

正規表現とgrep

ログをフィルタリングして目的のデータポイントを発見するには、通常、**正規表現(regex)**構文を使用したある種の文字列検索が必要です。正規表現とは、与えられた文字列の中でマッチする検索パターンです。検索パターンは 正規表現の構文から作られます。この構文は、検索演算子、数量詞、論理ステートメント、アンカー //バウンダリーとして機能するメタ文字を定義します。次のリストは、正規表現の構文でよく使われるいくつかの要素を示しています

- [...]は、角括弧内の1文字の单一インスタンスにマッチします。これにはリテラル、[a-z]などの範囲、および[\s] (空白) または[\d] (1桁の数字) などのトークンマッチが含まれます。
- +は1回以上の繰り返しにマッチします。数量詞は条件の後ろに置いてマッチさせます。例えば、\s+は1文字以上の空白文字にマッチします。
- *は0回以上の繰り返しにマッチします。
- ?は、1回または0回の繰り返しにマッチします。
- { }は、括弧内の回数にマッチします。例えば、{ 2 }は2回の繰り返しに、{ 2 , }は2回以上の繰り返しに、{ 2 - 5 }は2 ~ 5回の繰り返しにマッチします。



正規表現の構文の完全な説明は本コースの範囲から外れていますが、regexpr.comやrexegg.comなどのオンライン資料を用いて学ぶことができます。

grepコマンドは、簡素な文字列マッチングまたは正規表現の構文を呼び出し、テキストファイルを検索して特定の文字列を探します。これにより、テキストファイルの中身全体を検索して各行の特定の文字列を探し、そのパターンをスクリーンに表示させたり、別のファイルにダンプしたりすることができます。grepの簡単な使用例を以下に示します。

```
grep -F 192.168.1.254 access.log
```

これはテキストファイルaccess.logを検索し、リテラル文字列パターン192.168.1.254の何らかのバリエーションを含むすべての行を探した上で、それらの行だけを端末にプリントします。-Fスイッチは、そのパターンをリテラルとして扱うようgrepに指示するものです。

次の例は、正規表現の構文を用いて192.168.1.0/24サブネット内のすべてのIPアドレスを検索し、現在のディレクトリ内の全てのファイルの中にあるパターンを探すものです（それぞれのピリオドがエスケープされていることに注意してください）。-rオプションで反復が可能になりますが、ターゲット部分のピリオドは現在のディレクトリを指し示しています。

```
grep -r 192\.168\.1\.\[\d\]\{1,3\}.
```

レビュー アク ティビティ： SIEMの使用

次の質問にお答えください。

1. SIEMの目的は何ですか？
2. SIEMのコンテキストにおいて、センサーとコレクターの違いは何ですか？
3. SyslogはSIEMのすべての機能を実行しますか？
4. ダッシュボード内の/var/log/auditにあるログファイルの最後の5行を表示させるシェルスクリプトをあなたは記述しています。これを実行するLinuxコマンドは何ですか？
5. ログファイルに関するgrepの主な使用例は何ですか？

レッスン10

概要

セキュアなネットワーク設計を実装するために、ファイアウォール、プロキシ、IDS、SIEMの収集/集約機能などのネットワークアプライアンスを使用できる必要があります。

ネットワークセキュリティアプライアンスを実装する際のガイドライン

新規のセキュリティアプライアンス、またはアップグレードしたセキュリティアプライアンスをデプロイする際は、次のガイドラインに従います。

- ネットワークゾーンまたはエリアのセキュリティ要件を特定し、以下に挙げる適切なセキュリティ技術を選んで使用する。
 - 出入りするトラフィックにACLを適用するネットワークファイアウォール。
 - シグチャ型や振る舞い検知型の脅威検知を実装するためのIDS、IPS、または次世代ファイアウォール。
 - 外部のサイトやサービスなどへのユーザーアクセスを制御するためのコンテンツフィルター。
 - 単一のアプライアンスやレポートインターフェイス内にさまざまな制御を実装するためのUTM。
- ホスト型ファイアウォール、WAF、またはファイル完全性の監視など、追加のセキュリティでゾーン内のエンドポイントを保護すべきかどうかを評価する。
- 商用モデルを評価し、プロプライエタリとオープンソースのどちらが自分の要件に最も適しているかを判断する
- 設計目標を充足することを保証するためにデバイスを実装する際に、ACLまたはその他のセキュリティ構成を文書化してテストする。
- ログの記録、ネットワークデータの収集と集約について、以下に挙げる適切な方式を実装し、セキュリティイベントの監視とレビューを確実に行う。
 - syslogとファイル操作ツールを用いた手入力方式(head, tail, cat, grep, logger)。
 - SIEM (Security Information and Event Management)製品。
 - SOAR (Security Orchestration, Automation, and Response)製品。

レッスン11

セキュアなネットワークプロトコルを実装する

レッスン概要

ネットワークに参加する各ホストは、そのネットワークに適した設定で構成する必要があります。DHCPやDNSなど、それらの設定を提供するサービスは安全にデプロイ（展開）しなければなりません。Web/HTTP、電子メール、VoIPなどのサーバーアプリケーションを用いてホストがデータにアクセスする際、クライアントとサーバー間の通信は、セキュアなバージョンのアプリケーションプロトコルを使用して管理しなければなりません。また、ユーザーがネットワーク、ホストデスクトップ、またはアプリケーション構成インターフェイスにリモートでアクセスできるようにする、セキュアなプロトコルを構成する必要があります。

レッスンの目的

このレッスンの内容は、以下のとおりです。

- セキュアなネットワークオペレーションプロトコルを実装する。
- セキュアなアプリケーションプロトコルを実装する。
- セキュアなリモートアクセスプロトコルを実装する。

トピック11A

セキュアなネットワークオペレーションプロトコルを実装する



対象試験範囲

1.4与えられたシナリオに基づいて、ネットワーク攻撃に関連する可能性のあるインジケーターを分析することができる。

3.1与えられたシナリオに基づいて、セキュアなプロトコルの実装を行うことができる。

セキュアでないプロトコルは脅威アクターによって悪用され、データのセキュリティやシステムの完全性が侵害される可能性があります。このトピックでは、アドレス指定、名前解決、ディレクトリサービス、時刻同期、監視サービスをネットワークホストに提供するプロトコルとサービスの一部を検証します。これらのネットワークオペレーションプロトコルは、Webサーバーやメールサーバーほど目に見えるものではありませんが、セキュアなネットワークインフラストラクチャに不可欠なものです。

ネットワークアドレス割り当て

ほとんどのネットワークは、静的アドレス割り当てと動的アドレス割り当てを組み合わせて使用しています。ルーター、ファイアウォール、一部のタイプのサーバーのインターフェイスアドレスは、手動で割り当てと管理を行うのが最善です。その他のサーバーサービスやクライアントサービスには動的IP構成を割り当て、名前解決を用いてアクセスすることができます。

ダイナミックホストコンフィグレーションプロトコル(DHCP)は、ネットワークアドレス割り当てを自動で行う方法を提供します。DHCPについて重要なのは、1つのホストグループに対して、1つのサーバーのみがアドレスを提供すべきであるという点です。不正なDHCPサーバーがセットアップされると、DoSを実行したり（クライアントマシンが正しくないTCP/IP構成を取得するため）、ネットワーク情報を盗聴するためにそれを用いることができます。DHCP飢餓攻撃(starvation attack)はDoS攻撃の一種であり、IPアドレスプールを枯渇させる目的で、不正なクライアントが偽造したMACアドレスを用いて新規のIPアドレスを繰り返しリクエストします。これにより、アドレスのリースを求めるクライアントが不正なDHCPサーバーを使用する可能性が高くなります。

スイッチ上でDHCPスヌーピングポートセキュリティ機能を有効化することで、不正なDHCP攻撃を軽減できます。AD環境にあるWindows DHCPサーバーは、不正なDHCPサーバーからのトラフィックが検知されると自動でログに記録します。より一般的に言えば、DHCPサーバー自体の管理を慎重にコントロールし、定期的に設定をチェックする必要があります。脅威アクターがDHCPサーバーを侵害すると、ネットワーククライアントを不正なDNSサーバーを使用させ、なりすましたWebサイトにユーザーを誘導する手段として用いることができます。別の攻撃として、デフォルトゲートウェイを変更することで、トラフィックが脅威アクターのマシンを通過するようリダイレクトするというものがあり、脅威アクターはすべてのネットワークトラフィックを盗聴することができます。

The screenshot shows a Kali Linux terminal window with four tabs. The tabs are labeled 'root@KALI: ~', 'root@KALI: ~', 'root@KALI: ~', and 'root@KALI: ~'. The first tab contains msf auxiliary(dhcp) commands to set up a DHCP server on port 192. The second tab shows dnsspoof listening on eth0 for DNS requests. The third tab lists various DHCP Discover messages. The fourth tab shows the exploit completed with a message about the DHCP pool being exhausted.

```

root@KALI: ~
msf auxiliary(dhcp) > set dhcpielnd 10.1.0.210
dhcpielnd => 10.1.0.210
msf auxiliary(dhcp) > set netmask 255.255.255.0
netmask => 255.255.255.0
msf auxiliary(dhcp) > set dnsserver 10.1.0.192
dnsserver => 10.1.0.192
msf auxiliary(dhcp) > set router 10.1.0.192
router => 10.1.0.192
msf auxiliary(dhcp) > set srvhost 10.1.0.192
srvhost => 10.1.0.192
msf auxiliary(dhcp) > show options

Module options (auxiliary/server/dhcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
BROADCAST          no        The broadcast address to send to
DHCPIPEND       10.1.0.210  no        The last IP to give out
DHCPIPSTART     10.1.0.200  no        The first IP to give out
DNSERVER        10.1.0.192  no        The DNS server IP address
DOMAINNAME          no        The optional domain name to assign
FILENAME          no        The optional filename of a tftp boot
server
HOSTNAME          no        The optional hostname to assign
HOSTSTART          no        The optional host integer counter
NETMASK        255.255.255.0 yes      The netmask of the local subnet
ROUTER         10.1.0.192  no        The router IP address
SRVHOST        10.1.0.192  yes      The IP of the DHCP server

Auxiliary action:
Name      Description
----      -----
Service

[*] Auxiliary module execution completed
[*] Starting DHCP server...
[*] msf auxiliary(dhcp) >

```

ネットワークアドレス割り当てへの攻撃 – スクリプトがDHCPプールを枯渇させる一方、別のスクリプトが不正なDHCPサーバーを動作させています。不正なDHCP構成により、攻撃側のマシンをDNSサーバーとして使用するよう偽装された情報をクライアントに送信することにより、3番目のツールは不正なDNSを実行します。

ドメイン名解決

DNS (Domain Name System)は、完全修飾ドメイン名(FQDN)からIPアドレスへの名前解決を実行します。DNSは、ドメインとそれらドメイン内のホストに関する情報を含む分散型データベースシステムを使用しており、そのデータベースを構成する多数のネームサーバーに情報が分散されます。それらのネームサーバーはポート53で動作します。ドメイン名解決はセキュリティが重要なサービスであり、ローカルネットワークやインターネットに対する数多くの攻撃の標的となっています。

ドメインハイジャッキング

ドメインハイジャッキングは、ある企業の商号または商標を含むドメイン、もしくはそれと似たスペルのドメインを取得する攻撃です。多くの場合、それを禁じる商標および知的財産関連の法律があるものの、企業は引き続き使用したいドメイン名を更新し、登録の管理に用いる認証情報を保護するよう注意する必要があります。ドメイン名は毎年再登録しなければなりません。

ドメインハイジャッキング攻撃において、脅威アクターはドメイン名の登録に関するコントロール権を取得し、脅威アクターが選んだIPアドレスにホストレコードを構成できるようにします。これは、新規のドメイン名を申請する際、または既存のドメインを再登録する際、ドメイン登録機関に偽の認証情報を提供することで実行されることがあります。また脅威アクターは、ドメインの管理に用いられる正当なアカウントを（脆弱なパスワードを介して、またはクライアントコンピューター上にインストールしたマルウェアを介して）悪用したり、さらには何らかの方法でドメイン登録機関のセキュリティ手順を侵害したりすることができるかもしれません(upguard.com/blog/domain-hijacking)。

ドメインがハイジャックされた企業は、自分たちが登録機関の管理コンソールから締め出されている、またはそのドメインが、多くの場合他国で運営されている別の登録機関に移されたことを知ることになるでしょう。whoisコマンドを使用することでドメインの登録情報を検索し、他のケースで悪用されているかどうかを突き止めることができます。

URL (Uniform Resource Locator)リダイレクト

URL (Uniform Resource Locator)は、Webサイトとして公開されるページやファイルのアドレスです。URLはFQDN、ファイルパス、そして多くの場合スクリプトパラメータから構成されます。URLリダイレクトとは、ユーザーがリクエストしたページ以外のページを開くためにHTTPリダイレクトを使用することを指します。これは多くの場合、ユーザーをログインページに誘導する、またはモバイルデバイスのブラウザをそのサイトの対応するバージョンに誘導するなど、合法的目的で用いられます。リダイレクトがWebアプリケーションによって正しく検証されていないと、脅威アクターは何も知らないユーザーにとって正当に見える次のようなフィッシングリンクを作成することができます。

```
https://trusted.foo/login.php?url="https://
trusted.foo"
```

また脅威アクターがWebサーバーを侵害し、.htaccessファイルにリダイレクトを追加することもあります。さらに、リダイレクトはサーバーの侵害を通じて、または十分な検証がなされていないフォーム経由でスクリプトをアップロードすることにより、JavaScriptとして挿入されることもあります。

ドメインの評判

あなたのドメイン、Webサイト、またはメールサーバーがハイジャックされると、スパムやマルウェアの配布のために用いられる可能性があります。これはクレームと、そのドメインがブラックリストに掲載される可能性につながります。悪用を早期に検知するために、talosintelligence.com/reputation_centerなどのサイトを用いて監視をセットアップする必要があります。

DNSポイズニング

DNSポイズニングは、クライアントがネームサーバーにクエリを行ってFQDNをIPアドレスに名前解決するプロセスを侵害する攻撃です。DNSポイズニング攻撃を仕掛けるにはいくつかの方法があります。

中間者攻撃

脅威アクターが被害者と同じローカルネットワークへのアクセスを有している場合、脅威アクターは、ARPポイズニングを利用して、被害者からのDNSクエリに偽装した返信をすることができます。被害者の正当なDNSサーバーに対するサービス拒否攻撃がこれと組み合わされることもあります。不正なDHCPは、不正なDNSリゾルバーのアドレスをクライアントに設定するためには使用されることがあります。

DNSクライアントキャッシュポイズニング

1980年代にDNSが開発されるまで、名前解決はHOSTSという名称のテキストファイルを用いて行われていました。名前とIPアドレスの対応はこのファイルに記録され、システム管理者は最新のコピーをダウンロードし、手動で各インターネットクライアントまたはサーバーにインストールする必要がありました。現在、すべての名前解決はDNSを通じて実行されますが、HOSTSファイルは今なお存在しており、大半のオペレーティングシステムはDNSを使用する前にそのファイルをチェックします。その内容は、名前と既知の対応IPアドレス情報としてキャッシュにロードされていて、名前がキャッシュされていない場合のみ、クライアントはDNSサーバーに問い合わせを実行します。従って、HOSTSファイルに名前と偽の対応IPアドレス情報を挿入し、効果的にDNSキャッシュをポイズニングできれば、脅威アクターはトライックをリダイレクトできること

になります。HOSTSファイルを修正するには管理者アクセスが必要となります。UNIXやLinuxシステムでは/etc/hostsとして保存されており、Windowsでは%SystemRoot%\System32\Drivers\etc\hostsに置かれています。

DNSサーバーキャッシュポイズニング

DNSサーバーキャッシュポイズニングは、DNSサーバー自体が保持しているレコードを改変するのが目的です。これは、許可されたドメインのレコードを保持するサーバーに対してDoS攻撃を実行し、その他のネームサーバーからのリクエストに対する応答を偽装することで行われます。もう一つの攻撃は、攻撃側のホストからの再帰的なクエリに対して、被害側のネームサーバーが応答するように仕向けることです。再帰クエリにより、DNSサーバーはクライアントの代わりに権威ネームサーバーにクエリを行い、回答を求めます。脅威アクターのDNSは、権威あるネームサーバーを装い、クエリに対する回答を返しますが、他のドメインに関するドメインと偽の対応IPアドレス情報を多数含めておき、被害者のDNSがそれらを正当なものとして受け入れます。nslookupまたはdigツールを用いることで、サーバーが保持する名前レコードとキャッシュされたレコードのクエリを行い、偽のレコードが挿入されたかどうかを突き止めることができます。

DNSセキュリティ

DNSは重要なサービスであり、障害に耐えるよう構成しなければなりません。インターネット名前解決を行うサーバーにDoS攻撃を実行することは困難ですが、脅威アクターがプライベートネットワーク上のDNSサーバーを標的にすると、そのネットワークの運用を著しく妨げることができます。

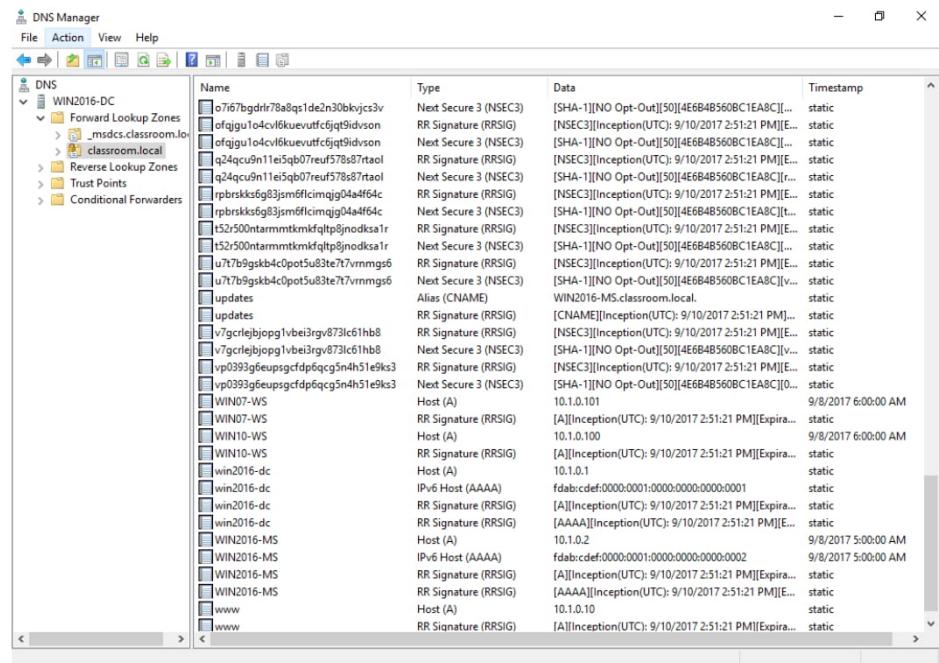
プライベートネットワーク上でDNSのセキュリティを保証するにあたり、ローカルDNSサーバーはローカルホスト（できれば認証されたローカルホスト）からのみ再帰クエリを受け入れ、インターネットからは受け入れないようにする必要があります。または、サーバー上でアクセス制御を実行し、悪意のあるユーザーが手作業でレコードを改変するのを防ぐことも必要です。同様に、許可されたリゾルバーを使って名前解決を行うようクライアントを制限しなければなりません。

DNSに対する攻撃は、サーバーアプリケーションやサーバー構成を標的とすることもあります。多くのDNSサーバーは、Internet Software Consortium (isc.org)が配布するBIND (Berkeley Internet Name Domain)上で動作します。BINDサーバーの多数のバージョンには既知の脆弱性があるので、サーバーに最新バージョンのパッチを適用することが重要です。この一般的なアドバイスは、Microsoftのものなどその他のDNSサーバーソフトウェアにも当てはまります。セキュリティに関するアナウンス入手・チェックし、セキュリティ関連の重要なパッチとアップグレードをテストした上で適用してください。

DNSフットプリントイングとは、DNSサーバーから不正なDNSへのゾーン転送を行うことにより、または単にnslookupやdigなどのツールを用いてDNSサービスにクエリを行うことにより、プライベートネットワークに関する情報を取得することを意味します。これを防ぐため、アクセス制御リストを適用して不正なホストまたはドメインへのゾーン転送を防止し、外部サーバーがプライベートネットワークアーキテクチャに関する情報を取得するのを防止することができます。

DNSSEC (DNS Security Extensions)は、DNSレスポンスの検証プロセスを提供することで、スプーフィング（なりすまし）またはポイズニング攻撃を軽減します。DNSSECが有効になると、そのゾーンの権威サーバーは秘密鍵（ゾーン署名鍵）で署名されたリソースレコードのパッケージ（RRsetと呼ばれます）を作成します。別のサーバーがセキュアなレコード交換をリクエストすると、権威サーバーはパッケージをその公開鍵と共に返すので、署名を検証するために使用できます。

パブリックゾーン署名鍵自体も個別の鍵署名鍵で署名されています。個別の鍵を用いることで、ゾーン署名鍵に何らかの侵害がある場合でも、ドメインは侵害された鍵を無効化して新しい鍵を発行することで、引き続きセキュアな形で動作することができます。



DNSSECが有効になっているWindows ServerのDNSサービス。
(スクリーンショットはMicrosoftからの許可を得て使用。)

特定のドメイン用の鍵署名鍵は、親ドメインまたはホストISPによって検証されます。トップレベルドメインの信頼性は地域インターネットレジストリによって検証され、DNSルートサーバーはM-of-Nコントロール型のグループ鍵署名を用いて自己検証が行われます。これによってルートサーバーから特定のサブドメインに至るまでの「信頼の連鎖」が確立されます。

セキュアなディレクトリサービス

ネットワークディレクトリは主体（主にユーザー、コンピューター、サービス）と、ネットワーク上で利用可能な対象（ディレクトリやファイルなど）に加え、主体が対象に対して有する許可をリストにしたもので、ディレクトリは認証と認可を容易にするものなので、安全性の高いサービスとして維持されることが重要です。大半のディレクトリサービスは**LDAP (Lightweight Directory Access Protocol)**を基礎としてポート389で動作します。基本プロトコルはセキュリティを一切提供せず、すべての通信は平文で行われるので、スニッフィングや中間者攻撃に対して脆弱です。認証（「サーバーにバインドする」と呼ばれます）は次の方法で実装できます。

- 無認証 – ディレクトリへの匿名のアクセスが許可されます。
- シンプルバインド – クライアントは識別名(DN)とパスワードを提供しなければなりませんが、それらは平文で渡されます。
- 簡易認証セキュリティレイヤー (SASL) – Kerberosなどサポートされている認証メカニズムの使用を、クライアントとサーバーが取り決めます。STARTTLSコマンドを用いると、暗号化（封印）とメッセージの完全性（署名）を要求できます。これは、MicrosoftのLDAPの実装であるActive Directory (AD)で推奨されている仕組みです。
- LDAPS (LDAP Secure)** – サーバーはデジタル証明書と共にインストールされ、その証明書を用いてユーザー認証情報の交換を行うセキュアなトンネルをセットアップします。LDAPはポート636を使用します。

セキュアなアクセスが必要な場合は、匿名と単純な認証アクセス方法をサーバー上で無効にしなければなりません。

一般的に、2つのレベルのアクセス、つまり読み取り専用アクセス（クエリ）と読み取り/書き込みアクセス（アップデート）がディレクトリ上で許可される必要があります。これはアクセス制御ポリシーを用いて実装されますが、詳細なメカニズムはベンダーごとに異なっており、LDAPの標準文書によって規定されているわけではありません。

パブリックサービスをホストするのでない限り、LDAPディレクトリサーバーはプライベートネットワークからのみアクセスできるようにしなければなりません。このことは、LDAPポートはファイアウォールによって、パブリックインターフェイス経由のアクセスからブロックされなければならないことを意味します。インターネット経由で他のサービスと統合されている場合は、できれば承認されたIPだけを許可すべきです。

時刻同期

ネットワークの多くのアプリケーションは時刻に依存しており、正確な時刻が不可欠です。これには認証とセキュリティメカニズム、スケジュールアプリケーション、バックアップソフトウェアが含まれます。NTP (Network Time Protocol)は、時間に依存するこれらアプリケーションを同期させる手段を提供します。NTPはポート123のUDP上で機能します。

トップレベルのNTPサーバー（ストラタム1）は、原子時計など極めて正確なクロックソースから協定世界時(UTC)を取得します。次に下位のサーバーが複数のストラタム1サーバーからUTCを取得し、その結果をサンプリングして権威のある時刻を取得します。大半の組織はこれらストラタム2サーバーの1つを用いて、LAN上で使用する時刻を取得します。さらに下位のサーバーは同様のサンプリングを行い、信号の伝搬による遅れを調整し、クライアントに時刻を提供します。通常、クライアント自身もプロトコルの修正版(Simple NTP)を用いて時刻を取得します。

歴史的に、NTPにはいかなる種類のセキュリティメカニズムもありませんが、Network Time Securityと呼ばれるセキュリティ拡張を制定する動きがあります(blog.cloudflare.com/secure-time)。

簡易ネットワーク管理プロトコルセキュリティ

SNMP (Simple Network Management Protocol : 簡易ネットワーク管理プロトコル)は、幅広く使用されている管理とモニタリングのフレームワークです。SNMPはSNMPモニターとエージェントで構成されています。

- このエージェントは、スイッチやルーター、サーバー、その他のSNMP対応ネットワークデバイスで実行されるプロセス（ソフトウェアまたはファームウェア）です。
- このエージェントは、デバイスのアクティビティに関する統計（スイッチが処理する1秒あたりのフレーム数など）を保持する管理情報ベース(MIB)という名のデータベースを維持します。また、エージェントには管理システムに注目すべきイベント（ポートエラーなど）を通知するトラップ操作を開始する機能もあります。トラップをトリガーするしきい値は、値ごとに設定できます。デバイスクエリはポート161(UDP)を介して行われ、トラップはポート162（同じくUDP）を介して通信されます。
- SNMPモニター（ソフトウェアプログラム）では、ネットワーク活動を監視することができます。すべてのエージェントを定期的にポーリングして監視することで、MIBの情報を取得し、表示します。また、トラップ操作をアラートとして表示することで、ネットワーク管理者が評価し、必要に応じて対処できるようにします。

SNMPを使用しない場合は、必ずデフォルト構成パスワードを変更し、ネットワークに追加するすべてのSNMP対応デバイスでデフォルトパスワードを無効にしなければなりません。SNMP v1またはv2cを動作させている場合は、次のガイドラインを守ってください。

- SNMPコミュニティ名は平文で送信されるので、傍受されるリスクがある場合はネットワーク経由で送信しない。

- 推測するのが難しいコミュニティ名を使用し、コミュニティ名を空白にしたり、デフォルトに設定したりしない。
- アクセス制御リストを用いることで、管理オペレーションを既知のホストに制限する（つまり、1つまたは2つのホストIPアドレスに制限する）。
- SNMP v3は暗号化と強力なユーザーベース認証をサポートしています。エージェントはコミュニティ名の代わりに、ユーザー名とアクセス許可のリストで構成されます。認証が必要な場合は、ユーザーのパスフレーズのハッシュでSNMPメッセージの署名を行えます。エージェントはその署名を検証し、それ自身が持つパスフレーズのレコードを用いてユーザーを認証することができます。

レビューアク ティビティ：

セキュアなネットワークオペ レーションプロトコル

次の質問にお答えください。

1. 不正なDHCPサーバーはユーザーをどのような脆弱性に晒しますか？
2. 組織のDNSサービスのセキュリティを保証するのが重要なのはなぜですか？
3. 次の記述は正しいですか、誤りですか？HOSTSファイルの内容は、DNSサービスが正しく構成されている限り重要ではない。
4. DNSサーバーキャッシュポイズニングとは何ですか？
5. 次の記述は正しいですか、誤りですか？DNSSECはルートサーバーから下の信頼の連鎖に依存している。
6. SASLがLDAPsに勝っている点は何ですか？
7. SNMPv2サービスを保護するにはどのような手順を踏むべきですか？

トピック11B

セキュアなアプリケーション プロトコルを実装する



対象試験範囲

2.1 エンタープライズ環境におけるセキュリティコンセプトの重要性を説明することができる。
3.1 与えられたシナリオに基づいて、セキュアなプロトコルの実装を行うことができる。

スイッチ、ルーター、アクセスポイント、セキュアなホストから成るネットワークインフラストラクチャは、サービスを実行する目的で実装されます。Web、電子メール、VoIPを可能にするアプリケーションプロトコルもセキュアな構成を必要とします。

ハイパーテキスト転送プロトコルとWebサービス

Webテクノロジーの土台をなすのはハイパーテキスト転送プロトコル(HTTP)です。HTTPにより、クライアント(通常はWebブラウザー)はHTTPサーバーに対してリソースを要求できるようになります。クライアントは適切なTCPポート(既定はポート80)を使用してHTTPサーバーに接続し、Uniform Resource Locator(URL)を使用してリソースの要求を送信します。サーバーはその要求を受け取り、データ(またはエラーメッセージ)で応答します。

応答と要求のフォーマットはHTTPヘッダー内で定義されます。HTTPペイロードは通常、HTMLのWebページを提供するために使用されます。これは、ページがどのように表示されるべきかを記述するコード化されたタグ(HyperText Markup Language)を持つプレーンテキストファイルです。Webブラウザーはタグを解釈し、そのページに関連するテキストや他のリソース(そのHTMLページとリンクしているバイナリ画像ファイルやサウンドファイルなど)を表示することができます。

HTTPはまた、ユーザーがクライアントからサーバーにデータを送信できるフォームメカニズム(POST)などの便利な機能を備えています。通常、HTTPはステートレスプロトコルです。このことは、サーバーがセッションの途中でクライアントに関する情報を一切保持しないことを意味します。とは言え、HTTPサーバーの基本機能は、スクリプト機能やプログラマブル機能(Webアプリケーション)をサポートすることで拡張できます。さらに、サーバーはテキストファイルCookieを設定することでセッション情報を保持できます。これらのコーディング機能と、データベースとの統合により、柔軟性と対話性が向上しているものの、同時に攻撃対象領域も増えることになり、より多くの脆弱性に晒されます。

! HTTPはステートフルプロトコルであると多くの人が主張しています。HTTPのバージョン2には、より多くのステート保存機能が追加されています(blog.zamical.com/2017/05/is-http2-stateful-protocol-application.html)。

トランSPORTレイヤーセキュリティ

初期における他のTCP/IPアプリケーションプロトコル同様、HTTP通信は保護されていません。HTTPにおけるセキュリティの欠如に対処すべく、SSL(Secure Sockets Layer)がNetscapeによって1990年代に開発されました。SSLは業界で非常な好評を博し、すぐさま**TLS (Transport Layer Security : トランSPORTレイヤーセキュリティ)**という標準名で採用されるようになりました。通常、HTTPアプリケーション(HTTPsまたはHTTP Secureと呼ばれる)で使用されま

ですが、他のアプリケーションプロトコルの保護や、仮想プライベートネットワーク(VPN)ソリューションとして使用することも可能です。

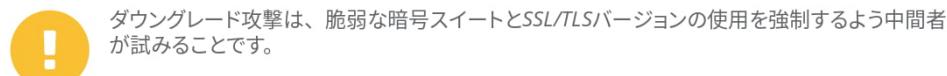
TLSを実装するにあたり、信頼できる認証局(CA)によって署名されたデジタル証明書がサーバーに割り当てられます。(クライアントがその認証機関を信頼していることを前提に) 証明書はサーバーの身元を証明し、サーバーの公開鍵と秘密鍵のペアを認証します。サーバーは鍵のペアとTLSプロトコルを使用して、双方でサポートされている暗号アルゴリズムをクライアントと取り決め、暗号化された通信セッションをやり取りします。



サーバーがクライアントを信頼できるように、クライアントに証明書をインストールすることも可能です。この方法はWebでは一般的でありませんが、相互認証を必要とするVPNと企業ネットワークには標準搭載されています。

SSL/TLSのバージョン

SSLという略称が依然として使用されていますが、安全に使用できるのはトランスポートレイヤーセキュリティだけです。サーバーはレガシークライアントをサポートできますが、これは明らかに安全度が下がります。一例を挙げると、クライアントがTLS 1.2をサポートしていない場合、TLS 1.1または1.0に、さらにはSSL 3.0にダウングレードできるよう、TLS 1.2サーバーが構成されていることもあります。



TLSバージョン1.3は2018年に承認されました。TLS 1.3の主な特徴の1つに、以前のバージョンのセキュアでない機能やアルゴリズムの使用を防ぐことで、ダウングレード攻撃を実行する余地を取り除いたことがあります。またハンドシェイクプロトコルも変更されており、メッセージの数を減らして接続を高速化しています。

暗号スイート

暗号スイートはクライアントとサーバーの両方がサポートしているアルゴリズムであり、プロトコルが必要とするさまざまな暗号化やハッシュ化を行います。TLS 1.3の登場前、暗号スイートは次の形式で記述されました。

ECDHE-RSA-AES128-GCM-SHA256

これは、サーバーが楕円曲線ディ菲ー・ヘルマンエフェメラル（一時的）モード（セッション鍵の共有用）、RSA署名、128ビットAES-GCM (Galois Counter Mode)（対称型一括暗号化用）、および256ビットSHA (HMAC機能用) を使用できることを意味します。サーバーが選択するスイートは、それがサポートする暗号アルゴリズムのリストの最初の方に列挙されています。

TLS 1.3は単純化して短縮したスイートを使用します。典型的なTLS 1.3暗号スイートは、次のような形で記述されます。

TLS_AES_256_GCM_SHA384

1.3では一時的な鍵共有だけがサポートされており、署名のタイプは証明書の中で提供されるため、この暗号スイートは一括暗号化鍵の強度と動作モード(AES_256_GCM)、および新型のハッシュ鍵導出関数(HKDF)内で用いられる暗号ハッシュアルゴリズム(SHA384)だけを列挙しています。HKDFは、D-H鍵合意によって確立された共有共通鍵を使用して、対称セッション鍵を導出するメカニズムです。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.106	172.217.20.132	TCP	66	53476 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
2	0.016952	172.217.20.132	192.168.0.106	TCP	66	443 → 53476 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0
3	0.017028	192.168.0.106	172.217.20.132	TCP	54	53476 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
4	0.018272	192.168.0.106	172.217.20.132	TLSv1.3	688	Client Hello
5	0.036762	172.217.20.132	192.168.0.106	TCP	60	443 → 53476 [ACK] Seq=1 Ack=635 Win=62208 Len=0
6	0.036763	172.217.20.132	192.168.0.106	TLSv1.3	266	Server Hello, Change Cipher Spec, Application Data
7	0.037274	192.168.0.106	172.217.20.132	TLSv1.3	118	Change Cipher Spec, Application Data
8	0.038669	192.168.0.106	172.217.20.132	TLSv1.3	224	Application Data

> Frame 6: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits) on interface \Device\NPF_{DC478856-D898-4

> Ethernet II, Src: Tp-LinkT_cf:ea:cb (60:e3:27:cfa:ea:cb), Dst: Tp-LinkT_15:af:e4 (c4:e9:84:15:af:e4)

> Internet Protocol Version 4, Src: 172.217.20.132, Dst: 192.168.0.106

> Transmission Control Protocol, Src Port: 443, Dst Port: 53476, Seq: 1, Ack: 635, Len: 212

▼ Transport Layer Security

 ▼ TLSv1.3 Record Layer: Handshake Protocol: Server Hello

 Content Type: Handshake (22)

 Version: TLS 1.2 (0x0303)

 Length: 128

 ▼ Handshake Protocol: Server Hello

 Handshake Type: Server Hello (2)

 Length: 124

 Version: TLS 1.2 (0x0303)

 Random: dba516a7b5f5b3d4f95453c6bbdf85d73a1db4632640372...

 Session ID Length: 32

 Session ID: 011fa8811607e422d8a3d92ecdd135e6da77498d8b64f75d...

 Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)

 Compression Method: null (0)

 Extensions Length: 52

 > Extension: pre_shared_key (len=2)

 > Extension: key_share (len=36)

 ▼ Extension: supported_versions (len=2)

 Type: supported_versions (43)

 Length: 2

 Supported Version: TLS 1.3 (0x0304)

Wiresharkパケットキャプチャ内のTLSハンドシェイクを表示しています。TLS 1.3と、短縮した暗号スイートの1つ(TLS_AES_256_GCM_SHA384)をこの接続に使用していることに注意してください。

APIの検討事項

現在、HTTPが静的なWebページのサービスに使用されることは少なくなり、Webアプリケーション（多くの場合クラウド製品の一部）の作成に使われることが多くなっています。企業はパブリックWebアプリケーションをインターネットとプライベートネットワークの両方で使用しています。Webアプリケーションの構成と管理は主に**API (Application Programming Interface)**を介して行います。例えば、アプリケーションが次のURL経由でユーザー アカウントを作成するのを許可しているとします。

```
https://example.foo/api/users?api_key=123456
```

デベロッパーは、必要なパラメータを（多くの場合JavaScript Object Notation [JSON] で）リクエストボディにコーディングした上で、POSTメソッドを使用してデータをこのURLに送信します。

```
POST /api/users HTTP/1.1
Content-Type: application/json
{
  "user": {
    "name": "James",
    "email": "jpengelly@comptia.org"
  }
}
```

これらのAPIはトークンまたは秘密鍵を介して許可されます。APIに関する秘密情報はさまざまに侵害やデータ詐取を実行するために広く用いられてきたので、それらを有効に管理することが現代のネットワークにおける主要な検討事項になっています。例えば、URLに鍵を入れること

は、深刻な暴露のリスクを引き起こします。APIはSAMLやOAuthなど、よりセキュアな認証と認可の方式を使用できますが、そこにもやはり秘密管理の要件が必要となります。APIに関するもう1つの検討事項として、使用状況を監視することで、許可されたエンドポイントだけがトランザクションを行っていることを確認しなければならないというものがあります。

サブスクリプションサービス

従業員がありとあらゆるサブスクリプションサービスへのアクセスを必要とすることもあります。次はその例です。

- ・ 市場情報や財務情報。
- ・ セキュリティ脅威に関するインテリジェンスと情報。
- ・ さまざまなフォーマット（電子ブックやビデオなど）の参考資料と教材。
- ・ 永久ライセンスでなくサブスクリプションとして購入されるソフトウェアアプリケーションとクラウドサービス。

この種のコンテンツの大半は、セキュアなWebサイトまたはクラウドアプリケーションによって提供されます。そうしたサービスへのエンタープライズシングルサインオン(SSO)アクセスのために、認証メカニズムを設定する必要があるかもしれません。

サブスクリプションのもう1つの使用例としてWebフィードがあり、更新された記事またはニュース項目がクライアントもしくはブラウザに送信されます。WebフィードはRSS (Really Simple Syndication)またはAtomフォーマットを土台としていますが、いずれもXMLを用いており、フィードが提供する1つ1つのドキュメントをマークアップしています。そうしたフィードは**XMLインジェクション**型の攻撃に対して脆弱である可能性があり、脅威アクターが悪意のあるリンクを表示させたり、さらにはファイルシステムとやり取りを行ったりすることが可能になります (<https://mikeknoop.com/lxml-xxe-exploit>)。



サブスクリプションサービスが、ネットワークおよびセキュリティのコンポーネントと手順のアウソーシングを意味することもあります。またエンタープライズクラウドアプリケーションがサブスクリプションで使用されることもあり、その場合はアクセスプローラーによって仲介されます。

ファイル転送サービス

ネットワーク間でファイルを転送する多数の方法があります。ネットワークオペレーティングシステムは共有フォルダとファイルをホストでき、ローカルネットワークまたはリモートアクセス経由で（例えばVPNを介して）それらをコピーしたり、アクセスしたりするのを可能にしています。メールアプリやメッセージアプリは添付の形でファイルを送信できます。HTTPはファイルのダウンロード（およびさまざまなスクリプトメカニズムを介したアップロード）をサポートしています。またピアツーピアのファイル共有サービスも存在します。そうした新しいプロトコルやサービスの可用性にもかかわらず、ファイル転送プロトコル（FTP）は効率的で、プラットフォームをまたいだ幅広いサポートがあるために、依然として大きな人気があります。

ファイル転送プロトコル

通常、**FTP (File Transfer Protocol : ファイル転送プロトコル)** サーバーはいくつかのパブリックディレクトリ、ホストしているファイル、ユーザー アカウントで構成されます。また大半のHTTPサーバーはFTPサーバーとしても機能し、Webサーバーをインストールする際にFTPのサービス、アカウント、ディレクトリがデフォルトでインストールされ、有効化されます。FTPはファイル添付やHTTPファイル転送よりも効率的ですが、セキュリティメカニズムがありません。すべての認証とデータ転送は平文で通信され、FTPトラフィックを傍受すれば認証情報を簡単に引き出せます。



ユーザーが自分のPCに許可されていないサーバー（不正なサーバー）をインストールしていないことをチェックする必要があります。例えば、HTTP、FTP、SMTPサーバーを含むIISのバージョンがWindowsのクライアントバージョンと共に出荷されていますが、これはデフォルトではインストールされていません。

SSH FTP (SFTP)とFTP Over SSL (FTPS)

SSH FTP (SFTP)は、クライアントとサーバー間の認証とデータ転送を暗号化することで、FTPが持つ機密性と完全性の問題に対処しています。SFTPにおいては、TCPポート22でSecure Shell (SSH)を使用することにより、クライアントとサーバーの間でセキュアなリンクが作成されます。そして一般的なFTPコマンドやデータ転送を、盗聴や中間者攻撃のリスクがない状況で、セキュアなリンク経由で送信できます。このソリューションにはSFTPをサポートするSSHサーバーとSFTPクライアントソフトウェアが必要になります。

FTPを保護するもう1つの方法としては、接続セキュリティプロトコルSSL/TLSの使用があります。これを行うには2つの方法があります。

- 明示的なTLS (FTPS) – AUTH TLSコマンドを使用して、ポート21で確立されたセキュアでない接続をセキュアなものにアップグレードします。これは認証情報を保護します。実際のファイル転送に用いられるデータ接続を (PROTコマンドを用いて) 暗号化することもできます。
- **暗黙のTLS (FTPS)** – FTPコマンドを交換する前に、SSL/TLSトンネルを取り決めます。このモードはコントロール接続用のセキュアポート990を使用します。

クライアントとサーバーの間にファイアウォールがある場合、FTPSの構成は難しくなります。結果として、通常はFTPSの方が好まれています。

メールサービス

メールサービスでは2種類のプロトコルが用いられます。

- **簡易メール転送プロトコル(SMTP)**は、メールがあるシステムから別のシステムへどう送信されるかを規定します。
- メールボックスプロトコルはユーザー宛てのメッセージを保存し、クライアントコンピューターにダウンロードしたり、サーバー上で管理したりできるようにします。

セキュアSMTP (SMTPL)

メッセージを配達するにあたり、送信側のSMTPサーバーはメールアドレスのドメイン名部分を使用して、受信側のSMTPサーバーのIPアドレスを見つけます。そのドメインのSMTPサーバーはMail Exchanger (MX)レコードを用いてDNSに登録されます。

SMTP通信はTLSを用いて保護することができます。これは、証明書を持つHTTPSと同じように、SMTPサーバー上で機能します。SMTPによるTLSの使用には2つの方法があります。

- STARTTLS – 既存のセキュアでない接続をアップグレードしてTLSを使用するコマンドです。これは明示的TLSまたは日和見TLSとも呼ばれます。
- SMTPL – SMTPコマンド (HELOなど) が交換される前にセキュアな接続を確立します。これは暗黙のTLSとも呼ばれます。

一般的に、STARTTLSの方がSMTPLよりも幅広く実装されています。典型的なSMTP構成では、次のポートとセキュアなサービスが使用されます。

- ポート25 – (SMTPサーバーまたはメッセージ転送エージェント [MTA] 間の) メッセージの中継に用いられます。両方のサーバーがセキュリティを必要とし、それをサポートしている場合、STARTTLSコマンドを使用してセキュアな接続をセットアップすることができます。
- ポート587 – SMTPサーバーによって配達されるメッセージを送信するために、メールクライアント (メッセージ送信エージェント [MSA]) によって使用されます。ポート587をサポートするよう構成されたサーバーは、メッセージを送信する前にSTARTTLSを使用し、認証を要求しなければなりません。
- ポート465 – プロバイダーやメールクライアントの中にはこのポートを使い、暗黙のTLS (SMTPL)でメッセージを送信するものもありますが、現在その使用は標準文書によって非推奨とされています。

セキュアPOP (POP3S)

ポストオフィスプロトコルv3 (POP3)は、サーバー上のSMTPによって配達されるメッセージの保管を目的としたメールボックスプロトコルです。クライアントがメールボックスに接続すると、POP3は受信側のメールクライアントにメッセージをダウンロードします。

```
GNU nano 2.2.2          File: /etc/dovecot/dovecot.conf      Modified

protocols = imap imaps
#protocols = none

# A space separated list of IP or host addresses where to listen in for
# connections. "*" listens in all IPv4 interfaces. "[::]" listens in all IPv6
# interfaces. Use "*", "[::]" for listening both IPv4 and IPv6.
#
# If you want to specify ports for each service, you will need to configure
# these settings inside the protocol imap/pop3/managesieve { ... } section,
# so you can specify different ports for IMAP/POP3/MANAGESIEVE. For example:
protocol imap {
    listen = *:143
    ssl_listen = *:943
}
protocol pop3 {
    listen = *:10100
    ...
}
protocol managesieve {
    listen = *:12000
    ...
}
listen = *

# Disable LOGIN command and all other plaintext authentications unless
[ Read 1280 lines ]
[G] Get Help [O] WriteOut [R] Read File ^Y Prev Page [K] Cut Text [C] Cur Pos
^X Exit [J] Justify [W] Where Is [N] Next Page [U] UnCut Text [T] To Spell
```

サーバー上でメールボックスアクセスプロトコルを構成する。

Microsoft OutlookやMozilla ThunderbirdなどのPOP3クライアントアプリケーションは、ポート110でPOP3サーバーへのTCP接続を確立します。ユーザーは（ユーザー名とパスワードによって）認証され、そのユーザーのメールボックスの中身がローカルPCにダウンロードされてそこで処理されます。POP3Sはこのプロトコルの保護されたバージョンであり、デフォルトではTCPポート995で動作します。

セキュアIMAP (IMAPS)

POP3と比較した場合、**インターネットメッセージ アクセスプロトコルv4 (IMAP4)**はサーバーへの永続的な接続と、複数のクライアントによる同じメールボックスへの同時接続をサポートしています。また、クライアントはサーバー上でメールフォルダーを管理できます。クライアントはTCPポート143でIMAPに接続します。それらのクライアントは認証の後に、指定されたフォルダーからメッセージを読み出します。他のメールプロトコルと同じく、SSL/TLSトンネルを確立することで接続を保護することができます。IMAPSのデフォルトポートはTCPポート993です。

Secure/Multipurpose Internet Mail Extensions

接続のセキュリティは、電子メールアカウントの侵害や電子メールのなりすましを防ぐのに大いに役立ちますが、エンドツーエンドの暗号化は通常保証されません。結果として、メッセージごとに認証と機密性を適用する必要が依然あります。それを行う手段の1つが、**Secure/Multipurpose Internet Mail Extensions (S/MIME)**と呼ばれるものです。S/MIMEを使用するにあたっては、ユーザーの公開鍵の正当性を証明するためにCAによって署名されたデジタル証明書がユーザーに発行されます。公開鍵は、ユーザーが秘密にしている秘密鍵とペアになっています。セキュアなメールの交換を確立するにあたり、両方のユーザーは次のようにS/MIMEと交換用の証明書を使用する必要があります。

1. アリスがボブに、彼女の公開鍵と正当なデジタルID（メールアドレス）を含むデジタル証明書を送信します。アリスは自分の秘密鍵を使ってこのメッセージに署名します。
2. ボブは証明書の中の公開鍵を使って、彼女の署名と、デジタル証明書とデジタルIDを検証するCA（またはCAの連鎖）の署名を復号し、アリスと彼女の電子メールアドレスを信頼できるかどうかを判断します。
3. ボブは自分のデジタル証明書と公開鍵を用いて応答し、アリスは同じプロセスに従ってボブを信頼するかどうかを判断します。
4. この段階でアリスとボブは、それぞれが持っている信頼できる証明書の保存場所に、互いの証明書を有しています。
5. アリスがボブに秘密のメッセージを送信する場合、彼女はメッセージのハッシュを作成し、自分の秘密鍵でそのハッシュに署名します。次にアリスは、ボブの公開鍵を用いてメッセージ、ハッシュ、そして彼女の公開鍵を暗号化し、S/MIMEの添付という形でこのデータと共にメッセージをボブに送信します。
6. ボブはこのメッセージを受信し、彼の秘密鍵を使って添付内容を復号します。ボブはアリスの公開鍵を使ってメッセージを復号し、彼女のハッシュ値と彼が作成したハッシュ値を比較することで、そのメッセージの署名と完全性を検証します。

ボイスサービスとビデオサービス

VoIP (Voice over IP)、Web会議、ビデオ遠隔会議(VTC)ソリューションは、ビジネス向けの通信を提供する標準的な手段となっています。これらのアプリケーションに共通する主な課題として、リアルタイムのデータが送信されるので、異なるネットワーク上のホスト間にポイントツーポイントのリンクを作成しなければならない、ということが挙げられます。

インターネットの電話・ビデオ会議の実装には、それ自体が持つセキュリティ上の懸念が数多く残ります。通信メディアネットワークインフラストラクチャの各部分を評価し、脅威と脆弱性を突き止める必要があります。これにはプロトコル、サーバー、ハンドセット、ソフトウェアが含まれます。リアルタイムサービスをサポートするよう設計されたプロトコルは、次の機能の1つまたは複数をカバーしています。

- セッション管理 – 通信セッションを確立、管理、終了するために使用されます。これはユーザーの発見（ネットワーク上でユーザーを見つける）、状況の広告（ユーザーが通話を受信する準備ができているかどうか）、セッションパラメータの取り決め（オーディオ/ビデオの使用など）、セッションの管理と終了といったタスクを処理します。
- データ通信 – 実際の映像または音声情報の配送を処理します。
- サービスの品質(QoS) – QoSシステムへの接続に関する情報を提供し、廃棄パケット、遅延、またはジッターなどの問題が音声または映像通信にないことを保証します。

SIP (Session Initiation Protocol)は、最も幅広く使用されているセッション管理プロトコルの1つです。SIPのエンドポイントは、IP対応のハンドセットや、クライアントとサーバーのWeb会議ソフトウェアなど、エンドユーザーのデバイス（ユーザーエージェントとも呼ばれます）です。デバイス、会議、または電話の各ユーザーには、sip:bob.dobbs@comptia.orgなど、SIP Uniform Resource Indicator (URI)という名の独自のSIPアドレスが割り当てられます。

SIPエンドポイントはピアツーピアアーキテクチャの中で通信を直接確立できますが、仲介サーバーやディレクトリサーバーを用いる方が一般的です。また、SIPネットワークはゲートウェイや構内交換機(PBX)アプライアンスを用いて、VoIPネットワークと外部の電話や携帯電話ネットワーク間のインターフェイスを提供することもあります。

SIPがセッション管理機能を提供する一方、リアルタイムデータの実際の配送では別のプロトコルが使用されます。その主要なものとして**RTP (Real-time Transport Protocol)**があります。

脅威アクターは暗号化されていない音声通信や映像通信を悪用することで、パスワードやクレジットカード情報を傍受しようと試みるかもしれません。強力な相互認証がない場合、接続は中間者攻撃に対しても脆弱です。