Real-time Cyber-attack Detection Method based on Darknet Traffic Analysis by Graphical Lasso

Chansu Han*†, Jumpei Shimamura‡, Takeshi Takahashi*, Daisuke Inoue*, Masanori Kawakita†,
Jun'ichi Takeuchi†*, and Koji Nakao*
*National Institute of Information and Communications Technology, Japan. {han, takeshi_takahashi, dai, ko-nakao}@nict.go.jp
†Kyushu University, Japan. {tak}@inf.kyushu-u.ac.jp
‡clwit Inc., Japan. {shimamura}@clwit.co.jp

***Abstract*—An increasingly evolving of cyber-attacks from malware has recently been causing severe security incidents, making cyberspace less secure.** 亜種や未知の攻撃を迅速かつ正確に対応可能な攻撃検知手法の研究が重要となっている．**Our earlier method monitors network traffic arriving at a darknet, estimates cooperativeness of the source host pairs of this traffic and** 異常に協調性が多いトラフィックを外れ値検知する．ダークネットトラフィック上で異常なほどのホスト同士の協調性が見られるということの原因の一つは，インターネット上を無差別にネットワークスキャンするような攻撃である．手法は攻撃検知に有効だが，リアルタイムに分析ができない．本研究では，オンライン処理アルゴリズムを提案し，リアルタイムに攻撃検知可能にする．また，いくつか重要なパラメータのチューニングを行い，提案手法を実装運用し，本手法のパフォーマンス評価を行う．限定的な **ground truth** を用いて，攻撃検知精度を評価した結果，正解率 **91.2%**，適合率 **100%**，再現率 **91.2%**，**F** 値 **95.4%**という結果を得た．

***Index Terms*—Real-time cyber-attack detection, Darknet traffic analysis, Outlier detection, Cooperativeness.**

I. はじめに

Cyber-attacks have continued to grow and become diverse recently, and the number of variants and unknown attacks is increasing. どのようなサイバー攻撃でも，その対策として，インターネット上で実際に行われている攻撃活動を迅速かつ正確に把握することは重要である．ネットワークベースでのアクセス制御及び観測により，実際の攻撃を迅速かつ正確に対応する従来方法として，IDS,FW によるシグネチャーやホワイトリスト/ブラックリストベースのアクセス制御方法がある．しかし，一般にこの従来方法では亜種や未知の攻撃への対応は期待できない．また，従来方法としてセキュリティオペレーターなど専門家によるヒューリスティックなルールベースの観測･対応方法もあるが，人為的ミスが起こり得る．上記のような理由から近年，亜種や未知の攻撃を迅速かつ正確に対応可能，かつ人為的ミスもない攻撃検知手法の研究が重要である．

そこで，我々は不特定多数のインターネットユーザーに対し攻撃･侵入するような無差別型攻撃の検知に注目した．(Internet-Wide Scan) 我々が保有している未使用 IP アドレスブロックのダークネットに応答を返さない受信機 (センサ) を設置し，届く全てのトラフィックを観測し，データセットとして用いる．現在世の中に流行しているサイバー攻撃をより流行らすために，無差別にばらまかれるネットワークスキャンが，ダークネットに多く届く．そのため，ダークネットを分析すると，どのようなネットワークサービスが狙われているのか，大局的なサイバー攻撃の傾向を把握することが容易にできる．ただし，我々が運用しているダークネットセンサは応答を返さないため，人間の目では受信したパケットが攻撃を意図したものか判別することは難しい．従って，本研究ではダークネットに届くネットワークスキャンのような無差別型攻撃を機械的に検知する研究に取り組んだ．本研究はダークネットに届くネットワークスキャンのような攻撃を，亜種や未知の攻撃を含み，人為的ミスもないように教師なし機械学習手法を用いて，迅速かつ正確に検知する手法を提案する．

我々の先行研究 [8] ではボットネットを成す感染ホストが C2 サーバーから司令を受け取ると同期して振る舞う特徴があることから [1]，マルウェアに感染された複数のホストが関係しあって，同時期にネットワークスキャンするような攻撃の検知に着目している．本手法は “glasso”という R 言語のライブラリを使っていることから [7]，便宜上以下からは本手法を GLASSO エンジンと呼ぶ．GLASSO エンジンは，スパース構造学習アルゴリズム “Graphical Lasso”を用いて，ダークネットトラフィックにおけるある時間帯の中での，各送信元ホストから受信したパケット数の時間傾向 (時刻パターン?) から，全送信元ホスト対の協調性を推定する [6], [11]．そして，その時間帯における協調性の度合いを数値化し，他の時間帯と比べてその数値が異常に高い (外れ値) 時間帯に対して，複数のホストが強く協調したイベントが行われたと判定する異常検知手法である [8]．ダークネットトラフィック上で異常なほどのホスト同士の協調性が見られるということの原因の一つは，インターネット上を無差別にネットワークスキャンするような攻撃であり，GLASSO エンジンはそのような攻撃が検知できる．ここで言うホスト対の協調性とは，ある 2 つホスト対のパケット受信数の時間傾向が条件付き独立である場合，そのホスト対に協調性はないことを意味する．この協調性は Graphical Lasso アルゴリズムで推定できる．また，Graphical Lasso は，たまたま関係し合ったホスト対の弱い協調性は削ぎ落とすことが期待できる．従って本エンジンは，ダークネットに届く誤設定による通信と，たまたま関係し合ったホスト対の弱い協調性は考慮されないようになり，より本質的な協調性を推定することが期待できる．

しかし，まだ先行研究 [8] での GLASSO エンジンはリアルタイム処理ができず，処理するためには 3 日分のダークネットトラフィックデータを要するため，処理時間まで合わせると，結果の出力まで 3 日以上遅延していた．つまり，

先行研究では迅速な対応ができるとは言えない．従って，本稿では GLASSO エンジンのオンライン処理アルゴリズムを提案し，逐次的かつリアルタイムにダークネットトラフィックの異常検知を可能にする．また，処理時間を短縮し，よりスケーラブルに処理できるように様々な工夫を行う．そして，GLASSO エンジンを実際にリアルタイムに運用し，その検知結果の評価を宛先 TCP ポート別 (サービス別) で行う．ダークネットに届くパケットが攻撃を意図したものか判別することは難しいため，通常ダークネットにおける攻撃の Ground Truth を出すことは難しいが，評価のために我々の分かる範囲での正解表を作成し，限定的な Ground Truth を用いて評価を行う．その結果，検知正解ポートは 31 個，見逃しポートは 3 個，誤検知ポートは 0 個となり，ダークネットにおいて GLASSO エンジンは次のようなネットワークスキャンを検知できることが分かった: 自己増殖するようなワーム型マルウェアや Mirai，Hajime のようなボットネットを形成し IoT 機器を狙うマルウェアに感染された多数の機器が次の感染対象を探索するスキャン (distributed scanning from infected botnet hosts)．本研究はネットワークスキャンのような無差別攻撃をリアルタイムかつ自動で迅速かつ正確に把握することが可能であり，ネットワークオペレーションの負担の軽減に繋がると考えている．

II. 背景知識

本節で理解を深めるためにダークネットと graphical Gaussian model と Graphical Lasso アルゴリズムを説明する．

A. ダークネット

インターネットは大きく使用中 IP アドレスブロックのライブネットワークと未使用 IP アドレスブロックのダークネットに分かれる．我々の組織が保有するダークネット空間に応答しない受信機 (センサ) を設置し，届く全てのトラフィック (raw network packets) を pcap 形式でキャプチャーしている．ダークネットに届くトラフィックの大部分は TCP パケットであり，そのほとんどは SYN フラグのパケットである．ダークネットに届く SYN フラグのパケットには，誤設定などによる何かの間違いやインターネット空間を無差別に攻撃･侵入･調査を試すネットワークスキャンがある．ダークネットに届く SYN フラグのパケット以外には，誤設定などによる何かの間違いや IP spoofing を原因とするバックスキャッターがある．本稿では無差別に攻撃･侵入を試すようなネットワークスキャンをサイバー攻撃と呼び，shodan のような調査目的組織により無差別に調査を試すネットワークスキャンは survey scan と呼ぶ．

B. Graphical Gaussian Model

A graphical Gaussian model(GGM) is a probabilistic model for which a graph expresses the dependence structure between random variables given a multivariate Gaussian distribution. 変数間の依存構造を測る方法として相関係数を求めるのは一つの方法だが，変数間の疑似相関 (spurious relationship) が含まれる問題がある．変数間の疑似相関を含まない依存構造を測る方法として，変数対の条件付き独立性が分かる精度行列 Σ^{-1} を求める方法がある．If and only if when $\Sigma^{-1}_{ij} = 0$, then x_i and x_j are independent conditioned on all the other variables. N 次元多変量正規分布に従う確率変数列の精度行列 $\Sigma^{-1} \in \mathbb{R}^{N\times N}$ を用いた GGM におけるグラフの定義は，N 個の変数のそれぞれを頂点とし，Σ^{-1} の行列要素がゼロなら辺なし，非ゼロなら辺ありである [11]．つまり，この精度行列を用いた GGM におけるグラフは全変数対の条件付き独立性を表すグラフになる．

C. Graphical Lasso

精度行列 Σ^{-1} は標本共分散行列 S の逆行列である．我々の望みとして，精度行列 Σ^{-1} の要素は，本質的な依存関係にある変数対に対しては非ゼロの値を取り，おそらくはノイズにより弱く関係しあっているだけの変数対に対してはゼロとなるような，スパースな行列になることを期待する．しかし，一般に標本共分散行列 S の要素が厳密にゼロになることはありえず，また精度行列 Σ^{-1} も一般にはスパースにならない．そこで，スパース構造学習アルゴリズムである “Graphical Lasso” は，精度行列を 1 列 (1 行) ずつ ℓ_1 正則化項付き最尤方程式を解いて最適化し，スパースな精度行列 $\hat{\Sigma^{-1}}$ を明示的な逆行列計算なしに推定することができる [6]．Graphical Lasso アルゴリズムの入力パラメータは標本共分散行列 S と ℓ_1 正則化項係数 $r \in \mathbb{R}$ (≥ 0) である．ここで，r はどの程度の依存関係までノイズ由来のものとみなすか決める閾値であり，推定する精度行列のスパーシティを調整できる．以上より，Graphical Lasso アルゴリズムで推定した精度行列 $\hat{\Sigma^{-1}}$ を用いた GGM のグラフは，全変数対の条件付き独立かつより本質的な依存関係を表現することができる．

III. Related Work

Many studies using darknet have been carried out, and shew its usefulness on analyzing Internet-wide scanning. Dainotti *et al.* developed and evaluated a methodology for removing spoofed traffic from both darknets and live networks, and contributed to support census-like analyses of IP address space utilization [3]. Durumeric *et al.* analyzed a large-scale darknet to investigate scanning activities, and identified patterns in large horizontal scanning operations [4]. Also, they presented an analysis of the latest network scanning on the overall landscape, and its influence, and countermeasures of the defender in detail. Fachkha *et al.* devised inference and characterization modules for extracting and analyzing cyber-physical systems (CPS) probing activities towards ample CPS protocols by correlating and analyzing various dimensions of a large amount of darknet data [5].

GLASSO エンジンはダークネットに届くホスト間の協調性を捉え，攻撃を検知するエンジンである．Most similar to our work is a study by Ban *et al.* [2], who proposed an abrupt-change detection algorithm that can detect botnet-probe campaigns with a high detection rate by exploring the temporal coincidence in botnet activities visible in darknet traffic. しかし，この abrupt-change detection algorithm で用いるデータセットは宛先ポートを一つに絞ったトラフィックに対して処理を行い，botnet-probe 活動を検知する．GLASSO エンジンは宛先ポートを絞らずに処理可能であるため，abrupt-change detection algorithm とデータセットの範囲が異なる．実は，ダークネットを用いて GLASSO エンジンと同様なデータセットの範囲で，同様なスケールのサイバー攻撃を検知す

るような研究は，現状我々が知る限り存在せず，比較評価が難しい．

IV. Earlier Work

本節では，先行論文 [8] はどのように GGM をダークネットトラフィックデータに適用したか説明し，その時の GLASSO エンジンのアルゴリズムとその欠陥を述べる．

A. Applying GGM to Darknet Traffic

GLASSO エンジンはダークネットに届く送信元ホストのパケット数の時間傾向を変数とし，GGM を適用して変数間 (ホスト間) の依存関係を見たい．この変数は決してガウス分布で表されるようなものではないが，対数正規分布の形に近い変数が多いと想定できるため，対数変換 (log-transformation) すると，ガウス分布にある程度似ているようになる．また，変数がガウス分布に完全に従わなくてもある程度近似できていれば，GGM における変数間の依存関係は捉えられる．

まず，ダークネットトラフィックをどのようなデータセットに加工するか考える．一つのモデル学習に用いる T 秒間のダークネットトラフィックをタイムスロットと呼ぶ．あるタイムスロット t に N 個のユニークなホストがあるとする．各ホスト毎にあるサンプリング間隔で観測されたパケット数を計数した時系列データを作成する．ここで時系列サンプルの数が M 個にすると，サンプリング間隔は $T/M(sec.)$ となる．そうするとタイムスロット t からデータ行列

$$\boldsymbol{D}_t = [D_{mn}] \in \mathbb{R}^{M\times N},\quad D_{mn} := \log(x_n^{(m)}),\quad \boldsymbol{x}^{(m)} \in \mathbb{N}_0^N$$

に変換できる．ここで $\boldsymbol{x}^{(m)}$ はサンプル数 M 個の N 次元変数であり，$x_n^{(m)}$ は n 番目のホストの m 時点目のパケット数を表し，$x_n^{(m)} = 0$ の時は対数変換ができないため，適当な値 $x_n^{(m)} = 0.1$ に変換する．また，$\mathbb{N}_0 = \{0, 1, 2, \cdots\}$ である．

次に，Graphical Lasso アルゴリズムを用いてデータ行列 $\boldsymbol{D}_t$ から精度行列を得て GGM を適用する．そうすると，GGM におけるグラフのノード集合には送信元ホスト (変数) 集合が対応し，エッジ集合は送信元ホスト対 (変数対) の依存関係の有無が対応する．本稿でのホスト対の協調性の定義は，ある送信元ホスト対に GGM における依存関係がある場合，つまりその対のパケット受信数の時間傾向が条件付き独立ではある場合，そのホスト対に協調性があると呼ぶ．逆に，GGM における依存関係がない，条件付き独立である場合，協調性はないと呼ぶ．

B. GLASSO エンジンのアルゴリズム

長期間に渡り観測したダークネットトラフィックを用意し，T 秒ごとにトラフィックを分けると複数のタイムスロットができる．その時，TCP の SYN パケットのみを収集する．(we consider only SYN packets, TCP.) 次に，タイムスロットごとにデータ行列 $\boldsymbol{D} \in \mathbb{R}^{M\times N}$ を作成し，標本共分散行列 $S \in \mathbb{R}^{N\times N}$ を求める．この標本共分散行列 S と任意の正の実数 r を Graphical Lasso アルゴリズムに入力し，スパースに推定された精度行列 $\left(\hat{\Sigma^{-1}}\right)^{(r)} \in \mathbb{R}^{N\times N}$ を得る．ここで $r \in R(= \{r_1, r_2, \cdots, r_s\} \in \mathbb{R}^s\ (\geq 0))$ のように，いくつかの正の実数値で試す．その精度行列から GGM における無向グラフ $G = \{V, E\}$ はノード集合 $V = \{x_1, \cdots, x_N\}$，エッジ集合 $E = \{(i, j)|\Sigma_{ij}^{-1} \neq 0\}$ で表現することができる．そして，タイムスロットにおける全ホスト対間の協調性の度合いをスカラー値で表すために，グラフ密度値 $d^{(r)} = |E|/N(N-1)$ をタイムスロットごとに求める．グラフ密度値は完全グラフのエッジ数に対する実際のエッジ数の比率を表す．最後に，複数のグラフ密度値から異常に高い値を示すタイムスロット s を外れ値検知手法を用いて判定し，そのタイムスロット s は他の時間帯と比べて異常なほどホスト間に協調性が捉えられたことが分かる．

C. 先行論文の欠陥

先行論文 [8] での GLASSO エンジンのアルゴリズムは，ホスト間の協調性の異常検知ができ，その有用性を示せたが，3 日分のダークネットトラフィックデータを用意する必要があった．その 3 日分のダークネットトラフィックデータから複数のタイムスロットに分けて，まとめて一気にバッチ処理する．つまり，処理時間まで合わせると，異常検知の結果出力まで 3 日以上遅延することとなり，迅速な対応へ繋がる期待効果は薄いと思われる．また，GLASSO エンジンでは複数パラメータが存在するが，どのような基準でパラメータを選択したのか評価していない．最後に，異常検知した結果をいくつかケーススタディを述べてはいるが，ground truth を用いた評価がなく，実際の処理時間も評価されず，不十分な部分が多かった．従ってこれ以降の本稿では，上記のような先行論文の欠陥を改善していく．

V. Proposed Method

本節では GLASSO エンジンのオンライン処理のために新たに提案したアルゴリズムを紹介する:オンライン処理アルゴリズム，そのためのアラート判定法．ここから以降の GLASSO エンジンは新たに提案する GLASSO エンジンを指す．GLASSO エンジンの入力は T 秒間の PCAP 形式のダークネットトラフィックとその他パラメータ $M, r \in R(= \{r_1, r_2, \cdots, r_s\}), \boldsymbol{d}^{(r)}, K, \theta$，出力は外れ値だと判定されたタイムスロットから生成するアラート情報である．このアラート情報には時間帯 (timestamp) や狙われているポート番号 (service)，その宛先ポートでパケットを送信した送信元ホスト IP アドレスとその数など情報を含む．狙われているポート番号はそのタイムスロットの中での送信元ホストが一番多くパケットを送信した宛先ポートを指す．

A. Algorithm for Online Processing

我々は，先行論文 [8] のように複数のタイムスロットをまとめて一気にバッチ処理するのではなく，一つのタイムスロットごとに処理してグラフ密度値を得て，逐次的にアラート判定をする方法を考えた．まず，あるタイムスロット t を Section IV-B と同様に処理し，グラフ密度値 $d_t^{(r)}$ を得る．そして，過去のグラフ密度値の列 $\boldsymbol{d}^{(r)}$ に $d_t^{(r)}$ を加え，$\boldsymbol{d}^{(r)}$ の長さが K ならアラート判定 (外れ値検知) を行う．$\boldsymbol{d}^{(r)}$ の長さが K 未満の場合は，アラート判定を行わずに次のタイムスロットの更新まで待つ．詳しいアラート判定法は次の節で説明する．アラート判定されたタイムスロットは，アラートを出力し，それ以降のアラート判定に参照されないように $\boldsymbol{d}$ から削除する．また，どのタイムスロットもアラートに判定されなかったら，一番古いタイムスロットを一つ削除する．時間が立ち，新たなタイムスロットが更新されると，上記のステップを繰り返すことで，アラート判定に用いるタイムスロットの数を保ちながら，逐次的に処

Algorithm 1 The GLASSO Engine with Online Processing

Input: t (a time slot), $M, r \in R(=\{r_1, r_2, \cdots, r_s\}), \boldsymbol{d}^{(r)}, K, \theta$
Output: alerts or none

for a time slot t is updated newly **do**
  preprocess a time slot t
  make $\boldsymbol{D}_t$ from a time slot t
  compute S from $\boldsymbol{D}_t$
  for r in R **do**
    compute $(\hat{\Sigma^{-1}})_t^{(r)}$ using the graphical lasso (input: S, r)
    compute $d_t^{(r)}$ from $(\hat{\Sigma^{-1}})_t^{(r)}$
    add $d_t^{(r)}$ to $\boldsymbol{d}^{(r)}$
    if $length(\boldsymbol{d}^{(r)}) = K$ **then**
      run alert judgment method (Algorithm 2)
      if there are *outliers* **then**
        collect alert information from *outliers*
        output alerts
        remove *outliers* from $\boldsymbol{d}$
      else
        remove the most old time slot from $\boldsymbol{d}^{(r)}$
      end if
    end if
  end for
end for

理することが可能となる．そして，更新される一つのタイムスロットだけを処理して結果を得られるため，バッチ処理と比べて非常に短い所要時間で結果を得ることができる．The pseudocode for GLASSO engine with online processing is described in Algorithm 1. Here, *length*() function gets the length of vectors.

B. Alert Judgment Method for Online Processing

In this section, we propose an outlier detection method for discriminating alerts (outliers) from sequence of graph densities $\boldsymbol{d}$ during online processing. コンセプトして，グラフ密度値列 $\boldsymbol{d}$ で最大要素が標本分散にどれほど影響を及ぼすかを見て，外れ値を判定する．その詳しいアラート判定法の擬似コードをアルゴリズム 2 で示す．ここで $\sigma^2_{(i+1)}/\sigma^2_{(i)} < \theta$ は外れ値判定式であり，$\theta\,(0 \le \theta \le 1)$ は外れ値判定式のしきい値である．また，*order*() function returns a permutation which rearranges its first argument into ascending or descending order, *var*() function returns a sample variance. 外れ値だと判定されたタイムスロットから，時間帯 (timestamp) や狙われているポート番号 (service)，その時の送信元ホスト IP アドレスとその数など情報を含んだテキスト形式のアラート情報を取得する．以上より，GLASSO エンジンはどのように処理し，どんなアラートを出力するのか分かった．

VI. Parameter Tuning

GLASSO エンジンにはタイムスロットの長さ T，時系列サンプルの数 M，正則化項係数 r，そしてアラート判定に用いられるタイムスロット (グラフ密度値) の数 K とそのしきい値 θ のように，複数パラメータが存在する．本節では，どのようにこれら 5 つのパラメータを設定したか紹介する．

Algorithm 2 Pseudocode for Alert Judgment Method

Input: $\boldsymbol{d} \in \mathbb{R}^K, K, \theta$
Output: *outliers* or none

$i \leftarrow 0$
while TRUE **do**
  $i \leftarrow i + 1$
  $\boldsymbol{d}_{(i)} \leftarrow order(\boldsymbol{d}, decreasing = True)[i : K]$
  $\sigma^2_{(i)} \leftarrow var(\boldsymbol{d}_{(i)})$
  if $\sigma^2_{(i+1)}/\sigma^2_{(i)} < \theta$ **then**
    $outliers \leftarrow order(\boldsymbol{d}, decreasing = True)[1 : i]$
    return *outliers*
  end if
end while

これらパラメータは経験的ヒューリスティックな方法で決めている．

A. Length of Time Slot T

このタイムスロットの長さ $T(sec.)$ は 1 モデル学習に用いるデータの全観測時系列の長さを意味する．まず，上界 (upper bound) から考える．GLASSO エンジンの計算量は送信元ホストの数に大きく依存する．T が長くなると，一般に送信元ホストの数も増え，処理時間が指数的に増えてしまい，リアルタイム処理が困難になる．次に，下界 (lower bound) を考える．T 秒間でネットワークスキャンの 1 キャンペーンが全て観測されるような，豊富な観測ができていれば十分である．経験上，GLASSO エンジンで用いるダークネットで観測される多くのネットワークスキャンの 1 キャンペーンは，観測規模が最大 29,182IP アドレス (約/17) でも 5 分ほどでポートスキャンが終わる．従って，GLASSO エンジンでは問題なくリアルタイムに処理を行えて，豊富な観測ができると思われる，$T = 600(sec.)$ に設定している．

B. The Number of Time Series Samples M

この時系列サンプル数 M は，タイムスロットの長さ T を M 個に分割し，1 モデル学習に用いるデータの時系列サンプルの数を意味する．この意味は，データ行列 $\boldsymbol{D}_t$ からホストから受信したパケット数の時間傾向を測るときの 1 区間を M 個に設定し，1 区間の長さであるサンプリング間隔 $T/M(sec.)$ をどれぐらいの長さに設定するかを意味する．一般に厳しく区切るほど，パケット数の時間傾向をより厳し目に測ることとなり，良い精度の学習が行われる．しかし，あまりにも区切り過ぎると，本来は協調して動くホスト間に依存関係はないと推定する恐れがある．逆に区切りが少な過ぎると，どんなホスト間にも協調性があると推定される恐れがある．従って，M を適当な数に設定する必要があり，最適な値を探すことは難しいことだが，幸いなことに次の節で紹介する正則化項係数 r はこの M と同様な働きができる．つまり，M を極端な数に設定しなければ，どのように設定しても正則化項係数 r でカバーすることができる．経験上，サンプリング間隔 $T/M = 50(sec.)$ 程度あれば，十分なパケット数の時間傾向を測れることから，$M = 12$ と設定している．

C. Regularization Coefficient r

正則化項係数 $r \in \mathbb{R}$ (≥ 0) は Graphical Lasso アルゴリズムで用いられる入力パラメータであり，どの程度の依存関係までノイズ由来のものとみなすか決めるしきい値であり，推定する精度行列 Σ^{-1} のスパーシティを調整する．時系列サンプル数 M との関係を述べると，r を調整することで弱く関係し合った依存関係を削ぎ落とすことができるため，M を調節してパケット数の時間傾向をより厳し目に測ることと同様な意味を持つ．従って，我々は M を 12 に固定して，r の値を調整することにした．

r の特徴として，一般に r が 0 に近いほど $d^{(r)}$ は 1 に近づき，r が大きくなるほど $d^{(r)}$ は 0 に近づく．また，graphical lasso アルゴリズムは r の値を大きくするほど，ゼロ要素が多くなるため，計算時間が短くなる特徴がある．最後に $r \in R(= \{r_1, r_2, \cdots, r_s\} \in \mathbb{R}^s)$ のように設定し，何度でも試行可能だが，その試行回数分だけ GLASSO エンジンの処理時間は伸びる．

我々は GLASSO エンジンを $T = 600(sec.), M = 12$ に設定し，r の値を少数点 6 桁まで微調整を行ってみた．小数点 6 桁から 2 桁までいろいろ値を変えながら試してみても，グラフ密度値は大きく変わることはなかった．例えば，$r = 0.55$ でアラート判定されるタイムスロットは，ほとんどの場合 r =0.5 か 0.6 でもアラートに判定されることが経験から分かった．このことから小数点 1 桁で値を変えながら試行すれば良いことが分かった．また，$r \geq 1$ からはゼロ行列が多くなり，$d^{(r)} = 0$ となる場合が多かった．そして，$r < 0.4$ では処理時間が長くかかること，十分にスパースな精度行列が推定されないこと，ほとんどの場合外れ値が出てこないことから，最終的に $r \in R(= \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ に設定することにした．

D. The Number of Graph Densities K and Threshold θ for Alert Judgement

アラート判定に用いるグラフ密度値の数 K としきい値 θ は，時系列サンプル数 M と正則化項係数 r の関係と同様に，K を極端な数に設定しなければ，θ でカバーすることができる．K は外れ値検知するときに用いるデータの数を意味する．このデータの数が少な過ぎると，参照できるデータが少なくなるため，外れ値検知が不安定になりやすい．逆に多すぎると，安定し過ぎて外れ値が検知されにくくなる．しかし，この安定さは外れ値判定式 $\sigma^2_{(i+1)}/\sigma^2_{(i)} < \theta$ のしきい値 $\theta (0 \leq \theta \leq 1)$ でも調整できる．θ は 1 に近いほど外れ値を緩く判定し，0 に近いほど外れ値を厳しく判定する．我々は 3 日分のデータを外れ値検知に用いることにし ($K = 432$)，θ は試行錯誤の上，現在 0.98 に設定している．

VII. Performance Evaluation

本節では GLASSO エンジンのパフォーマンス評価を行う．最初に，GLASSO エンジンのパフォーマンスを向上させるために，我々にとって興味のないホスト間の協調性を推定することなく前処理段階であらかじめ除外する工夫を述べる．次に，実際にリアルタイムに GLASSO エンジンの運用を行い，検知したアラート結果を分析する．そして，攻撃検知精度を評価し，検知した攻撃の詳細を述べる．

TABLE I
IPs Size and the Number of Alerts of Each Darknet Sensors

Sensor	IPs Size	# of Alerts	Sensor	IPs Size	# of Alerts
A	29,182	122	E	8,188	198
B	14,593	199	F	16,384	115
C	4,098	146	G	2,044	118
D	4,096	460	H	2,045	276

A. 前処理の強化

あるタイムスロット t に対して，既存と同様に TCP の SYN パケットだけを収集する．次に，ある TCP ポートに対して長期間に渡り定常的に膨大な数のパケットや送信元ホストからパケットが観測されている，もしくは複数回に渡りアラートとして観測されているポート宛のパケットは除外する．定常的に膨大な数のパケット･ホストが見られる宛先ポートは誰にでも簡単に気付くことができ，一般にそのような宛先ポートはしばらくの間は注目して観測･オペレーションを行う．また，複数回に渡りアラートとして観測されいる宛先ポートは，GLASSO エンジンでの学習対象にしなくてもしばらくの間は注目して観測･オペレーションを行うことが想定される．さらに，このような宛先ポートが GLASSO エンジンのモデル学習時に含まれていると，推定するホスト間の協調性の大半をこれらが占めてしまい，それより小さい規模の協調性を見逃す恐れがある．GLASSO エンジンにおいて学習に悪影響を及ぼす，かつ興味のないこのようなパケットはノイズとしてみなし，除外すべきである．このようにパケットを除外することは，送信元ホストの数がある程度減ることになり，処理時間の短縮にも繋がる．このように除外する宛先ポートを定期的に自動更新する．

B. GLASSO エンジンのリアルタイム運用

本節では，2018 年 10 月の 1 ヶ月間運用を行った GLASSO エンジンの結果を示す．実験で使った GLASSO エンジンのパラメータは $T = 600$ 秒，$M = 12$，$K = 432$，$\theta = 0.98$，$r \in R(= \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ に設定した．前処理として，TCP ポート 22，23，80，81，445，2323，3389，5555，8080，50382，50390，52869 番を予め除外した．これらは 2018 年 10 月 1 日の時点で定常的に膨大なホストまたはパケットが観測されるポートである．我々は 8 つの異なるダークネットセンサを用いて，それぞれに対して GLASSO エンジンを 1 ヶ月間リアルタイム運用した．このダークネットセンサはそれぞれ異なる IP アドレスブロックで観測しており，IP アドレス観測規模や設置国 (source country) も異なる．運用結果，合計 1,634 個のアラートをリアルタイムに得ることができた．8 つの各センサの観測 IP アドレス規模とセンサ別アラート数を表 I に示す．

C. アラート結果分析

アラート情報には一つの狙われているポート番号情報が含まれていて，1,634 個のアラートの中で合計 128 種類のポートが得られた．我々はポート別にアラートを調べることにした．2018 年 10 月の 1 ヶ月間 GLASS エンジンから得られたアラートをポート別に調べると，大きく次のような 3 種類が分けれることが分かった．

TABLE II
GLASSO エンジンの 2018 年 10 月における運用結果をポート別に 3 種類を分けた結果

Alert Type	TCP ports (The Number of Alerts, First Detected Date)
Cyber-attack (1,482 Alerts) (31 Ports)	21(13, 13:40 20th), 82(56, 10:10 7th), 83(9, 20:00 11th), 84(8, 00:30 12th), 85(25, 11:40 6th), 88(100, 18:20 1st), 110(3, 14:50 17th), 443(143, 19:30 12th), 1701(1, 04:30 9th), 2480(4, 20:10 14th), 5358(309, 21:30 24th), 5379(27, 13:50 31st), 5431(26, 10:20 3rd), 5900(2, 21:40 31st), 5984(3, 03:20 20th), 6379(4, 18:20 26th), 7379(25, 13:30 31st), 7547(27, 09:50 20th), 8000(78, 21:40 5th), 8001(47, 22:40 5th), 8081(267, 21:00 10th), 8088(7, 06:10 2nd), 8181(69, 01:30 1st), 8291(17, 14:40 5th), 8443(47, 02:40 20th), 8888(31, 20:30 5th), 9000(11, 01:30 2nd), 23023(5, 07:20 14th), 37215(100, 01:30 1st), 49152(11, 01:10 14th), 65000(7, 03:00 14th)
Survey Scan (57 Alerts) (16 Ports)	17(1, 21:00 31st), 53(12, 01:10 20th), 102(6, 18:12 12th), 111(6, 00:00 27th), 990(1, 21:00 28th), 1900(4, 16:00 28th), 3128(1, 18:20 26th), 3780(2, 21:00 25th), 4567(1, 05:40 28th), 5000(2, 01:30 31st), 5357(1, 17:30 31st), 5560(1, 10:30 30th), 7657(1, 16:00 25th), 9200(2, 22:40 28th), 9981(1, 02:30 26th), 11211(15, 00:50 25th)
One-dst Centralized (95 Alerts) (81 Ports)	99(1, 21:00 25th), 139(3, 14:00 28th), 321(1, 17:20 26th), 792(1, 20:40 25th), 1678(1, 20:10 28th), 1859(1, 19:20 25th), 3227(1, 23:00 24th), 3407(1, 19:30 27th), 4466(5, 19:40 31st), 5601(1, 17:00 27th), 5777(1, 20:00 28th), 6821(1, 04:40 27th), 7199(1, 00:10 27th), 8096(1, 22:30 31st), 8185(1, 19:20 28th), 8983(1, 04:10 31st), 10994(1, 14:10 30th), 11647(1, 01:11 31st), 11876(1, 10:00 25th), 12385(1, 18:40 28th), 13750(1, 20:20 31st), 13804(1, 06:20 26th), 14401(1, 17:20 31st), 16964(1, 13:20 25th), 17396(1, 15:00 28th), 17502(1, 06:50 20th), 19533(1, 05:10 26th), 20340(1, 23:30 26th), 20382(1, 19:30 31st), 20405(1, 21:40 28th), 21221(1, 03:30 27th), 21490(1, 21:00 27th), 22063(1, 01:10 26th), 24357(1, 14:40 26th), 25024(1, 17:50 25th), 25476(1, 05:20 28th), 26137(1, 01:30 29th), 26644(1, 22:40 27th), 26934(1, 23:40 24th), 27200(1, 14:50 14th), 27910(1, 14:20 20th), 29217(1, 17:10 25th), 31632(1, 19:20 29th), 34149(1, 01:00 30th), 35669(1, 17:10 30th), 35927(1, 23:10 25th), 36064(1, 16:20 26th), 36678(1, 06:10 25th), 37822(1, 20:40 25th), 38718(1, 05:30 29th), 39420(1, 00:40 28th), 40500(4, 14:40 27th), 41939(1, 19:30 26th), 43160(6, 12:00 28th), 43361(1, 21:40 29th), 43566(1, 08:30 30th), 46928(1, 17:30 30th), 47149(1, 18:40 20th), 48449(1, 14:40 25th), 49328(1, 05:40 25th), 49516(1, 02:00 25th), 52204(1, 20:20 27th), 52854(1, 15:30 30th), 53518(1, 19:10 29th), 53557(1, 02:40 20th), 55186(1, 08:20 26th), 56011(1, 00:50 21st), 56409(1, 20:50 27th), 56499(1, 21:10 24th), 57343(1, 20:50 29th), 57762(1, 07:10 26th), 59751(1, 00:50 28th), 60850(1, 22:30 24th), 60917(1, 10:20 20th), 60928(1, 05:10 31st), 62627(1, 14:40 29th), 63591(1, 13:50 26th), 63918(1, 01:00 26th), 65032(1, 18:00 29th), 65165(1, 01:20 28th), 65238(1, 14:10 28th)

1) Cyberattack: 既にマルウェアに感染された複数のホストが次の感染対象を探索するために，脆弱なポートに対して無差別にネットワークスキャンする行為
2) Survey scan: Shodan, Censys など組織が複数のホストを用いてあるポートを調査･研究目的に無差別にネットワークスキャンする行為
3) One-dst centralized: 何らかの原因により突発的に一つのダークネット宛先 IPs のあるポートに複数のホストからパケットが集中する現象

1 つのアラートにある複数の送信元ホストは異常なほど協調性があることに注意し，以下にどのように上記のような 3 種類にアラートを分けたか述べる．

上記 3 種類のうち，一点集中型以外の 2 種類は主にスキャンが目的であるため基本幅広い宛先 IPs で観測されるが，一点集中型は特定の宛先 IPs に複数のホストからパケットが集中するため，明らかに他の 2 種類と傾向が異なる．すなわち，一点集中型とその他 2 種類の間に包含関係はない．あるアラートの中で，最も多い dst IP 宛のパケット数と全パケット数の割合と最も多い dst IP 宛にパケット投げたホスト数と全ホスト数の割合が共に 70%より大きいとき，そのアラートは一点集中型だと判定する．この基準を全アラートに適用した結果，計 1,634 個のアラートのうち 95 個のアラートが一点集中型によるアラートであることが分かり，計 128 個のうち 81 個のポートを一点集中型グループに分けれた．今回この一点集中型が何を意図した通信か，我々の手元のダークネットだけでは正確に把握することはできず，ルーティングミスの可能性があるというぐらいの弱い推測しかできない．しかし，本稿は攻撃検知を対象としているため，この一点集中型は考慮対象外とする．

次に，残りのアラート 1,539 個 (ポート別 47 個) は複数のホストが協調して複数の宛先 IPs のある特定のポートへパケットを送っていることからネットワークスキャンであることが想定できる．ダークネットに届く TCP の SYN パケットのネットワークスキャンには攻撃と survey scan，この 2 種類がある．Survey scan も攻撃と同様にネットワークスキャンではあるが，それぞれ通信の意図･目的が異なるため，攻撃と survey scan を区別する必要がある．survey scanner だけによるアラートの場合は，攻撃によるアラートと比べてホストの規模が小さい傾向があることから，我々はホストの規模に着目し，残りのアラート 1,539 個を攻撃と survey scan に分けることを考えた．Survey scanner の多くの場合，ホスト IPs から逆引きして得られるホスト名や HTTP 接続することから survey scanner を判別できる．あるアラートの送信元ホストの中で survey scanner の割合が多い場合，そのアラートは survey scanner によるアラートだと考えられる．survey scanner の割合が 5 割を超えるアラートを調べた結果，ホストの規模が 20 個より小さいアラートは survey scan，ホストの規模が 20 個以上のアラートに survey scanner の割合は多くて 2 割程度であったため攻撃，のように分けることができた．その結果，Survey scan によるアラートは 57 個 (ポート 16 個)，攻撃によるアラートは 1,482 個 (ポート 31 個) に分けれた．2018 年 10 月の 1 ヶ月間で得た全アラートをポート別に 3 種類に分けた結果を表 II に示し，次の節で本題の攻撃検知精度の評価を行う．

D. 攻撃検知精度評価

前節より 2018 年 10 月における GLASSO エンジンの全アラートのうち攻撃対象ポート 31 個だけを分けることができた．本節では，分けたその 31 個の攻撃対象ポートがどれだけ正しいのか，検知精度を測る．正解･不正解を確認するために 2018 年 12 月上旬の時点に従来のヒューリスティックな方法に基づいて我々組織のオペーレーターより分かる範囲で，我々が運用･観測しているダークネットにおける 2018 年 10 月時点での 1 ヶ月間の攻撃を受けたポートの正解表を作成した．その限定的な正解表を用いて，GLASSO エンジンからリアルタイムに検知した攻撃対象ポートの答え合わせを行った．その答え合わせを行った結果を表 III に示す．

TABLE III
攻撃の特徴別に正解表を用いた答え合わせ

攻撃の特徴	正解ポート	見逃しポート	誤検知ポート
IoT マルウェア	82,83,84,85,88, 2480,5358,5984, 7547,8000,8088, 8443,8888,9000 (計 14 個)	444, 8010 (計 2 個)	None
ルータ関連脆弱性	21,110,443,5431, 8001,8081,8181, 8291,23023, 37215,65000 (計 11 個)	None	None
その他脆弱性	1701,5379,5900, 6379,7379,49152 (計 6 個)	2004 (計 1 個)	None
合計	31 個	3 個	0 個

TABLE IV
GLASSO エンジンの攻撃検知精度

Accuracy $\frac{TP}{TP+FP+FN}$	Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{TP+FN}$	F-measure $\frac{2TP}{2TP+FP+FN}$
91.2%	100%	91.2%	95.4%

表 III は攻撃の特徴別に，正解 (True Positive)，見逃し (False Negative)，誤検知 (False Positive) のポート情報を示す．正解表は攻撃を受けたポート，つまり正解だけを記録しているため，このような評価の場合，True Negative はない．その結果，正解ポート 31 個，見逃しポート 3 個，誤検知ポート 0 個という結果になり，表 IV で示されたように，GLASSO エンジンの 2018 年 10 月における攻撃検知の精度は正解率 91.2%，適合率 100%，再現率 91.2%，F 値 95.4%という結果となった．また，正解ポートの中には GLASSO エンジンだけが検知した攻撃ポートが 1 つあった (ポート番号 1701)．これは人為的ミスによりオペレーターが見逃した攻撃事象であった．最後に，現在把握している見逃しが 3 個あるだけで，未知だった攻撃が今後明らかになり，見逃しが増える可能性はあることに注意して欲しい．

E. 検知した攻撃の詳細

本節では GLASSO エンジンで 2018 年 10 月に検知した攻撃の詳細を 3 つの特徴別に紹介する．本稿での攻撃とは前述のように，既にマルウェアに感染された複数のホストが次の感染対象を探索するために，脆弱なポートに対して無差別にネットワークスキャンする行為を指す．攻撃に用いられるマルウェアの種類には，ボットネットを形成し C2 サーバーから司令を受けネットワークスキャンを仕掛けるマルウェアや自己増殖するために広域に渡りネットワークスキャンを仕掛けるワーム，コンピュータウイルスなどのマルウェアがあると考えられる．

1) IoT マルウェア*:* 表 III の IoT マルェアタイプの正解ポート計 14 個は大きく Mirai，Hajime，HNS(Hide and Seek) の 3 つマルウェア種別に分かれる．まず Mirai から見ると，ポート 80 番，8000 番台の Web 系ポートへのスキャン活動が多く，新たな Web 系ポートへスキャン活動を行うように益々攻撃が拡散している．次に Hajime は定常的にポート 5358, 9000 番へスキャンを行っている．最後に HNS はポート 23, 80, 8080, 2480, 5984 番とランダムポートをスキャンすることが報告されている [12].

2) ルータ関連脆弱性*:* 表 III のルータ関連脆弱性の正解ポート計 11 個は大きく 5 つの製造社のルータ製品に存在する脆弱性に対する攻撃に分かれる [9], [13], [14]．A 社のルータ製品に脆弱性が存在し，たくさんのルータがマルウェアに感染した結果，短時間で多数のホストによるネットワークスキャンが 2018 年 10 月にいくつかのポート (21, 110, 443, 8291, 23023, 65000) に対して観測された．他にも B，C，D 社のルータ製品の脆弱性を利用して乗っ取られたルータから Mirai の特徴持つネットワークスキャンが観測された．(ポート 8001, 8081, 8181, 37215) 最後に E 社の UPnP を利用する多数の製造社のルータ製品が，E 社の UPnP の脆弱性を利用しルータ製品を乗っ取るようなネットワークスキャンが観測された．(ポート 5431)

3) その他脆弱性*:* 表 III のその他脆弱性の正解ポート計 6 個は大きく 4 つのサービスに対する脆弱性に対する攻撃に分かれる [10]．ある NoSQL データベースに脆弱性が存在し，そのデータベースを探索するためのスキャン活動が観測された．(ポート 5379, 6379, 7379) その他にも L2TP VPN(Layer 2 Tunneling Protocol Virtual Private Network)，VNC(Virtual Network Computing)，Supermicrio BMC(Baseboard Management Controller) といったサービスを探索するためのスキャン活動が観測された．(ポート 1701, 5900, 49152)

2018 年 10 月における攻撃の多くは，それより以前から恒常的 or 定期的に観測されていた既知のマルウェアもしくは既知の脆弱性を狙った攻撃である．そのような場合，2018 年 10 月の中で攻撃を検知した時期が適切かどうか語るのは大きな意味を持たない．ただし，2018 年 10 月において新たな傾向を示す攻撃に対しては検知時期を評価するのは意味がある．そのような意味では，A 社のルータ製品がいくつかのポートにスキャンを仕掛けたのは 2018 年 10 月において新たな傾向の攻撃であり，GLASSO エンジンはホスト数がピークを迎えたときを，全て適切な時期に検知している．

VIII. 見逃しに対する考察

本節では今回見逃した 3 個のポートに関連する攻撃に対して考察を行う．ポート 444，8010 番に対しては Mirai の特徴を持ったスキャン活動が観測され，ポート 2004 番に対してはある CMS(Content Management System) サービスを探索するためのスキャン活動が観測された．10 分間のユニークホスト数をダークネットセンサ別に見てみると，8010 番は多いときは 161 ホストも観測されている反面，444,2004 番はそれぞれ高々 19,23 個に達しない．

まず，8010 番を見逃した原因を考えると，現在はアラートが発行されたタイムスロットに対してホストの数が 1 番多いポートに対してのみアラートとして扱っていることが考えられる．1 番目以降にホストの数が多いポートが 1 番目のポートと比べてホスト数に差があまりないのであれば，1 番目以降のポートに対してもアラートとして扱うようにすると，ポート 8010 番は簡単に検知できる．しかし，この対策は誤検知を増やす可能性がある．

ホスト数が少ないと協調性は薄くなり，GLASSO エンジンで 444, 2004 番のような事象を捉えることは難しくなる．

このような見逃しの対策として，ホストの数を増やせるような方法や少ない数のホスト間の協調性も捉えられるような方法を考えると，以下の 3 点が考えられる．

1) ホストの数が 1 番多いポート以降の 2 番目
2) ホストの数を縮小せず，第 4 オクテットまでの IP アドレスを 1 つのホストとする．
3) より大きい規模のダークネットセンサを用いる．
4) ポート除外をよりこまめに行う．

上記の対策を適用し，今回見逃した 3 個のポートに関する攻撃が GLASSO エンジンで捉えられるのか確かめることは今後の課題とする．

IX. Conclusion

先行研究での GLASSO エンジンではできなかったリアルタイム処理を可能にし，より迅速な対応が可能なネットワークスキャン攻撃検知エンジンを提案した．GLASSO エンジンから得られるアラートは 3 タイプ (攻撃，survey scan，一点集中型) あることが分かり，適当な基準を立て分別することで，攻撃だけのアラートを判別することができた．我々は限定的な ground truth を作成し，GLASSO エンジンの攻撃検知精度を評価し，その攻撃の詳細を紹介した．また，従来の方法では検知できなかったものを検知することもできた．GLASSO エンジンはいくつかの攻撃を見逃したが，考察を行い対策を考えた．

今後の予定として，見逃しを減らすように GLASSO エンジンを調整し，長期間に渡る運用を行うことで，検知時期の適切さを適当な基準で評価を行いたい．そして，一点集中型がどういうものかダークネット以外の別のデータセット (ハニーポットなど) を用いて真相を調べたい．最後に，GLASSO エンジンは攻撃検知のみならず，Survey scanner の検知ができることが分かったため，拡張研究として survey scanner の検知を行いたい．

Acknowledgement

The authors thank associate Prof. K. Yoshioka from the Yokohama National University and Prof. N. Murata from the Waseda Univeristy for their valuable comments.

References

[1] M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi. A Proposal of Metrics for Botnet Detection based on Its Cooperative Behavior. In *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops(SAINT-W'07)*, pp.82-85, 2007.

[2] T. Ban, L. Zhu, J. Shimamura, S. Pang, D. Inoue, and K. Nakao. Detection of botnet activities through the lens of a large-scale darknet. In *International Conference on Neural Information Processing, Springer*, 2017.

[3] A. Dainotti, K. Benson, A. King, K. Claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos. Estimating Internet address space usage through passive measurements. *ACM SIGCOMM Computer Communication Review*, 44(1):42-49, 2013.

[4] Z. Durumeric, M. Bailey, and J.A. Halderman. An Internet-wide view of Internet-wide scanning. *23rd USENIX Security Symposium*, pp.65-78, 2014.

[5] C. Fachkha, E. Bou-Harb, A. Keliris, N. Memon, and M. Ahamad. Internet-scale probing of CPS: inference, characterization and orchestration analysis. In *Proceedings of NDSS*, 2017.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 2008.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Graphical Lasso: Estimation of Gaussian Graphical Models. https://cran.r-project.org/web/packages/glasso/glasso.pdf, [Accessed Dec. 2018].

[8] C. Han, K. Kono, S. Tanaka, M. Kawakita, and J. Takeuchi. Botnet detection using graphical lasso with graph density. In *International Conference on Neural Information Processing, Springer*, 2016.

[9] Huawei, Security Notice - Statement on Remote Code Execution Vulnerability in Huawei HG532 Product. https://www.huawei.com/en/psirt/security-notices/huawei-sn-20171130-01-hg532-en, [Accessed Dec. 2018].

[10] Imperva, RedisWannaMine Unveiled: New Cryptojacking Attack Powered by Redis and NSA Exploits. https://www.imperva.com/blog/rediswannamine-new-redis-nsa-powered-cryptojacking-attack/, [Accessed Dec. 2018].

[11] T. Ide, A.C. Lozano, N. Abe, and Y. Liu. Proximity-Based Anomaly Detection Using Sparse Structure Learning. In *Proceedings of 2009 SIAM International Conference on Data Mining*, 2009.

[12] Netlab 360, HNS Botnet Recent Activities. https://blog.netlab.360.com/hns-botnet-recent-activities-en/, [Accessed Dec. 2018].

[13] Netlab 360, BCMPUPnP_Hunter: A 100k Botnet Turns Home Routers to Email Spammers. https://blog.netlab.360.com/bcmpupnp_hunter-a-100k-botnet-turns-home-routers-to-email-spammers-en/, [Accessed Dec. 2018].

[14] Netlab 360, 7,500+ MikroTik Routers Are Forwarding Owners' Traffic to the Attackers, How is Yours?. https://blog.netlab.360.com/7500-mikrotik-routers-are-forwarding-owners-traffic-to-the-attackers-how-is-yours-en/, [Accessed Dec. 2018].