2008－CSEC－43 (6)
2008／12／5

パッケージマネージャと連携したNIDSの誤検知削減

北野雄大　　嶋村 誠　　河野 健二

慶應義塾大学 理工学部 情報工学科

E-mail: kitano@sslab.ics.keio.ac.jp, tima@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

現在のシグネチャ型ネットワーク侵入検知システム (NIDS) は無害なメッセージに対して警告を発する誤検知が多発している．その原因の一つにインストールされていないソフトウェアに対する攻撃の誤検知が挙げられる．このような誤検知への対策は既存手法では不十分である．本研究では監視対象サーバのシステムの情報を考慮して NIDS が発した警告が本当に必要な警告かを選別し誤検知を減らすシステムを提案する．提案機構では監視対象サーバのパッケージマネージャを利用し，監視対象サーバにインストールされているソフトウェアの情報をあらかじめ取得しておく．NIDS が攻撃を検知すると，その攻撃が対象とするソフトウェアの情報と取得した監視対象サーバのソフトウェア情報を照合し，一致した場合にのみ警告を発する．実際に研究室内 DMZ を流れるパケットデータを用いた実験の結果，提案機構は全警告の 49%に当たる 289 件の false positive を削減できた．

Reduction of NIDS False Alerts Using Package Manager

Takehiro Kitano　　Makoto Shimamura　　Kenji Kono

Department of Information and Computer Science, Keio University

E-mail: kitano@sslab.ics.keio.ac.jp, tima@sslab.ics.keio.ac.jp, kono@ics.keio.ac.jp

Current signature-based Network Intrusion Detection Systems (NIDSs) generate many false alarms for normal messages. This is partly because an NIDS detects attacks to software not installed. We propose the system that uses software information of the monitored server to decrease false alarms. Software information means the name and version of the software installed on the monitored server. Our system uses a package manager to obtain software information. When NIDS detects attack messages, our system refers to the software information and generates an alert only when the target software is installed on the monitored machine. We conducted an experiment using the packet data from our laboratory's DMZ and the results suggest that our system decreases 289 false alarms, 49% of all.

1 はじめに

現在，インターネットを介してサーバを攻撃する攻撃者が依然として存在している．そこでサーバ管理者は，攻撃者からの攻撃メッセージを検知するネットワーク侵入検知システム (NIDS) を用いて，攻撃に対応している．

現在主流である NIDS は，シグネチャと呼ばれる攻撃メッセージの特徴を記述したものを用いて，インターネットから来るメッセージを検査することによって検知を行う．多くの NIDS はシグネチャとして攻撃メッセージ中のバイト列を用いる [1, 2]．シグネチャに記述されているバイト列とメッセージが一致すると，メッセージが攻撃であると判断され，NIDS は管理者に対して警告を発する．このようにシグネチャを用いた NIDS を Signature-based NIDS と呼ぶ．

しかし，現在の Signature-based NIDS は無害なメッセージに対して警告を発する誤検知 (*false positive*) が多発しており，管理者が本当の攻撃メッセージに対する警告を見逃してしまうという問題が発生している．Julisch の論文 [3] では，標準の設定の NIDS においては，一日に発生する数千件の警告の 99 %以上が false positive であると報告されている．また，実際に我々の研究室の DMZ において取得した 2 週間分のパケットを著名な NIDS である Snort [1] で検査したところ 587 件の警告が発生したが，それらは全て false positive であった．このように false positive が多発すると，管理者が NIDS を信頼しなくなってしまう．

このような false positive の原因は，以下の 2 種類に分類できる．

- **攻撃かどうかわからないメッセージを検知するシグネチャによる誤検知**　シグネチャの中には，特定の攻撃メッセージを検知するのではなく，機械語命令列を含むメッセージやプロトコルに違反したメッセージを検知するシグネチャがある．しかし，これらのシグネチャは通常のメッセージを攻撃メッセージであると誤検知してし

まう可能性が高く，多くの false positive を発生させる原因となっている．例えば，Snort のシグネチャID:651 [4] は x86 の jmp 命令に使われるバイト列を検出することで，機械語命令列の検知を行う．しかし，このシグネチャは動画のストリーミング再生などで，容易に false positive を発生することが知られている．

- **インストールされていないソフトウェアに対する攻撃の誤検知**　Signature-based NIDS では監視対象となるサーバにどのようなソフトウェアがインストールされているかを考慮していない．このため，攻撃対象サーバにインストールされていないソフトウェアに関連する攻撃メッセージに対しても警告を発する．例えば，サーバプログラムとして Apache Web Server を使っているにもかかわらず，NIDS が Microsoft IIS Web Server に対する攻撃を検知してしまうことがある．この検知された攻撃はサーバで成功することはないため，NIDS の発した警告は false positive になる．

攻撃かどうかわからないメッセージを検知するシグネチャによる誤検知は，機械語命令列やプロトコル違反などの攻撃を示唆する異常なメッセージをシグネチャを用いて検知しようとするために発生する．前述のように，このようなメッセージをシグネチャを用いて検知することには限界がある．機械語命令列やプロトコル違反といった攻撃を示唆するメッセージは，シグネチャを使わない NIDS を用いることで，高い精度で検出できる．例えば，メッセージ中の機械語命令列を検知する NIDS [5, 6, 7] やプロトコルの定義を用いてプロトコル違反を検知する NIDS [8] がある．よって本研究ではこのような誤検知を対象としない．

インストールされていないソフトウェアに対する攻撃の誤検知を減らすために，現在は管理者が人手で監視対象サーバにインストールされていないソフトウェアへの攻撃を検知するシグネチャを削除している．しかし，NIDS の監視対象となるサーバが複数ある場合はそれぞれ様々なソフトウェアが稼働しているため，シグネチャの削除を適切に行って false positive を減らすことは難しい．また，誤ったシグネチャを削除してしまうと攻撃メッセージを無害なメッセージと判断してしまう誤検知 (*false negative*) が発生してしまう．

一方で，攻撃の成否を考慮して false positive を削減する研究 (*Alert Verification*) [9, 10, 11] が進められている．このような技術を用いることで，インストールされていないソフトウェアに対する攻撃の誤検知を削減できる．既存の Alert Verification は脆弱性検査の結果や攻撃の痕跡，攻撃に対するサーバのレスポンスを検査して攻撃の成否を判断する．しかし，このような Alert Verification では脆弱性検査手法の用意されていない攻撃や，痕跡を残さない攻撃，サーバレスポンスを偽造する攻撃は，攻撃の成功を判断することができない．これによって，false negative が発生するという問題がある．

本研究では以上の考察より，監視対象サーバにインストールされているソフトウェアなどのシステムの情報を考慮して NIDS が発した警告が本当に必要な警告かを選別し false positive を減らすシステムを提案する．近年のオペレーティングシステム (OS) では RPM [12] などのパッケージマネージャによってインストールされているソフトウェアの情報が管理されている．提案機構では監視対象サーバのパッケージマネージャを利用し，監視対象サーバにインストールされているソフトウェアの情報をあらかじめ取得しておく．NIDS が攻撃を検知すると，その攻撃が対象とするソフトウェアの情報と取得した監視対象サーバのソフトウェア情報を照合し，一致した場合にのみ警告を発する．これにより，既存の Alert Verification システムを回避しようとする攻撃に対しても正しく検知を行うことができる．

提案機構を Linux 2.6.11 上で動作する Snort 2.8.0.1 を用いて実装した．我々の研究室の DMZ に設置されたサーバが送受信したパケットを 2 週間キャプチャした 2.8GB のログを用いて行った実験の結果，提案システムは 587 件の false positive のうち，289 件 (49%) の false positive を削減できた．

本論文では 2 章に Alert Verification およびパッケージマネージャに関する関連研究について論じ，3 章で提案機構について述べる．4 章は実装について述べ，5 章では NIDS をテストするツールである Nessus による実験と実際のパケットログを用いた実験について述べる．7 章で本論文をまとめる．

2　関連研究

2.1　Alert Verification

近年では，NIDS が検知した攻撃の成否を確認し，攻撃が成功した場合のみ警告を出すことで false pos-

itive を減らす研究 (*Alert Verification*) が進められている．

Kruegel らの Snort Alert Verification [9] では，攻撃が対象とする脆弱性の存在や，攻撃が成功した証拠を検出することで攻撃の成否を確認する．例えば，Slammer ワームによる攻撃を検知した場合，/tmp/.uubugtraq というファイルが作られたかどうかを調べ，ファイルがあれば警告を発する．しかし，彼らのシステムでは，攻撃の成否の確認方法を用意することが難しい．それに対し，提案機構は攻撃に関連するソフトウェアの情報を用いて Alert Verification を行う．

Zhou らの研究 [10] ではサーバからのレスポンスがプロトコルに従っているかを分析することで，Alert Verification を行う．レスポンスがプロトコルに違反する場合は攻撃が成功したと判断し，管理者に警告を発する．しかし，Todd ら [13] は，このような Alert Verification はプロトコルに従うレスポンスを偽造することで，回避可能であることを示した．提案機構ではレスポンスを参照する必要がないので，このような手法で回避されることはない．

3 提案機構

提案機構では監視対象サーバのパッケージマネージャを利用し，監視対象サーバにインストールされている OS 名，ソフトウェア名やそのバージョン (システム情報) をあらかじめ取得しておく．NIDS が攻撃を検知すると，その攻撃が対象とする脆弱性のある OS，ソフトウェアやそのバージョン (脆弱性情報) と取得した監視対象サーバのシステム情報を照合し，一致した場合にのみ警告を発する．逆に一致しない場合は脆弱性のあるソフトウェアにパッチが当てられていたり，監視対象サーバ上で脆弱性のあるソフトウェアが稼働していないなどと推測することができ，攻撃は失敗するとみなすことができるため，警告を破棄する．

パッケージマネージャとは各種ソフトウェアの導入と削除，ソフトウェア同士やライブラリの依存関係を管理するシステムである．パッケージマネージャはインストールされているソフトウェアや依存情報などの情報を保持している．代表的なパッケージマネージャとして主に RedHat 系 Linux や SUSE Linux や Vine Linux などで使われている RPM [12]，Debian 系 Linux で使われている apt [14, 15]，Gentoo Linux で使われている Portage [16] が挙げられる．

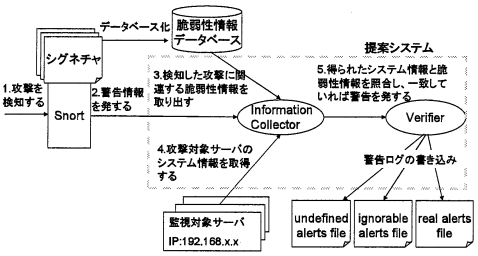

図 1: 提案機構の設計図

提案機構の設計図を図 1 に示す．提案機構は監視対象サーバのシステム情報と検知した攻撃に関する脆弱性情報を集める Information Collector 部と，集めた情報を元に警告の選別を行う Verifier 部に分けることができる．

3.1 Information Collector

Information Collector は攻撃対象となる脆弱性情報をまとめたデータベース (脆弱性情報データベース) と，監視対象サーバにインストールされているシステム情報ファイルから必要な情報を取り出す．

3.1.1 脆弱性情報データベース

Signature-based NIDS では，あるシグネチャを用いて検知できる攻撃がどのような脆弱性を突いた攻撃なのかが理解できるように，各シグネチャについてのドキュメントを用意している．提案機構ではこの情報を基にして，各シグネチャとそのシグネチャによって検知できる攻撃が対象とする脆弱性を対応づけ，脆弱性情報データベースを作成する．脆弱性情報データベースはシグネチャID，攻撃対象となる脆弱性のあるソフトウェアや OS 名とそのバージョン，攻撃対象となるポート番号といった情報を保有している．

例えば，Snort のシグネチャID:1634 [17] は攻撃者に任意のコードを実行させるような buffer overflow 攻撃を検出するシグネチャである．この攻撃は Windows 上で稼働している Xtramail 1.11 mailserver の脆弱性を利用する．

この情報を基に，図 2 に示すような脆弱性情報を生成する．この脆弱性情報はシグネチャID:1634 が検知する攻撃はソフトウェア名が Xtramail でそのバージョンが 1.11，かつ OS が Windows でポート 110 番が開いているときに脆弱性があることを示す．

sid=1634 app_name=xtramail
app_ver=1.11 os_name=windows port=110

図 2: シグネチャID:1634 の脆弱性情報の例

3.1.2 監視対象サーバのシステム情報の取得

監視対象サーバにインストールされているソフトウェアの情報はパッケージマネージャを利用して取得する．例えば RedHat 系 Linux などで使用されている RPM ではインストールされているソフトウェア名とバージョンのリストを `rpm -qa` というコマンドで取得できる．

また，監視対象サーバのカーネル情報やディストリビューション情報といった OS に関する情報も取得可能である．例えば Linux ではカーネル情報を表示させるコマンドとして `uname` が用意されている．さらに各ディストリビューションの情報は`/etc` 以下の `redhat-release` (Fedora) や `vine-release` (Vine Linux) や `debian_version` (Debian) などのファイルに記述されている．

これらのシステム情報を監視対象サーバが起動した際およびソフトウェアの更新が行われた際に毎回取得し保持することによって，最新のシステム情報での Alert Verification が可能である．

3.2 Verifier

Verifier は Information Collector で集めた脆弱性情報と監視対象サーバのシステム情報を照合する．また，攻撃が成功する要件にポート番号が指定されている場合は，監視対象サーバに対してポートスキャンを行い，指定ポートが開放されているかどうかを確認する．

情報が一致した場合は，検知した攻撃に対して脆弱性があると判断できるので必要な警告 (real_alert) として警告のログをとる．一致しなかった場合は，検知した攻撃は監視対象サーバにインストールされているシステムと無関係の攻撃であると判断できるので必要ない警告 (ignorable_alert) として警告を分類する．

例えば，シグネチャID:1634 のシグネチャによって警告が生成された場合，Verifier は検知されたパケットのあて先IP アドレスを基に攻撃対象となっている監視サーバのシステム情報を取得し，図 2 の脆弱性情報と照合する．監視下にある攻撃対象サーバの OS が Windows で Xtramail のバージョン 1.11 以下がインストールされておりかつ，ポートスキャンを行った結果，ポート 110 番が開放されていた場合はその警告を real_alert ファイルに書き込む．監視下にある攻撃対象サーバが脆弱性情報をひとつでも満たしていない場合は ignorable_alert として警告を分類する．

しかし，すべてのシグネチャの脆弱性情報が得られるとは限らない．例えば，シグネチャID:486 [18] はあて先 IP アドレスが不明であるときに警告を生成するシグネチャであるが，このシグネチャによって検知されたメッセージはあるソフトウェアの脆弱性を突いた攻撃ではないので，脆弱性情報が得られない．このような理由で脆弱性情報が取得できない警告は定義できない警告 (undefined_alert) として警告を分類する．これらの警告は攻撃を示唆するメッセージや攻撃準備のためのスキャンニングに対して発せられる警告である．そのため提案機構ではこのような警告は対象外とし，false positive の削減は行わない．

4 実装

3 章で述べた提案機構を Linux 2.6.11 上で動作する Snort 2.8.0.1 を用いて実装した．Snort のシグネチャのドキュメントを参考にして脆弱性情報データベースを作成したところ，全シグネチャ13199 個中，10766 個のシグネチャについて脆弱性情報をデータベース化できた．

また我々の研究室内 DMZ に設置されている Web サーバ，メールサーバ，DNS サーバのパッケージマネージャはすべて RPM 4.4.2 を用いた．

5 実験

4 章で実装した提案機構の効果を検証するため，脆弱性検査ツールである Nessus を用いて発生した警告と，研究室内の DMZ に設置されているサーバが送受信した実際のパケットキャプチャデータ (リアルトレースデータ) を用いて発生した警告に対して false positive が削減できていることを確認する実験を行った．提案機構を実装したマシンの CPU は Pentium 4 CPU 3.40GHz，メモリは 256MB，OS は Linux2.6.11 (Fedora Core 4)，IDS は Snort 2.8.0.1 を利用した．

5.1 Nessus を用いた実験

Nessus [20] は擬似攻撃を仕掛けることにより，リモートサーバの脆弱性を検査するスキャンニング

分類	内訳	件数	合計
real_alert	—	0	0
ignorable_alert	SNMP に関する攻撃	11	
	FTP に関する攻撃	3	14
undefined_alert	宛先 IP アドレス不明	891	
	スキャンニング	6	
	ファイルシステムへのアクセス	3	
	プロトコル違反	3	903

表 1: Nessus を用いた実験結果

ツールである．Nessus を利用して提案機構を実装したマシンへスキャンニングを行い，発生した警告を選別した結果を表 1 に示す．Nessus のバージョンは 3.2.2.1 を利用し，スキャンポリシは default scan policy を利用した．

実験の結果，オリジナルの Snort は 917 件の警告を発した．警告を精査した結果，917 件全てが false positive であった．提案機構はその内，14 件 (1.5%) を ignorable_alert に分類し，903 件 (98.5%) を undefined_alert に分類した．real_alert に分類された警告はなかった．

ignorable_alert に分類された 14 件の警告は SNMP に関する攻撃に対する警告と FTP に関する攻撃に対する警告であったが，どの攻撃も対象サーバには無関係の攻撃であった．そのため，これらの警告は全て false positive と考えられる．

undefined_alert に分類された 903 件の警告は宛先 IP アドレス不明に対する警告，スキャンニングに対する警告，利用していないファイルシステムへの不正アクセスに対する警告，プロトコル違反に対する警告が含まれていた．利用していないファイルシステムへの不正アクセスに対する警告は false positive になりうる警告であるが，脆弱性情報データベースの拡張によって ignorable_alert に分類することは可能である．その他の警告は攻撃準備のためのスキャンニングに対して発せられる警告である．特定の脆弱性をついた攻撃ではないため提案機構ではこのような警告は対象外とし，false positive の削減は行わない．

5.2 リアルトレースを用いた実験

リアルトレースデータを取得するため，我々の研究室の DMZ に設置されている Web サーバ，メールサーバ，DNS サーバが送受信したパケットを tcpdump [19] を用いて 2 週間キャプチャした．その結果取得した約 670 万個のパケット (2.8GB) を含むリアルトレースデータを，提案機構を用いて検査した．また実験で利用するために，我々の研究室の Web サーバ，メールサーバ，DNS サーバのシステム情報を RPM を用いて取得した．それぞれ 1003 個，849 個，1159 個のインストール済みソフトウェアの情報を取得した．

リアルトレースデータでの実験結果を表 2 に示す．オリジナルの Snort は 587 件の警告を発した．警告を精査した結果，587 件全てが false positive であった．提案機構はそのうち，14 件 (3%) の警告を real_alert，289 件 (49%) の警告を ignorable_alert，284 件 (48%) の警告を undefined_alert に分類した．

real_alert に分類された 14 件の警告は全て存在しない Web ページへのアクセスを警告するものであった．存在しない Web ページへのアクセスは Web サーバへの攻撃のためのスキャンニングである可能性はあるが，この警告自体が Web サーバに対する攻撃を警告するものではない．そのため本研究の対象外の警告であると考えられる．しかし提案機構では Web サービスを提供している (ポート 80 番が開放されている) サーバに対して存在しない Web ページのアクセスが行われた場合はその警告は real_alert に分類した．

ignorable_alert に分類された 289 件の警告は Web サーバやメールサーバへの攻撃に対する警告であった．しかし，これらの攻撃は全てインストールされていないソフトウェアに対する攻撃または，古いバージョンのソフトウェアに対する攻撃であった．そのため，これらの警告は全て false positive と考えられる．

undefined_alert に分類された 284 件の警告には宛先 IP アドレスが不明であることに対する警告やポートスキャンや Web サーバへのスキャンニングに対する警告や snort_decoder が検知したプロトコル違反に対する警告が含まれていた．これらの警告は本研究の対象外の警告であり，提案機構では Alert Verification ができないので undefined_alert に分類されている．

以上よりリアルトレースを用いた結果，本研究が

分類	内訳	件数	合計
real_alert	存在しない Web ページへのアクセス	14	14
ignorable_alert	Web サーバへの攻撃	14	289
	メールサーバへの攻撃	275	
undefined_alert	宛先 IP アドレス不明	4	284
	スキャンニング	249	
	プロトコル違反	31	

表 2: リアルトレースを用いた実験結果

対象とする警告に対して全ての false positive を削減できた.

6 まとめ

現在の Signature-based NIDS は無害なメッセージに対して警告を発する誤検知 (*false positive*) が多発するという問題が発生している. このような false positive が発生する原因は, 攻撃かどうかわからないメッセージを検知するシグネチャによる誤検知とインストールされていないソフトウェアに対する攻撃の誤検知の 2 つに分類される. 攻撃かどうかわからないメッセージを検知するシグネチャによる誤検知にはシグネチャを用いない IDS を使うことで対策できるが, インストールされていないソフトウェアに対する攻撃の誤検知は対策が不十分である.

そこで本研究では, 監視対象サーバのシステム情報を考慮して NIDS が発した警告が本当に必要な警告かを選別し false positive を減らすシステムを提案する. 提案機構では監視対象サーバのパッケージマネージャを利用し, 監視対象サーバにインストールされているソフトウェアの情報をあらかじめ取得しておく. NIDS が攻撃を検知すると, その攻撃が対象とするソフトウェアの情報と取得した監視対象サーバのソフトウェア情報を照合し, 一致した場合にのみ警告を発する. リアルトレースを用いて行った実験の結果, 提案機構が false positive の削減に有用であることを示した.

今後は攻撃の見逃しが起こらないことを証明する実験やオリジナルの Snort と比較したオーバーヘッドの測定実験といった追加実験を行う予定である.

参考文献

[1] Roesch, M.: Snort: Lightweight Intrusion Detection for Networks, *Proc. of the 13th USENIX Conference on Systems Administration* (*LISA '99*), pp. 229–238 (1999).

[2] Paxson, V.: Bro: a system for detecting network intruders in real-time, *Computer Networks*, Vol. 31, No. 23–24, pp. 2435–2463 (1999).

[3] Julisch, K.: Mining Alarm Clusters to Improve Alarm Handling Efficiency, *Proc. of the 18th Annual Computer Security Applications Conference* (*ACSAC '01*), pp. 12–21 (2001).

[4] Snort-SID651: SHELLCODE x86 stealth NOOP.

[5] Wang, X., Pan, C.-C., Liu, P. and Zhu, S.: SigFree: A Signature-free Buffer Overflow Attack Blocker, *Proc. of the 15th Usenix Security Symposium*, pp. 225–240 (2006).

[6] Polychronakis, M., Anagnostakis, K. G. and Markatos, E. P.: Network-Level Polymorphic Shellcode Detection Using Emulation, *Proc. of the 3rd Detection of Intrusions and Malware and Vulnerability Assessment* (*DIMVA '06*) , pp. 54–73 (2006).

[7] Polychronakis, M., Anagnostakis, K. G. and Markatos, E. P.: Emulation-Based Detection of Non-self-contained Polymorphic Shellcode, *Proc. of the 10th International Symposium on Recent Advances in Intrusion Detection* (*RAID '07*), pp. 87–106 (2007).

[8] Kono, K., Shinagawa, T. and Kabir, R.: Improving Internet Server Security by Filtering on TCP Streams, *Transactions of Information Processing Society of Japan*, pp. 33–44 (2005).

[9] Kruegel, C., Robertson, W. and Vigna, G.: Using Alert Verification to Identify Successful Intrusion Attempts, *Practice in Information Processing and Communication*, Vol. 27, No. 4, pp. 219–227 (2004).

[10] Zhou, J., Carlson, A. J. and Bishop, M.: Verify Results of Network Intrusion Alerts Using Lightweight Protocol Analysis, *Proc. of the 21st Annual Computer Security Applications Conference* (*ACSAC '05*), pp. 117–126 (2005).

[11] Xiao, M. and Xiao, D.: Alert Verification Based on Attack Classification in Collaborative Intrusion Detection, *Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing*, pp. 739–744 (2007).

[12] RPM: RedHat Package Manager, `http://rpm.org/`.

[13] Todd, A. D., Raines, R. A., Baldwin, R. O., Mullins, B. E. and Rogers, S. K.: Alert Verification evasion Through Server Response Forging, *International Symposium on Recent Advances in Intrusion Detection (RAID'07)*, pp. 256–275 (2007).

[14] apt get: APT-GET, `http://www.apt-get.org/`.

[15] apt rpm: APT-RPM, `http://apt-rpm.org/`.

[16] Progect, G. L.: Gentoo Linux Portage Development, `http://www.gentoo.org/proj/en/portage/`.

[17] Snort-SID1634: POP3 PASS overflow attempt, `http://www.securityfocus.com/bid/791`.

[18] Snort-SID486: ICMP Destination Unrachable Communication with Destination Host is Administratively Prohibited.

[19] TCPDUMP/LIBPCAP public repository: TCPDUMP, `http://www.tcpdump.org/`.

[20] Nessus: Nessus Vulnerability Scanner, `http://www.nessus.org/`.