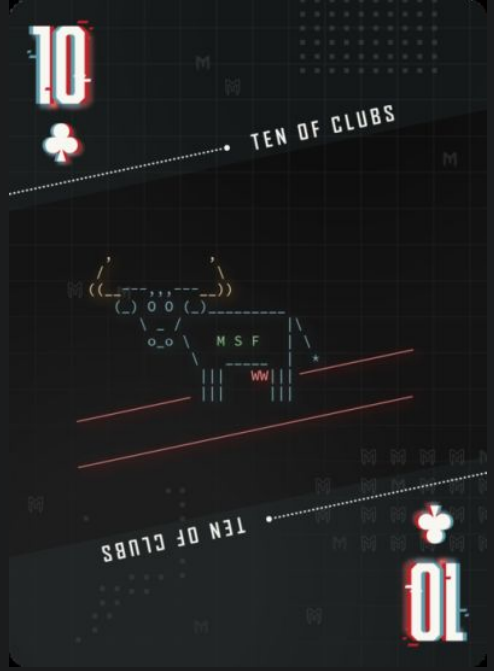


Resource Script for Automating Exploits

Automating Exploits from Project 2 Through Metasploit

Presented by Room 2: Alula Tesfay, Brandon Schiller, Brian Hensien, Faven Getachew, Jason Lanctot, Shamil Ranasinghe, Sharmarke Omar, Travel Clemon, Vincent Musyoka, and Tianna Nguyen

Resource Scripts in Metasploit

Definition

A resource script usually refers to a type of script commonly used in software development for creating, managing, or loading various resources for an application. Resource scripts within Metasploit can be used to ease the process of setting options for multiple exploits as well as execute several exploits at once.

Benefits

Ethical hackers can use scripts for efficient testing.

Attackers can automate malicious activities.

Controlled environments are crucial.

Ruby

Ruby is a high-level programming language that can be used within resource scripts. It directly utilizes Metasploit's API to interact with framework modules, exploit settings, payloads etc. Ruby code used within a resource script can streamline exploitation and can execute commands within meterpreter as well as the target system's shell.

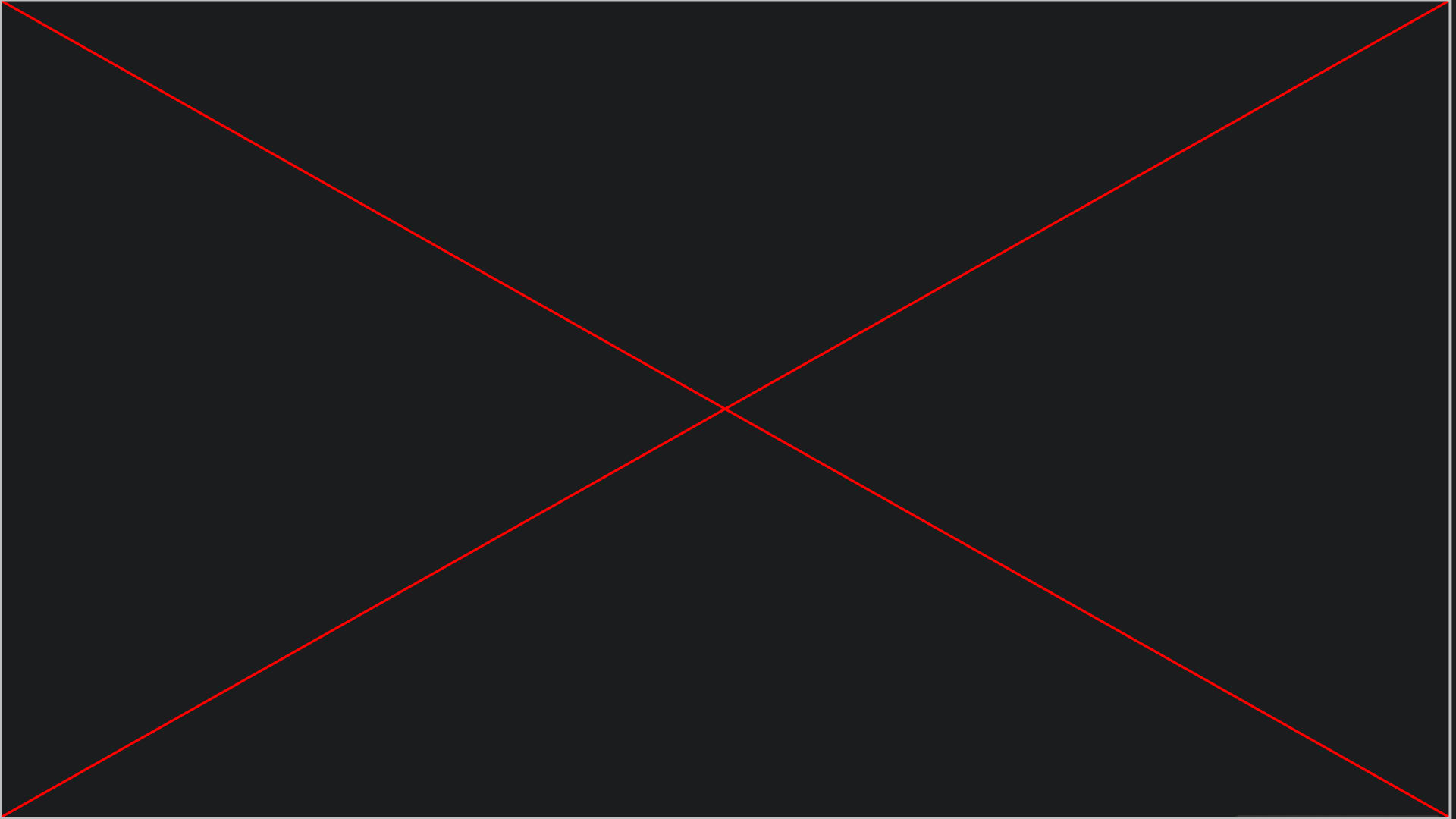
Since Ruby interacts with Metasploit's API directly, the use of loops, conditionals and error handling is made available. An example for the use of a loop would be to design the script to loop through a list of IPs and run one or more exploits on each IP. An example for error handling would be to catch exceptions and modify the flow of the script based on the output or success of a command (like using an "if" statement).

For the purposes of this presentation, we will only be using Linux commands.

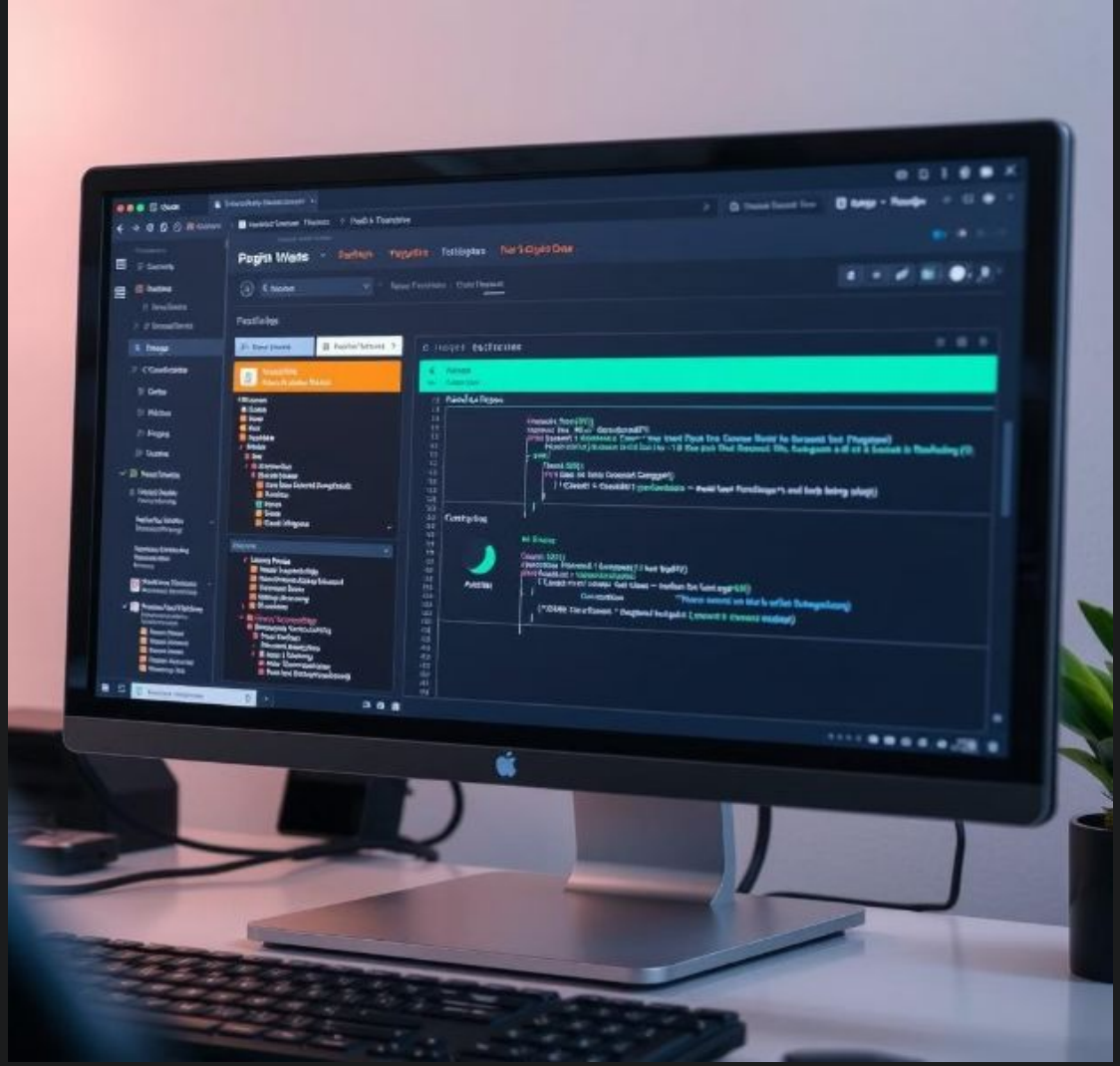
How we used it in Metasploit

We designed a resource script with the goal of automating five separate exploits (from Project 2) while ensuring each runs in its own session in parallel execution. This was achieved by setting the options of each exploit followed by the “sleep” command as well as the “exploit -j” command. “session -i” at the tail end of the script will show us the list of active sessions. After the successful execution of the script, we will go into each session to find the flag pertaining to that exploit.

The sleep command forces the script to hang for a set limit of time. This was used in order to allow enough time for the successful execution of each exploit. The “exploit” command used with the option “-j” executes each exploit and backgrounds the session.



Other Use Cases



Interactive Demonstrations and Training

- Use Case:
  - Creating a "showcase" demonstration of a successful attack, where you can pause to explain each step.
  - Highlighting specific payloads, exploits, or tactics in a controlled environment.

- Example:
  - Resource scripts can be used for training sessions, creating pre-scripted attacks or payload delivery demonstrations. Pausing the script with sleep or gets allows the user to follow along or observe the process step-by-step.

Simulating Attack Scenarios (Red Teaming)

Use Case:

- Running a series of attacks that mimic an advanced persistent threat (APT).
- Automating lateral movement across different systems after an initial compromise.

Example:

- For red team operations, resource scripts can be used to simulate specific attack scenarios, automate lateral movement, and exploit multiple vulnerabilities in an environment to mimic the actions of a real adversary.

Credential Reuse and Privilege Escalation

Use Case:

- After gaining a foothold, a resource script can attempt different privilege escalation techniques or use different exploits to escalate privileges.
- Automating the process of reusing credentials on other services (e.g., SMB, RDP, SSH).

Example:

- If you are conducting a pentest and successfully gain access with low-level user credentials, you might want to automate the process of attempting privilege escalation or reusing credentials across different services.

Scenario

How it could be used by Attacker?

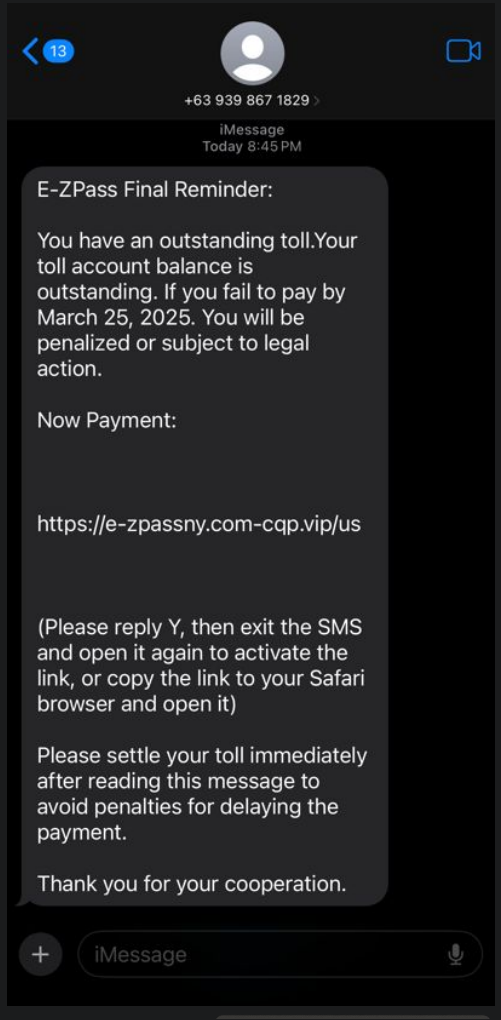
Picture this...
You're a hardworking employee—early mornings, skipped lunches, and still behind on deadlines.

You're exhausted.
Your phone buzzes with a message.

You think, *"Finally, a quick break."*
You glance around—no boss in sight—and open the message.

It says you've missed a toll payment. You panic.
You click the link without thinking.

In that moment...
You've just given an attacker access to your company's network.



Common goal: Steal credentials, money, or sensitive data.

How it could be used by Attacker?

The attacker now has access to your company's Wi-Fi.

They run an **Nmap scan**, find outdated services, and check **Metasploit** for known exploits.

With a **resource script ready**, they wait for the office to clear.

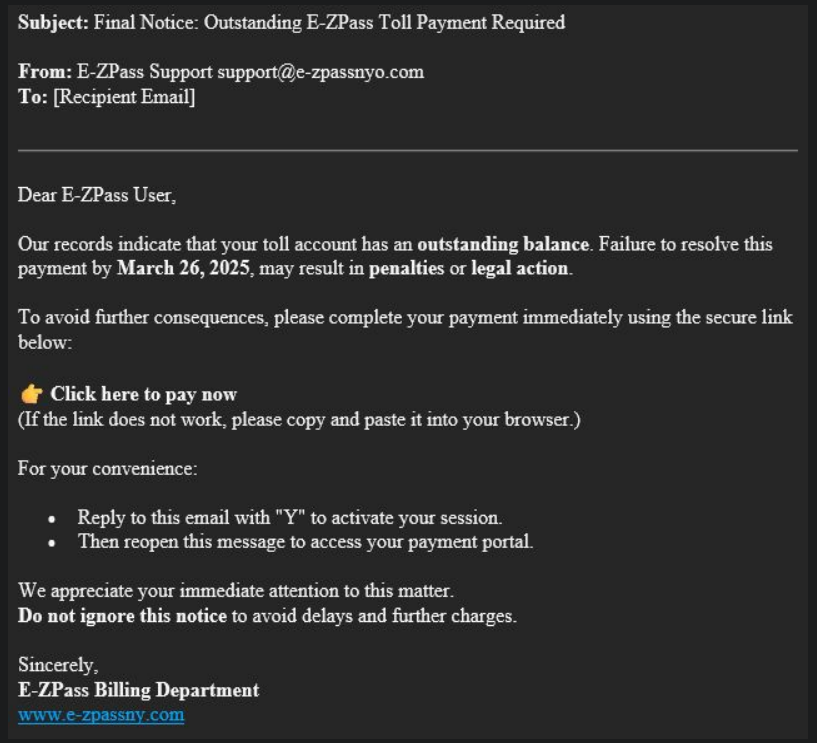
Then—with **one command**—your network is compromised.

🚨 The Reality

Anyone can be a target.
Any connected device can be a doorway.
With tools like Metasploit, attacks can happen in **seconds**.

Stay alert. Stay secure.

Common goal: Steal credentials, money, or sensitive data.



Subject: Final Notice: Outstanding E-ZPass Toll Payment Required

From: E-ZPass Support support@e-zpassnyo.com
To: [Recipient Email]

Dear E-ZPass User,

Our records indicate that your toll account has an **outstanding balance**. Failure to resolve this payment by **March 26, 2025**, may result in **penalties** or **legal action**.

To avoid further consequences, please complete your payment immediately using the secure link below:

👉 **Click here to pay now**
(If the link does not work, please copy and paste it into your browser.)

For your convenience:

- Reply to this email with "Y" to activate your session.
- Then reopen this message to access your payment portal.

We appreciate your immediate attention to this matter.
Do not ignore this notice to avoid delays and further charges.

Sincerely,
E-ZPass Billing Department
www.e-zpassny.com

Key Takeaways and Next Steps

Automation Benefits

Resource scripts enhance efficiency and consistency in vulnerability assessments.

Ethical Use

Use scripts responsibly for ethical hacking and security testing.

Further Exploration

Explore additional Metasploit modules and scripting techniques for advanced automation.

Remediation

Make sure all devices and software are up to date and permissions are set to only necessary personnel.