キャンパス情報ネットワークにおけるTCPポート開放の受動的検知
Passive Detection of Open TCP Ports in Campus Information Network

升屋　正人*，下園　幸一*
Masato MASUYA*, Koichi SHIMOZONO*

鹿児島大学 学術情報基盤センター*
Computing and Communications Center, Kagoshima University*

キャンパス情報ネットワークにおいて，内部ホストのTCPポートの開放状況を把握することはセキュリティ対策上不可欠である．学外のホストからポートスキャンツールを用いて調べればよいが，新たなホストや開放ポートが追加された場合に対応が遅れてしまう．そこで，能動的に開放ポートをスキャンする方法ではなく，受動的に開放ポートを調べることができるシステムを構築し，日々のポート開放状況を確認した．
キーワード：TCPポート，ファイアウォールポリシー，脆弱性診断，ポートスキャン

An investigation of open TCP ports of internal hosts in campus information network is essential for information security arrangement. Although open ports can be checked by port scaning from an external host, it will be delayed when a new host or open port is added. Therefore, instead of active scanning open TCP ports, we developed a system that can passively check open TCP ports of internal hosts and have been monitored on a daily basis.
Keywords : TCP ports, firewall policy, vulnerability diagnosis, port scan

1. はじめに

情報セキュリティ対策を実施する上で，内部ホストのTCPポートの開放状況に関する知見は不可欠である．外部に対するポート開放はファイアウォールのポリシーで制御できるが，ポリシー設定のミスにより開放すべきポートが開放されていない場合や，開放するべきでないポートが開放されてしまっている場合がある．ポリシー設定に際して十分な確認を行うことでこうしたミスは軽減できるが，ヒューマンエラーを完全にゼロにするのは難しい．また，不正プログラムによるバックドアの設置や，利用者が意図しないプログラムの動作により，ホストにおいて意図しないポート開放が発生する場合もある．

鹿児島大学においては，全ポートを閉鎖し申告によりTCPポートを開放する設定と，一部を除いて全ポートを開放し申告によりTCPポートを閉鎖する設定を併用した運用を行っている．前者においてファイアウォール設定のミスによるポート未開放，後者において利用者が意図していないポート開放を実際に経験している．

設定ミスや意図しないTCPポートの開放がないかどうかを調べるため，nmap[1]をはじめとするポートスキャンツールやnessus[2]などの脆弱性診断ツールを用いることができる．これらのツールを用いる方法では，ファイアウォールの外部のグローバルIPアドレスを持つホストから調査を行う必要がある．しかし，民間の回線やホスティングサービス，VPSサービスを利用した場合，サービス提供事業者に攻撃を行っていると見なされ警告が来たり，通信を遮断されたりする場合がある[3]．調査を受けた側が調査を嫌い，調査を行っているホストのIPアドレスを指定して遮断を実施する場合もあるほか，自らが設置したファイアウォールにおいて自動的に遮断されてしまう場合もある．

また，ツールを用いた能動的診断には，新たに

* 〒890-0065　鹿児島市郡元 1–21–35
1–21–35, Korimoto, Kagoshima 890-0065
E-mail: {masatom, simozono}@cc.kagoshima-u.ac.jp

開放ポートが追加された場合に次回の診断まで開放ポートの追加を知ることができないという問題もある．診断の頻度を高くすればよいが，ネットワークやサーバに負荷がかかり，正常なサービスを妨害してしまう．

そこで，開放されている TCP ポートを能動的にスキャンする方法ではなく，キャンパス情報ネットワークと外部の間の通信を監視することにより，受動的に開放ポートを調べることにした．例えば，Nessus Network Monitor[4] など，ネットワークモニタリングを行う製品で同様の機能を持つものはある．しかし，知りたいのが開放ポートの情報だけの場合，こうしたツールが持つ可視化機能や脆弱性診断機能は必要ない．ツールの導入に費用がかかるほか，保守や維持管理の費用，使用方法の学習も必要となる．準備や維持の手間が少なく簡単な，開放ポートの情報だけを知ることができるシステムを目指した．

2. 受動的検知の仕組み

本システムでは，TCP ポートが開放されているかどうかを，TCP の 3 ウェイハンドシェイクの 2 番目の手順である，SYN ビットと ACK ビットをセットした TCP パケット（SYN+ACK パケット）を内部ホストが外部に送信しているかどうかで判断する．

SYN+ACK パケットは，開放している TCP ポートに対するアクセスが生じないと送信されないパケットである．ところが，グローバル IP アドレスを持つホストに対しては，Web サーバ等に対する正規のアクセスのほか，日常的に外部からのポートスキャンが行われている．Shodan[5] や Censys[6] など悪意ではない意図でスキャンを行っているサービスのほか，不正プログラムに感染し，ボットネットに参加したホストによるスキャン攻撃が日常的に行われている．TCP の 3 ウェイハンドシェイクを確立される TCP スキャンのほか，TCP セッションの確立を行わず，SYN ビットをセットした TCP パケットを送り，SYN+ACK パケットが返ってくるかどうかで開放ポートを調べる SYN スキャン（ステルススキャン，ハーフスキャンとも言われる）もある．

本システムではこれらを利用することにし，ファイアウォールの外側でキャンパス情報ネットワーク内外の通信をキャプチャして，内部から外部に向けて送信される SYN+ACK パケットを観測することで開放ポートを検知することにした．ポートスキャン等の能動的操作を必要とせず，ほかの通信に影響を与えることもない，受動的検知となる．

3. 検知システム

本システムでは，観測用サーバとして HP ProLiant DL320e Gen8 v2（Xeon E3-1240 v3 3.40GHz，16GB メモリ）に Intel X520-SR2（10GbE MMF×2）を搭載したものを用い，OS として Debian GNU/Linux 9（stretch）をインストールした．観測用サーバの 10GbE ポートを，キャンパス情報ネットワークのファイアウォールの外側に接続しているポートのミラーポートに接続した．

パケットの観測と保存には，鹿児島大学において以前から監視に利用している[7] ネットワーク監査ツール argus[8] を用いた．

TCP パケットが SYN+ACK パケットであるかどうかの判断は，tcpdump のフィルタであれば，`tcp[tcpflags] & 255 == 18` となる．argus では-に続いて tcpdump のフィルタを記述できるため，デーモンモードで起動する`-d` オプションと合わせて，

```
$ argus -d - tcp[tcpflags] & 255 == 18\
  and src net 163.209.0.0/16
```

と指定した．`163.209.0.0/16` は鹿児島大学の内部ホスト全体を示すネットワークアドレスであるので，この指定により内部から外部への SYN+ACK パケットのみがファイルに保存される．保存先は，`/var/log/argus/argus.out` である．

このファイルを argus の付属コマンドである `argusarchive` を用いて 1 時間おきに日付別ディレクトリに bzip2 圧縮形式（拡張子.bz2）で保存する．2019 年 1 月の場合，1 日あたり 83 MB から 502 MB，1 ヶ月では 3,821 MB（= 3.8 GB）となった．1 月あたり 4 GB としても，1 年間保存して 50 GB 以下となるため，例えば 1 TB の容量のドライブであればハードウェア

の寿命を超える 20 年間分の蓄積が可能である．キャンパス情報ネットワークにおける通信量は膨大であるため，すべての通信をキャプチャするとデータ保存領域を大量に必要とする．しかし，本システムでは内部から外部への SYN ビットと ACK ビットがセットされた TCP パケットのみを保存することで保存領域の削減を果たした．

保存したパケットの情報は argus の付属コマンド `ra` を用いて，

```
$ ra -nn -r file
```

とすることで確認できる．ここで，`file` はファイル名である．

内部ホストの IP アドレス，開放 TCP ポート，外部ホストの IP アドレス，送信元 TCP ポートの情報を表示するには，

```
$ ra -nn -r file -s saddr sport daddr\
  dport
```

とすればよい．本システムで用いるのは，内部ホストから外部ホストに送信される SYN+ACK パケットの情報であるため，内部ホストの IP アドレスは送信元 IP アドレス（`saddr`），開放 TCP ポートは送信元 TCP ポート（`sport`），外部ホストの IP アドレスは宛先 IP アドレス（`daddr`），送信元 TCP ポートが宛先 TCP ポート（`dport`）と逆になる．

本システムでは，この出力の各行に文字列処理を行って日時の情報を付加したものを 1 レコードとし，フィールドを，**日** (date)，**時** (hour)，**内部ホスト** IP(saddr)，**開放ポート** (sport)，**外部ホスト** IP(daddr)，**送信元ポート** (dport) の 6 つとしてデータベース SQLite[9] に格納し，SQL 文を用いて集計した．

3.1 FTP に関する例外処理

FTP プロトコルによるファイル転送は，制御用のコントロールコネクションと，データの転送に用いるデータコネクションの，2 つの TCP コネクションにより行われる．コントロールコネクションの通信は，FTP クライアントから FTP サーバに対し，TCP21 を用いた TCP コネクションを確立して行われる．これは他の TCP 通信と同様の動作であるため，本システムで TCP21 の開放が確認できた内部のホストは FTP サーバが稼働していることになる．コントロールコネクションについては，他の TCP 通信と同様に扱えばよい．

一方，データコネクションについては内部ホストの開放ポートがコネクションを確立するたびに変更になり，また，当該 FTP セッションについてのみ有効であるため，他の TCP 通信と同様の開放ポートとして扱う必要がない．特段のセキュリティ対策が必要ないことから，確認すべき開放ポートを減らすため検知の対象からは除外する．このため，例外的な処理を行う．データコネクションについては，FTP の 2 つの動作モード，アクティブモードとパッシブモードで動作が異なるため，以下，それぞれのモードにおける例外処理について述べる．

3.1.1 アクティブモードの開放ポートの除外

アクティブモードでは，FTP サーバから FTP クライアントに対して TCP セッションが確立される．このため，内部ホストから外部ホストに対して FTP による通信を行った場合に，外部ホストからの TCP セッションが確立されることになり，本システムでは内部ホストで開放ポートが検知される．アクティブモードの TCP コネクションでは，FTP サーバ側の送信元 TCP ポートは TCP20 であるため，本システムにおいて送信元ポートが TCP20 として検知された場合，内部ホストから外部の FTP サーバに対して FTP プロトコルによる通信が行われ，データコネクションが確立されたと見なすことができる．ただし，FTP のアクティブモードのデータコネクションと同様，送信元ポートを TCP20 として，TCP22 や TCP80 などをスキャンする攻撃方法がある．このため，送信元ポートが TCP20，かつ，宛先 TCP ポート番号が 1024 以上，かつ，TCP3389 などよく用いられるポート以外の場合について，FTP のデータコネクションと見なすこととし，開放ポート検知の対象から除外することにした．

3.1.2 パッシブモードの開放ポートの除外

パッシブモードでは，FTP クライアントから FTP サーバに対して任意のポートで TCP セッションが確立される．このため，FTP クライアントが動作している外部ホストから内部

のFTPサーバに対してFTPによる通信を行った場合に，内部ホストで開放ポートが検知されることになる．

ポートの情報はデータコネクションに含まれるため，ファイアウォールやIDS/IPSではデータコネクションの通信の中身を見てデータコネクション用のポートを判断し，FTPクライアントに対してのみポートを開放する．当該FTP通信にのみ一時的に使用されるポートであることから，パッシブモードのデータコネクションで用いられるポートもアクティブモードのデータコネクションのポート同様，開放ポート検知の対象から除外する．

パッシブモードのデータコネクションで使用されるポート番号を知るにはコントロールコネクションの通信を解析する必要がある．しかし，3ウェイハンドシェイクの2番目の手順のSYN+ACKパケットのみで開放ポートの検知を行う本システムでは，パケットの中身の解析は行わない．そこで，内部ホストにおけるTCP21の開放の検知，すなわち，内部のFTPサーバとのコントロールコネクションの確立に続く，同じ外部ホストとの間で検知された通信を，パッシブモードのデータコネクションと見なして開放ポートの検知から除外した．2019年1月に開放ポートとしてTCP21が検知され，FTPサーバが稼働していると思われる内部ホスト19台のうち，この方法で除外対象となったデータコネクションを検知したのは2台であった．

4. 検知結果

本システムは2018年秋より常時稼働中である．ここでは，2019年1月1日から1月31日までの31日間の検知結果について述べる．本システムが検知の対象としている内部ホストは鹿児島大学のグローバルIPを持つすべてのホストである．なお，ハニーポットや特殊な実験用機材など，十分な注意を払いつつ一般的なサーバとは別用途で運用していることが既知のホストについては，集計の対象から除外した．

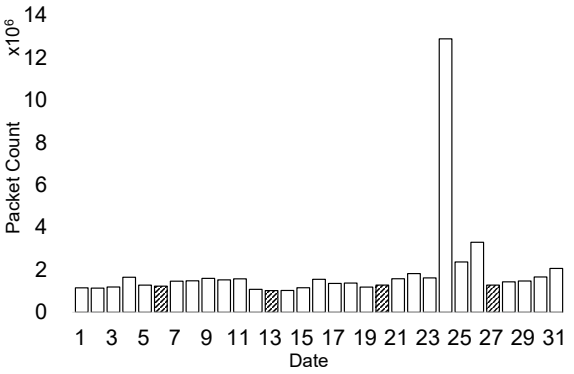

図1 2019年1月の日別検知パケット数．日曜日は，6日，13日，20日，27日（斜線）．

4.1 検知パケット数

期間内の日別の検知パケット数を図1に示す．この期間に検知されたパケット数の合計は57,866,550パケット，1日あたりの検知パケット数の最小値は1,022,500パケット（1月13日），最大値は12,891,423パケット（1月24日），平均は1,866,663パケット，中央値は1,467,102パケットであった．また，日曜日は検知パケット数が少なくなる傾向が見られ，中央値に対して，69.7%（1月13日）から87.4%（1月20日）のパケット数であった．

日毎の検知パケット数を見ると，1月24日に検知パケット数が極端に増加している．これは，米国のCDN事業者であるCloudflareに割り当てられているIPアドレス，`104.27.144.254`と`104.27.145.254`から行われた，TCP22，TCP443，TCP8080に対するスキャンによるもので，内部ホスト95台がSYN+ACKパケットを返している．これらの2台の外部ホストのみで1月24日の検知パケット数の82.8%にあたる10,678,115パケットが記録された．これら2台の外部ホストから意図的なスキャンが行われたものと思われる．スキャンは1月26日まで継続し，1月26日は検知パケット数の11.0%がこれら2台の外部ホストからのものであった．想定外の大量通信であったが，月間の中央値の8.8倍のパケットが検知されても本システムの動作に問題がないことが意図せず示された．

なお，エンドユーザに近い設備からコンテンツの配信を行うCDNサービスを行っているCloudflareは，日本国内にも設備を有していると言われており，1月24日に大量スキャンが観測されたこれらのIPアドレスを持つホスト

も ping による往復遅延時間を見ると日本国内にあると思われる．

4.2 開放ポートが検知された内部ホスト

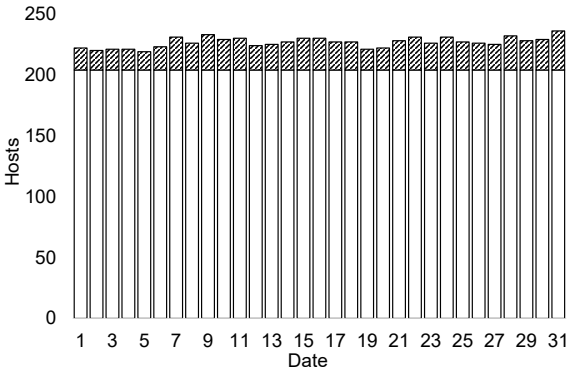

図 2 2019 年 1 月の日別検知ホスト数．毎日検知されたホスト（204 台）以外を斜線で示した．

期間内の日別検知ホスト数を図 2 に示す．この期間に検知されたホスト数の合計は 270 台，1 日あたりの検知ホスト数の最小値は 219 台（1 月 5 日），最大値は 236 台（1 月 31 日），平均は 226.7 台，中央値は 227 台であった．全検知ホスト 270 台の 75.6%にあたる 204 台が毎日検知され，それ以外の 66 台（24.4%）は日によって検知されたりされなかったりした．

毎日検知されていない 66 台に限ると，日別の検知台数の最小値は 15 台，最大値は 32 台，平均は 22.7 台，中央値は 23 台であった．これらの 66 台の内部ホストについて，検知された日数のヒストグラムを図 3 に示す．

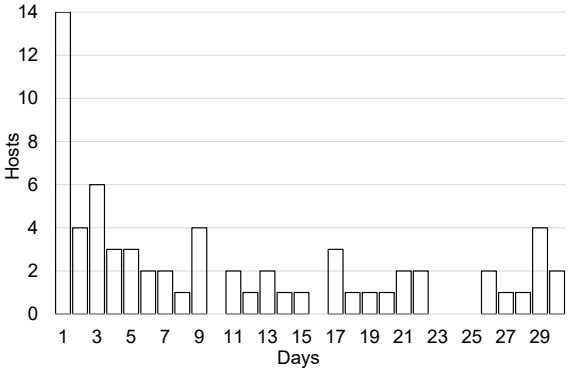

図 3 2019 年 1 月の毎日検知（検知日数が 31 日間）を除く，検知日数 1 日間から 30 日間のヒストグラム．

毎日（31 日間）検知されたホスト 204 台に次いで，最も多かったのは 1 日間しか検知されなかったホストで 14 台，次が 3 日間で 6 台，その次が 9 日間と 29 日間で 4 台ずつであった．

検知日数が多いホストについて，数日のみ検知されず，毎日検知されなかった理由を開放ポートの情報に基づいて考察する．

検知日数が 26 日間から 30 日間であったホスト 10 台の開放ポートを見ると，半数の 5 台が TCP53 と管理用と思われるポートの 2 ポートのみであり，DNS サーバとして動作しているものと思われる．TCP53 は DNS メッセージが 512 バイトを超える場合の DNS 応答に使用されるポートである．これら 5 台は，問い合わせ頻度が多くない DNS サーバであるため，たまたま TCP53 による通信が発生しない日があったものと考えられる．

また，10 台のうち 4 台は，一般的でない TCP ポートのみが開放され，特定の相手方とのみ通信を行っているホストであった．公開サーバではなく個人もしくは研究室内の研究用サーバであり，研究を実施しない日など，通信が行われない日があったものと思われる．

残る 1 台については，TCP80 が検知されており Web サーバであるが，期間中の土日（大学入試センター試験）の 2 日間に通信が検知されていない．休日にあたり機器を停止させていたものと考えられる．

4.3 新規検知ホスト数の変化

開放ポートが検知される内部ホスト数には毎日増減があり，2019 年 1 月には 219 台から 236 台の範囲で推移した（図 2）．しかし，月間では合計 270 台の内部ホストが検知されており，毎日新たなホストが検知されている．日別の新たに開放ポートが検知されたホストの数と累積ホスト数の変化を図 4 に示す．累積ホスト数は，1 月 1 日の 222 台からの累積値である．

毎日新たにに検知されたホスト数は 0 台から 8 台まで日によって異なり，平均は 1.6 台であった．月の前半 15 日までを平均すると 2.7 台，月の後半 16 日以降を平均すると 0.6 台であることから，検知開始から半月が経過すると新規の検知ホストが減る傾向にあると言える．

導入費用や管理者の数が限られていることから内部ホスト数には限りがあり，累積検知ホスト数はいずれは一定値に収束すると考えたが，

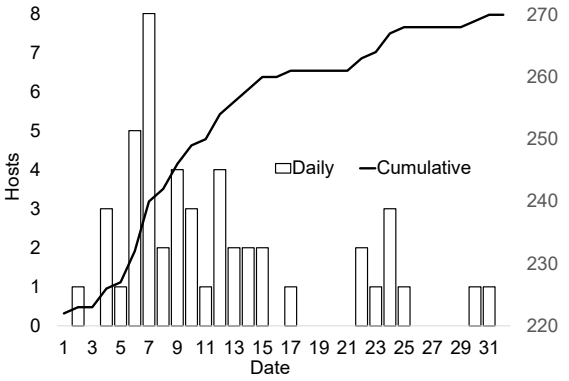

図 4 2019 年 1 月の日別新規検知ホスト数（棒グラフ・左軸）と累積検知ホスト数の変化（折れ線グラグフ・右軸）.

増加の傾向は止まらなかった．これは，稼働を終了するホストがある一方で，新たに別の IP アドレスで稼働するホストもあるためと思われる．このように，数は少なくなっても日々新たな開放ポートを有するホストが検知される．本システムを常時稼働することで，こうした現状にも対応できる.

4.4 検知された開放ポート

本システムで検知された開放ポートは，ホストによって異なる．検知された開放ポートのうち，検知されたホスト数が多い 8 ポートについて，検知されたポート番号と検知ホストの台数，そして，検知ホストのうち毎日検知されたホストを表 1 に示す.

表 1 開放ポートごとの検知台数（上位 8 ポート）.

ポート番号	検知台数	毎日検知
80	169	151
443	106	100
25	51	51
22	39	35
53	38	32
110	37	37
143	31	31
587	27	27

HTTP で用いられる TCP80 が検知されたホストが最も多く，検知された 270 台の内部ホストの 62.6%であった．毎日検知されたホスト台数との差にあたる 18 台のホストは，期間の中で一部の日のみ稼働していたことになる．HTTPS で用いられる TCP443 は次に多く，検知ホストのほとんど（106 台中 100 台）が毎日稼働していた.

メールの送受信で用いられる TCP25（SMTP），TCP110（POP3），TCP143（IMAP4），TCP587（Submission）は常時稼働しないと利用者にメールサービスを提供できないことから，すべての検知ホストで毎日検知されている．期間内に新たにメールサービスを開始したホストはなかったものと思われる.

DNS で用いられる TCP53 については，前述のとおり，応答頻度の低い DNS サーバにおいて，TCP による DNS 応答が発生しない日があったものと思われる．また，SSH で用いられる TCP22 については，管理用に常時ポートを開放している 35 台のホストのほか，一時的にサービスを稼働したホストが 4 台あったものと思われる.

4.5 検知された外部ホスト

本システムで検知された外部ホストは，TCP の 3 ウェイハンドシェイクの最初の手順である SYN ビットがセットされた TCP パケットを鹿児島大学のグローバル IP アドレスネットワーク（163.209.0.0/16）に送り，2 番目の手順である SYN+ACK パケットを受け取ったと思われるインターネット上のホストである．期間内に検知された日別外部ホスト数を図 5 に示す.

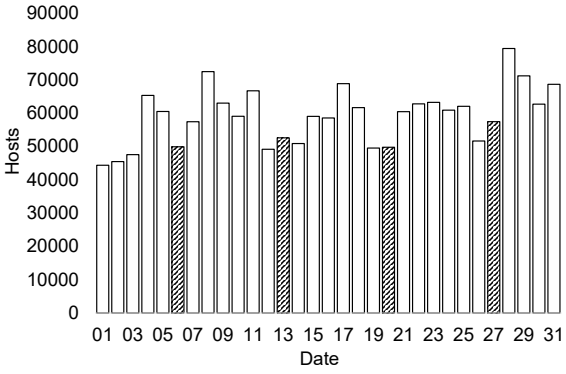

図 5 2019 年 1 月の日別検知外部ホスト数．斜線は日曜日.

検知された外部ホストの合計は 876,688 台，1 日に検知された外部ホスト数の最小値は 44,371 台（1 月 1 日），最大値は 79,452 台（1 月 28 日），平均は 59,134.9 台，中央値は 60,468 台であっ

た．検知される外部ホスト数は日曜日と年始の休暇期間に少なくなる傾向が見られた．図 2 を見る限り，内部ホストでは日曜日や年始に検知数が減少する傾向は小さいため，検知パケット数の日曜日と年始休暇期間中の減少（図 5）は外部ホストの大学に対するアクセスが休日に減少するためと考えられる．

4.5.1 ポートごとの外部ホストアクセス状況

外部ホストがアクセスした開放ポート上位 16 ポートを表 2 に示す．

表 2 外部ホストがアクセスした開放ポート上位 16 ポート．

順位	開放ポート	外部ホスト数	内部ホスト数
1	80	445,034	169
2	443	327,223	106
3	25	214,526	51
4	8081	26,065	3
5	22	20,325	39
6	8080	13,609	19
7	587	8,285	27
8	995	8,104	19
9	81	7,482	26
10	993	4,720	20
11	110	3,785	37
12	53	2,655	38
13	3389	1,890	5
14	465	1,792	20
15	143	1,280	31
16	21	1,185	19

HTTP が使用する TCP80，HTTPS が使用する TCP443，SMTP が使用する TCP25 に対してアクセスする外部ホストが多い．これらのポートには，多くの内部ホストで稼働している Web サーバ，メールサーバに対する通常アクセスが多く行われるほか，スキャン攻撃の対象となっているものと思われる．

外部ホスト数に比べて内部ホスト数が少ない TCP8081 は，ライブカメラを運用している内部ホストである．TCP3389 はリモートデスクトップサービス（RDP）が使用するポートで，マルウェアにより多数のスキャンが行われていることが知られている．リモートデスクトップサービスを有効にしている内部ホストに対して，多数のスキャンが行われていることが考えられる．

また，TCP81 はいくつかの内部ホストで Web サーバが稼働している．運用は行われていないが外部ホストからのスキャンの対象になっているものと思われる．その他の上位ポートはいずれも内部ホストで運用されている TCP アプリケーションに対するアクセスであり，表 2 に挙げていない下位ポートも含め現在のところ脆弱性やセキュリティ侵害は確認されていない．

4.6 ホストごとの開放ポート数

開放ポートが検知されたホストごとに，検知された開放ポートや開放ポートの数は異なるため，ホストごとの開放ポートの数の分布を調べた．結果を表 3 に示す．

表 3 1 台あたりの開放ポート数と該当ホストの台数

開放ポート数	ホスト台数
1	88
2	74
3	32
4	15
5	4
6	4
7	10
8	13
9	2
10	8
11	1
12	1
13	12
14	5
59	1

ホストごとの開放ポート数の最小は 1 ポート，最大は 59 ポート，平均は 3.8 ポートであった．開放ポートが 1 ポートと 2 ポートで全検知ホストの 60.0%を占めた．また，開放ポートの数が 10 ポート以上の開放数が同じホストにおいて，同一のポートを開放している複数のホストがある．これらのホストについては，同一組

織もしくは同一管理者によるサーバ運用が推定される．また，開放ポート数が 59 ポートのホストは，他のホストと利用形態が異なる例外的なホストである．こうしたホストが検知された場合には，利用者と連絡を密にして詳細の把握に努めることにしている．

5. 脆弱性診断との比較

鹿児島大学では数カ月に 1 回，インターネット上のホストから Nessus[2] を用いた脆弱性診断を行っている[3]．これによる開放ポートの検知について，本システムとの比較を行う．Nessus は開放ポートの検知に際して，ハーフスキャンを行うため，脆弱性診断に用いるインターネット上のホストに返る通信は本システムですべて観測されることになる．このため，本システムで記録されたパケットのうち，脆弱性診断に用いるホストの IP アドレスが外部ホストとなっているパケットと，そうでないパケットを比較した．

5.1 開放ポートが検知されたホスト数

脆弱性診断：162
本システム：270

脆弱性診断は対象セグメントを予め定めて，ホストスキャンとポートスキャンを行う．時間がかかることもあり，対象セグメントを必要最小限としている．このため，対象外となっているホストがある．これに対して，対象ホスト数が増えても要する時間は変わらない本システムは，すべてのホストを対象としている．この違いが現れたものと思われる．

5.2 検知されたポート数

脆弱性診断：147
本システム：207

ポートスキャンツールによるスキャンは，高速化のために待ち時間を短くするなどの工夫がされている場合が多い．本システムは実際に SYN+ACK パケットが返ったかどうかを確認しているので確実に開放ポートを観測できる．脆弱性診断に際しては，ポートスキャンにおいてタイムアウトにより開放ポートを見逃したか，診断時には開放されいなかったポートが新たに開放されたものと思われる．こうした場合でも本システムなら開放ポートを観測できる．

5.3 ポートあたりホスト数

脆弱性診断：1〜89
本システム：1〜169

いずれも最大値は開放ポートとして TCP80 が検知された内部ホストの数である．本システムによる値の一部については表 1 にも示している．本システムは，検知されたホスト数，ポート数とも脆弱性診断より多く，その結果，ポートあたりのホスト数もより多くなった．

5.4 ホストあたりポート数

脆弱性診断：1〜57
本システム：1〜59

最大値は 4.6 節で述べた特殊なホストである．脆弱性診断では 57 ポートしか検知されなかったが，本システムでは 59 ポートを検知できた．差の 2 ポートについては，間欠的に記録された以下の 3 パケットにより検知された（当該ホストの IP アドレスは `163.209.x.y` としている）．

```
date,hour,saddr,sport,daddr,dport
13,00,163.209.x.y,33020,185.176.27.26,40269
15,22,163.209.x.y,60912,5.8.18.62,46266
16,11,163.209.x.y,60912,176.119.4.26,45228
```

これら 3 つのパケットのうち，1 番目のパケットの外部ホスト `185.176.27.26` はロシア，2 番目のパケットの外部ホスト `5.8.18.62` はモルドヴィア，3 番目のパケットの外部ホスト `176.119.4.26` はウクライナのホストである．ロシアとウクライナのホストについてはスパムブラックリストに登録がある IP アドレスであるため，マルウェア等に感染したホストであるものと思われる．

以上の脆弱位診断との比較では，ポート数，ホスト数，ポートあたりホスト数，ホストあたりポート数のすべてについて，本システムの方が検知数が多い結果となった．開放ポートの検

知であれば，脆弱性診断ツールを用いるより，本システムを用いる方がよい．

6. まとめ

開放されているTCPポートを能動的にスキャンする方法ではなく，キャンパス情報ネットワークと外部の間の通信を監視することにより，ネットワーク装置やセキュリティ装置に負荷を与えることなく受動的に開放ポートを調べる方法を確立し，システムを構築して開放ポートの検知を行った．

システムは有効に機能し，能動的方法に比べてホスト，ポートとも，より多くを検知することができた．本システムは開放ポートの検知に有効であると言える．また，本システムは，新たな開放ポートが生じた場合に利用者からの申告なしに検知することができる．このため，組織のポリシーに応じた早期の対策，対応を実現することもできる．鹿児島大学においては，申告によりTCPポートを閉鎖する設定による運用を廃止し，すべての内部ホストについて，全ポートを閉鎖し申告によりTCPポートを開放する設定による運用にすることを検討しており，本システムで得られた情報はその準備に利用することもできる．

なお，本システムはパケットの記録までを行うものであり，任意の方法やタイミングで記録を確認できる．本論文における集計はデータベース（SQLite）を用いてSQL文を実行することで行ったが，データを形式を変更して移動することで，別のデータベースや統計ソフトウェア等での集計も実現できる．

今後は引き続き日々の運用と随時の分析を継続すると共に，分析を自動化し，新規開放ポート検知時や開放ポートの状況に変化が生じた場合に通知を行う機能を追加することにしている．

参考文献

(1) Nmap Security Scanner, https://nmap.org

(2) Nessus Professional, https://www.teneble.com/products/nessus/nessus-professional

(3) 相羽俊生，川原智徳，高橋至，小田謙太郎，古屋 保，下園幸一，佐藤豊彦，升屋正人，森邦彦，“学内サーバの脆弱性診断と診断結果の解析方法”，学術情報処理研究，No.20，pp. 105–111，2016.

(4) Nessus Network Monitor, https://www.teneble.com/products/nessus/nessus-network-monitor

(5) Shodan, https://www.shodan.io

(6) Censys, https://censys.io

(7) Masato Masuya, Takashi Yamanoue, Shinichiro Kubota, “An Experience of Monitoring University Network Security Using a Commercial Service and DIY Monitoring,” Association for Computing Machinery (ACM) Special Interest Group on University and College Computing Services (SIGUCCS) 2006 Fall Conference, Nov. 5–8, 2006.

(8) ARGUS – Auditing Network Activity, https://qosient.com/argus/

(9) SQLite, https://www.sqlite.org

著者略歴

升屋　正人　1991年東京大学理学部卒業，1996年同大学院農学生命科学研究科博士課程修了，同年4月岡崎国立共同研究機構分子科学研究所非常勤研究員，1997年11月鹿児島大学工学部情報工学科助手，2000年4月同大学総合情報処理センター助教授，2003年4月同大学学術情報基盤センター助教授，2006年11月同教授．博士（農学）．

下園　幸一　1991年九州大学工学部情報工学科卒業，1993年同大学院工学研究科情報工学専攻修了，同年4月九州大学情報処理教育センター助手，1998年2月鹿児島大学法文学部経済情報学科講師，2000年4月同助教授，2007年7月同大学学術情報基盤センター准教授．修士（工学）．