FTP サーバーインスペクタ

FTP サーバーインスペクタの概要

タイプ	インスペクタ（サービス）
使用方法	検査
インスタンス タイプ	マルチトン
その他のインスペクタが必要	`ftp_client`、`stream_tcp`
有効	`true`

File Transfer Protocol（FTP）は、TCP/IP を介してクライアントとサーバー間でファイルを転送するために使用されるネットワークプロトコルです。クライアントとサーバーが接続を確立すると、クライアントはサーバーにコマンドを発行してファイルをサーバーにアップロードするか、またはサーバーからファイルをダウンロードし、サーバーからの応答を解釈します。

`ftp_server` インスペクタは、FTP コマンドチャネル上のコマンドを確認して正規化します。

FTP コマンドチャネルバッファを指定すると、`ftp_server` インスペクタは FTP コマンドとパラメータを識別し、パラメータの正確性を適用します。`ftp_server` は、FTP コマンド接続がいつ暗号化され、FTP データチャネルがいつ開かれるかを決定します。

FTP サーバーインスペクタのパラメータ

FTP サーバーポートの設定

binderインスペクタは、FTPサーバーの設定を定義します。詳細については、『バインダインスペクタの概要』を参照してください。

例：

```
[
    {
        "when": {
            "role":"any",
            "service":"ftp",
            "ports": ""
        },
        "use": {
            "type":"ftp_server"
        }
    }
]
```

chk_str_fmt

書式文字列攻撃を確認するFTPコマンドのリストを指定します。ルール125:5を有効にしてアラートを生成し、インライン展開で、インスペクタがこの状態を検出したときに問題のあるパケットをドロップすることができます。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：有効なFTPコマンドのリスト。

デフォルト値：なし

data_chan_cmds

正しい書式設定を確認するFTPコマンドのリストを指定します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：PORT PASV LPRT LPSV EPRT EPSVの1つ以上のコマンドのリスト。

デフォルト値：なし

data_xfer_cmds

データ転送コマンドのリストを指定します。コマンドの正しい書式設定を確認します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：RETR STOR STOU APPE LIST NLSTの1つ以上のコマンドのリスト。

デフォルト値：なし

file_put_cmds

PUT コマンドのリストを指定します。コマンドの正しい書式設定を確認します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：STOR STOU APPE の 1 つ以上のコマンドのリスト。

デフォルト値：なし

⚠

注意 サポートからの指示がない限り、file_put_cmds パラメータは変更しないでください。

file_get_cmds

GET コマンドのリストを指定します。コマンドの正しい書式設定を確認します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：GET コマンド（RETR など）のリスト。

デフォルト値：なし

⚠

注意 サポートからの指示がない限り、file_get_cmds パラメータは変更しないでください。

encr_cmds

セキュア接続に関連するコマンドのリストを指定します。コマンドの正しい書式設定を確認します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：セキュア接続に関連するコマンド（AUTH など）のリスト。

デフォルト値：なし

login_cmds

ログインプロセスに関連するコマンドのリストを指定します。コマンドの正しい書式設定を確認します。複数のコマンドは余白文字で区切ります。

型：文字列

有効な値：USER、PASS など 1 つ以上のコマンドのリストを指定します。

デフォルト値：なし

check_encrypted

暗暗号化されたセッションで暗号化を終了するコマンドを確認するかどうかを指定します。`encrypted_traffic` パラメータとともに使用します。

このパラメータに対してルール 125:7 を有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

cmd_validity[]

FTP コマンドの配列と、インスペクタがそれらを検証するために使用する基準。これらの妥当性検査は、`ftp_server` インスペクタ（RFC 959）によって実行されるデフォルトの検査をオーバーライドします。

イベントを生成し、ライン展開ではこのパラメータの問題のあるパケットをドロップするには、ルール 125:2 と 125:4 を有効にします。

型：配列（オブジェクト）

例：

```
{
   "cmd_validity": [
       {
           "command": "CWD",
           "format": "abc",
           "length": 250
       }
   ]
}
```

cmd_validity[].command

検証する FTP コマンドの名前を指定します。

型：文字列

有効な値：二重引用符で囲まれた有効な FTP コマンド。

デフォルト値：なし

cmd_validity[].format

`cmd_validity[].command` の有効な形式について説明します。

型：文字列

有効な値：次のいずれかの形式。

- `int`：このパラメータは整数にする必要があります。
- `number`：このパラメータは 1 ～ 255 の整数にする必要があります。

- char *chars*：このパラメータは *chars* の 1 文字にする必要があり、1 文字以上のリストには文字間に区切り文字は使用しません。
- date *datefmt*：このパラメータは指定した形式に従います。*datefmt* は次の要素を使用して構成します。
  - # = 数字
  - c = 文字
  - [] = オプションの形式を囲みます
  - | = または
  - {} = 囲まれた形式の選択
  - .+- リテラル文字
- string：このパラメータは制限のない文字列です。
- host_port：このパラメータは、RFC 959 に従って、ホストポートの指定子にする必要があります。
- long_host_port：このパラメータは、RFC 1639 に従って、ホストポートの長い指定子にする必要があります。
- extended_host_port：このパラメータは、RFC 2428 に従い、拡張ホストポート指定子にする必要があります。
- {},|：このパラメータは、中括弧で囲まれ、| で区切られた選択肢の 1 つにする必要があります。
- {}, []：このパラメータは、中括弧で囲まれた選択肢の 1 つにする必要があります。オプションの値は角括弧で囲みます。

デフォルト値：なし

cmd_validity[].length

cmd_validity[].command パラメータの最大長をバイト単位で指定し、def_max_param_len で定義されているデフォルト値をオーバーライドします。FTP コマンドのパラメータが cmd_validity[].length を超え、ルール 125:3 が有効になっている場合、Snort はアラートを生成します。cmd_validity[].length を使用して、特定のコマンドを小さなパラメータ値に制限します。

長さに制限がないことを示すには、0 を指定します。

型：整数

有効な範囲：0 ～ 4,294,967,295 (max32)

デフォルト値：0

def_max_param_len

サーバーが処理するすべてのFTPコマンドに対してインスペクタが許可するデフォルトの最大長をバイト単位で指定します。基本的なバッファオーバーフロー検出には `def_max_param_len` を使用します。（これは、`cmd_validity[].length` を使用すると個々のコマンドでオーバーライドすることができます。）このパラメータに対してルール 125:3 を有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

長さに制限がないことを示すには、`0` を指定します。

型：整数

有効な範囲：`0` 〜 `4,294,967,295 (max32)`

デフォルト値：`100`

encrypted_traffic

暗号化された FTP トラフィックを確認するかどうかを指定します。`check_encrypted` パラメータとともに使用します。このパラメータに対してルール 125:7 を有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

ftp_cmds

RFC 959 で説明されている FTP コマンド以外にサーバーがサポートする FTP コマンドのリスト。（たとえば、インストールで RFC 775 で指定されている「X」コマンドを使用している場合は、このパラメータを使用してそれらをインスペクタに追加できます。）

型：文字列

有効な値：二重引用符で囲まれ、スペースで区切られた有効な FTP コマンドのリスト。

デフォルト値：なし

ignore_data_chan

FTP データ チャネルを無視するかどうかを指定します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

ignore_telnet_erase_cmds

FTP コマンドチャネルを正規化するときに、消去文字（TNC EAC）と消去行文字（TNC EAL）の Telnet エスケープシーケンスを無視するかどうかを指定します。`ignore_telnet_erase_cmds` を設定して、FTP サーバーが Telnet 消去コマンドを処理する方法を照合します。通常、新しい

FTP クライアントはこれらの Telnet エスケープシーケンスを無視しますが、レガシークライアントは通常、それらを処理します。

telnet 消去コマンドが無視されず、ルール 125:1 が有効になっている場合、Snort はイベントを生成し、インライン展開では問題のあるパケットをドロップします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

print_cmds

初期化時にこのサーバーの各 FTP コマンドの設定を出力するかどうかを指定します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

telnet_cmds

FTP コマンドチャネルで Telnet コマンドを確認するかどうかを指定します。このようなコマンドがある場合は、FTP コマンドチャネルでの回避試行を示している可能性があります。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

FTP サーバーインスペクタのルール

`ftp_server` インスペクタのルールを有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

表 *1 : FTP* サーバーインスペクタのルール

GID:SID	ルール メッセージ
125:1	FTP コマンドチャネルの TELNET コマンド（TELNET command on FTP command channel）
125:2	無効な FTP コマンド（invalid FTP command）
125:3	FTPコマンドパラメータが長すぎる（FTP command parameters were too long）
125:4	FTP コマンドパラメータの形式が正しくない（FTP command parameters were malformed）

GID:SID	ルール メッセージ
125:5	FTP コマンドパラメータに文字列形式が含まれている可能性がある（FTP command parameters contained potential string format）
125:7	FTP トラフィックが暗号化されている（FTP traffic encrypted）
125:9	FTP コマンドチャネルに回避（不完全）Telnet コマンドがある（evasive (incomplete) TELNET cmd on FTP command channel）

FTP サーバーインスペクタの侵入ルールのオプション

`ftp_server` インスペクタには侵入ルールのオプションはありません。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。