

自分専用のピボット トラボを構築する

内部ネットワーク探索の出発点



\$ whoami

Fran Canteli

オフセック責任者 @ Hackmetrix

OSCP, CEHv11, eJPT

国際講演者

Hacker Rank @ Hack The Box



アジェンダ

導入

方法論

ピボットツール

ツールの比較

結論

導入

ピボットとは何ですか？

ピボットとは、侵入テストにおいて、直接アクセスできないシステムやネットワークへのアクセスを取得するために使用される手法です。あるマシンが侵入されると、そこを起点として他のマシンや内部ネットワークにアクセスできるようになります。このプロセスは、最終ターゲットに到達するまで再帰的に繰り返すことができます。

方法論

方法論

01. 最初のマシンの妥協(ピボットポイント)

02. 侵入されたマシンの偵察

03. 権限昇格(必要な場合)

04. ネットワークインターフェースを識別する

05. 設置されている便利なツールを特定する
土地の暮らし(LOTL)

06. ピンスweep

07. 偵察で収集した情報に基づいてポート転送を作成する

ピボットシナリオ



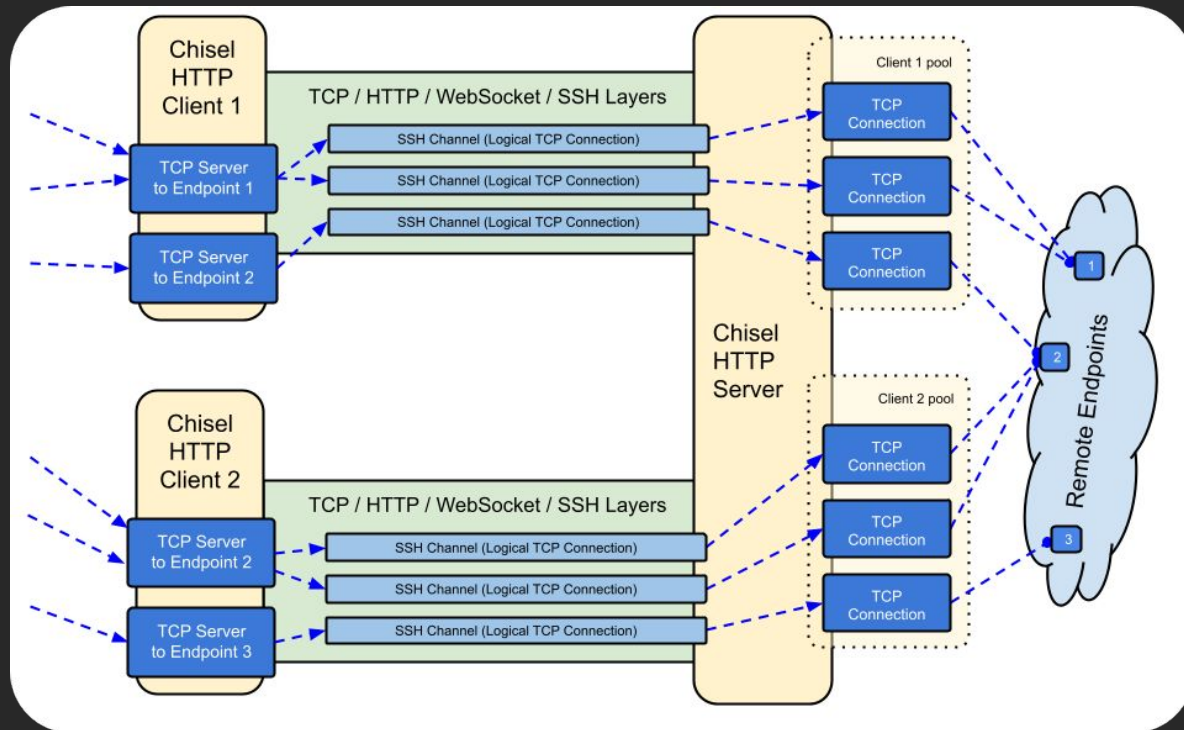
ピボットツール

Chisel

- HTTP 経由で転送され、SSH 経由で保護される TCP/UDP トンネル。
- これはクライアントとサーバーの両方を含む単一の実行可能ファイルです。
- Go (golang) で書かれています。
- ファイアウォールをバイパスするのに役立ちますが、ネットワークへの安全な接続ポイントを提供するためにも使用できます。

ソース: <https://github.com/jpillora/chisel>

Chisel



Chisel Server

Chisel Server: クライアントからの接続を受信できるようになります。

```
$ chisel server -p <port>
```

Chisel Client

Chisel Client: Chiselサーバーに接続できるようになります。クライアント側では、作成したいポートフォワーディングまたはSocksプロキシの種類を指定します。

\$ chisel client <server-ip>:<port> [remote]s

- remote/s: これらはサーバーを経由するトンネル接続であり、次の形式を持ちます。 :
[<local-host>:<local-port>:<remote-host>:<remote-port>/<protocol>]

Chisel Reverse Mode

- Chisel サーバーに接続するクライアントがリバース トンネルを定義できるようにします。
- 接続するクライアントは、サーバーが実行されているマシン上のリスニング ポートを開くことができます。
- リモートには、リバース トンネルであることを示すために、プレフィックスとして R を付けることができます。
- サーバーは接続をリスンして受け入れ、これらの接続はリモートを指定したクライアントを通じてプロキシされます。

```
$ chisel server -p <port> --reverse
```

```
$ chisel client <server-ip>:<port> R:[remote]
```

ポート転送

- 攻撃者が侵入したマシンを介して内部ネットワークにアクセスできるようにする
 - ◆ ローカルポート転送
 - ◆ リモートポート転送
 - ◆ ダイナミックポートフォワーディング

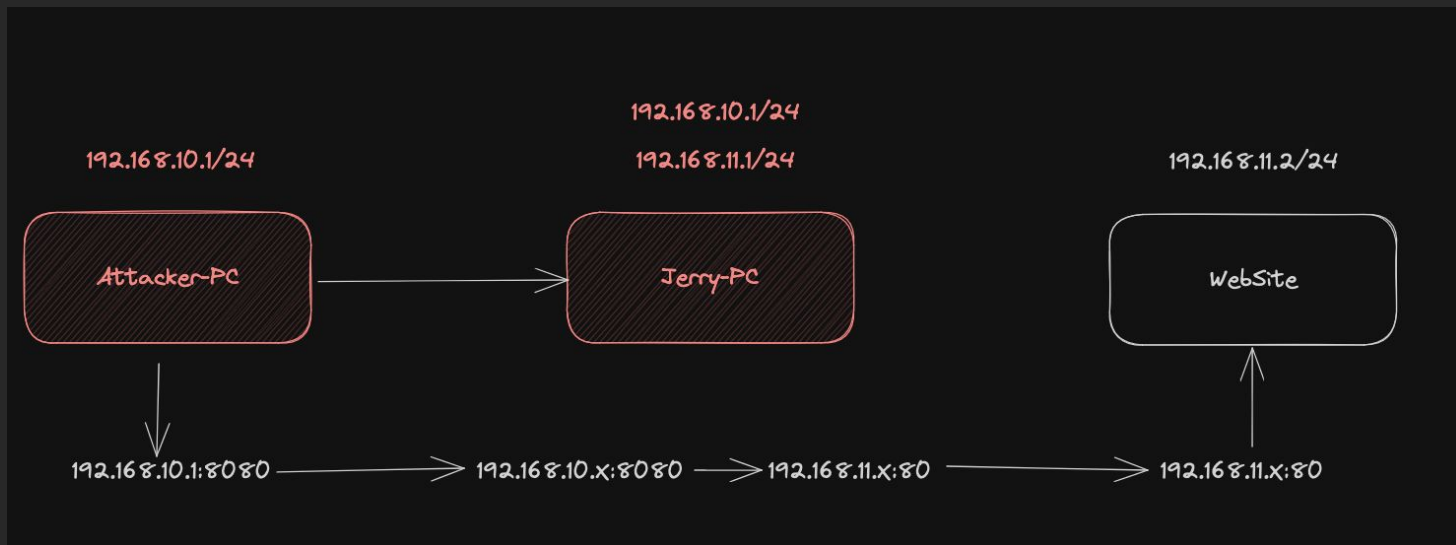
ローカルポート転送

ローカルポートフォワーディングは、攻撃マシンから侵入マシンへ、そして標的マシンへとポートをリダイレクトするために使用されます。リッスンされているポートは侵入マシン上にあります。このポートは標的マシン(同じくリッスンしている)へリダイレクトされます。

ローカルポートフォワーディングの例として、社内ウェブサイトアクセスする場合などが挙げられます。

ローカルポート転送 - シナリオ A

「Jerry-PC のポート 8080 で受信したすべての接続をポート 80 の Web サイトにリダイレクトします。」



ローカルポート転送 - デモChisel

The screenshot shows a web browser displaying a presentation slide. The slide title is "Local Port Forwarding - Scenario A". Below the title, it says "Redirect all connections received on Jerry-PC at port 8080 to Website on port 80." The diagram illustrates the following setup:

- Attacker-PC** (IP: $M2.16\ \$.10.1/24$) is connected to **Jerry-PC** (IP: $M2.16\ \$.10.1/24$).
- Jerry-PC** is connected to **WebSite** (IP: $M2.16\ \$.11.2/24$).
- Below the network diagram, a sequence of IP addresses is shown: $M2.16\ \$.10.1/80 \rightarrow M2.16\ \$.10.1/80 \rightarrow M2.16\ \$.11.1/80 \rightarrow M2.16\ \$.11.1/80$.

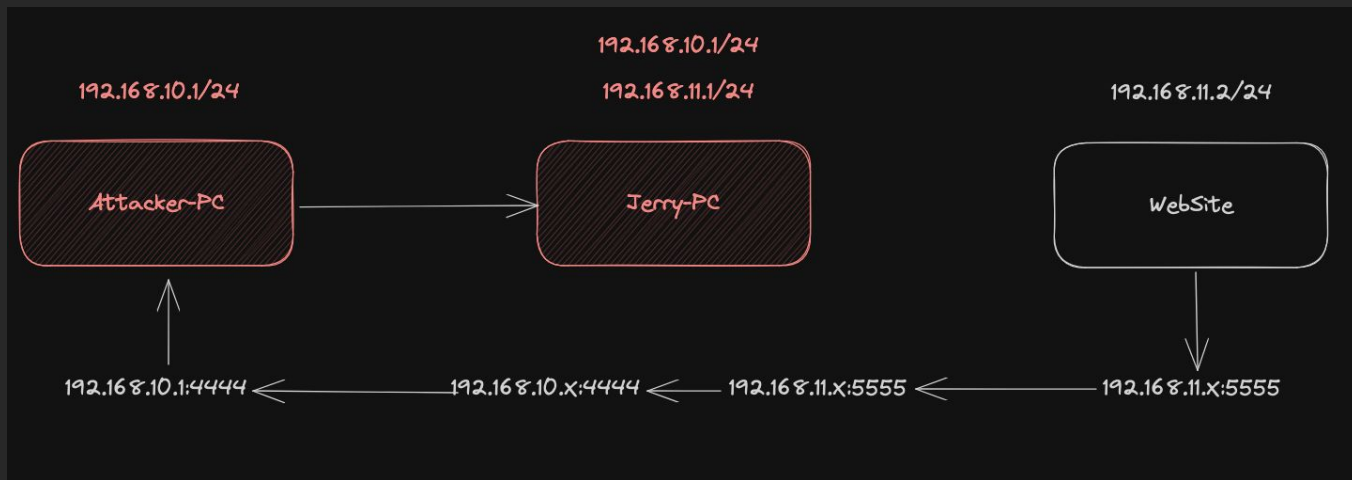
リモートポート転送

リモートポートフォワーディングは、ターゲットマシンから侵入先マシンへ、そして攻撃元マシンへとポートをリダイレクトするために使用されます。リッスンされるポートは攻撃元マシン上のポートであり、侵入先マシンから転送されるすべてのリダイレクトを受信します。

リモートポートフォワーディングの例として、リバースシェルを受信する場合は挙げられます。

リモートポート転送

「ポート 5555 の Jerry-PC で受信したすべての接続をポート 4444 の Attacker-PC にリダイレクトします。」



リモートポート転送 - デモChisel

The screenshot shows a Google Slides presentation titled "Remote Port Forwarding" by Fran Cantelli. The slide content includes the following text and diagram:

hackmetrix

Remote Port Forwarding

"Redirect all connections received on Jerry-PC at port 5555 to Attacker-PC at port 4444."

The diagram illustrates the network setup for Chisel port forwarding:

- Attacker-PC** (IP: $M2.16\ \#.10.1/24$)
- Jerry-PC** (IP: $M2.16\ \#.10.1/24$)
- WebSite** (IP: $M2.16\ \#.1.2/24$)

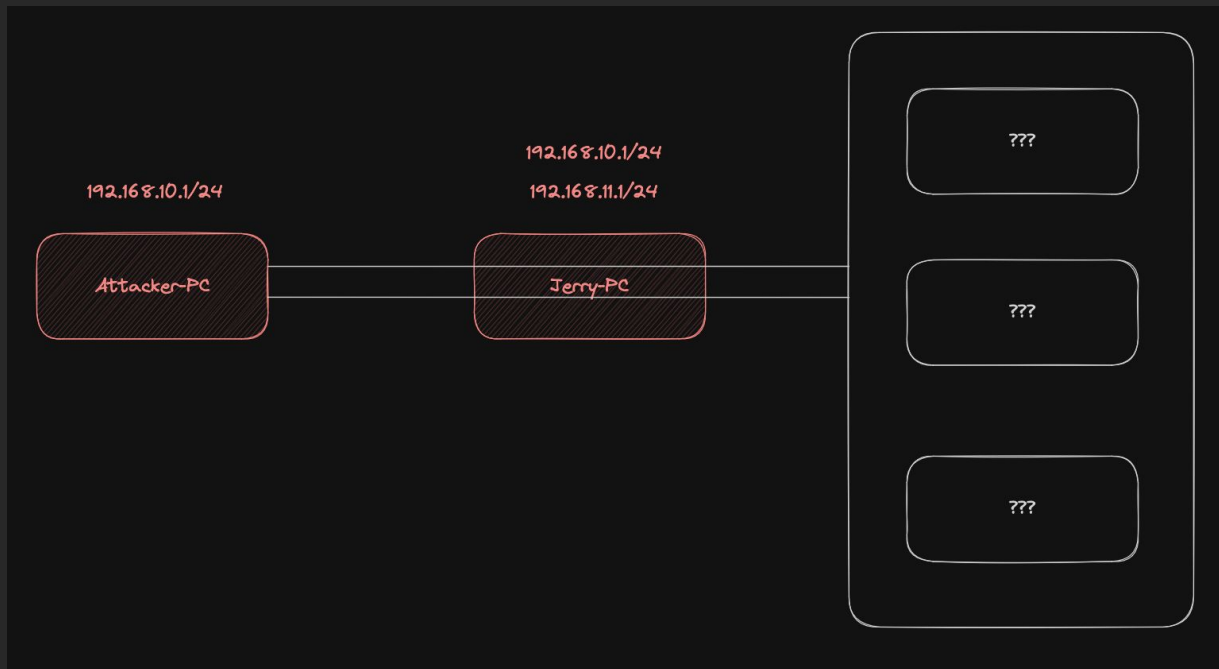
Network connections are shown as follows:

- Attacker-PC connects to Jerry-PC.
- WebSite connects to Jerry-PC.
- Jerry-PC connects to Attacker-PC (port 4444).
- Attacker-PC connects to WebSite (port 5555).

ダイナミックポートフォワーディング

- ダイナミックポートフォワーディングは、攻撃マシンと侵入先マシンの間に SOCKS5トンネルを作成します。これにより、攻撃者は内部ネットワーク上の複数のポートにアクセスできるようになります。
- SOCKS5 は、クライアントとサーバー間のトラフィックのルーティングに使用されるレイヤー 5 プロトコルです。
- すべてのネットワーク プロトコルが SOCKS5 と互換性があるわけではありません。
- 互換性の問題により、一部のプロトコルは SOCKS5 では動作しない可能性があります。

ダイナミックポートフォワーディング



SOCKS トンネル経由で ping できますか？

簡潔な答え: いいえ。

なぜSOCKS5トンネル経由でpingができないのでしょうか？

SOCKS(レイヤー5)はTCPとUDPのトラフィックをルーティングできますが、ICMP(レイヤー3)はルーティングできないためです。

SOCKS5は、セッション管理とプロキシ経由の TCPおよびUDPデータの転送に重点を置いたプロトコルです。TCPやUDPのようなコネクション指向プロトコルとは異なり、ICMPプロトコルは同じように動作しないため、SOCKS5セッションモデルには自然には適合しません。

SOCKS トンネル経由でポートをスキャンできますか ?

簡潔な答え: はい、ただし制限事項があります。

Nmapでは、完全なTCPスキャンを示す `-sT` フラグを使用してください。このタイプのスキャンは、ターゲットとの完全なTCP接続を確立します。つまり、接続要求が送信され、応答が待機されます。

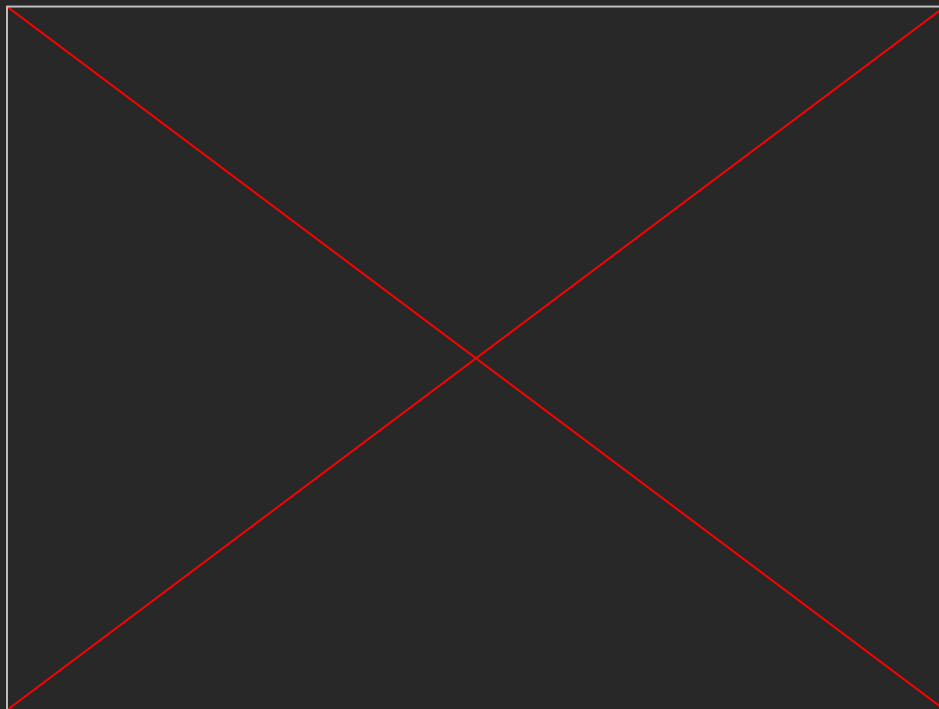
-sT フラグを使用しないとどうなりますか ?

簡潔に答えると、基本的には動作しません。

-sT フラグを使用しない場合、Nmap はデフォルトで SYN スキャン (-sS) を実行しようとする可能性があります。SYN スキャンでは、完全な応答を待たずに SYN パケットをターゲットに送信します。

これは SOCKS5 プロキシとは互換性がありません。SOCKS5 プロキシは通常、TCPトラフィックのルーティングに重点を置いており、確立された接続に含まれていない SYN パケットを適切に処理できない可能性があるためです。

ダイナミックポートフォワードイング - デモチゼル

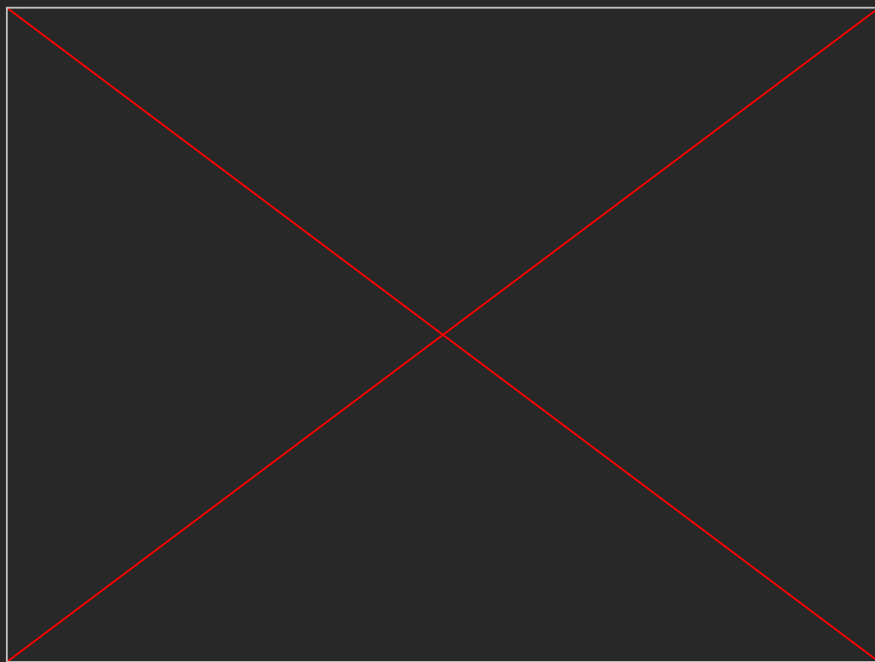


Ligolo-ng

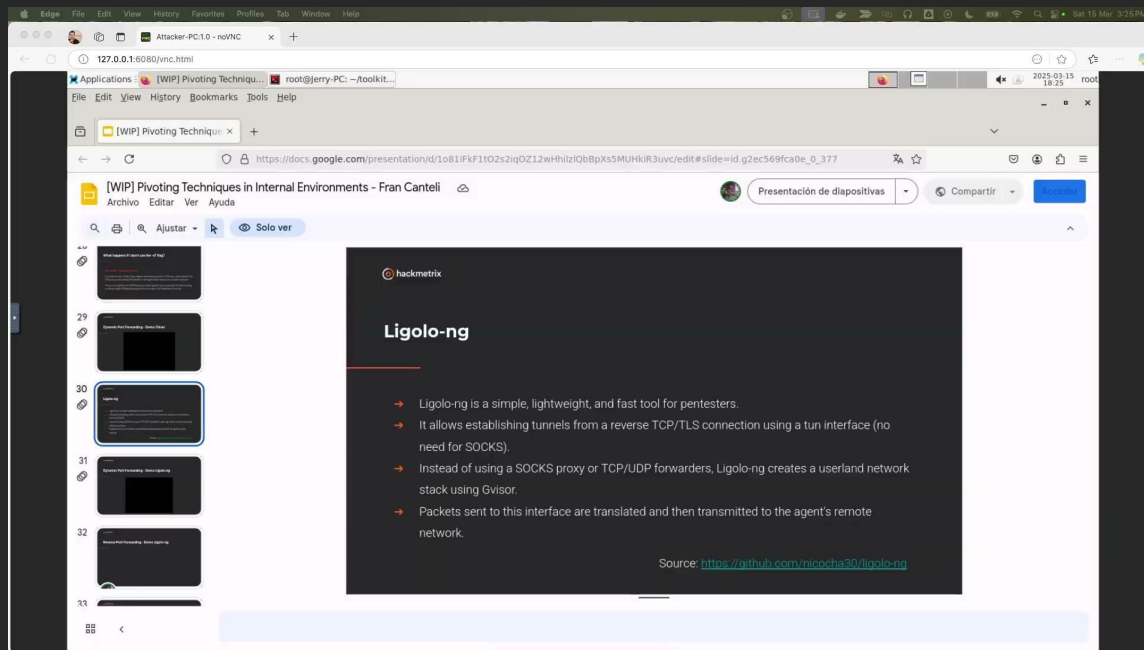
- Ligolo-ng は、ペンテスター向けのシンプルで軽量、かつ高速なツールです。
- tun インターフェースを使用してリバース TCP/TLS 接続からトンネルを確立できます (SOCKS は不要)。
- Ligolo-ng は、SOCKS プロキシまたは TCP/UDP フォワーダーを使用する代わりに、Gvisor を使用してユーザーランド ネットワーク スタックを作成します。
- このインターフェースに送信されたパケットは変換され、エージェントのリモート ネットワークに送信されます。

ソース: <https://github.com/nicocha30/ligolo-ng>

動的ポート転送 - デモ ライブラリ



リバースポートフォワーディング - デモライブラリ



ツールの比較

ツールの比較

Characteristic	Ligolo-Ng	Chisel	Traditional Tools (Metasploit, Cobalt Strike, SSH, etc)
Ease of Use	Medium, requires setup	Very easy, minimal config	Varies, can be complex (Metasploit, Cobalt Strike)
Detection Evasion	High, blends with normal traffic	Very high, mimics HTTP/S	Medium, often flagged by security tools
Tunneling Method	Not limited to SOCKS (full TCP tunneling)	SOCKS proxy-based	Mostly SOCKS (Metasploit, Cobalt Strike, SSH)
NAT Traversal	Excellent, usually bypasses restrictions	Good, but less flexible	Limited, requires extra setup (Metasploit, SSH)
Resource Usage	Low	Low	Medium to high (Metasploit, Cobalt Strike)
Cost	Free	Free	Can be expensive if using a Pro version.

实践！

ラボのセットアップ！



<https://github.com/franc205/pivoting-lab>

結論

結論

- Ligolo-NG は Chisel よりも柔軟性が高く、SOCKS を超える完全な TCP トンネリングをサポートし、制限された環境に最適です。
- Chisel はシンプルで効果的ですが、SOCKS プロキシに限定されています。
- 従来のツールは一般的に機能が豊富ですが、重く、検出されやすく、有料版では高価になることが多いです。

本当にありがとうございました！

ご質問はありますか？

