CRTE Modular Lab - User Guide

Overview

The CRTE Modular Lab is a scenario-based implementation of the CRTE lab environment that integrates with the GOAD framework. This modular approach allows you to deploy only the VMs needed for specific attack scenarios, making the lab practical to run on systems with limited RAM.

Key Features

- **Scenario-Based Deployment**: Deploy only the VMs needed for specific attack scenarios
- **Resource Efficiency**: Minimize RAM usage by deploying only necessary components
- **GOAD Integration**: Seamlessly integrates with the GOAD framework
- **Comprehensive Coverage**: Includes all attack techniques required for CRTE certification

Installation

1. **Copy the CRTE-Modular directory to your GOAD installation**: `bash cp -r CRTE-Modular /path/to/GOAD/ad/`

2. **Add the CRTE-Modular integration to GOAD**: `bash # Add the following line to your GOAD/goad.sh file before the line that calls goad.py source /path/to/GOAD/ad/CRTE-Modular/scripts/goad_integration.sh`

Usage

Starting GOAD with CRTE Lab

```
./goad.sh -p vmware -l CRTE
```

Selecting a Scenario

At the GOAD prompt, use the `set_scenario` command to select a specific attack scenario:

```
CRTE/vmware/local/192.168.56.X > set_scenario kerberoasting
```

To see a list of available scenarios:

```
CRTE/vmware/local/192.168.56.X > list_scenarios
```

Installing the Selected Scenario

After selecting a scenario, install it using:

```
CRTE/vmware/local/192.168.56.X > install
```

This will deploy only the VMs required for the selected scenario.

Available Scenarios

The CRTE Modular Lab includes the following attack scenarios:

1. **Kerberoasting** (8GB RAM)
2. Practice extracting and cracking service account hashes
3. **AS-REP Roasting** (8GB RAM)
4. Exploit accounts with "Do not require Kerberos preauthentication"
5. **Unconstrained Delegation** (10GB RAM)
6. Exploit servers with unconstrained delegation enabled
7. **Constrained Delegation** (10GB RAM)
8. Exploit servers with constrained delegation
9. **Resource-Based Constrained Delegation** (10GB RAM)
10. Exploit RBCD misconfigurations
11. **ACL Abuse** (10GB RAM)

12. Exploit misconfigured access control lists

13. **Domain Trust Attacks** (8GB RAM)

14. Exploit parent-child domain trust relationships

15. **Forest Trust Attacks** (14GB RAM)

16. Exploit forest trust relationships

17. **SQL Server Link Attacks** (14GB RAM)

18. Exploit linked SQL servers for lateral movement

19. **Exchange Server Attacks** (12GB RAM)

    - Exploit Exchange server vulnerabilities

20. **Azure AD Connect Attacks** (10GB RAM)

    - Exploit Azure AD Connect for credential extraction

21. **Local Privilege Escalation** (8GB RAM)

    - Practice local privilege escalation techniques

Validation

The CRTE Modular Lab includes a validation script that verifies the functionality of the scenario selector and configuration generation:

```
cd /path/to/GOAD/ad/CRTE-Modular/scripts
python3 validate.py
```

Troubleshooting

Common Issues

1. **Scenario selection fails**:
2. Ensure the CRTE-Modular directory is correctly placed in your GOAD installation

3. Check that the scenario ID is correct (use `list_scenarios` to see available scenarios)

4. **VM deployment fails**:

5. Verify that you have sufficient system resources for the selected scenario

6. Check the Vagrant and VirtualBox/VMware logs for specific errors

7. **GOAD integration issues**:

8. Ensure the integration script is correctly sourced in your goad.sh file
9. Check that the paths in the integration script match your installation

Support

For issues or questions about the CRTE Modular Lab, please refer to the documentation or contact the maintainer.

Credits

The CRTE Modular Lab is built on top of the GOAD (Game of Active Directory) framework, with modifications to support scenario-based deployment for CRTE certification practice.