

セキュアシェル(SSH)パケット交換について

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[SSHプロトコル](#)

[SSH交換](#)

[関連情報](#)

はじめに

このドキュメントでは、セキュアシェル(SSH)ネゴシエーション中のパケットレベル交換について説明します。

前提条件

要件

基本的なセキュリティの概念に関する知識があることが推奨されます。

- [Authentication]
- 機密保持
- 整合性
- キー交換方式

使用するコンポーネント

このドキュメントは、特定のハードウェアバージョンに限定されるものではありません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな(デフォルト)設定で作業を開始しています。

SSHプロトコル

SSHプロトコルは、1台のコンピュータから別のコンピュータに安全なリモートログインを行うための方法です。SSHアプリケーションはクライアント/サーバアーキテクチャに基づいており、SSHクライアントインスタンスをSSHサーバに接続します。

SSH交換

1. SSHの最初のステップが Identification String Exchange.

a. クライアントはパケットを作成し、それを次の情報を含むサーバに送信します。

- SSHプロトコルバージョン
- [Software Version]

```
323 5.946818 10.65.54.8 10.106.51.72 SSHv2 82 Client: Protocol (SSH-2.0-PuTTY_Release_0.76)
> Frame 323: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
> Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
> Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1, Ack: 1, Len: 28
v SSH Protocol
  Protocol: SSH-2.0-PuTTY_Release_0.76
```

クライアントプロトコルのバージョンはSSH2.0、ソフトウェアのバージョンはPutty_0.76です。

b. サーバは、SSHプロトコルバージョンとソフトウェアバージョンを含む独自のID文字列交換で応答します。

```
326 6.016955 10.106.51.72 10.65.54.8 SSHv2 73 Server: Protocol (SSH-2.0-Cisco-1.25)
> Frame 326: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
> Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
> Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
> Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1, Ack: 29, Len: 19
v SSH Protocol
  Protocol: SSH-2.0-Cisco-1.25
```

サーバのプロトコルバージョンはSSH2.0で、ソフトウェアバージョンはCisco1.25です。

2. 次のステップは「Algorithm Negotiation.」です。このステップでは、クライアントとサーバの両方が次のアルゴリズムをネゴシエートします。

- 鍵交換
- 暗号化
- HMAC (ハッシュベースのメッセージ認証コード)
- 圧縮

1. クライアントは、サポートするアルゴリズムを指定して、Key Exchange Initメッセージをサーバに送信します。アルゴリズムは優先順位に従ってリストされます。

```
329 6.021990 10.65.54.8 10.106.51.72 SSHv2 238 Client: Key Exchange Init
> Frame 329: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits) on interface 0
> Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
> Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
> Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1101, Ack: 20, Len: 184
> [3 Reassembled TCP Segments (1256 bytes): #327(536), #328(536), #329(184)]
v SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 1252
    Padding Length: 11
  v Key Exchange
    Message Code: Key Exchange Init (20)
    > Algorithms
```

キー交換の初期化

```

 Algorithms
 Cookie: 47a96215afc92003180b60342970a105
 kex_algorithms length: 315
 kex_algorithms string [truncated]: curve448-sha512,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,dif
 server_host_key_algorithms length: 123
 server_host_key_algorithms string: rsa-sha2-512,rsa-sha2-256,ssh-rsa,ssh-ed448,ssh-ed25519,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-dss
 encryption_algorithms_client_to_server length: 189
 encryption_algorithms_client_to_server string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305
 encryption_algorithms_server_to_client length: 189
 encryption_algorithms_server_to_client string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305
 mac_algorithms_client_to_server length: 155
 mac_algorithms_client_to_server string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm
 mac_algorithms_server_to_client length: 155
 mac_algorithms_server_to_client string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm
 compression_algorithms_client_to_server length: 26
 compression_algorithms_client_to_server string: none,zlib,zlib@openssh.com
 compression_algorithms_server_to_client length: 26
 compression_algorithms_server_to_client string: none,zlib,zlib@openssh.com

```

クライアントがサポートするアルゴリズム

b.サーバは自身のKey Exchange Initメッセージで応答し、サポートするアルゴリズムをリストします。

c.これらのメッセージは同時に交換されるため、両方のパーティがアルゴリズムリストを比較します。両方の側でサポートされているアルゴリズムに一致がある場合は、次のステップに進みます。完全に一致するアルゴリズムがない場合、サーバはクライアントのリストから同じくサポートする最初のアルゴリズムを選択します。

d.クライアントとサーバが共通のアルゴリズムに合意できない場合、キー交換は失敗します。

```

 334 6.093250 10.106.51.72 10.65.54.8 SSHv2 366 Server: Key Exchange Init
 > Frame 334: 366 bytes on wire (2928 bits), 366 bytes captured (2928 bits) on interface 0
 > Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
 > Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
 > Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 20, Ack: 1285, Len: 312
 ✓ SSH Protocol
   ✓ SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
     Packet Length: 308
     Padding Length: 4
     ✓ Key Exchange
       Message Code: Key Exchange Init (20)
       > Algorithms

```

サーバーキー交換の初期化

3. この後、両側でKey Exchange フェーズに入り、DHキー交換を使用して共有秘密を生成し、サーバを認証します。

a.クライアントはキーペアを生成しPublic and Private、DH Group Exchange InitパケットでDH公開キーを送信します。このキーペアは、秘密キーの計算に使用されます。

```

 337 6.201114 10.65.54.8 10.106.51.72 SSHv2 326 Client: Diffie-Hellman Group Exchange Init
 > Frame 337: 326 bytes on wire (2608 bits), 326 bytes captured (2608 bits) on interface 0
 > Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
 > Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
 > Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1309, Ack: 612, Len: 272
 ✓ SSH Protocol
   ✓ SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
     Packet Length: 268
     Padding Length: 6
     ✓ Key Exchange
       Message Code: Diffie-Hellman Group Exchange Init (32)
       Multi Precision Integer Length: 256
       DH client e: 1405ab00ff368031363467ad6653967d5a64eac4734e5dc6...
       Padding String: 5c81f2cfff95

```

クライアントDH公開キーとDiffie-Hellmanグループ交換の初期化

b.サーバはそれ自体のPublic and Private キーペアを生成します。クライアントの公開キーと独自のキ

ーペアを使用して共有秘密を計算する

c.サーバは、次の入力を使用してExchangeハッシュも計算します。

- クライアント識別文字列
- サーバID文字列
- クライアントKEXINITのペイロード
- サーバKEXINITのペイロード
- ホストキーからのサーバ公開キー (RSAキーペア)
- クライアントDH公開キー
- サーバDH公開キー
- 共有秘密キー

d.ハッシュの計算後、サーバはRSA秘密キーを使用してハッシュに署名します。

e.サーバは、次の内容を含むメッセージDH_Exchange_Replyを作成します。

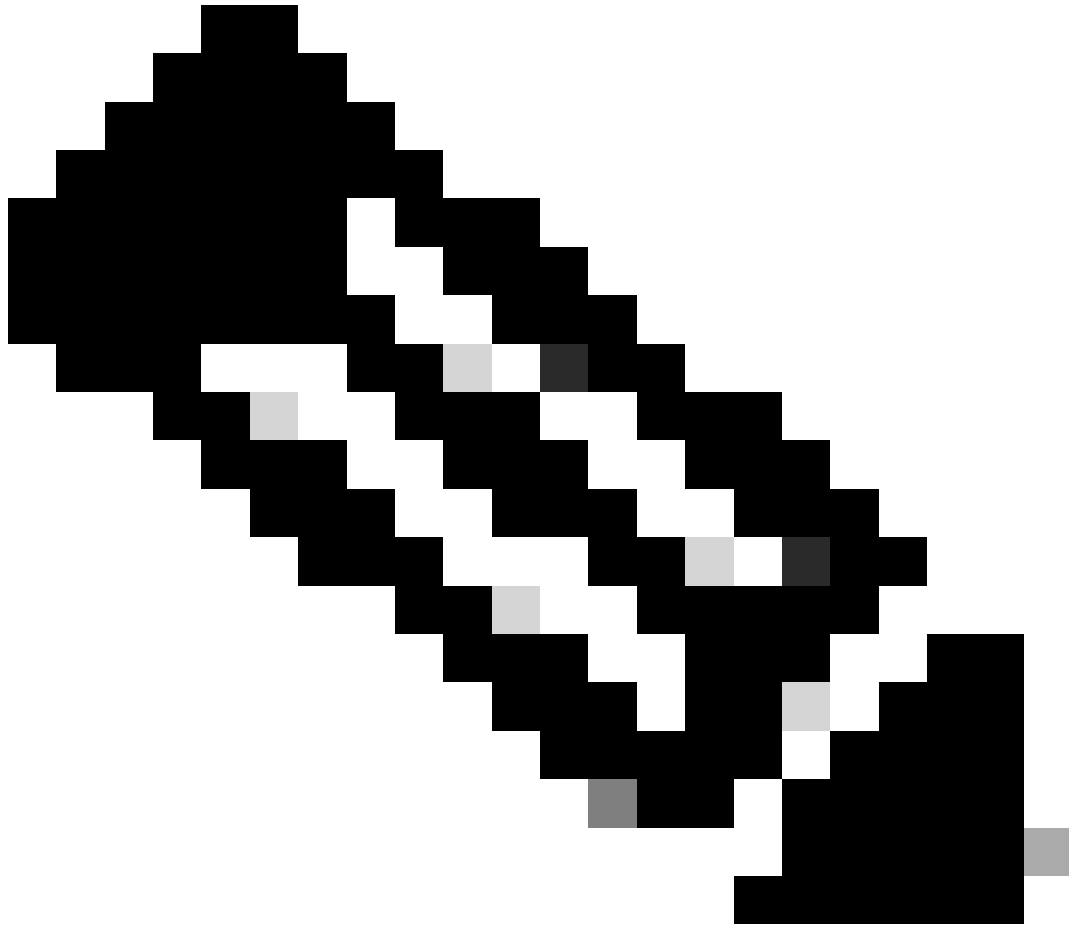
- サーバのRSA公開キー (クライアントがサーバを認証するのを支援するため)
- サーバのDH公開キー (共有秘密を計算するため)
- HASH (秘密キーはハッシュ計算の一部であるため、サーバを認証し、サーバが共有秘密を生成したことを証明するため)

```
343 6.330017 10.106.51.72 10.65.54.8 SSHv2 350 Server: Diffie-Hellman Group Exchange Reply
Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1148, Ack: 1581, Len: 296
[2 Reassembled TCP Segments (832 bytes): #342(536), #343(296)]
SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 828
    Padding Length: 8
    Key Exchange
      Message Code: Diffie-Hellman Group Exchange Reply (33)
      KEX host key (type: ssh-rsa)
        Host key length: 279
        Host key type length: 7
        Host key type: ssh-rsa
        Multi Precision Integer Length: 3
        RSA public exponent (e): 010001
        Multi Precision Integer Length: 257
        RSA modulus (N): 0098c7d23c9ababd730f07b5c2aee1e4e51bac67970aa5af...
        Multi Precision Integer Length: 256
        DH server f: 3a17a0995531f12d629a48ab6f25715bc181ea3deb6c6793...
        KEX H signature length: 271
        KEX H signature: 000000077373682d72736100000100691d2c896761bc7481...
        Padding String: 0000000000000000
```

サーバDH公開キーとDiffie-Hellmanグループ交換応答


f. DH_Exchange_Replyの受信後、クライアントは同じ方法でハッシュを計算し、受信したハッシュと比較し、サーバのRSA公開キーを使用して復号化します。

g.受信したHASHを復号化する前に、クライアントはサーバの公開キーを確認する必要があります。この検証は、認証局(CA)によって署名されたデジタル証明書を使用して行われます。証明書が存在しない場合は、サーバの公開キーを受け入れるかどうかをクライアントが決定します。



注：デジタル証明書を使用しないデバイスに初めてSSH接続する際に、サーバの公開キーを手動で受け入れるよう求めるポップアップが表示されることがあります。接続するたびにポップアップが表示されないようにするには、サーバのホストキーをキャッシュに追加することを選択できます。

Warning ? X



Continue connecting to an unknown server and add its host key to a cache?

The server's host key was not found in the cache. You have no guarantee that the server is the computer you think it is.

The server's RSA key details are:

Algorithm: ssh-rsa 2048
 SHA-256: [REDACTED]
 MD5: [REDACTED]

If you trust this host, press Yes. To connect without adding host key to the cache, press No. To abandon the connection press Cancel.

[Copy key fingerprints to clipboard](#)

サーバのRSAキー

4. これで共有秘密が生成されたため、両方の端末が共有秘密を使用してこれらの鍵を取得します (共有秘密は暗号化の対象となります)。

- 暗号化キー
- IVキー：これらは、セキュリティを強化するために対称アルゴリズムへの入力として使用される乱数です
- 整合性キー

キー交換の終了は、NEW KEYS' メッセージの交換によって通知されます。このメッセージによって、各当事者は以降のすべてのメッセージがこれらの新しいキーを使用して暗号化および保護されることを知らされます (新しいキーによって、新しいキーは暗号化されません)。

346	6.330368	10.106.51.72	10.65.54.8	SSHv2	70	Server: New Keys
347	6.365552	10.65.54.8	10.106.51.72	SSHv2	70	Client: New Keys

```

> Frame 346: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
> Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
> Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1444, Ack: 1581, Len: 16
✓ SSH Protocol
  ✓ SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 12
    Padding Length: 10
    ✓ Key Exchange
      Message Code: New Keys (21)
      Padding String: 00000000000000000000
  
```

クライアントとサーバの新しいキー

5. 最後のステップはサービスリクエストです。クライアントは、SSHサービスリクエストパケットをサーバに送信してユーザ認証を開始します。サーバはSSH Service Acceptメッセージで応答し、クライアントにログインを要求します。この交換は、確立された安全なチャネルを介して行われます。

関連情報

- <https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html>
- <https://datatracker.ietf.org/doc/html/rfc4253>
- [シスコのテクニカルサポートとダウンロード](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。