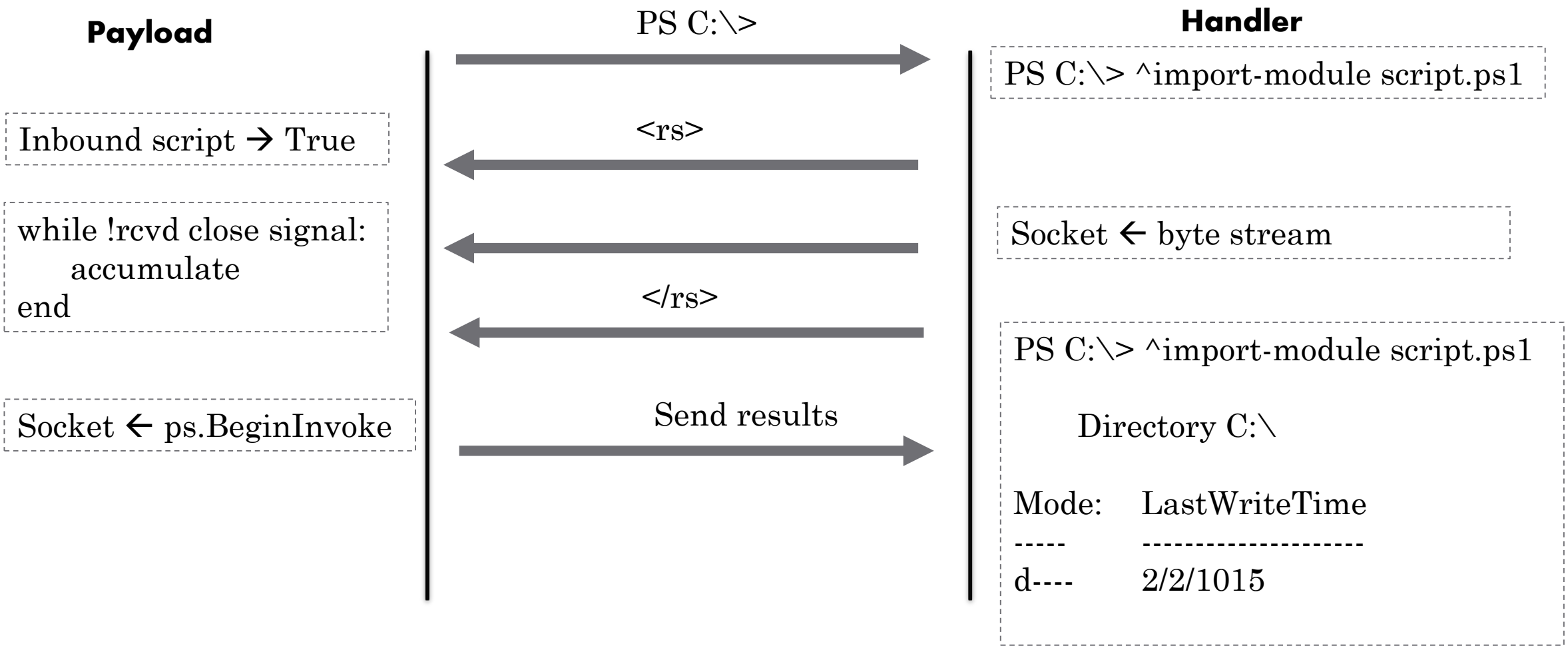
Under the hood

- Listener/Framework Requirements
  - Python 3.4
    - Asyncio
  - Linux
    - Tested on Kali
- Why Python? Why not Ruby? Why not Metasploit?
  - Mostly for the learning experience
  - I prefer Python to Ruby (calm down)
  - Should be simple enough to port to Metasploit module

Under the hood



Under the hood



Questions?