

Forum:[Hack Forums](#)

Thread: [./Dark Legion/Bloodhound: Unleashing the Power of Graph Theory](#)

Dread.Reaper - March 31, 2023, 10:26 am

@Genuine@

Bloodhound: Unleashing the Power of Graph Theory for Maximum Network Domination

Written By: Dread.Reaper – Dark Legion

Introduction

Bloodhound is an exceptional tool that provides us hackers with a revolutionary approach to exploit networks. By leveraging graph theory, it is able to map out intricate Active Directory (AD) infrastructures and identify the perfect attack paths and vulnerabilities that we could leverage to gain access. It has turned out to be a valuable asset for our unethical endeavors. However, we must always maintain our caution and use the tool within ethical boundaries.

The tool itself has an exceptional interface which allows us to visualize the complete AD graph with a number of options to filter out unwanted information. As hackers, we can use this information to easily identify privileged accounts and escalate our attack vectors.

Installation and configuration of the tool on Windows-based systems is simple and straightforward. By following the necessary steps, we can quickly start using the tool to search for vulnerabilities. Once installed, we can use SharpHound, the data collection tool, to gather AD object information and their properties in a stealthy manner, thereby keeping our identity hidden. The tool also supports various query languages, which allow us to simulate various attack scenarios to identify high-value targets and potential privilege escalation paths.

To further assist us hackers, Bloodhound provides us with a list of commonly used commands that can help us to search for sensitive files in domain shares and retrieve information about AD objects. These commands include Get-DomainComputer, Get-DomainUser, Get-DomainGroup, Find-InterestingDomainShareFile, and Invoke-SharpHound. We can use these commands to gather information that can be used to identify vulnerabilities in the network and exploit them to gain access.

By using Bloodhound, we can identify potential attack paths and vulnerabilities in target networks, making it an indispensable tool in our arsenal. However, we must always use it responsibly and ethically, as it is not intended to be used for harm or malicious activities. We must take care to only use it within the legal and ethical boundaries.

This technical manual will cover the following topics:

- Explanation of Bloodhound's functionality and application
- Installation and configuration process for Bloodhound on Windows-based systems
- Use of various commands with examples to retrieve information about AD objects and search for sensitive files in domain shares
- A conclusion that summarizes the benefits of using Bloodhound and the importance of ethical use.
- Bloodhound's Functionality and Application
- Bloodhound uses graph theory to visualize and analyze relationships between AD objects, including users, groups, and computers. It includes a data collection tool called SharpHound that can collect information about AD objects and their properties in a stealthy manner. The tool also supports various query languages and can simulate various attack scenarios to identify high-value targets and potential privilege escalation paths.

Installation and Configuration

To use Bloodhound on Windows-based systems, users need to follow the following steps:

Install the required software dependencies:

- PowerShell version 3.0 or later
- .NET Framework version 4.7.2 or later

Download and extract the Bloodhound tool from its official GitHub page.

<https://github.com/BloodHoundAD/BloodHound>

Configure the tool settings using PowerShell by running the following commands:

- Set-ExecutionPolicy Unrestricted -Force
- cd <Bloodhound_directory>
- .\Start-Bloodhound.ps1

Once the settings are configured, use the SharpHound tool to collect data about AD objects in the target domain by running the following command:

- Invoke-SharpHound -CollectionMethod <method> -SearchForest -LDAPFilter <filter>

-OutputFileName <filename>.json

This command runs the SharpHound data collection tool and generates a JSON report file that contains the collected data.

Import the collected data into the Bloodhound GUI tool to visualize and analyze the AD graph.

Commands and Examples

Here are some of the most commonly used commands with examples:

Get-DomainComputer

This command retrieves a list of all computers in the target domain.

Syntax:

```
Get-DomainComputer [[-Domain] <String>] [-DomainController <String>] [-LDAPFilter <String>]
[-Properties <String[]>] [-SearchBase <String>] [-Server <String>] [-Credential <PSCredential>]
[-UseSSL] [<CommonParameters>]
```

Example:

```
Get-DomainComputer -Domain example.com -LDAPFilter "(operatingSystem=*Server*)"
-Properties Name,OperatingSystem
```

This command retrieves a list of all computers in the example.com domain that have "Server" in their operating system name and returns their name and operating system properties.

Get-DomainUser

This command retrieves a list of all user accounts in the target domain.

Syntax:

```
Get-DomainUser [[-Domain] <String>] [-DomainController <String>] [-LDAPFilter <String>]
[-Properties <String[]>] [-SearchBase <String>] [-Server <String>] [-Credential <PSCredential>]
[-UseSSL] [<CommonParameters>]
```

Example:

```
Get-DomainUser -Domain example.com -LDAPFilter "(memberOf=CN=Domain
Admins,CN=Users,DC=example,DC)"
```

This command retrieves a list of all user accounts in the example.com domain that are members of the "Domain Admins" group and returns their name, SamAccountName, and description properties.

Get-DomainGroup

This command retrieves a list of all group accounts in the target domain.

Syntax:

```
Get-DomainGroup [[-Domain] <String>] [-DomainController <String>] [-LDAPFilter <String>]
[-Properties <String[]>] [-SearchBase <String>] [-Server <String>] [-Credential <PSCredential>]
[-UseSSL] [<CommonParameters>]
```

Example:

```
Get-DomainGroup -Domain example.com -LDAPFilter
```

"(groupType:1.2.840.113556.1.4.804:=2147483648)" -Properties Name,Description,Member
This command retrieves a list of all domain local groups in the example.com domain and returns their name, description, and members properties.

Find-InterestingDomainShareFile

This command searches for sensitive files in domain shares that can be accessed by the current user.

Syntax:

```
Find-InterestingDomainShareFile [-Path <String[]>] [-ExcludePattern <String[]>] [-IncludePattern <String[]>] [-User <String>] [-Computer <String>] [-MaxFileSize <Int64>] [-Recurse] [-ErrorAction <ActionPreference>] [-ErrorVariable <String>] [<CommonParameters>]
```

Example:

```
Find-InterestingDomainShareFile -Path "\\example.com\finance" -IncludePattern "*.xlsx" -User "exampleuser" -Computer "COMPUTER01" -MaxFileSize 10MB -Recurse
```

This command searches the \example.com\finance domain share for all Excel files (.xlsx) that can be accessed by the "exampleuser" account on the "COMPUTER01" computer, with a maximum file size of 10 megabytes (MB), and recursively searches all subdirectories.

Invoke-SharpHound

This command runs the SharpHound data collection tool and generates a JSON report.

Syntax:

```
Invoke-SharpHound [-CollectionMethod] <String> [-SearchForest] [-LDAPFilter <String>] [-OutputFileName <String>] [-ZipFileName <String>] [-CollectionName <String>] [-SkipExfiltration] [-ForceCollection] [-HoundSPN] [-Stealth] [-Verbose] [-Debug] [-ErrorAction <ActionPreference>] [-ErrorVariable <String>] [<CommonParameters>]
```

Example:

```
Invoke-SharpHound -CollectionMethod "Default" -SearchForest -LDAPFilter "(userAccountControl:1.2.840.113556.1.4.803:=512)" -OutputFileName "BloodhoundData.json" -ZipFileName "BloodhoundData.zip" -CollectionName "MyCollection" -SkipExfiltration -Verbose
```

This command runs the SharpHound data collection tool using the "Default" collection method to search the entire forest for all enabled user accounts and generates a JSON report file named "BloodhoundData.json" and a ZIP file named "BloodhoundData.zip". The data collection is named "MyCollection" and the exfiltration phase is skipped. Verbose output is enabled.

Conclusion

Bloodhound is a powerful tool that can help attackers and hackers identify potential attack paths and vulnerabilities in target networks. However, it is important to use the tool responsibly and only for ethical purposes. This technical manual provided an overview of Bloodhound's functionality and application, as well as a step-by-step guide for its installation and configuration on Windows-based systems. It also included various commands with examples to retrieve information about AD objects and search for sensitive files in domain shares. The commands covered were Get-DomainComputer, Get-DomainUser, Get-DomainGroup, Find-InterestingDomainShareFile, and Invoke-SharpHound.

By using Bloodhound, security professionals can identify vulnerabilities in their networks and take steps to mitigate them before attackers can exploit them. It is important to note that Bloodhound is a tool for defensive purposes and should not be used to harm any networks or individuals.

In summary, Bloodhound is an extremely useful tool for identifying potential attack paths and vulnerabilities in AD infrastructures. By using the tool responsibly and ethically, security professionals can better protect their networks and prevent cyber attacks.