NEC

WebSAM

Network Flow Analyzer 3.2

リリースメモ

WebSAM

著作権

輸出時の注意

本製品を輸出する場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問合せください。

商標

- 本製品には、Visigoth Software Society (http://www.visigoths.org/) によって開発されたソフトウェアが含まれています。
- そのほかの会社名ならびに商標名は各社の商標または登録商標です。
- 本文中では™や ® は明記していません。

はじめに

このたびは、WebSAM Network Flow Analyzer 3.2 (以降、NFA と略記します) をお買い求めいただき、誠にありがとうございます。NFA では、ネットワークを流れる通信のフロー情報を分析することで、様々な通信の状況を可視化することができます。

本書では、NFA のリリース項目、および、NFA のインストールメディアの収録内容について説明しています。NFA をご使用になる前に本書の内容を確認してください。

本書の表記規則

本書では、注意すべき事項や補足事項について、以下の表記を用います。

表　注意補足事項の表記

表記	説明
⚠ **注意**	製品機能の設定、操作を行う上で守らなければならない事柄や特に注意すべき点を示します。
ヒント	知っておくと役に立つ便利な情報を示します。

本書では、以下の表記規則に従って記述しています。

表　表記規則

表記	説明	例
[]	ダイアログ、タブ、メニュー、項目名、ボタンなどの画面要素を示します。	[**ダッシュボート**]タブ、[**OK**]ボタン
<*userinput*>	ユーザー環境により変化する項目、および入力値を示します。	<*%インストールディレクトリ%*>、<*filepath*>
`configuration file`	設定ファイルの記述内容を示します。	以下の値を設定します。 `port = 27120`
`command line`	コマンドライン操作を示します。	以下のコマンドを実行します。 `$ rpm -q nec-nfa-controller`

本書では、以下の略称を用いて記述しています。

表　略称表現

正式表記	略称表現
WebSAM Network Flow Analyzer	NFA
WebSAM Integrated Management Server	IMS
WebSAM NetvisorPro V	NetvisorPro
WebSAM Network Flow Analyzer Security Monitoring ライセンス	Security Monitoring ライセンス

本製品は、デフォルトでは、以下のディレクトリにインストールします。

デフォルトのインストール先:

```
/opt/nec/nfa
```

本書では、上記のインストール先を*<%インストールディレクトリ%>*と記述します。インストール先を変更している場合は、適宜読み替えてください。

インストールの際に、本製品で管理するデータの格納先をインストール先とは異なるディレクトリに設定することができます。本書では、この場合のデータの格納先を*<%データディレクトリ%>*と記述します。インストール先とデータ格納先を分離していない場合は、*<%データディレクトリ%>*と*<%インストールディレクトリ%>*は、同じディレクトリを指します。

目次

第 1 章
製品概要

NFA の製品概要について説明します。

目次

1.1　製品の特長

NFA では、ネットワークを流れる通信のフロー情報を、直感的で簡単な操作で分析していき、通信状況を様々な視点で可視化することができます。

NFA は、どこから、どこ宛に、何の通信が、どれだけ行われているのか を細かく分析、表示することで、ネットワークの安定運用をサポートします。

フロー情報(NetFlow、IPFIX、sFlow)から通信状況を詳細に分析

ネットワークの通信状況を調べる方法として、一般的に SNMP が多く用いられています。しかし、SNMP では、スイッチやルーターの各インターフェイスを流れる通信量を調べることはできても、その通信量の内訳を調べることは困難です。

NFA では、SNMP ではなく、フロー情報(NetFlow、IPFIX、sFlow)を用いて通信状況を分析します。フロー情報を用いた分析により、SNMP では調べることができなかった、どこから、どこ宛に何の通信がどれだけ行われているのかの通信量の内訳を細かく調べることが可能です。通信状況を詳細に把握することで、ネットワーク障害の原因調査やキャパシティ管理業務を効率的に行えるようになります。

簡単な操作でドリルダウン分析が可能

NFA では、画面上のグラフ、一覧の情報をクリック 1 つで、簡単に絞り込んでいくことができます。

例えば、以下のように、画面に表示した情報に対し、直感的で簡単な操作を行っていくことで、より細かな通信状況を即座に確認していくことができます。

操作例:

1. 各インターフェイスを流れる通信量の表示から、特定のインターフェイス(仮に Ethernet1/1)を選択します。

   (選択した Ethernet1/1 を流れる通信の表示に絞り込まれます。)

2. 各アプリケーションの通信量の表示から特定のアプリケーション(仮に http)を選択します。

3. Ethernet1/1 を流れる http 通信量に関する分析結果が表示されます。

表示内容の自由なカスタマイズ機能を提供

NFA では、可視性の向上を図るために表示内容を自由にカスタマイズすることができます。

例えば、以下のように、運用環境に合わせて、表示、分析のカスタマイズを行っていくことで、ネットワークの状況を正確に把握できるようになります。

カスタマイズ例:

- NFA にログインするユーザー毎に、ダッシュボード(メイン画面)で表示するグラフや一覧の内容を定義し、運用することができます。
- 独自の業務アプリケーション通信の定義や IP アドレスの範囲指定による部門の定義を行うことで、分析結果をより分かり易く表現することができます。

1.2　機能概要

NFA が提供する機能概要について説明します。

ダッシュボード

- NFA にログインしたユーザーが担当するネットワーク範囲について、現在の通信状況やイベント発生状況をリアルタイムに表示します。
- 表示するすべての分析結果を CSV ファイル形式で外部出力することができます。
- グラフや一覧を表示する構成要素である[**ウィジェット**]をドラッグ&ドロップの操作で自由に配置でき、ユーザー毎の運用に合わせたダッシュボード定義を簡単に作成することができます。

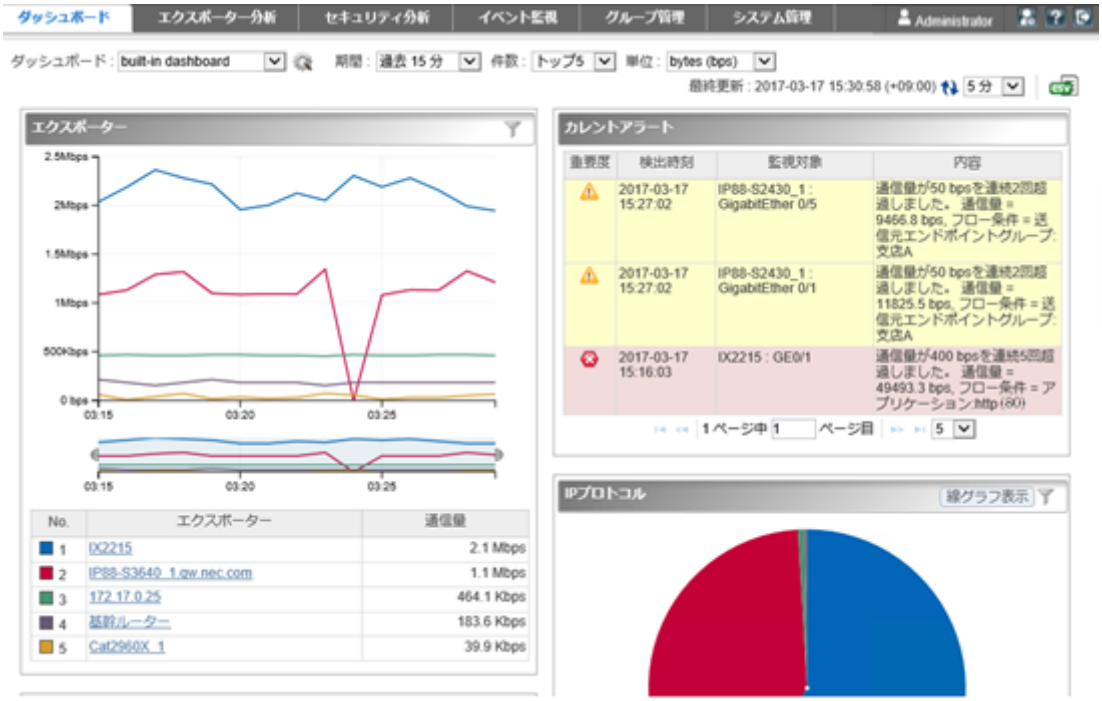

図 1-1　ダッシュボード表示

エクスポーター分析

- フロー情報を送信してくるエクスポーターやそのインターフェイスを絞りこんで、詳細な通信状況を分析することができます。

- 現在の通信状況だけではなく、過去の通信状況も分析することができ、中長期的な通信状況の変化の推移を確認することができます。
- ダッシュボード画面と同様に、各分析結果を CSV ファイル形式で外部出力することができます。

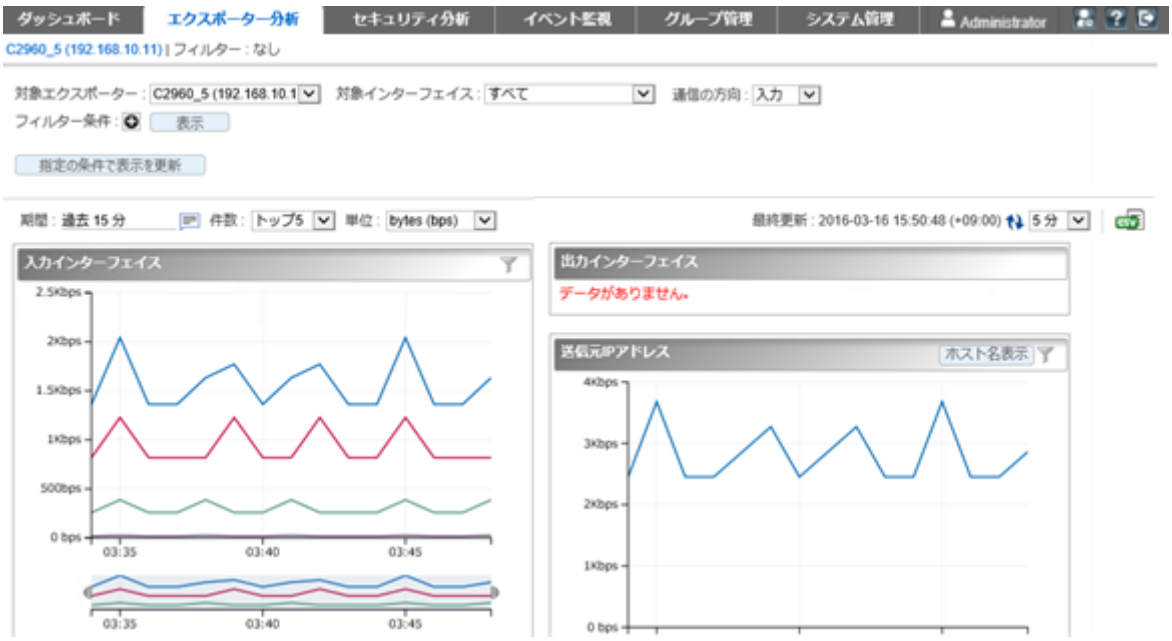

図 1-2　エクスポーター分析

セキュリティ分析

- 受信したフロー情報をセキュリティの観点で分析・監視し、DoS/DDoS やスキャンの攻撃の疑いを検知することができます。
- 検知の履歴はイベントとして確認することができ、SNMP トラップ形式で別の管理システムに送信することができます。
- 本機能を利用するためには Security Monitoring ライセンスが必要です。

図 1-3　インシデント履歴

イベント監視

- 送信元や宛先の IP アドレス、アプリケーションなどの条件で絞り込んだ通信量に対し、しきい値監視を行うことができます。
- しきい値超過、回復に関するイベントの発生履歴を一覧で表示します。ダッシュボード画面にカレントアラートウィジェットを配置した場合は、現在のイベントの発生状況をダッシュボード画面で見ることができます。
- しきい値超過、回復のイベントは、SNMP トラップ形式で、別の管理システムに送信することができます。

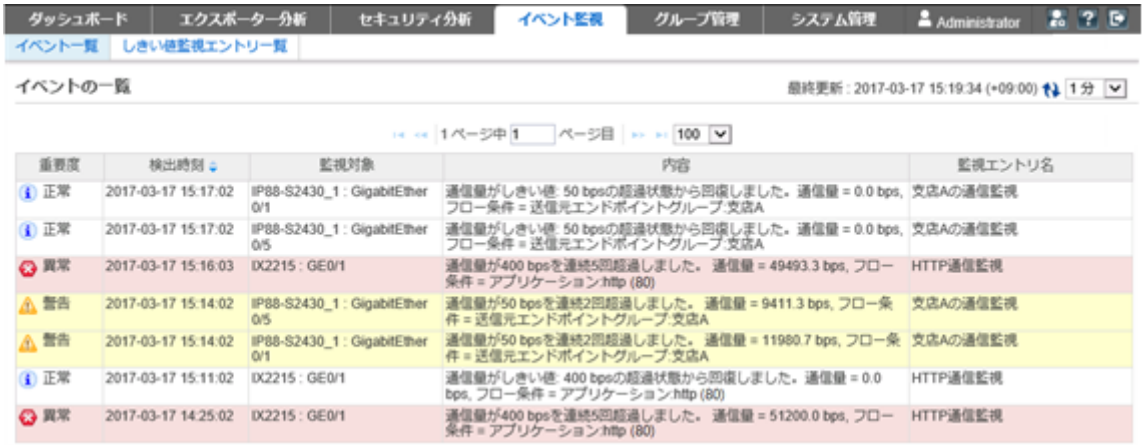

図 1-4　イベント一覧

グループ管理

- 通信のエンドポイント(送信元、または宛先)である複数の IP アドレスまたはネットワークアドレスを部門単位などでグルーピングすることで、グループ単位での通信量の分析を行うことができます。
- LAG(Link Aggregation)を構成する複数のインターフェイスをグルーピングすることで、1 つの LAG インターフェイスとして通信量を分析することができます。

図 1-5　エンドポイントグループ一覧

システム管理

- 通信状況の分析で利用するアプリケーションの定義を行うことができます。アプリケーションの定義は、IP プロトコルとポート番号の組み合わせの情報に送信元、または、宛先にあたる IP アドレスを組み合わせることで、細分化したアプリケーション定義を行うことができます。
- フロー情報を送信するエクスポーターやそのインターフェイスの情報、ライセンスの割り当て状況を一覧で管理することができます。
- NFA にログインするユーザーのパスワードやデフォルトで表示するダッシュボードの定義の情報を管理することができます。

図 1-6　アプリケーション定義

第 2 章
動作環境

NFA の動作環境について説明します。

目次

2.1　システム構成

NFA のシステム構成について説明します。

NFA のシステム構成

NFA の運用環境は、「図 2-1 システム構成図（8 ページ）」に示した通り、NFA をインストールしたサーバー(NFA サーバー)、および、NFA の利用者の端末のほか、エクスポーター、エンドポイントで構成されます。

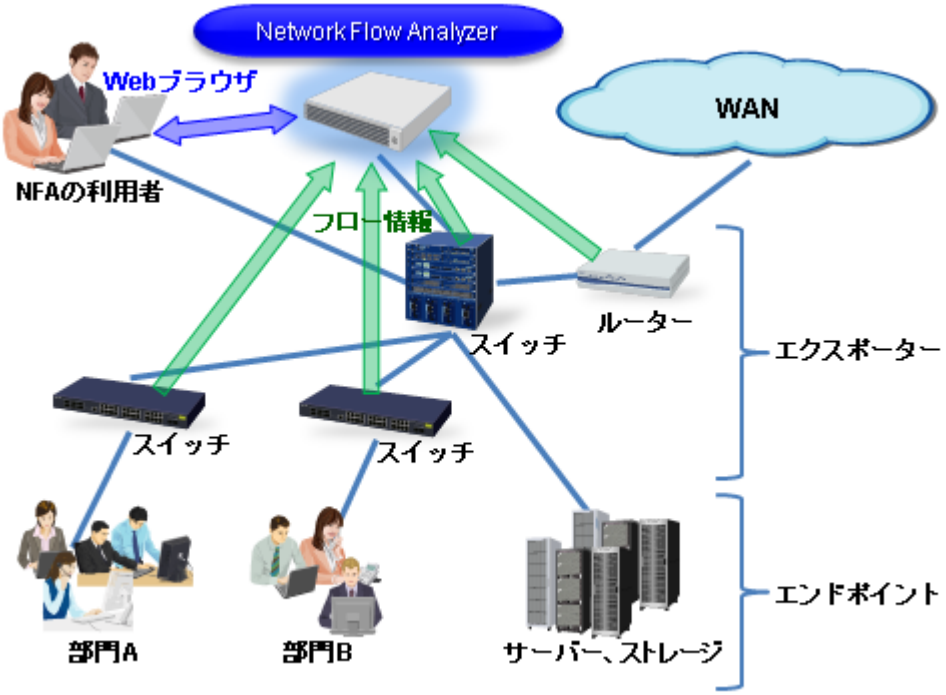

図 2-1　システム構成図

NFA は、フロー情報を受信・蓄積するフローコレクターとしての役割と、蓄積したフロー情報から通信状況を分析するフローアナライザーとしての役割の 2 つを持ちます。また、NFA の利用者向けの画面を提供する Web サーバーの機能も内蔵しています。NFA では、フローコレクター部分を 「コレクター」 (collector)、フローアナライザー部分と Web サーバーを合わせて 「コントローラー」 (controller) と呼びます。

NFA の利用者は、手元にある端末から Web ブラウザーを起動して、NFA の Web コンソールに接続します。

ヒント

- NFA では、ネットワークに接続し、通信を行う端末やサーバーなどの機器のことを総称してエンドポイントと呼んでいます。
- エンドポイント間の通信内容をフロー情報に変換し、NFA に送信することができるスイッチやルーターなどの機器のことを総称してエクスポーターと呼んでいます。

IMS コンポーネント利用時のシステム構成

IMS コンポーネントを利用することで、複数配置した NFA の統合運用や、NFA と NetvisorPro との統合運用が可能になります。統合運用時のシステム構成例を「図 2-2 統合運用時のシステム構成例（9 ページ）」に示します。

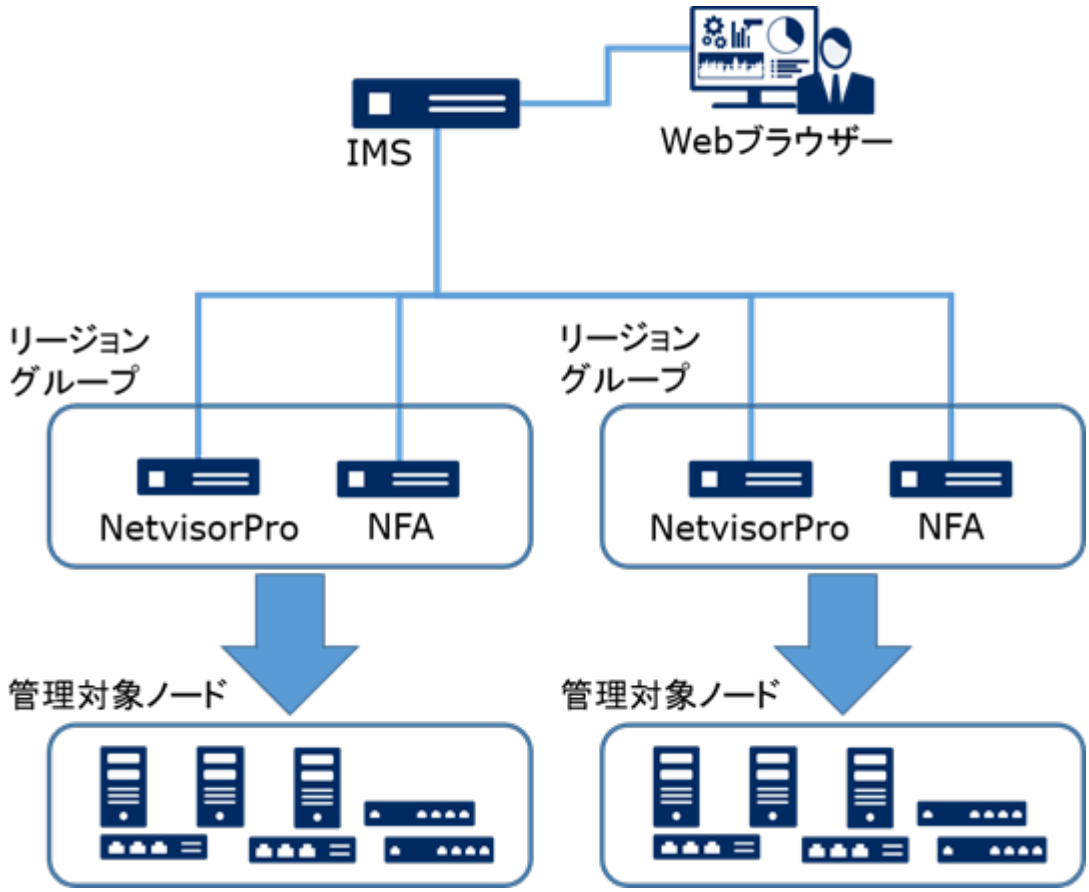

図 2-2　統合運用時のシステム構成例

「図 2-2 統合運用時のシステム構成例（9 ページ）」に示すように、同一ノード(エクスポーター)を管理する NFA と NetvisorPro は、リージョンというグループでグルーピングします。IMS コンポーネントの Web コンソールでは、同一リージョングループ内の各製品が管理する同一ノード(エクスポーター)の情報を統合して表示します。

ヒント

NFA と IMS コンポーネントとを同じサーバーにインストールするができます。ただし、この場合、操作に対する応答が遅いなどの問題が発生する可能性があります。十分に検証した上で、運用を開始してください。また、可能な限り、別のサーバーに分散してインストールする構成を推奨します。

2.2　システム要件

NFA の動作に必要なシステム要件、および、サポート環境について以下に示します。

表 2-1　サーバーのシステム要件

項目	内容
CPU	Intel クアッドコア Xeon 以上、または、同等の互換プロセッサを推奨 注 1
システムメモリ	最低 4GB 以上 (16GB 以上を推奨) 注 1
ディスク容量	インストールディレクトリ: 5GB 以上
	データディレクトリ: 最低 100GB 以上 注 2 注 3
OS	• Red Hat Enterprise Linux 9 (x86_64) 注 4 注 6 (9.2 以上をサポート) • Red Hat Enterprise Linux 8 (x86_64) 注 5 注 6
フロープロトコル	• NetFlow (v5、v9) • IPFIX • sFlow (v4、v5) NetFlow、IPFIX はサンプリングにも対応

注

1. 仮想化環境で運用する場合、オーバーコミットの影響を受けずに、確実に指定した CPU リソース、および、メモリリソースが利用できるように、仮想化基盤の設定を行ってください。
2. 製品仕様上、ハードディスクへのアクセスが頻繁に行われます。利用環境に合わせて、「SAS 15,000rpm」などのアクセス性能の高いハードディスクを利用することを強く推奨します。また、「RAID 5」,「RAID 50」,「RAID 10」 のいずれかの構成で運用することを推奨します。
3. 仮想化環境で運用する場合、他の仮想マシンの動作の影響を受けて、ハードディスクへの十分なアクセス性能が得られない場合があります。SSD (Solid State Drive) を利用するなどして、ハードディスクへの十分なアクセス性能を確保してください。
4. 以下のパッケージをインストールする必要があります。
   - python3
   - bzip2
   - chkconfig
   - initscripts
5. 以下のパッケージをインストールする必要があります。
   - python3
   - bzip2
6. SELinux を無効（disabled）に設定する必要があります。

表 2-2　Web ブラウザーの要件

項目	内容
対応ブラウザー	Windows 上で動作する以下のブラウザー • Microsoft Edge 104 以上 • Google Chrome 104 以上
CPU	Intel Core i3 以上、または同等の互換プロセッサを推奨

項目	内容
システムメモリ	1GB 以上

ヒント

- ブラウザーに最新の修正プログラムを適用した上でご利用いただくことを推奨します。修正プログラム未適用の場合、一部機能が正常動作しない場合があります。
- ブラウザーによっては、Unicode のサロゲートペア文字が 2 文字として扱われることがあります。この場合、各入力欄に実際に入力できる文字数は少なくなります。

2.3　フローデータの管理について

NFA では、受信したフローデータをデータベースを用いて管理しています。ここでは、フローデータの管理の仕組みについて説明します。

2.3.1　フローデータの保持期間と丸め処理について

NFA では、大量のフローデータを限られたディスク容量の中で長期間保持するために、受信したフローデータを以下の「表 2-3 フローデータの粒度と保持期間（11 ページ）」で示す単位時間ごとに集約(丸め処理)し、データの粒度を変えて保持しています。また、NFA では、データの粒度ごとに保持期間を設けており、保持期間を超えたデータを破棄します。保持期間はユーザーが変更することもできます。

表 2-3　フローデータの粒度と保持期間

データの粒度(単位時間)	デフォルトの保持期間	保持期間の変更可能範囲
1 分	24 時間	2～168 時間
10 分	72 時間	12～336 時間
60 分	14 日間	4～60 日間
6 時間	60 日間	14～365 日間
24 時間	365 日間	60～1095 日間
7 日	1095 日間	365～2190 日間

フローデータの集約処理では、単位時間ごとに以下の 7 つのフローキーがすべて同一のフローデータを集約して 1 つにまとめます。

1. 送信元 IP アドレス
2. 宛先 IP アドレス
3. 送信元ポート番号
4. 宛先ポート番号
5. IP プロトコル
6. ToS バイト(DSCP)

7. 入力インターフェイス

さらに、NFA では、フローデータの蓄積に必要なディスク使用量を一定に抑えるため、上記の集約処理に加えて、以下のような処理を行います。

- 単位時間ごとに、通信量の多い上位 1,000 フローまでのデータのみを詳細な分析対象として管理します。
- 上位 1,000 フローに含まれない下位のフローデータについては、「その他」のフローとして、集約して管理します。

2.3.2　ディスク使用量の見積もり方法

受信したフローデータを蓄積、管理するために必要なディスク使用量の見積もり方法について説明します。

フローデータの蓄積、管理に必要なディスク使用量は、NFA が管理するエクスポーターの台数、および、フローの発生頻度に関係しています。また、「2.3.1　フローデータの保持期間と丸め処理について（11 ページ）」で示した通り、フローデータに対する保持期間、および単位時間ごとの最大フロー数は、NFA で規定されています。そのため、フローデータの蓄積に必要なディスク使用量の目安は、これらを踏まえた計算式から算出することができます。

⚠ 注意

- エクスポーターの台数が多い場合など、フローデータのサイズは非常に大きくなるため、ディスクの空き容量が枯渇する可能性があります。ディスクが枯渇すると、新規のフローデータが受信できない他、全体として正常に動作できなくなります。ディスク容量が枯渇しないよう、最大フロー数は、少し余裕を持たせて計算することを推奨します。
- 以下で説明する見積もり内容には、ローデータを外部出力した際に必要となるディスク容量は含まれていません。ローデータを外部出力する運用を実施する場合は、ローデータの外部出力設定を行う NFA が受信したすべてのフローデータを、集約前の状態で外部出力するための設定について説明します。の内容を参照し、ローデータの外部出力で必要となるディスク容量の見積もりも行ってください。

具体的な算出方法を以下に説明します。

1. NFA で管理するエクスポーターの台数を確認します。

   今後の運用において増加する予定があれば、最終的な管理数を明確にします。

2. フローの保持期間を確認し、ディスク容量算出で使用する係数を以下の計算式から算出します。

   保持期間係数 P = P1 × 60 + P2 × 6 + P3 × 24 + P4 × 4 + P5 + P6 ÷ 7

   - P1: 1 分粒度データの保持期間(単位：時)
   - P2: 10 分粒度データの保持期間(単位：時)
   - P3: 60 分粒度データの保持期間(単位：日)
   - P4: 6 時間粒度データの保持期間(単位：日)

- P5: 24 時間粒度データの保持期間(単位：日)
- P6: 7 日粒度データの保持期間(単位：日)

計算結果の小数点以下は切り上げてください。

保持期間がデフォルト値のままであれば、係数は 2970 となります。

ヒント

フローデータに対する保持期間の変更については、「2.3.1　フローデータの保持期間と丸め処理について（11 ページ）」を参照してください。

3. 運用環境におけるフローの発生頻度(1 分間の平均フロー数)を確認します。

   フローの発生頻度は、運用環境において 1 分間に平均何セッションの通信が発生しているのか をおおよその数値で求めます。

4. 以下の計算式にあてはめて、ディスク容量の目安を算出します。

   ディスク使用量の目安[MB] = (N + 5) × P × L × 0.000415 + A × 0.15 + 10,000 [MB]

   - N: NFA が管理するエクスポーターの台数

     手順 1 で確認した値を代入して計算します。

   - P: NFA の保持期間に影響を受ける係数

     手順 2 で確認した値を代入して計算します。

   - L: 単位時間ごとに保持する最大フロー数

     デフォルトでは、最大で上位 1,000 フローを保持するため、1,000 を指定します。

   - A: NFA が受信した 1 分間の平均フロー数

     手順 3 で確認した値を代入して計算します。

計算例

エクスポーターの台数が 50 台 、フローデータに対する保持期間・単位時間ごとの最大フロー数がデフォルト値、1 分間の平均フロー数が 600,000 フローの場合は、以下のような計算結果になります。

- N = 50
- P = 2,970 (24 × 60 + 72 × 6 + 14 × 24 + 60 × 4 + 365 + 1095 ÷ 7)
- L = 1,000
- A = 600,000
- ディスク使用量の目安 = (50 + 5) × 2,970 × 1,000 × 0.000415 + 600,000 × 0.15 + 10,000 ≒ 163.9GB

第 3 章
ドキュメント一覧

本バージョンで提供する NFA のドキュメントについて説明します。

表 3-1　NFA のドキュメント一覧

タイトル (ファイル名)	概要
WebSAM Network Flow Analyzer 3.2 リリースメモ (nfa-release.pdf)	NFA 3.2 のリリース内容を示したドキュメント(本書)です。
WebSAM Network Flow Analyzer 3.2 スタートアップガイド (nfa-startup.pdf)	NFA 3.2 のセットアップ方法を示したマニュアルです。 新規インストールや、古いバージョンからのアップグレード (バージョンアップ) の手順を記載しています。
WebSAM Network Flow Analyzer 3.2 リファレンスマニュアル (nfa-reference.pdf)	NFA 3.2 の操作マニュアルです。
WebSAM Network Flow Analyzer 3.2 オープンソースソフトウェアのライセンス条文 (nfa-oss-license.pdf)	NFA 3.2 が利用しているオープンソースソフトウェアのライセンス条文および著作権表示です。

第 4 章
リリース内容

リリース内容について説明します。

目次

4.1　バージョン 3.2 でのリリース内容

NFA 3.2 において、機能追加、修正した内容を説明します。

4.1.1　セキュリティ監視機能対応

NFA が受信した集約前のフローデータを用いて、セキュリティ観点での分析・監視を行う機能を提供します。

本機能では、監視対象のインターフェイスを通るフローの TCP フラグ値を分析します。これにより、DoS/DDoS 攻撃、および、スキャン攻撃の傾向を検知することができます。

監視精度の高いセキュリティ監視専用製品は、導入コストが高い場合が多いため、外部ネットワークとの境界部分を重点的に監視するように配置することが一般的です。これに対し、NFA のセキュリティ監視機能は、フロー情報を元にした簡易的な監視である一方、既存のエクスポーターを活用した監視であるため、低コストで、かつ、様々な箇所でのセキュリティ監視を実現することができます。

⚠ 注意

本機能を利用するためには Security Monitoring ライセンスが必要です。

4.1.2　証跡ログ対応

各ユーザーのログインの状況を証跡ログとして記録するようにしました。

証跡ログを活用することで、誰がどのような頻度でネットワークフローの状況を確認しているかを追跡調査することができます。詳細は、リファレンスマニュアルを確認してください。

4.1.3　アプリケーション定義の追加

NFA が標準で提供するアプリケーション通信を識別するためのアプリケーション定義を追加しました。

追加したアプリケーション定義は以下の通りです。

- IANA が公開している情報に基づく 23 件のアプリケーション定義

4.1.4　サポートする動作環境の追加

NFA の動作をサポートする OS の種類が増えました。

NFA では、新たに以下の動作環境をサポートします。

- OS：

  Red Hat Enterprise Linux 9 (x86_64)

(9.2 以上をサポート)

4.1.5　バージョン 3.2 における機能改善

バージョン 3.2 で改善された機能の一覧を記載します。

1. ダッシュボード定義および、エクスポーター分析画面での監視対象の指定において、インターフェイスに対する通信フローの向き(入力、出力、両方向)を明示的に指定できるようになりました。
2. Microsoft 365 定義更新用の通信でプロキシサーバーを利用する場合に、プロキシ認証(Basic または Digest 認証)を利用できるようになりました。設定方法はスタートアップガイドを参照してください。
3. NFA サービスが起動している状態で OS の DNS サーバー設定を変更した場合に、NFA サービスの再起動をせずに DNS サーバー設定を反映するためのコマンドを用意しました。

   詳細はリファレンスマニュアルの `nfa_reload_dnssetting` コマンドの説明を参照してください。

4.1.6　バージョン 3.2 における仕様変更

バージョン 3.2 で変更された仕様の一覧を記載します。

1. セキュリティ監視機能の対応に伴い、一部の画面構成を変更しました。
   - SNMP トラップの通知先の設定は、従来、しきい値監視エントリ一覧画面から遷移する SNMP トラップ通知設定画面から行っていましたが、NFA3.2 からは[**システム管理**]タブの環境設定画面から行うように変更しました。
2. IANA が公開している情報に基づいて、製品が標準で提供しているアプリケーション定義の内容を以下のように変更しました。
   - アプリケーション名: geognosisman, ポート番号: 4325, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**UDP**] に変更
   - アプリケーション名: citysearch, ポート番号: 3974, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「xk22」 に変更
   - アプリケーション名: cm, ポート番号: 5910, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ats-atn」 に変更
   - アプリケーション名: cpdlc, ポート番号: 5911, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ats-acars」 に変更
   - アプリケーション名: fis, ポート番号: 5912, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ais-met」 に変更
   - アプリケーション名: ads-c, ポート番号: 5913, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「aoc-acars」 に変更

- アプリケーション名: pando-pub, ポート番号: 7680, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ms-do」 に変更
- アプリケーション名: pando-sec, ポート番号: 8276, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ms-mcc」 に変更

3. 対応する Web ブラウザーとして、Mozilla Firefox のサポートを終了しました。

4.1.7　バージョン 3.2 における修正項目

バージョン 3.2 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
    - AdoptOpenJDK (1.8.0_332 へ更新)
    - Apache Commons IO (2.6 へ更新)
    - Apache Qpid JMS (client) (0.61.0 へ更新)
    - Apache Qpid Proton-J (0.33.10 へ更新)
    - Apache Struts2 (2.5.27 へ更新)
    - Apache Tomcat (8.5.77 へ更新)
    - fasteners (0.17.3 へ更新)
    - FreeMarker (2.3.30 へ更新)
    - Jackson (2.13.2 へ更新)
    - Javassist (3.20.0 へ更新)
    - Log4j (2.17.2 へ更新)
    - Lombok (1.18.22 へ更新)
    - Netty (4.1.72 へ更新)
    - monotonic (1.6 へ更新)
    - OGNL (3.1.29 へ更新)
    - PostgreSQL (14.2 へ更新)
    - PostgreSQL JDBC Driver (42.3.3 へ更新)
    - psycopg2 (2.9.3 へ更新)
    - pytz (2021.3 へ更新)
    - six (1.16.0 へ更新)
    - SLF4J-API (1.7.32 へ更新)
    - Spring Framework (5.3.16 へ更新)

- SQLAlchemy (1.4.32 へ更新)
- SQLite (3.38.1 へ更新)
- tzlocal (4.1 へ更新)

2. sFlow の解析において、カウンタサンプルを 1 分間受信しなかった場合に、フローサンプルが受信できていたとしてもグラフの値が 0 と表示されてしまう問題を修正しました。

4.2　バージョン 3.1 でのリリース内容

NFA 3.1 において、機能追加、修正した内容を説明します。

4.2.1　ローデータの出力対応

NFA が受信したすべてのフローデータを集約前の状態で保存､出力するための仕組みを提供します。

NFA では､フロー数の上限値の設定に従い､通信量の少ないフローデータを 1 つに集約してデータベースに記録します。そのため、通信量の少ないフローの記録が残らず、セキュリティ問題が発生した場合などに、細かな通信の挙動を調べることができませんでした。

バージョン 3.1 では、受信したすべてのフローデータを、集約処理を行う前のローデータとして保存し、外部出力することができるようになりました。

保存したローデータの外部出力は、15 分毎に自動で行われます。エクスポーター毎に 15 分間に発生した通信フローのローデータを 10 万件/ファイルで CSV ファイルに記録し､1 つの圧縮ファイルとして出力します。また､出力ファイルの保持期間を管理し､ディスク使用量を一定に保つ仕組みも提供します。

本対応により､NFA が管理する通信フローのデータを簡易的なネットワークフォレンジックとして活用することができるようになります。

ヒント

- ローデータには、NFA の Web コンソールでは表示していない、各フローの TCP フラグの情報も含まれています。
- 本機能を利用するためには、以下の設定ファイルにおいて、一時記録用データベースへの書き込みを有効にしておく必要があります。
  - 設定ファイル:

    `<%`***データディレクトリ***`%>/collector/conf/collector.conf`
  - 設定パラメーター:

```
rawdb.switch = 1
```

- 本機能を有効にした場合、環境によりローデータの外部出力に遅延が発生する可能性があります。通常ローデータは 15 分毎に出力されますが、遅延が発生した場合、出力に 15 分以上かかる可能性があります。

  遅延の有無は、送受信フロー数、エクスポーター数及び、CPU スペック、ディスク I/O 性能に大きく影響されます。弊社検証では、以下の条件下で、ローデータの出力を確認しております。

表 4-1　ローデータの外部出力の動作実績環境(パターン 1)

項目	内容
CPU	Intel Xeon Gold6136 (3.0G GHz) * 8 コア
システムメモリ	32GB
ディスク	2.5 型 SAS (15,000rpm), RAID 1+0
OS	Red Hat Enterprise Linux 8.5 (x86_64), on VMware ESXi 6.5
送受信フロー	500 万フロー/分
エクスポーター数	80 エクスポーター x 2IF

表 4-2　ローデータの外部出力の動作実績環境(パターン 2)

項目	内容
CPU	Intel Xeon Gold6136 (3.0G GHz) * 8 コア
システムメモリ	32GB
ディスク	2.5 型 SAS (15,000rpm), RAID 1+0
OS	Red Hat Enterprise Linux 8.5 (x86_64), on VMware ESXi 6.5
送受信フロー	600 万フロー/分
エクスポーター数	500 エクスポーター x 2IF

表 4-3　ローデータの外部出力の動作実績環境(パターン 3)

項目	内容
CPU	Intel Xeon E5-2690v4 (2.6G GHz) * 4 コア
システムメモリ	6GB
ディスク	2.5 型 SAS (10,000rpm), RAID 5+0
OS	Red Hat Enterprise Linux 8.5 (x86_64), on VMware ESXi 6.5
送受信フロー	30 万フロー/分
エクスポーター数	5 エクスポーター x 4IF

4.2.2　アプリケーション定義の強化

アプリケーション通信を識別するためのアプリケーション定義の内容および処理を強化しました。

アプリケーション定義のインポート/エクスポート対応

アプリケーション定義をインポート、および、エクスポートするためのコマンド `nfa_application_conf` を提供します。

nfa_application_conf は、別環境の NFA へアプリケーション定義を移行したい場合や、アプリケーション定義の変更作業前のバックアップなどで活用することができます。

アプリケーション定義の追加

NFA が標準で提供するアプリケーション通信を識別するためのアプリケーション定義を追加しました。

追加したアプリケーション定義は以下の通りです。

- IANA が公開している情報に基づく 12 件のアプリケーション定義

4.2.3　フローレートの表示対応

監視対象となるネットワークを流れる通信フローの流量(フローレート)を確認できるようになりました。

フローレートは以下の方法で確認することができます。

- エクスポーター管理画面のエクスポーターの一覧

  直近 7 日間におけるエクスポーター全体、および、エクスポーター毎のフローレートの最大値とその発生日時の情報を表示します。

- <%*データディレクトリ*%>/collector/flowrate_log 配下

  各エクスポーターの 1 分単位のフローレートの記録を CSV ファイルに記録しています。

フローレートは、NFA の諸元と運用状況を比較確認する場合などで活用します。

ヒント

NFA では、エクスポーター 1 台あたり最大 600,000 フロー/分、全体で約 6,000,000 フロー/分のフローデータを処理することができます。 *1

4.2.4　保守ツールの提供

NFA の運用維持や障害調査を行うためのツールを提供します。

ディスク使用率の監視ツール(nfa_diskcheck コマンド)

nfa_diskcheck コマンドを利用することで、指定パスのディスク使用率を監視することができます。また、指定したしきい値を超えていた場合は、メールや Syslog で通知を行うことができます。

nfa_diskcheck コマンドは、NFA のデータディレクトリのディスク容量が十分に確保できているかを監視する際に活用します。

*1　ディスク I/O 性能などハードウェアの十分な処理性能を確保する必要があります。

ヒント

nfa_diskcheck コマンドは、cron を用いて定期実行する運用を想定しています。

ディスクの性能評価ツール(fio コマンド)

本バージョンでは、オープンソースソフトウェアのディスク性能評価ツール fio コマンドを以下の場所に配置しています。

- <%*インストールメディア内*%>/NFA/tools/fio
- <%*インストールディレクトリ*%>/collector/bin/fio

fio コマンドを利用することで、NFA のデータを記録するハードディスクの I/O 性能を測定することができます。fio コマンドは、以下の 3 つの指標で I/O 性能を数値化します。

- IOPS(Input Output Per Second):

  1 秒間の I/O 数を測定します。

- スループット:

  1 秒間に正常処理が行えるデータサイズを測定します。

- レイテンシー:

  I/O リクエストを完了させるまでにかかる時間を測定します。

fio コマンドは、NFA が利用するハードディスクの性能が十分かを確認する際に活用します。

保守用の情報採取ツール(nfatech コマンド)

nfatech コマンドを利用することで、NFA、および、OS の動作環境情報を一括して収集することができます。

nfatech コマンドは、NFA の動作において不具合が発生した場合に、NEC カスタマーサポートセンターへ送付する情報を採取する際に活用します。

4.2.5　バージョン 3.1 における機能改善

バージョン 3.1 で改善された機能の一覧を記載します。

1. しきい値監視エントリの監視対象の指定において、インターフェイスに対する通信フローの向き(入力、出力、両方向)を明示的に指定できるようになりました。
2. アプリケーション定義で指定したドメイン名に対応する IP アドレス情報をメモリに保持するだけではなく、データベースに記録することで、NFA の起動直後から、受信したフローデータに対するアプリケーション通信の識別が行えるようになりました。
3. ダッシュボード関連操作で表示するダイアログ内のメッセージにおいて、読みやすさの観点で表現を見直しました。

4.2.6　バージョン 3.1 における仕様変更

バージョン 3.1 で変更された仕様の一覧を記載します。

1. IANA が公開している情報に基づいて、製品が標準で提供しているアプリケーション定義の内容を以下のように変更しました。
    - アプリケーション名: surfpass, ポート番号: 5030, IP プロトコル: [**TCP または UDP**] の定義を削除

4.2.7　バージョン 3.1 における修正項目

バージョン 3.1 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
    - AdoptOpenJDK (1.8.0_332 へ更新)
    - Apache Commons IO (2.6 へ更新)
    - Apache Qpid JMS (client) (0.61.0 へ更新)
    - Apache Qpid Proton-J (0.33.10 へ更新)
    - Apache Struts2 (2.5.27 へ更新)
    - Apache Tomcat (8.5.77 へ更新)
    - fasteners (0.17.3 へ更新)
    - FreeMarker (2.3.30 へ更新)
    - Jackson (2.13.2 へ更新)
    - Javassist (3.20.0 へ更新)
    - Log4j (2.17.2 へ更新)
    - Lombok (1.18.22 へ更新)
    - Netty (4.1.72 へ更新)
    - monotonic (1.6 へ更新)
    - OGNL (3.1.29 へ更新)
    - PostgreSQL (14.2 へ更新)
    - PostgreSQL JDBC Driver (42.3.3 へ更新)
    - psycopg2 (2.9.3 へ更新)
    - pytz (2021.3 へ更新)
    - six (1.16.0 へ更新)
    - SLF4J-API (1.7.32 へ更新)

- Spring Framework (5.3.16 へ更新)
- SQLAlchemy (1.4.32 へ更新)
- SQLite (3.38.1 へ更新)
- tzlocal (4.1 へ更新)

2. セキュリティ脆弱性を含んでいる TLS 1.0、および、1.1 を用いた接続を許可しないように修正しました。
3. 個人設定画面、ユーザー追加画面、および、ユーザー編集画面の[**デフォルトのダッシュボード**]のプルダウンリストにおいて、ダッシュボード名の文字が切れて適切に表示されない場合がある問題を修正しました。

4.3　バージョン 3.0 でのリリース内容

NFA 3.0 において、機能追加、修正した内容を説明します。

4.3.1　ユーザー管理の強化

ユーザー管理が強化され、Web コンソールへのログイン操作のセキュリティが向上しました。

具体的には、以下の強化を実施しています。

- Web コンソールへのログインにおいて、パスワード誤りを 5 回連続で検出した場合に、当該ユーザーの情報をロックし、当該ユーザーでのログインを 10 分間行えない状態にします。
- 十分なパスワード強度を確保するために、設定するパスワードに対し以下の条件を設けました。
  - 大文字、小文字、数字、記号の中から 3 種以上の文字を含んでいること
  - パスワード変更の際に、過去 10 回分のパスワードと一致していないこと

  なお、バージョン 3.0 から、パスワードの最大文字数を 32 文字から 64 文字に拡大しています。

4.3.2　アプリケーション定義の強化

アプリケーション通信を識別するためのアプリケーション定義の内容および処理を強化しました。

ドメインによるアプリケーション通信の識別

アプリケーション通信を識別する処理において、[**ポート番号**]と[**IP プロトコル**]に加えて、[**ドメイン**]を指定することができるようにりました。[**ドメイン**]の指定においては、ワイルドカードとしてアスタリスク(*) *2 を利用することも可能です。

ドメインが同一で IP アドレスだけが定期的に変更になるシステムや、ドメインだけが公開されているクラウドサービスなどへのアプリケーション通信を識別する際に、本仕組みを活用することができます。

アプリケーション定義の高度な設定

アプリケーション定義において、従来からの設定に加え、[**高度な設定**]を有効にすることで、以下の設定が行えるようになりました。

- 通信の向きを意識したアプリケーションの識別を行うための設定
- 宛先、送信元それぞれの通信ポート番号、IP アドレスを明確に指定した識別条件の設定
- [**ポート番号**]、[**IP プロトコル**]、[**IP アドレス/ドメイン**]の条件を複数組み合わせた設定

アプリケーション定義の追加

NFA が標準で提供するアプリケーション通信を識別するためのアプリケーション定義を追加しました。

追加したアプリケーション定義は以下の通りです。

- IANA が公開している情報に基づく 265 件のアプリケーション定義
- 以下のクラウドサービス通信を識別するためのアプリケーション定義
  - Microsoft365 (Office365):
    * O365-Exchange

      Microsoft Exchange Online に関する通信
    * O365-SharePointAndOneDrive

      Microsoft SharePoint Online、および、Microsoft OneDrive for Business に関する通信
    * O365-SkypeAndTeams

      Microsoft Skype for Business Online、および、Microsoft Teams に関する通信
    * O365-Office

      Microsoft 365 Common、および、Microsoft Office Online に関する通信

*2　アスタリスク(*)を用いた指定を行う場合、対象のドメイン名は、IP アドレスからの逆引きで解決できる必要があります。

- Box [*3]
- Zoom

4.3.3　Microsoft 365 通信定義の自動更新対応

Microsoft 365 (Office 365)の通信に対するアプリケーション定義の内容を自動で更新するための仕組みを提供します。

Microsoft 365 (Office 365)が提供しているサービスのエンドポイントの IP アドレス、または、ドメイン名は、不定期に変更される場合があります。そのため、Microsoft 365 (Office 365)に対する通信を正確に識別するためには、NFA に登録しているアプリケーション定義の内容を運用の中で更新していく必要がありました。

バージョン 3.0 では、マイクロソフトが提供している REST API を利用して、Microsoft 365 (Office 365)サービスのエンドポイント情報の更新を検知し、NFA に登録しているアプリケーション定義の内容を自動的に更新します。これにより、運用の中で、Microsoft 365 (Office 365)の通信に対するアプリケーション定義の手動更新作業が不要になります。

ヒント

- 自動更新の機能を利用する場合は、NFA からマイクロソフトのサイトへの通信が可能な環境である必要があります。
- Microsoft 365 (Office 365)の通信に対する分析が不要な場合、または、NFA からマイクロソフトのサイトへの通信が行えない環境の場合は、自動更新の機能を設定ファイルの編集により無効にすることができます。

4.3.4　アプリケーション名表示の改善

NFA の運用性を向上させるため、アプリケーション通信の表示内容について改善を行いました。

以下に示す画面での表示において、アプリケーション名と共に通信ポート番号を表示します。

- ダッシュボード画面、および、エクスポーター分析画面:
  - アプリケーションウィジェットでの表示
  - CSV ファイル出力したアプリケーションウィジェットのデータ [*4]
- イベント一覧画面:
  - しきい値超過したアプリケーション通信の表示 [*5]

*3　各社固有のドメイン名宛(例：{yourcustomsubdomain}.box.com)の通信を識別する定義は含まれておりません。別途、追加して運用してください。

*4　フィルター条件のアプリケーション名には通信ポート番号は付加されません。

*5　しきい値超過の SNMP トラップの情報においてもアプリケーション名と共に通信ポート番号の情報が追加されます。

4.3.5　nfa_flow_export コマンドの改善

nfa_flow_export コマンドの操作性を改善しました。

以前のバージョンでは、-continue オプションを指定したコマンド実行をフローデータが存在しない期間に対して行った場合、CSV ファイルは出力せず、最後の出力時刻の記録更新のみを行っていました。-continue オプションで一度に出力できるデータの時間幅は、データ粒度のレベルごとに決まっているため、フローデータが存在しない期間がある場合、ユーザーは何度もコマンドを実行する必要がありました。

バージョン 3.0 では、-continue オプションを指定したコマンド実行をフローデータが存在しない期間に対して行った場合であっても、フローデータが存在する期間の最古のデータから CSV ファイルに出力するように処理を改善しました。これによって、システムメンテナンスのため、NFA を長期間停止させていた場合であっても-continue オプションを指定したコマンド実行の運用を簡単に開始することができるようになります。

4.3.6　バージョン 3.0 における仕様変更

バージョン 3.0 で変更された仕様の一覧を記載します。

1. ユーザー名の指定において、最大文字数を 32 文字から 255 文字に拡大しました。また、指定可能な文字として、アットマーク(@)を追加しました。
2. ユーザーのパスワードに対する最大文字数を 32 文字から 64 文字に拡大しました。
3. 以下に示す機能のパラメーターに対し、既存の入力制限に加え、一部の記号の入力を不可としました。

   入力不可の記号:

   \ $ " < > ? ^ ` { | } ~ = ! ' * + ;

   - ユーザー管理

     ユーザーの[**表示名**]

   - エクスポーター管理

     エクスポーターの[**表示名**]、インターフェイスの[**表示名**]

   - グループ管理

     [**IF グループ名**]、[**エンドポイントグループ名**]

   - ダッシュボード管理

     [**ダッシュボード名**]、ウィジェットの[**表示タイトル**]

   - アプリケーション定義

     [**アプリケーション名**]

   - しきい値監視エントリ

     [**エントリ名**]

4. アプリケーション定義において、製品が標準で提供する製品定義とユーザーが作成するユーザー定義を明確に区別し、管理する仕様に変更しました。これに伴い、製品定義を直接編集することはできなくなり、製品定義をコピーしてユーザー定義として編集する操作手順に変わります。

5. IANA が公開している情報に基づいて、製品が標準で提供しているアプリケーション定義の内容を以下のように変更しました。

   - アプリケーション名: imap, ポート番号: 143, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**TCP**] に変更
   - アプリケーション名: imaps, ポート番号: 993, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**TCP**] に変更
   - アプリケーション名: owamp-control, ポート番号: 861, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**TCP**] に変更
   - アプリケーション名: twamp-control, ポート番号: 862, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**TCP**] に変更
   - アプリケーション名: dbsa-lm, ポート番号: 1407, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名: tibet-server, IP プロトコル: [**TCP**] に変更
   - アプリケーション名: ibm-mqisdp, ポート番号: 1883, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「mqtt」 に変更
   - アプリケーション名: newheights, ポート番号: 2114, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ariascribe」 に変更
   - アプリケーション名: rockwell-csp1, ポート番号: 2221, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ethernet-ip-s」 に変更
   - アプリケーション名: hp-rda, ポート番号: 2371, IP プロトコル: [**TCP**] の定義において、アプリケーション名を 「RemoteDeviceAccess」 に変更
   - アプリケーション名: community, ポート番号: 2459, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「xrpl」 に変更
   - アプリケーション名: sai_sentlm, ポート番号: 2640, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「ami-control」 に変更
   - アプリケーション名: enc-eps, ポート番号: 3567, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「dof-eps」 に変更
   - アプリケーション名: enc-tunnel-sec, ポート番号: 3568, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「dof-tunnel-sec」 に変更
   - アプリケーション名: hp-dataprotect, ポート番号: 3612, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「dataprotector」 に変更
   - アプリケーション名: gmmp, ポート番号: 4183, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「cyborgnet」 に変更

- アプリケーション名: visicron-vs, ポート番号: 4307, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「trueconf」 に変更
- アプリケーション名: lisp-data, ポート番号: 4341, IP プロトコル: [**TCP または UDP**] の定義において、IP プロトコルを [**UDP**] に変更
- アプリケーション名: enc-eps-mc-sec, ポート番号: 5567, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「dof-dps-mc-sec」 に変更
- アプリケーション名: coap, ポート番号: 5683, IP プロトコル: [**UDP**] の定義において、IP プロトコルを [**TCP または UDP**] に変更
- アプリケーション名: coaps, ポート番号: 5684, IP プロトコル: [**UDP**] の定義において、IP プロトコルを [**TCP または UDP**] に変更
- アプリケーション名: dali-port, ポート番号: 5777, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「starfield-io」 に変更
- アプリケーション名: printercare-cc, ポート番号: 6716, IP プロトコル: [**TCP**] の定義において、アプリケーション名を 「princity-agent」 に変更
- アプリケーション名: smc-https, ポート番号: 6789, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名: radg, IP プロトコル: [**TCP**] に変更
- アプリケーション名: enc-tunnel, ポート番号: 8567, IP プロトコル: [**TCP または UDP**] の定義において、アプリケーション名を 「dof-tunnel」 に変更
- アプリケーション名: z-wave-s, ポート番号: 44123, IP プロトコル: [**TCP**] の定義において、アプリケーション名を 「z-wave-tunnel」 に変更
- アプリケーション名: balour, ポート番号: 4324, IP プロトコル: [**TCP または UDP**] の定義を削除

4.3.7　バージョン 3.0 における修正項目

バージョン 3.0 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
   - AdoptOpenJDK (1.8.0_292 へ更新)
   - Apache Qpid JMS (client) (0.57.0 へ更新)
   - Apache Qpid Proton-J (0.33.8 へ更新)
   - Apache Tomcat (8.5.69 へ更新)
   - Jackson (2.12.2 へ更新)
   - Log4j (2.14.1 へ更新)
   - Lombok (1.18.18 へ更新)
   - Netty (4.1.60 へ更新)

- PostgreSQL JDBC Driver (42.2.19 へ更新)
- SLF4J-API (1.7.30 へ更新)
- fasteners (0.16 へ更新)
- msgpack-python (0.5.6 へ更新)
- psycopg2 (2.8.6 へ更新)
- pytz (2021.1 へ更新)
- Spring Framework (5.3.5 へ更新)
- SQLAlchemy (1.4.17 へ更新)
- SQLite (3.35.3 へ更新)

2. OS 環境のファイル･モード作成マスク(umask)をデフォルト値から変更している場合に、NFA のインストールに失敗する場合がある問題を修正しました。

4.4　バージョン 2.2 でのリリース内容

NFA2.2 において、機能追加、修正した内容を説明します。

4.4.1　サポートする動作環境の追加

NFA の動作をサポートする OS、および、Web ブラウザーの種類が増えました。

NFA では、新たに以下の動作環境をサポートします。

- OS：

  Red Hat Enterprise Linux 8 (x86_64)

- Web ブラウザー：

  Microsoft Edge (Chromium)

4.4.2　フロー情報を取得する Web API のサポート

WebSAM Integrated Management Server (IMS)コンポーネントが提供する Web API を利用することで､NFA で管理するフローデータを他のアプリケーションで利用できるようになりました。

IMS コンポーネントが提供する Web API を利用することで､各ウィジェットで表示する同等の情報を外部利用することができます。また、併せて、Microsoft Excel で動作するサンプルプログラムも提供します。サンプルプログラムを利用することで､プログラムを作成しなくても NFA のフローデータを元にしたレポートを Microsoft Excel で作成することができます。

詳細は、「WebSAM Integrated Management Server 2.0 リリースメモ」を参照してください。

4.4.3　しきい値監視におけるフロー条件の複数指定のサポート

フローに対するしきい値監視において、フローを特定する条件を複数指定することができるようになりました。

本強化によって、例えば、特定アプリケーションの特定送信元のフローに対して、しきい値監視を行うことができます。

4.4.4　フロー情報の記録処理性能の安定化

NFA が定期的に行う内部処理の影響を受けないように、フロー情報の記録処理の仕組みを改善しました。

以前のバージョンでは、NFA が定期的に行う内部処理がフロー情報の記録処理に影響を与え、記録処理に時間がかかる場合がありました。

バージョン 2.2 では、フロー情報の記録処理の仕組みを見直すことで、NFA が定期的に行う内部処理の影響を受けないようになり、安定した処理性能を確保しました。

4.4.5　フローデータ集約(丸め処理)における基準時刻の変更機能

蓄積するフローデータを集約(丸め処理)する基準時刻を変更できるようになりました。

バージョン 2.1 以前においては、蓄積するフローデータを集約(丸め処理)する際の基準時刻は、必ず、UTC (協定世界時：Coordinated universal time)の 00:00 でした。そのため、日本国内での運用において、24 時間粒度のデータは、09:00 から翌日の 08:59 までのデータを集約したものとなっていました。

集約処理を行う基準時刻を変更することで、日本時間 (UTC+09:00) に合わせたデータの集約を行うことができます。例えば、24 時間粒度のデータにおいては、00:00 から 23:59 までのデータを集約したものにすることができます。

ヒント

本強化と関連して、NFA のインストール時に設定される集約処理の基準時刻の既定値を変更しています。詳細は、「4.4.8.1　フローデータ集約(丸め処理)の基準時刻の既定値変更（32 ページ）」を参照してください。

4.4.6　フロー情報の記録処理方式の改善

環境条件に合わせて、受信したフロー情報の記録処理の方式を切り替えることができる仕組みを追加しました。

以前のバージョンでは、1 分毎に、受信したフロー情報を以下のような処理の流れで、データベースに記録しています。

1. 1 分間に受信したすべてのフロー情報を一時記録のためのデータベースへ書き込みます。
2. 一時記録のデータベースのデータを元に、1 分間のフローデータとして集約処理(丸め処理)を実施します。
3. フロー数の上限設定の値に従い、下位のフローデータを「その他」のフローデータとして集計し、上位データと共にフローデータの管理用データベースに記録します。

定常的に受信するフロー情報が多く、かつ、NFA サーバーのディスク性能が十分ではない環境においては、「処理 1」でのデータベース書き込みに時間がかかり、この影響でメモリ使用量が増加し続ける可能性があります。メモリ使用量が増加し続けると、OS の制御によって、NFA のコレクタープロセスが強制停止される場合があります。

上記のような問題に対応するため、「処理 1」でのデータベース書き込みを停止し、メモリ上に保持するデータを元に「処理 2」の集約処理(丸め処理)を実施する新たな記録処理方式を追加しました。従来からの記録処理方式から、新たに追加した記録処理方式に切り替えることで、データベースへ書き込むデータ量を大幅に削減することができます。この効果として、ディスク性能が十分ではない環境であっても、フローデータの書き込みにかかる時間を短縮することができ、メモリ使用量が増加し続ける事象の発生を防止、または、発生頻度を軽減することができます。

4.4.7　WebSAM SystemManager G 連携対応

フローに対するしきい値監視のイベントを WebSAM SystemManager G (バージョン 10 以上)に連携できるようになりました。

WebSAM SystemManager G へのイベント連携を行うためには、WebSAM Integrated Management Server (IMS)コンポーネント (バージョン 2.0.1.8 以上) のイベントアクション機能の設定が必要になります。

詳細については、IMS コンポーネントの各種ドキュメントを参照してください。

4.4.8　バージョン 2.2 における仕様変更

バージョン 2.2 で変更された仕様について説明します。

4.4.8.1　フローデータ集約(丸め処理)の基準時刻の既定値変更

蓄積するフローデータを集約(丸め処理)する際の基準時刻において、既定値を NFA サーバーのタイムゾーンに合わせるように仕様を変更しました。

バージョン 2.1 以前においては、蓄積するフローデータを集約(丸め処理)する際の基準時刻の既定値は、UTC (協定世界時：Coordinated universal time)の 00:00 でした。

これに対し、バージョン 2.2 では、例えば、日本のタイムゾーンに設定しているサーバーに NFA をインストールした場合、集約処理の基準時刻の既定値は、日本時間 (UTC+09:00) の 00:00 となります。

ヒント

- バージョン 2.2 以降において、集約処理の基準時刻は、既定値から変更することができます。詳細は、リファレンスマニュアルを確認してください。
- 旧バージョンから最新バージョンにアップグレードした場合は、製品の既定値として、旧バージョンでの設定をそのまま引き継ぎます。そのため、バージョン 2.1 以前からアップグレードした場合の基準時刻は、UTC の 00:00 のままとなります。

  この場合、製品の既定値としては、設定を引き継ぎますが、ユーザーの設定値としては管理されません。明示的に集約処理の基準時刻の設定を行うことを推奨します。

4.4.8.2　記録対象フローの条件変更

フロー情報の記録・破棄の条件を見直しました。

以前のバージョンでは、sFlow v5 のフローパケットにおいて、出力インターフェイスの情報がなく、エクスポーター内で破棄されたことが示されている場合、sFlow v5 のプロトコルに則って、受信したフローを破棄し、記録していませんでした。

上記のフローパケットは、ルーターのミラーポートを介して、別のルーター(エクスポーター)で当該フローをキャプチャーした場合に、エクスポーターの仕様に依存して発生する場合があります。このような運用構成は、sFlow をサポートしていないルーターの通信内容を分析したい場合に用いられます。

バージョン 2.2 では、このような運用構成にも対応できるように仕様を変更しました。バージョン 2.2 では、受信したフローパケットに、出力インターフェイスの情報がなく、エクスポーター内で破棄されたことが示されていたとしても例外的に、それを破棄せず、記録します。

ヒント

NetFlow、および、IPFIX においては、上記のような場合に破棄することを示すプロトコルはありません。そのため、以前のバージョンにおいても該当するフローパケットが破棄されることはありません。

4.4.9　バージョン 2.2 における修正項目

バージョン 2.2 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
   - Apache Commons BeanUtils (1.9.4 へ更新)
   - Apache Qpid JMS (client) (0.52.0 へ更新)
   - Apache Qpid Proton-J (0.33.5 へ更新)

- Apache Tomcat (8.5.57 へ更新)
- Jackson (2.9.10 へ更新)
- Log4j (2.13.3 へ更新)
- Lombok (1.18.12 へ更新)
- Netty (4.1.50 へ更新)
- OkHttp (3.14.9 へ更新)
- PostgreSQL (9.2.24 へ更新)
- PostgreSQL JDBC Driver (42.2.13 へ更新)
- psycopg2 (2.8.5 へ更新)
- pytz (2020.1 へ更新)
- six (1.15.0 へ更新)
- Spring Framework (5.2.7 へ更新)
- SQLAlchemy (1.3.17 へ更新)
- SQLite (3.32.2 へ更新)
- tzlocal (2.1 へ更新)

2. 通信フローのエンドポイント(送信元/宛先)の IP アドレスとホスト名を管理しているデータベーステーブルのデータ件数が増加すると当該データベーステーブルへのアクセス性能が低下し、サーバーの CPU が高負荷状態になる問題を修正しました。
3. エクスポーター管理画面の[**SNMP 情報取得**]ボタンをクリックし、エクスポーターから SNMP 情報を取得した際に、エクスポーターの仕様により一部の情報取得が行えなかった場合、NFA と IMS コンポーネントとの間で構成情報の差分が生じる問題を修正しました。
4. NFA 2.2.0-7 以前において、Red Hat Enterprise Linux 8 (x86_64)へのインストールが行えない場合がある問題を修正しました。

4.5　バージョン 2.1 でのリリース内容

NFA2.1 において、機能追加、修正した内容を説明します。

4.5.1　フロー受信性能の向上

フロー受信処理の実装を見直し、フロー受信性能を向上しました。

処理並列化や一時データの蓄積方法などを変更することで、単位時間あたりにより多くのフローを受信できるようになりました。

弊社検証では、以下の条件下で、フロー受信性能の改善を確認しています。

表 4-4　フロー受信性能改善の動作実績環境

項目	内容
CPU	Intel E5-2690 v4 (2.60 GHz) * 8 コア
システムメモリ	16GB
ディスク	2.5 型 SAS (10,000rpm), RAID 5+0
OS	Red Hat Enterprise Linux 7.3 (x86_64), on VMware ESXi 6.5
フロー数 (改善前)	20,000 フロー/秒 (エクスポーター 1 台あたり最大 5,000 フロー/秒)
フロー数 (改善後)	100,000 フロー/秒 (エクスポーター 1 台あたり最大 10,000 フロー/秒)

⚠ 注意

改善の効果は、環境に応じて異なります。すべての環境で 5 倍の改善を保証するものではない点にご注意ください。

ヒント

フローの受信性能は、ディスク I/O 性能に大きく左右されます。ディスク I/O 性能のひとつの目安として、上記環境でディスク I/O ベンチマークである fio を実行した結果を以下に示します。

表 4-5　fio ベンチマーク [ブロックサイズ：4KB] の結果 (参考値)

種類	IOPS (IO 回数/s)	帯域幅 BW (MiB/s)	レイテンシー lat (usec)
random read	1,602	6.4	3,116
random write	2,669	10.4	1,867

表 4-6　fio ベンチマーク [ブロックサイズ：8KB] の結果 (参考値)

種類	IOPS (IO 回数/s)	帯域幅 BW (MiB/s)	レイテンシー lat (usec)
random read	1,596	12.5	3,123
random write	2,335	18.2	2,136

random read は以下のコマンドを実行することで計測できます。

```
# fio -filename=<%データディレクトリ%>/fio -direct=1 -rw=randread -bs=4k -size=20G
    -numjobs=5 -group_reporting -name=randomread
```

上記は、ブロックサイズを 4KB にした場合の実行例です。ブロックサイズを 8KB にする場合は、`-bs` オプションに、「8KB」を指定してください。

random write は以下のコマンドを実行することで計測できます。

```
# fio -filename=<%データディレクトリ%>/fio -direct=1 -rw=randwrite -bs=4k -size=20G
    -numjobs=5 -group_reporting -name=randomwrite
```

上記は、ブロックサイズを 4KB にした場合の実行例です。ブロックサイズを 8KB にする場合は、`-bs` オプションに、「8KB」を指定してください。

それぞれのコマンドを実行した後、出力された内容の以下の部分を読み取ります。

- `read:` または `write:` から始まる行の、IOPS の値、および BW の値(帯域幅)

- `lat (usec)`: から始まる行の、avg の値(レイテンシー平均値)

以下は random write (randwrite) の場合の出力例です。

```
randomwrite: (g=0): rw=randwrite, bs=(R) 4096B-4096B, (W) 4096B-4096B,
 (T) 4096B-4096B, ioengine=psync, iodepth=1
...
fio-3.15
Starting 5 processes
Jobs: 1 (f=1): [w(1),_(4)][99.9%][w=23.5MiB/s][w=6009 IOPS][eta 00m:07s]
randomwrite: (groupid=0, jobs=5): err= 0: pid=14996: Tue Jan 21 18:14:55 2020
  write: IOPS=2669, BW=10.4MiB/s (10.9MB/s)(100GiB/9819695msec)
    clat (usec): min=38, max=16726k, avg=1867.65, stdev=30805.21
     lat (usec): min=38, max=16726k, avg=1867.85, stdev=30805.21
    clat percentiles (usec):
（以下略）
```

コマンドの実行後、<%*データディレクトリ*%>/fio ファイルが残っていたら、削除します。

4.5.2　フロー受信におけるジャンボフレームサポート

NFA 2.1 にて、管理対象のエクスポーターからジャンボフレームとして送信されたフローの受信をサポートしました。

ヒント

ジャンボフレームとは、最大転送単位(MTU)が 1,500 バイトを超えるようなフレームを指します。NFA が受信できる最大の MTU は、9,000 バイトです。

4.5.3　バージョン 2.1 における仕様変更

バージョン 2.1 で変更された仕様について説明します。

4.5.3.1　SSL サーバー証明書を格納するキーストア形式の変更

`nfa_ssl_keytool` コマンドで作成するキーストアの形式を、Java keytool コマンドの推奨形式変更に伴い、Java KeyStore (JKS) から PKCS12 へ変更しました。

この変更に伴い、`nfa_ssl_keytool` コマンドの仕様もあわせて変更されています。詳細はリファレンスマニュアルの `nfa_ssl_keytool` コマンドの説明を参照してください。

ヒント

バージョン 2.0 以前で作成したキーストアは、2.1 以降にアップグレードした後も JKS のまま変更しません。JKS 形式のままで引き続きご利用いただけます。

4.5.4　バージョン 2.1 における修正項目

バージョン 2.1 で修正された問題の一覧を記載します。

1. バグ修正取り込みのために、Oracle Java を AdoptOpenJDK (1.8.0_202) へ置換しました。

2. 以下の OSS を、バグ修正取り込みのために更新しました。
    - Apache Tomcat (8.5.42 へ更新)
    - Apache Commons FileUpload (1.4 へ更新)
    - Apache Commons IO (2.6 へ更新)
    - Apache Commons Lang (3.8.1 へ更新)
    - OGNL (3.1.21 へ更新)
    - Log4j (2.11.2 へ更新)
    - OkHttp (3.14.2 へ更新)
    - PostgreSQL JDBC Driver (42.2.5 へ更新)
    - Qpid JMS (0.43.0 へ更新)
    - Qpid Proton-J (0.33.1 へ更新)
    - Spring Framework (5.1.8 へ更新)

4.6　バージョン 2.0 でのリリース内容

NFA2.0 において、機能追加、修正した内容を説明します。

4.6.1　IMS コンポーネントによる統合運用

WebSAM Integrated Management Server (IMS)コンポーネントを利用することで、複数配置した NFA の統合運用や、NFA と NetvisorPro との統合運用が可能になりました。

IMS コンポーネントが提供する Web コンソールを用いることで、以下の運用を行うことができます。

- 複数の NFA で運用している環境において、各 NFA のウィジェットを 1 つのダッシュボードに並べて配置することができます。これにより、管理対象となるネットワーク全体の状況を簡単に把握することができます。
- NetvisorPro による SNMP 監視の状況と NFA が収集したフロー情報を同時に確認することができます。これにより、ネットワーク障害の原因調査をスムーズに行うことができます。
- IMS コンポーネントが提供する Web コンソールから、NFA の Web コンソールをシングルサインオンで起動することができます。これにより、2 つの Web コンソールをシームレスに操作することができます。

上記以外にも IMS コンポーネントを利用することで、NFA のしきい値監視で検知したしきい値超過のイベントを、メール送信や任意のコマンドで通報することができます。

4.6.2　DSCP によるフロー分析

NFA 2.0 では、フロー情報に含まれている DSCP の値を用いたフロー分析が行えるようになりました。

DSCP 値を用いてフロー情報を分析することで、以下のような通信状況の確認を行うことができます。

- 経路上を流れるパケットに対し、意図した QoS 設定(DSCP によるマーキング)が行われているかの確認
- QoS 設定(DSCP の設定)を行ったことによる通信状況の変化の確認
- DSCP による優先度ごとの通信量の確認

具体的に DSCP によるフロー分析が行える機能は、以下の通りです。

- ダッシュボード表示

  円グラフ/折れ線グラフ表示タイプの[**DSCP**]ウィジェットが追加されました。各 DSCP 値(PHB)ごとの通信量の状況を確認することができます。

- エクスポーター分析

  [**DSCP**]ウィジェットで、フロー情報の分析結果を確認することができます。また、フィルター条件として DSCP 値(PHB)を指定することで、特定の DSCP 値(PHB)に対する通信量や通信内容を確認することができます。

- しきい値監視

  しきい値監視の対象となるフローの条件として、DSCP 値(PHB)を指定することができます。

- CSV ファイル出力

  ダッシュボード画面、および、エクスポーター分析画面で表示する[**DSCP**]ウィジェットで表示するフロー情報を他のウィジェットと同様に CSV ファイル形式で外部出力することができます。

  `nfa_flow_export` コマンドでは、出力する CSV ファイルに対し、今まで出力していた項目に加えて、DSCP 値(PHB)が追加されています。

4.6.3　バージョン 2.0 における修正項目

バージョン 2.0 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
   - Java Runtime (8u131 へ更新)
   - Apache Tomcat (8.5.35 へ更新)
   - Apache Struts2 (2.3.36 へ更新)
   - Apache Commons FileUpload (1.3.3 へ更新)

- Apache Commons Collections (3.2.2 へ更新)
- Apache Commons BeanUtils (1.9.3 へ更新)
- Apache Commons Logging (1.2 へ更新)

2. エクスポーターの表示名、エンドポイントグループ名、アプリケーション名に"*その他*"という名前を設定した場合に、[**エクスポーター**]ウィジェット、[**送信元エンドポイントグループ**]ウィジェット、[**宛先エンドポイントグループ**]ウィジェット、[**アプリケーション**]ウィジェットにおいて、名前としての"*その他*"に対し、リンクが設定されない問題を修正しました。
3. サービスの停止時に、nfa_collector プロセスが異常終了する場合がある問題を修正しました。
4. sFlow のエクスポーターにおいて、送信側のフローに対するモニタリングを設定した場合に、当該エクスポーターから取得したフロー情報に対するグラフ表示が正しく行えない場合がある問題を修正しました。
5. 受信したフローデータを蓄積するデータベースのデータサイズが肥大化する場合がある問題を修正しました。

4.7　バージョン 1.1 でのリリース内容

NFA1.1 において、機能追加、修正した内容を説明します。

4.7.1　フローデータのエクスポート機能

データベースに蓄積したフローデータを CSV ファイルとして出力する `nfa_flow_export` コマンドを追加しました。

本コマンドにより、蓄積したフローデータを粒度を落とすことなく、外部ファイルとして長期保存することができます。

本コマンドを用いた主な運用例は以下になります。

- 詳細な過去のデータを外部ファイルとして長期保存する。
- コマンドを cron などから呼び出すことより、分析レポート作成の元になるデータを定期的に生成する。
- 外部の運用管理ソフトウェアから本コマンドを呼び出すことにより、インシデント発生時の通信状況や、イベントの詳細を自動で保存する。

作成した CSV ファイルは外部の表計算ソフトに取り込むことで、自由に編集・分析することができます。CSV の作成は、分析を行いたい期間を指定したり、前回コマンドを実行した続きから CSV ファイルを出力するなど、運用に合わせた柔軟な操作が可能です。

本機能はコマンドラインのため、Web ブラウザーを必要とせずに実行することができます。

4.7.2　フローデータ保持期間の動的変更対応

NFA では、大量のフローデータを長期間保持するために、一定の期間ごとにデータを集約し、データの粒度を変えて保持しています。NFA1.1 では、この保持期間を変更できるように機能強化を行いました。

保持期間のデフォルト値と、変更可能な保持期間の範囲は以下の通りです。

表 4-7　フローデータの粒度と保持期間

データの粒度(単位時間)	デフォルトの保持期間	保持期間の変更可能範囲
1 分	24 時間	2～168 時間
10 分	72 時間	12～336 時間
60 分	14 日間	4～60 日間
6 時間	60 日間	14～365 日間
24 時間	365 日間	60～1095 日間
7 日	1095 日間	365～2190 日間

保持期間の変更は運用を停止することなく実施することが可能です。このため、運用中に監視エクスポーター数を増加したり、大量のフローデータを格納したことによりディスクの空き容量が少なくなった場合でも、保持期間を変更することで対処を行うことができます。

また、保持期間の設定は、データの粒度毎に異なる値を設定することができます。例えば、詳細なフローデータの保持期間を増加し、粒度の粗いフローデータの保持期間を短くするといった、運用の目的に合わせた設定変更を行うことができます。

4.7.3　しきい値監視機能の性能向上

しきい値監視機能の処理性能を向上しました。NFA1.0 では監視項目数を 150 項目以下にすることを推奨していましたが、NFA1.1 にてしきい値監視機能の処理を改善し、より多くの監視ができるようになりました。

弊社検証では、以下の環境で監視項目数 2000 が動作することを確認しています。

表 4-8　監視項目数 2000 の動作実績環境

項目	内容
CPU	Intel E5-2630 v3 (2.40 GHz) * 8 コア
システムメモリ	64GB
ディスク	2.5 型 SAS (15,000rpm)
OS	Red Hat Enterprise Linux 7.3 (x86_64)
フロー数	20,000 フロー/秒

4.7.4　対応フロープロトコルの強化

NFA1.1 において、NetFlow サンプリングに対応しました。また新規フロープロトコルとして IPFIX に対応しました。

NetFlow サンプリング

エクスポーターが、NetFlow Lite などのサンプリングした情報を送信した場合でも、フロー情報を分析できるように強化を行いました。サンプリング率はエクスポーターごとに手動で設定することができます。受信したフロー情報にサンプリング率が含まれている場合は、自動でサンプリング率を読み込みこともできます。

IPFIX

対応フロープロトコルに IPFIX を追加しました。サンプリングされたフロー情報を受信した場合でも手動でサンプリング率を設定することで、適切にフローデータを分析することができます。

4.7.5　グラフ表示タイプの切り替え機能

アプリケーション、IP プロトコルのウィジェットについては、円グラフと折れ線グラフの両方で表示できるように強化しました。

NFA1.0 では、アプリケーション、IP プロトコルのウィジェットは円グラフとしてのみ表示できましたが、NFA1.1 からは、円グラフと折れ線グラフの両方で表示できるように機能強化を行いました。

これにより、アプリケーション観点や、IP プロトコル観点で、時系列に沿ってフローを分析することができるようになります。

円グラフと折れ線グラフの切り替えは、ダッシュボード画面やエクスポーター分析画面から、動的に行うことができます。

ダッシュボード画面においては、円グラフと折れ線グラフのどちらをデフォルトのグラフとして表示するかを定義することができます。また、ウィジェットを複数定義することにより、円グラフと折れ線グラフを並べて表示することもできます。

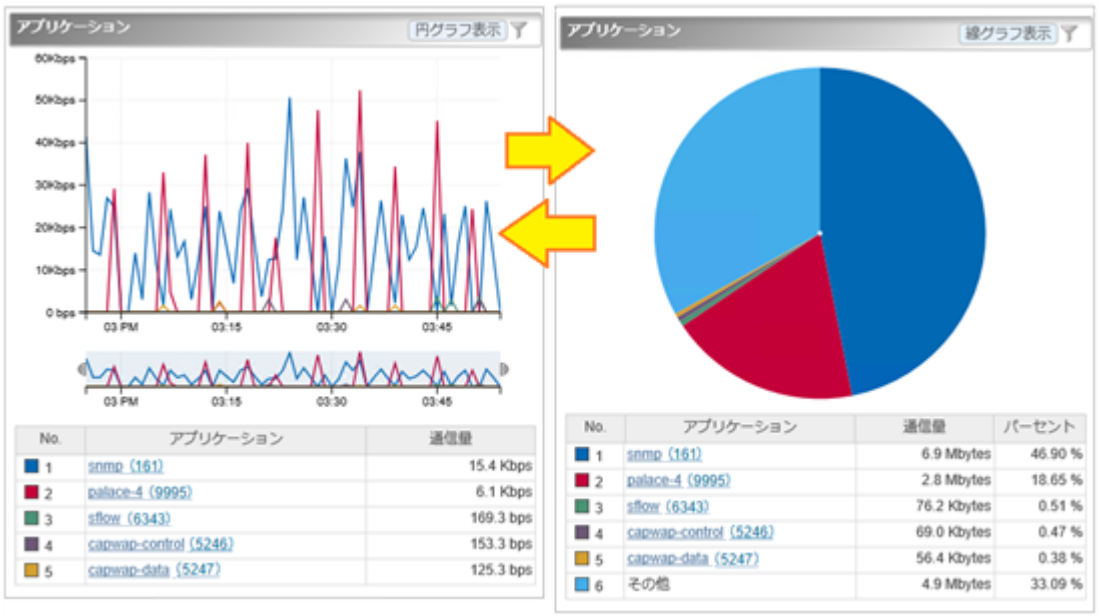

図 4-1　円グラフ/折れ線グラフ表示タイプのウィジェット

4.7.6　バージョン 1.1 における仕様変更

バージョン 1.1 で変更された仕様について説明します。

4.7.6.1　分析結果の CSV ファイル出力内容の変更

ダッシュボード画面やエクスポーター分析画面からの CSV 出力に関する仕様を、以下のように変更しました。

ファイル名に関する変更点

- ダウンロードファイル名

  エクスポーター分析画面での CSV 出力によるダウンロードファイル名の接頭辞を以下のように変更しました。

変更前	変更後
ExporterAnalyzeCSV_	ExporterAnalysisCSV_

- CSV ファイル名

  ダウンロードした zip ファイルに含まれる CSV ファイル名のうち、ウィジェットの名称にあたる部分を変更しました。

変更前	変更後
ExporterTraffic	Exporters
InterfaceInputTraffic	InInterfaces
InterfaceOutputTraffic	OutInterfaces

変更前	変更後
srcIPAddress	SourceIPAddresses
dstIPAddress	DestinationIPAddresses
Conversation	Conversations
srcEndPointGroup	SourceEndpointGroups
dstEndPointGroup	DestinationEndpointGroups
srcAS	SourceAS
dstAS	DestinationAS
Application	Applications
IPProtocol	IPProtocols
CurrentAlert	CurrentAlerts

CSV ファイルの内容に関する変更点

- 一部の項目名を変更しました。
  - 共通

変更前	変更後
StartTime	StartingTime
EndTime	EndingTime
Exporter	Exporters
Interface	Interfaces
FlowFilterCount	FilterCount
WidgetName	WidgetTitle

  - エクスポーター分析画面から出力されるウィジェット名称

変更前	変更後
ExporterTraffic	Exporters
InterfaceInputTraffic	InInterfaces
InterfaceOutputTraffic	OutInterfaces
srcIPAddress	SourceIPAddresses
dstIPAddress	DestinationIPAddresses
Conversation	Conversations
srcEndPointGroup	SourceEndpointGroups
dstEndPointGroup	DestinationEndpointGroups
srcAS	SourceAS
dstAS	DestinationAS
Application	Applications
IPProtocol	IPProtocols
CurrentAlert	CurrentAlerts

- エクスポーター分析画面でフロー条件を指定した場合の項目名

変更前	変更後
srcIPAddress	SourceIPAddress
dstIPAddress	DestinationIPAddress
srcEndPointGroup	SourceEndpointGroup
dstEndPointGroup	DestinationEndpointGroup
srcAS	SourceAS
dstAS	DestinationAS

- 一部の項目の値の出力形式を変更しました。
  - 以下の項目に出力される時刻情報を UNIX 時刻形式に変更しました。
    * Date
    * StartingTime
    * EndingTime
    * <データ行中の時刻を表す値>
  - Exporters 項目について、すべてのエクスポーターを表す値を変更しました。

変更前	変更後
(all)	(All)

  - Exporters および Interfaces 項目について、エクスポーター名に IP アドレスを付与するよう変更しました。

変更前	変更後
Exporter-001	Exporter-001 (192.168.10.1)

  - 削除されたインターフェイス、エンドポイントグループまたはアプリケーションを出力する場合の表現を変更しました。 対象ウィジェットは下記の通りです。
    * 入力インターフェイス
    * 出力インターフェイス
    * 送信元エンドポイントグループ
    * 宛先エンドポイントグループ
    * アプリケーション

変更前	変更後
deleted	(deleted)

  - 対象のフローデータが存在しない場合に、「No Data」を出力するように改善しました。 対象ウィジェットは下記の通りです。
    * アプリケーション

        * IP プロトコル

    - エクスポーター分析画面からの出力時の CsvType の値を変更しました。

変更前	変更後
ExporterAnalyze	ExporterAnalysis

- データ行の一部について、値の出力形式を変更しました。
    - 「その他」の名称を変更しました。 対象ウィジェットは下記の通りです。
        * アプリケーション
        * IP プロトコル

変更前	変更後
(Other)	Others

    - カレントアラートウィジェットのデータラベル名を変更しました。

変更前	変更後
OccurredTime,Severity,Target,Detail	Severity,DetectionTime,Targets,Content

4.7.7　バージョン 1.1 における修正項目

バージョン 1.1 で修正された問題の一覧を記載します。

1. 以下の OSS を、バグ修正取り込みのために更新しました。
    - Java Runtime (8u121 へ更新)
    - Apache Tomcat (8.0.39 へ更新)
    - Apache Struts2 (2.3.32 へ更新)
    - Apache Commons BeanUtils (1.9.2 へ更新)
    - Apache Commons FileUpload (1.3.2 へ更新)
    - Log4j2 (2.5 へ更新)
    - PostgreSQL (9.2.14 へ更新)
    - SQLite3 (3.10.2 へ更新)
    - ICU (58.2 へ更新)
2. UNIVERGE PF6800 Ver. 6.3 の WebGUI からの Network Flow Analyzer の画面起動が失敗する問題を修正しました。
3. エクスポーター分析画面の期間の指定において、起点の日時を現在時刻から 1 時間以内に設定し、かつ期間を「現在時刻まで」に設定した場合に、分析結果に起点の日時に指定した時刻の 1 分前のデータが含まれてしまう問題を修正しました。

4. エクスポーター分析画面の期間の指定における[**特定の日時と期間を指定**]にて、期間に「現在時刻まで」以外を選択した際に、指定された期間と、分析結果として表示される期間が異なる(指定された期間に対して、分析結果の期間が 1 つの単位時間の分だけ余分に表示される)問題を修正しました。

   単位時間についてはリファレンスガイドの「フローデータの保持期間と丸め処理について」をご参照ください。

5. エクスポーター分析画面での[**特定の日時と期間を指定**]の期間に「現在時刻まで」を指定し、表示された分析結果の画面で CSV 出力を実行した場合に、蓄積データの切り替えのタイミングによって線グラフのウィジェットのデータ粒度が画面上の粒度よりも細かくなる場合がある問題を修正しました。

6. エクスポーター分析画面の分析期間の指定において、起点の日付を現在時刻の 3 日前に指定した場合に、時刻指定ができない（プルダウンリストが選択できない）場合がある問題を修正しました。

7. 複数のエクスポーターに対して、DNS 情報取得または SNMP 情報取得が同時に実行されると、情報取得に失敗する、または、実際には情報取得が成功しているにも関わらず画面上に失敗と表示される場合がある問題を修正しました。

8. イベント一覧画面の表示処理性能を改善し、大量にイベントが登録されている場合でも、数秒で表示できるようになりました。

9. 入力側インターフェイスの識別子(IN ifIndex)の値が有効でない(0 もしくは値が含まれない)フローを受信した場合に発生する次の問題を修正しました。

   - 当該エクスポーターが sFlow エクスポーターの場合、グラフの値が 0 で表示される。
   - フローデータの集約(丸め処理)において、受信したフローを別のフローと同一とみなし集約してしまうため、グラフ表示の値が不正となる。

第 5 章
注意制限事項

NFA3.2 における注意制限事項について説明します。

目次

5.1　エクスポーター側の設定に対する注意制限事項

エクスポーター側の設定に対する注意制限事項について説明します。

5.1.1　SNMP ifIndex 持続性のための設定

NFA でフローを正しく分析するためには、分析対象のインターフェイスに対応する ifIndex の値が変化しないように、エクスポーター側の設定を行う必要があります。

エクスポーターを再起動すると、エクスポーターの仕様によっては、分析対象のインターフェイスに対応する ifIndex の値が変化する場合があります。この場合、NFA では、分析箇所のインターフェイスの特定が正しく行えないため、分析結果も正しく表示することができなくなります。

エクスポーターの仕様によっては、ifIndex 値を再起動後も持続するための設定が行える場合があります。運用を開始する前に、必ず、エクスポーターの ifIndex 値の持続性に関する仕様を確認し、ifIndex 値の持続性のための設定を行ってください。

以下にエクスポーター側での ifIndex 値の持続性のための設定例 (Cisco Catalyst 6500 シリーズ) を示します。

```
(config)# snmp-server ifindex persist
```

⚠ 注意

エクスポーターの設定を行うコマンドの仕様は、機種によって異なります。必ず、エクスポーター側の設定マニュアルを確認し、設定作業を実施してください。

5.1.2　NetFlow v9 および IPFIX 利用のための設定

NFA では、NetFlow v9、および、IPFIX に対して、特定のフォーマットのみをサポートしています。

NetFlow v9、または、IPFIX を利用する場合は、エクスポーター側の設定において、以下のフィールドタイプを含むフローレコード定義の作成を行ってください。

1. 送信元 IP アドレス / 宛先 IP アドレス [注 1]
2. 送信元ポート番号 / 宛先ポート番号 [注 1]
3. IP プロトコル [注 1]
4. ToS バイト(DSCP) [注 1]
5. 入力インターフェイス / 出力インターフェイス [注 2]
6. フローのバイト数、パケット数 [注 3]

注

1. 個々のフィールドタイプは必須ではありませんが、特別な理由が無い限りエクスポータ側でフローレコードに含める設定を行ってください。

   フローレコードに該当情報が存在しない場合は任意値(ゼロ)として扱います。 そのため、該当する widget が表示されない等の結果となり、フローを正しく分析出来ない場合があります。

2. エクスポーター側でフローレコードに含める設定を必ず行ってください。

   ライセンスを正しく付与するために必須な情報です。

3. エクスポーター側でフローレコードに含める設定を必ず行ってください。

   フローの通信量を統計分析するために必須な情報です。

以下にエクスポーター側でのフローレコードの設定例(Cisco Catalyst 3850 シリーズ)を示します。

```
(config)# flow record NetFlow-record
(config)# match ipv4 tos
(config)# match ipv4 protocol
(config)# match ipv4 source address
(config)# match ipv4 destination address
(config)# match transport source-port
(config)# match transport destination-port
(config)# collect interface input
(config)# collect interface output
(config)# collect counter bytes long
(config)# collect counter packets long
(config)# collect timestamp sys-uptime first
(config)# collect timestamp sys-uptime last
```

⚠ 注意

エクスポーターの設定を行うコマンドは、機種によって異なります。必ず、エクスポーター側の設定マニュアルを確認し、設定作業を実施してください。

5.1.3　IPv6 通信のフロー分析について

NFA 3.2 では、IPv6 通信のフローの分析に対応していません。

エクスポーター側の設定において、IPv6 通信のフローを監視対象とした場合、NFA では、そのフローデータを処理することができません。

不要な通信を避けるため、エクスポーター側の設定において、IPv6 通信のフローを監視対象としないように設定してください。

WebSAM
Network Flow Analyzer 3.2
リリースメモ

NFA00RJ0320-01

2023 年 10 月 01 版 発行

日本電気株式会社