ハニーポットを設置してみた
～ ENOG 33 Meeting ～

(株)NS・コンピュータサービス
江部 仁士

話の流れ

- ハニーポットについてちょっと
- ハエトリグサの栽培
- ハエトリグサの観察

ハニーポットとは

- 不正アクセスを受けることに価値を持つシステム

- 何らかの有益そうな情報や資源がありそうな場所を用意して、それにつられた者を観察したり、肝心な部分で被害を出さないために目をそらせたり、コンピュータ・フォレンジックスを行うための証拠を集めたりする、一種のおとり手法

By Wikipedia

ハニーポットの種類

- **高対話型ハニーポット**
  - ✓ 脆弱性を残した「本物」のOSやアプリケーションなどをハニーポットとして利用する
  - ✓ 高度な情報を得ることができる
  - ✓ 侵入されたときのリスクが高い

- **低対話型ハニーポット**
  - ✓ 特定のOSやアプリケーションをエミュレートし監視を行う
  - ✓ 機能制限が制限されているため、情報量は落ちる
  - ✓ 高対話型ハニーポットに比べて比較的安全に運用できる
  - ✓ nmapのスキャンでハニーポットと検出される可能性あり

ハニーポットの評価

- 2012年11月20日に欧州ネットワーク情報セキュリティ庁（ENISA：European Network Information Security Agency）から発行された
  「Proactive Detection of Security Incidents-Honeypots」
  で数多くのハニーポットが評価されている。

URL:
http://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots

ハニーポットの評価結果

NAME	DETECTION SCOPE	ACCURACY OF EMULATION	QUALITY OF COLLECTED DATA	SCALABILITY AND PERFORMANCE	RELIABILITY	EXTENSIBILITY	EASE OF USE AND SETTING UP	EMBEDDABILITY	SUPPORT	COST	USEFULNESS FOR CERT
LOW-INTERACTION SERVER-SIDE HONEYPOTS											
General purpose honeypots											
Amun	MULTI	★★	★★★	★★★	★★★★	★★★★	★★★	★★★	★	$	
Dionaea	MULTI	★★★	★★★★	★★★	★★★★	★★★★	★★★	★★★★	★★★★	$	
KFsensor	MULTI	★★	★★★	★★★★	★★★★	★★★	★★★★	★★★	★★★	$$	
Honeyd	MULTI	★★	★	★★★★	★★★★	★★★★	★★★	★★	★	$	
Honeytrap	MULTI	★★	★★	★★★★	★★★	★★★	★★	★	★★	$$	
Nepenthes	MULTI	★★	★★	★★★	★★★★	★★★★	★★★	★★	★	$$	
Tiny Honeypot	MULTI	★★★	★★	★★★	★★★★	★★★★	★★	★★	★	$$	
Web application honeypots											
DShield Web Honeypot	SPEC	★★	★★	★★★	★★★★	★★★	★★★	★★	★★	$$	
Google Hack Honeypot	SPEC	★★	★★★	★★★★	★★★	★★★★	★★★	★★★	★	$	
Glastopf	SPEC	★★★★	★★★	★★	★★★	★★★★	★★★	★★★	★★★★	$	
SSH Honeypots											
Kippo	SPEC	★★★	★★★	★★	★★★	★★	★★★	★★	★★★	$$	
Kojoney	SPEC	★★★	★★	★★★	★★	★★	★★	★★	★	$$$	
SCADA Honeypots											
SCADA HoneyNet Project	MULTI	★★	★	★★★	★★★	★★	★★	★★★	★	$	
SCADA HoneyNet (Digital Bond)	MULTI	★★	★	★	★★	★★	★★	★	★	$$	
VoIP Honeypots											
Artemisa	SPEC	★★★★	★★★★	★★	★★★	★★	★★★	★★	★	$$	
Bluetooth Honeypots											
Bluepot	SPEC	★★★	★★	★★★	★★	★★	★★★	★	★	$$$	
Sinkholes											
HoneySink	MULTI	★★★	★★★★	★★★	★★★★	★★	★★	★★★	★	$$	
USB Honeypots											
Ghost USB honeypot	SPEC	★★★	★★	N/A	★★★	★★	★★★	★★★★	★★	$$$	

Proactive Detection of Security Incidents- Honeypotsより

ハニーポットの選定

- パッと見で評価が高い「Dionaea」

評価項目	「Dionaea」の評価結果
検出できるサービス範囲	MULTI
エミュレーションの精度	☆☆☆　(Good)
収集されたデータの品質	☆☆☆☆　(Excellent)
拡張性とパフォーマンス	☆☆☆　(Good)
信頼性	☆☆☆☆　(Excellent)
機能の拡張性	☆☆☆☆　(Excellent)
使いやすさ、設定のしやすさ	☆☆☆　(Good)
他システムとの統合性	☆☆☆☆　(Excellent)
サポートの有無	☆☆☆☆　(Excellent)
導入における費用感	$　(Low Cost)

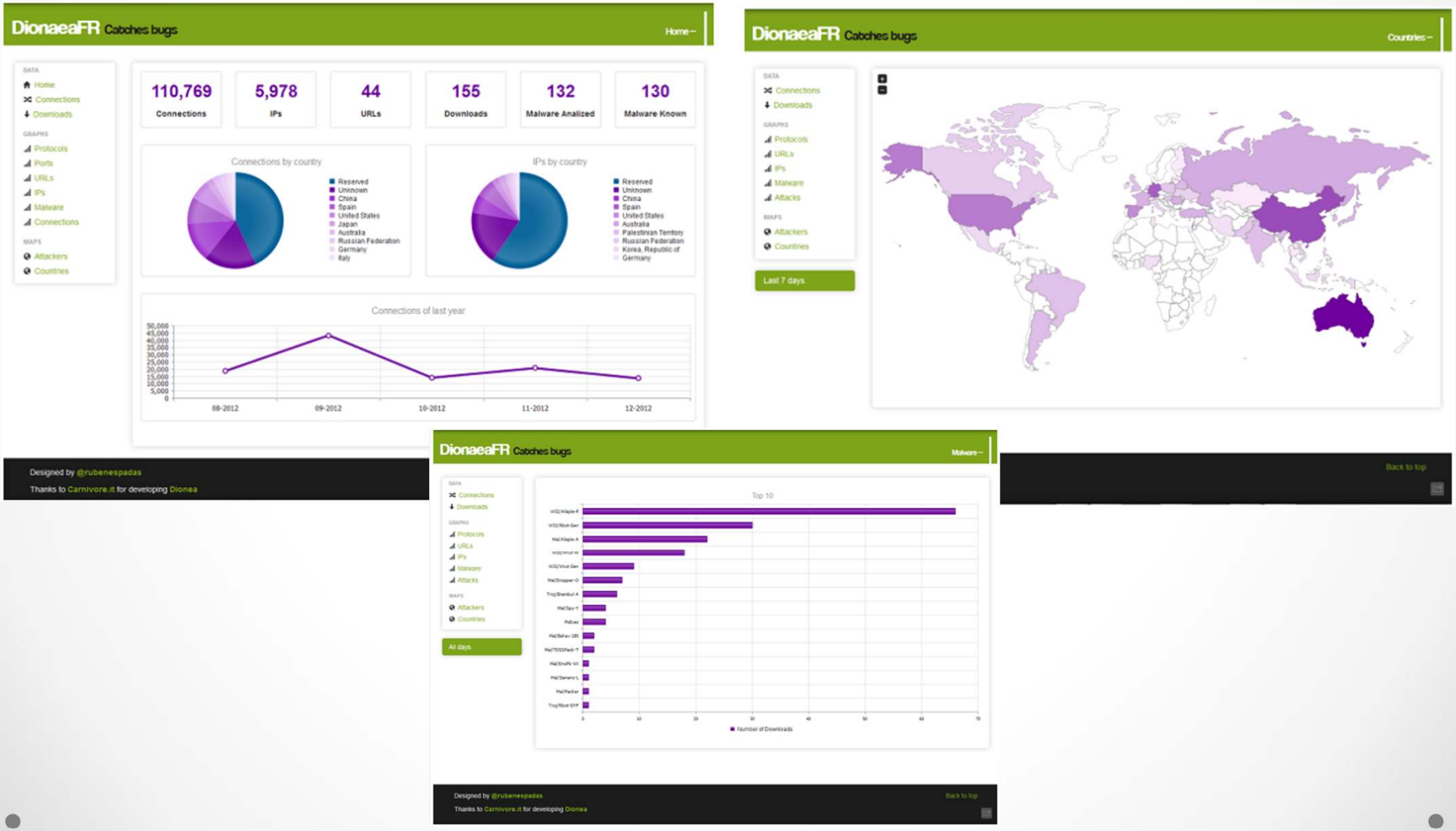
- じゃあコレにしよう。。。

Dionaea(ハエトリグサ)

- **Nepenthes(ウツボカズラ) の後継**
- **低対話型ハニーポット**
  - ✓ マルウェア収集向け
- **ご提供サービス**
  - ✓ SMB , HTTP , HTTPS , FTP , TFTP , MSSQL , MySQL , SIP
- **IPv6対応**
- **ログ解析ツール**
  - ✓ DionaeaFR
-

DionaeaFR(ログ解析ツール)

- ログ解析ツールはこんな感じ



Dionaea必須環境

- Libev >= 4.04
- Libglib >= 2.20
- Libssl
- libcfg
- libemu
- Python >= 3.2
  - Sqlite >= 3.3.6
  - readline >= 3
- Cython >=0.14.1
- Libudns
- Libcurl
- Libpcap >= 1.1.1
- Libnl
- libgc

befor

栽培に成功すると

after

DionaeaFR環境

- Python 2.7.3
- Django 1.4
- Jquery 1.7.2
- Bootstrap Framework 2.1.1
- jVectorMap 1.0
- Kendo-UI v2011.3.1129
- SQLite3

栽培開始

- ということで、CentOS6.6環境にDionaeaとDionaeaDFをインストールしてみることにした

Dionaeaインストール

- liblcfg

```
# git clone git://git.carnivore.it/liblcfg.git liblcfg
# cd liblcfg/code
# autoreconf –vi
# ./configure --prefix=/opt/dionaea
# make install
# cd ../..
```

- libemu

```
# git clone git://git.carnivore.it/libemu.git libemu
# cd libemu
# autoreconf –vi
# ./configure --prefix=/opt/dionaea
# make install
# cd ..
```

Dionaeaインストール

- libev

```
# wget http://dist.schmorp.de/libev/Attic/libev-4.04.tar.gz
# tar xzf libev-4.04.tar.gz
# cd libev-4.04
# ./configure --prefix=/opt/dionaea
# make install
# cd ..
```

Dionaeaインストール

- Python

```
# yum install zlib-devel ¥
  bzip2-devel ¥
  openssl-devel ¥
  ncurses-devel ¥
  sqlite-devel ¥
  readline-devel ¥
  k-deve
# wget http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz
# tar xfz Python-3.2.2.tgz
# cd Python-3.2.2/
# ./configure --enable-shared ¥
  --prefix=/opt/dionaea ¥
  --with-computed-gotos --enable-ipv6 ¥
  LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -L/usr/lib/x86_64-linux-gnu/“
# make
# make install
# cd ..
```

Dionaeaインストール

- Cython

```
# wget http://cython.org/release/Cython-0.15.tar.gz
# tar xzf Cython-0.15.tar.gz
# cd Cython-0.15
# /opt/dionaea/bin/python3 setup.py install
# cd ..
```

- udns

```
# wget http://www.corpit.ru/mjt/udns/old/udns_0.0.9.tar.gz
# tar xfz udns_0.0.9.tar.gz
# cd udns-0.0.9/
# ./configure
# make shared
# cp udns.h /opt/dionaea/include
# cp *.so* /opt/dionaea/lib/
# cd /opt/dionaea/lib
# ln -s libudns.so.0 libudns.so
# cd –
# cd ..
```

Dionaeaインストール

- libcurl

```
# git clone https://github.com/bagder/curl.git curl
# cd curl
# autoreconf –vi
# ./configure --prefix=/opt/dionaea
# make
# make install
# cd ..
```

- libpcap

```
# wget http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz
# tar xfz libpcap-1.1.1.tar.gz
# cd libpcap-1.1.1
# ./configure --prefix=/opt/dionaea
# make
# make install
# cd ..
```

Dionaeaインストール

- OpenSSL

```
# wget https://www.openssl.org/source/openssl-1.0.2c.tar.gz
# tar xzvf openssl-1.0.2c.tar.gz
# cd openssl-1.0.2c
# ./Configure shared --prefix=/opt/dionaea linux-x86_64
# make SHARED_LDFLAGS=-Wl,-rpath,/opt/dionaea/lib
# make install
# cd ..
```

Dionaeaインストール

- Dionaea

```
# yum –y install glib2-devel
# git clone git://git.carnivore.it/dionaea.git dionaea
# cd dionaea
# autoreconf –vi
# ./configure --with-lcfg-include=/opt/dionaea/include/ ¥
  --with-lcfg-lib=/opt/dionaea/lib/ ¥
  --with-python=/opt/dionaea/bin/python3.2 ¥
  --with-cython-dir=/opt/dionaea/bin ¥
  --with-udns-include=/opt/dionaea/include/ ¥
  --with-udns-lib=/opt/dionaea/lib/ ¥
  --with-emu-include=/opt/dionaea/include/ ¥
  --with-emu-lib=/opt/dionaea/lib/ ¥
  --with-gc-include=/usr/include/gc ¥
  --with-ev-include=/opt/dionaea/include ¥
  --with-ev-lib=/opt/dionaea/lib ¥
  --with-curl-config=/opt/dionaea/bin ¥
  --with-pcap-include=/opt/dionaea/include ¥
  --with-pcap-lib=/opt/dionaea/lib/ ¥
  --with-ssl-include=/opt/dionaea/include/ ¥
  --with-ssl-lib=/opt/dionaea/lib
# make
# make install
```

Dionaeaインストール

- Dionaeaの起動

```
# groupadd dionaea
# useradd -g dionaea -s /usr/sbin/nologin dionaea
# chown -R dionaea:dionaea /opt/dionaea/
# cd /opt/dionaea/bin/
# ./dionaea -c /opt/dionaea/etc/dionaea/dionaea.conf -u dionaea -g dionaea -D
```

Dionaeaの栽培完了

DionaeaFRインストール

- Python

```
# yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel
# wget http://www.python.org/ftp/python/2.7.6/Python-2.7.6.tar.xz
# tar -Jxvf Python-2.7.6.tar.xz
# cd Python-2.7.6
# ./configure --with-threads --enable-shared --prefix=/usr/local
# make
#  vi Modules/Setup
～以下のとおり修正～
#SSL=/usr/local/ssl
#_ssl _ssl.c ¥
#       -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl ¥
#       -L$(SSL)/lib -lssl –lcrypto
↓
SSL=/usr
 _ssl _ssl.c ¥
 -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl ¥
-L$(SSL)/lib -lssl -lcrypto
# make altinstall
# ln -s /usr/local/lib/libpython2.7.so.1.0 /lib64/
# cd ..
```

DionaeaFRインストール

- pip

```
# wget http://pypi.python.org/packages/source/d/distribute/distribute-0.6.49.tar.gz
# tar zxf distribute-0.6.49.tar.gz
# cd distribute-0.6.49
# python2.7 setup.py install
# easy_install-2.7 pip
```

- Django

```
# pip2.7 install Django pygeoip django-pagination django-tables2 ¥
  django-compressor django-htmlmin django-filter
# cd /opt
# wget https://github.com/benjiec/django-tables2-simplefilter/archive/master.zip ¥
 -O django-tables2-simplefilter.zip
# unzip django-tables2-simplefilter.zip
# mv django-tables2-simplefilter-master/ django-tables2-simplefilter/
# cd django-tables2-simplefilter/
# python2.7 setup.py install
```

DionaeaFRインストール

- python-netaddr

```
# cd /opt
# wget https://pypi.python.org/packages/source/n/netaddr/netaddr-0.7.11.tar.gz
# tar xzvf netaddr-0.7.11.tar.gz
# cd netaddr-0.7.11
# python2.7 setup.py install
```

- PySubnetTree

```
# cd /opt
# git clone git://git.bro-ids.org/pysubnettree.git
# cd pysubnettree/
# python2.7 setup.py install
```

DionaeaFRインストール

- Node.js

```
# cd /opt
# wget http://nodejs.org/dist/v0.8.16/node-v0.8.16.tar.gz
# tar xzvf node-v0.8.16.tar.gz
# cd node-v0.8.16
# ./configure
# make
# make install
# npm install -g less
# npm install -g promise
```

DionaeaFRインストール

- DionaeaFR

```
# cd /opt/
# wget https://github.com/RootingPuntoEs/DionaeaFR/archive/master.zip ¥
  -O DionaeaFR.zip
# unzip DionaeaFR.zip
# mv DionaeaFR-master/ DionaeaFR
# wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
# wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz
# gunzip GeoLiteCity.dat.gz
# gunzip GeoIP.dat.gz
# mv GeoIP.dat DionaeaFR/DionaeaFR/static
# mv GeoLiteCity.dat DionaeaFR/DionaeaFR/static
# cd /opt/DionaeaFR/
# cp DionaeaFR/settings.py.dist DionaeaFR/settings.py
# vi DionaeaFR/settings.py
～以下のとおり修正 18行目～
'NAME': '/var/lib/dionaea/logsql.sqlite',
↓
'NAME': '/opt/dionaea/var/dionaea/logsql.sqlite',
# mkdir /var/run/dionaeafr
# touch /var/run/dionaeafr/dionaeafr.pid
# python2.7 manage.py collectstatic
```

DionaeaFRインストール

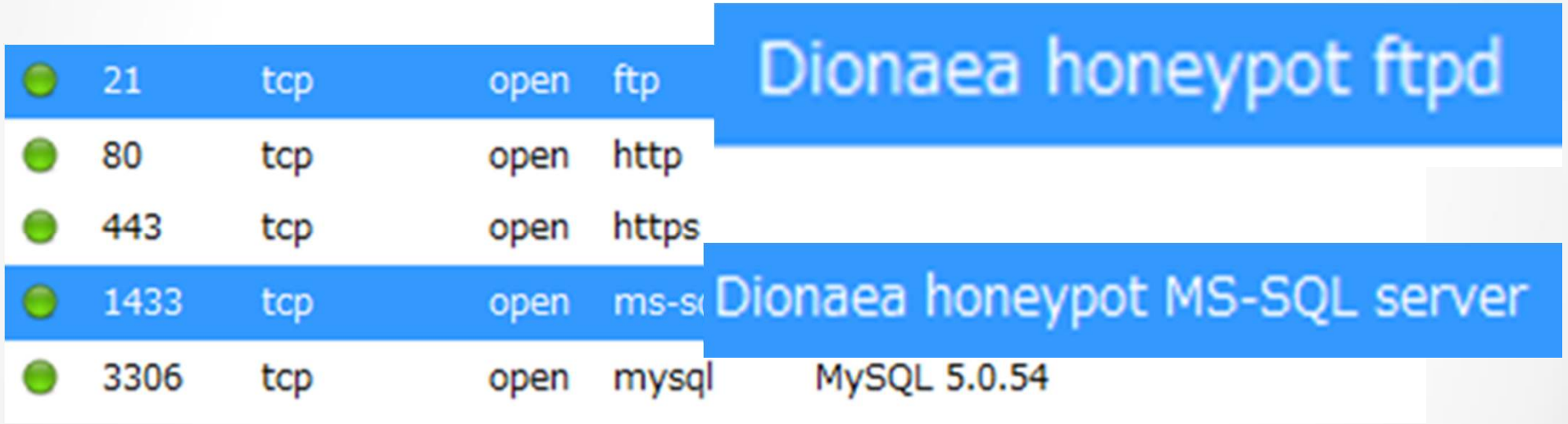
- DionaeaFR起動！

```
# cd /opt/DionaeaFR/
# python2.7 manage.py runserver 0.0.0.0:8000
```

- DionaeaFR起動確認
  - ✓ 以下のURLアクセスしてこんな感じの画面が出力されればOK！
  - ✓ http://サーバのIPアドレス:8000

ポートスキャンをかけてみた

- nmapでポートキャンをかけてみる。
  nmap -sT -sU -T3 -A -v xx.xx.xx.xx



- ・・・あれ？

nmapは知っていた

- nmapはハニーポットの挙動を知っている！！
  - ✓ nmap シグネチャ

    URL: https://svn.nmap.org/nmap/nmap-service-probes

  - ✓ FTPのシグネチャ

    match ftp m|^220 Welcome to the ftp service¥r¥n| p/Dionaea honeypot ftpd/

nmapではFTP接続時のバナーの文字列チェックでDionaeaを特定している

nmap対策

- **FTP**

/opt/dionaea/lib/dionaea/python/dionaea/ftp.pyの227行目を変更する。

self.reply(WELCOME_MSG, "Welcome to the ftp service")

↓

self.reply(WELCOME_MSG, "Microsoft FTP Service")

- **MS-SQL**

/opt/dionaea/lib/dionaea/python/dionaea/mssql/mssql.py
の151行目を変更する。

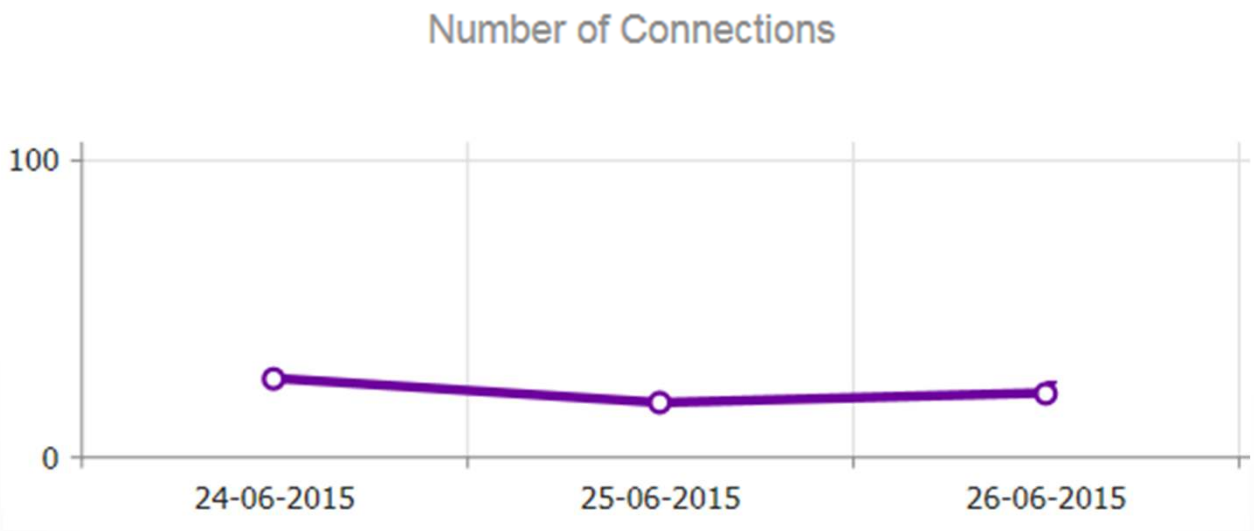
r.VersionToken.TokenType = 0x00

↓

r.VersionToken.TokenType = 0x01

苦悩

- 実際に動かしてみるもアクセス数が伸びない。。。



決断

- **iptables**

-A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP

-A INPUT -p tcp -m tcp ! --tcp-flags FIN,SYN,RST,ACK SYN -m state --state NEW -j DROP

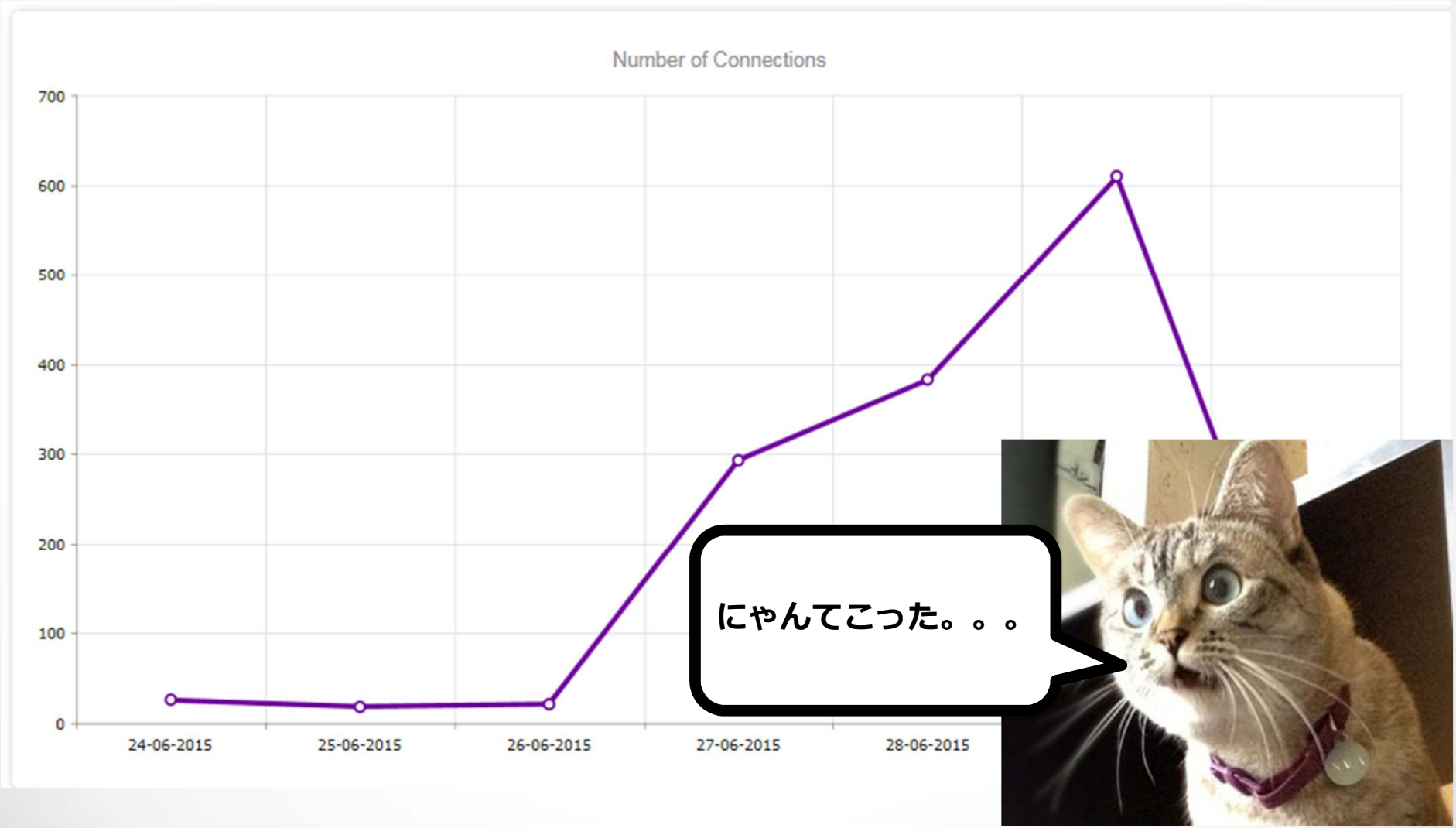
…

この時点ではSMB、SQLを止めていた。。。

IPtables全開！！

※全開は言い過ぎでした。さすがにSSH、管理画面は絞ってます。

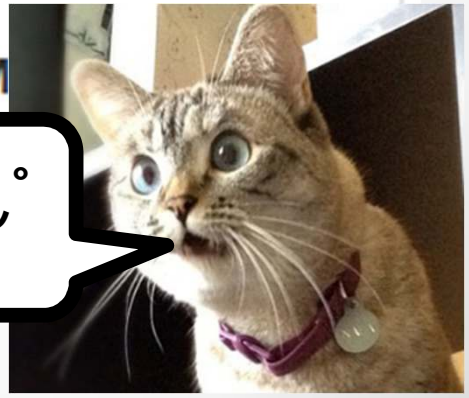
豊作


Number of Connections
700
600
500
400
300
200
100
0
24-06-2015
25-06-2015
26-06-2015
27-06-2015
28-06-2015
にゃんてこった。。。

アクセス集計

- 期間: 2015年6月19日~6月30日

2,310	414	99
Connections	IPs	URLs
189	0	0
Downloads	Malware Analized	M



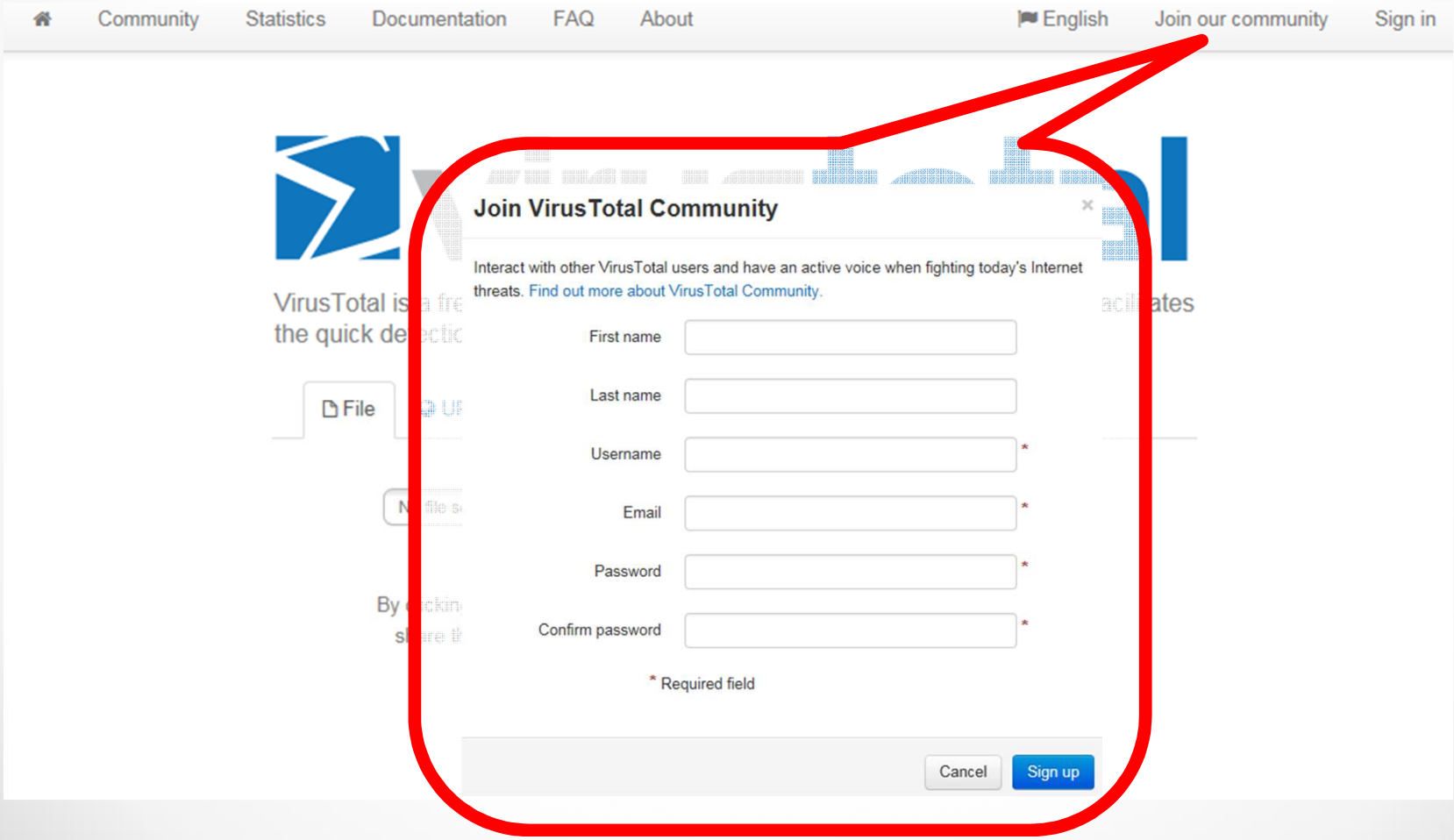
マルウエアを検出しない

- “Malware Analized”はDionaeaに設置されたバイナリのハッシュ値をvirustotalで検索した数を意味する

- Virustotalと連携が必要!!!
  - ✓ Virustotal:
    無料でファイルのウイルス検査を行ってくれるウェブサイト

Virustotalと連携

- まずはvirustotalでアカウントを作成する
https://www.virustotal.com/



Virustotalと連携

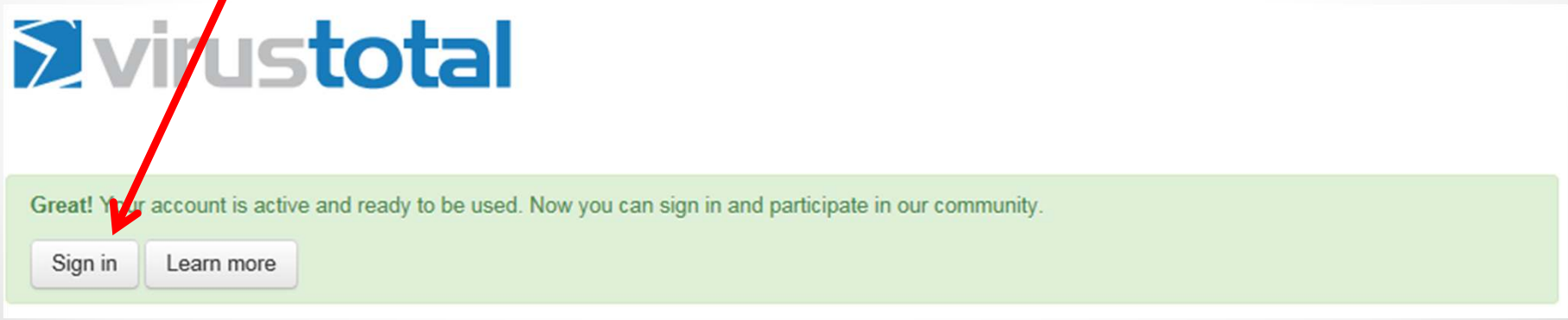
- こんなメールが届きますので、URLにアクセスしてアクティベーションする

Hi Yamada!

Welcome to VirusTotal Community.

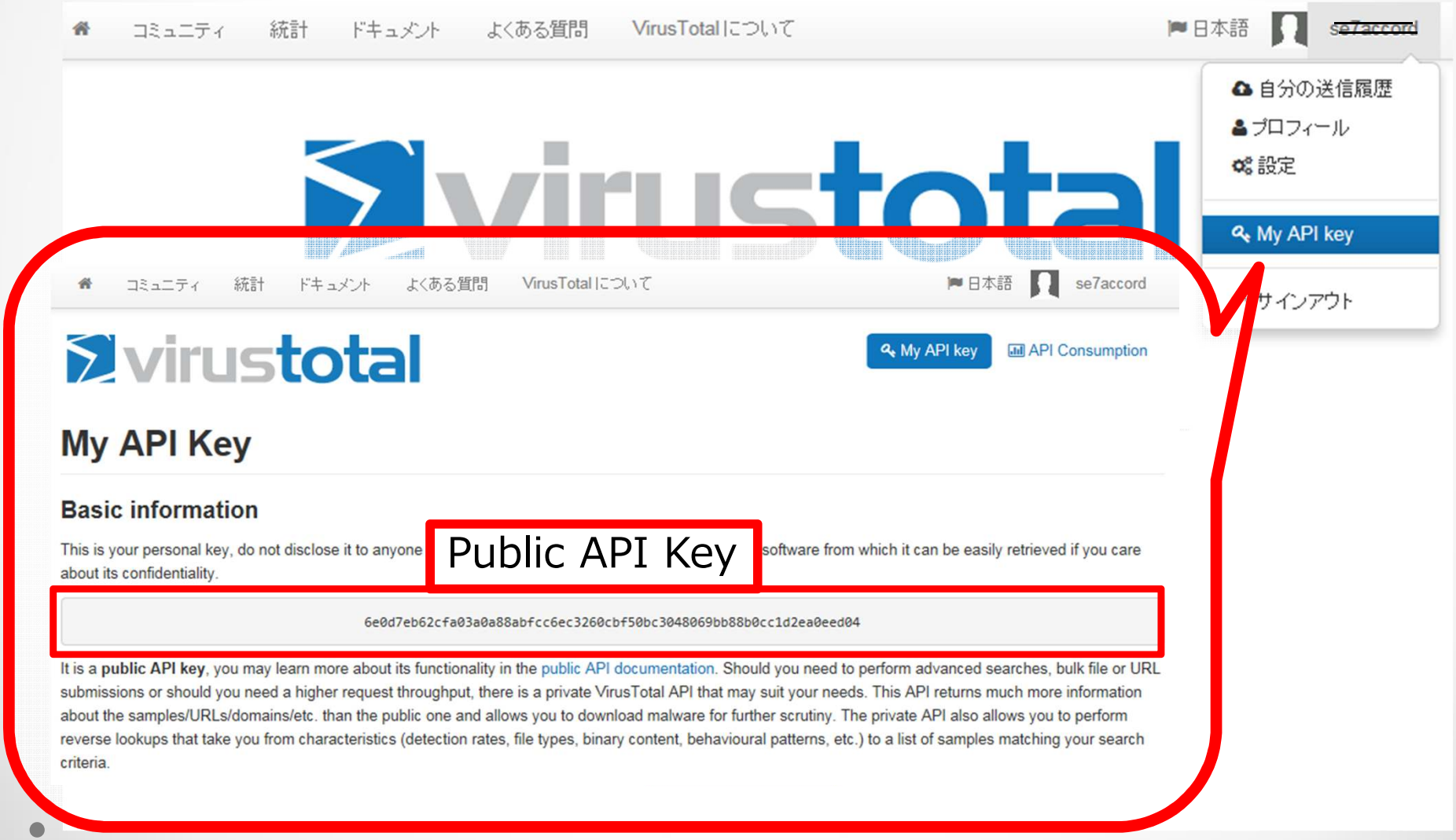
https://www.virustotal.com/en/account/activate/～～～～

The VirusTotal Team



Virustotalと連携

- Public API Keyを発行する



Virustotalと連携

- 設定ファイルの編集

  設定ファイル: /opt/dionaea/etc/dionaea/dionaea.conf

- 323行目のapikeyにPublic API Keyを入力

```
virustotal = {
        apikey = "........." // grab it from your virustotal account at My account -> Inbox ->
    Public API
```

Public API keyを入力

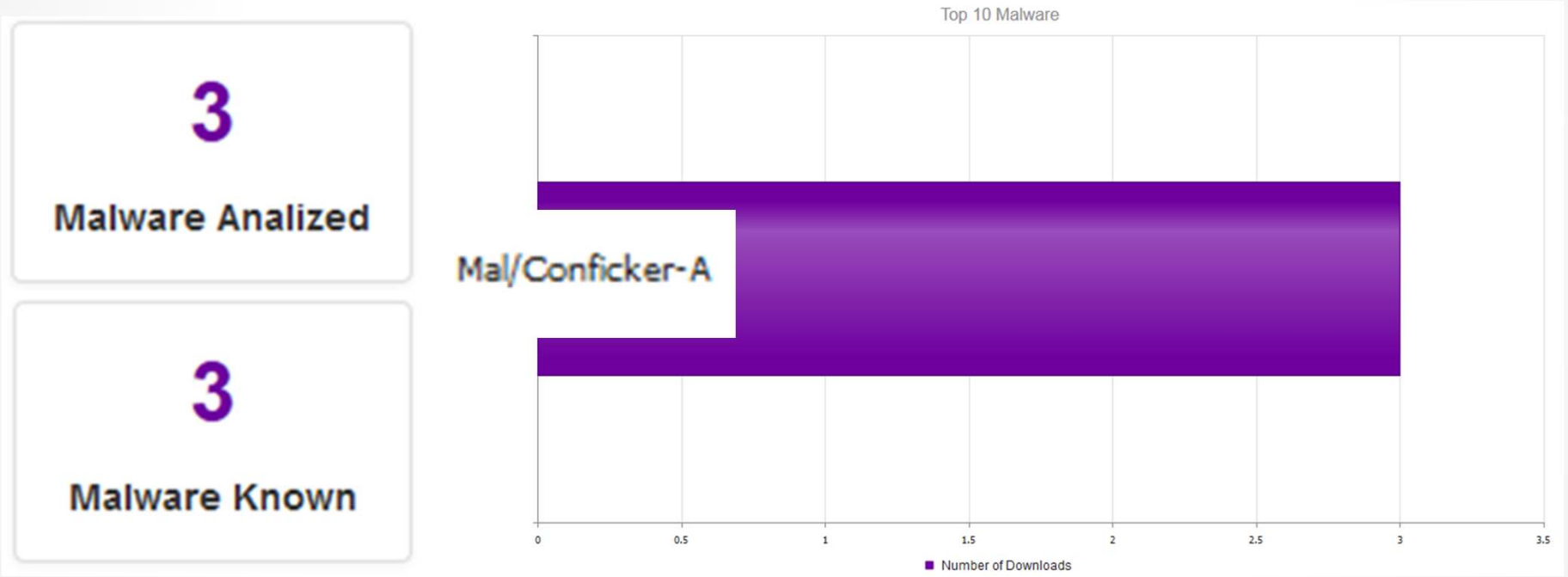
- 465行目の”virustotal”のコメントを解除

```
ihandlers = {
        handlers = ["ftpdownload", "tftpdownload", "emuprofile", "cmdshell", "store",
    "uniquedownload",
                    "logsql",
//                  "virustotal",
```

コメント解除

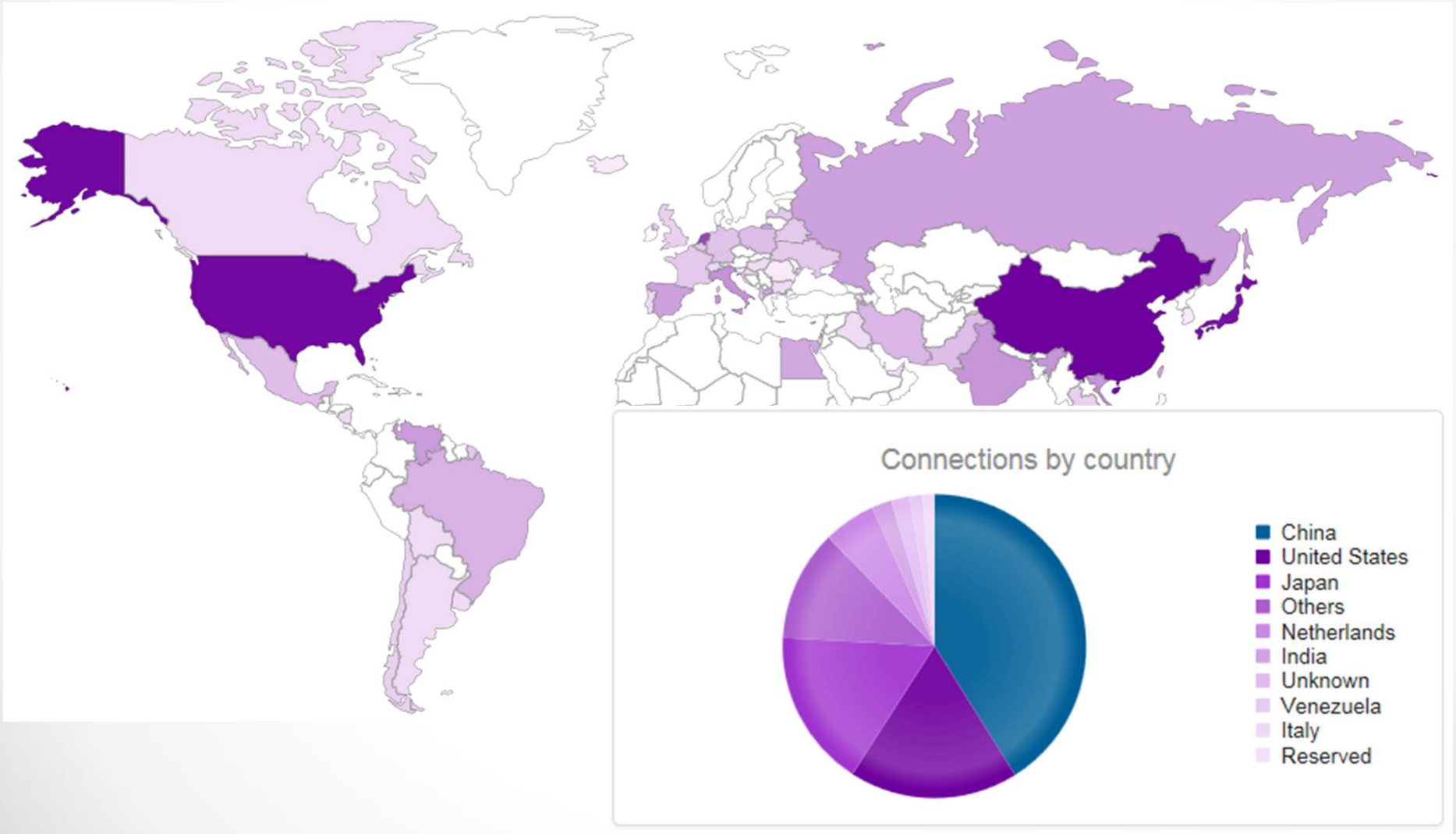
マルウェアの入手

- なんとか間に合った。。。



- ご提供いただいた3つのマルウェアは全てConfickerの亜種
- 2008年11月に発見され、猛威を振るったワーム
-

ご支援をいただいた国

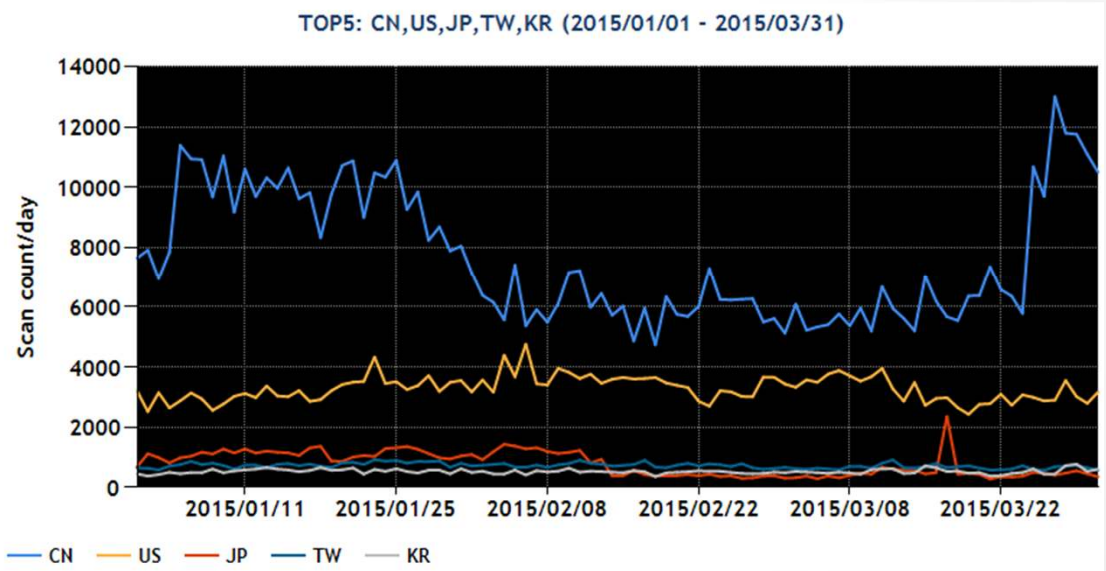

Connections by country
China
United States
Japan
Others
Netherlands
India
Unknown
Venezuela
Italy
Reserved

インターネット定点観測(送信元)

- JPCERTで公開されているインターネット定点観測レポート
- 2015年1月~3月に観測された日本宛のパケット分析結果

[表2:送信元地域トップ5]

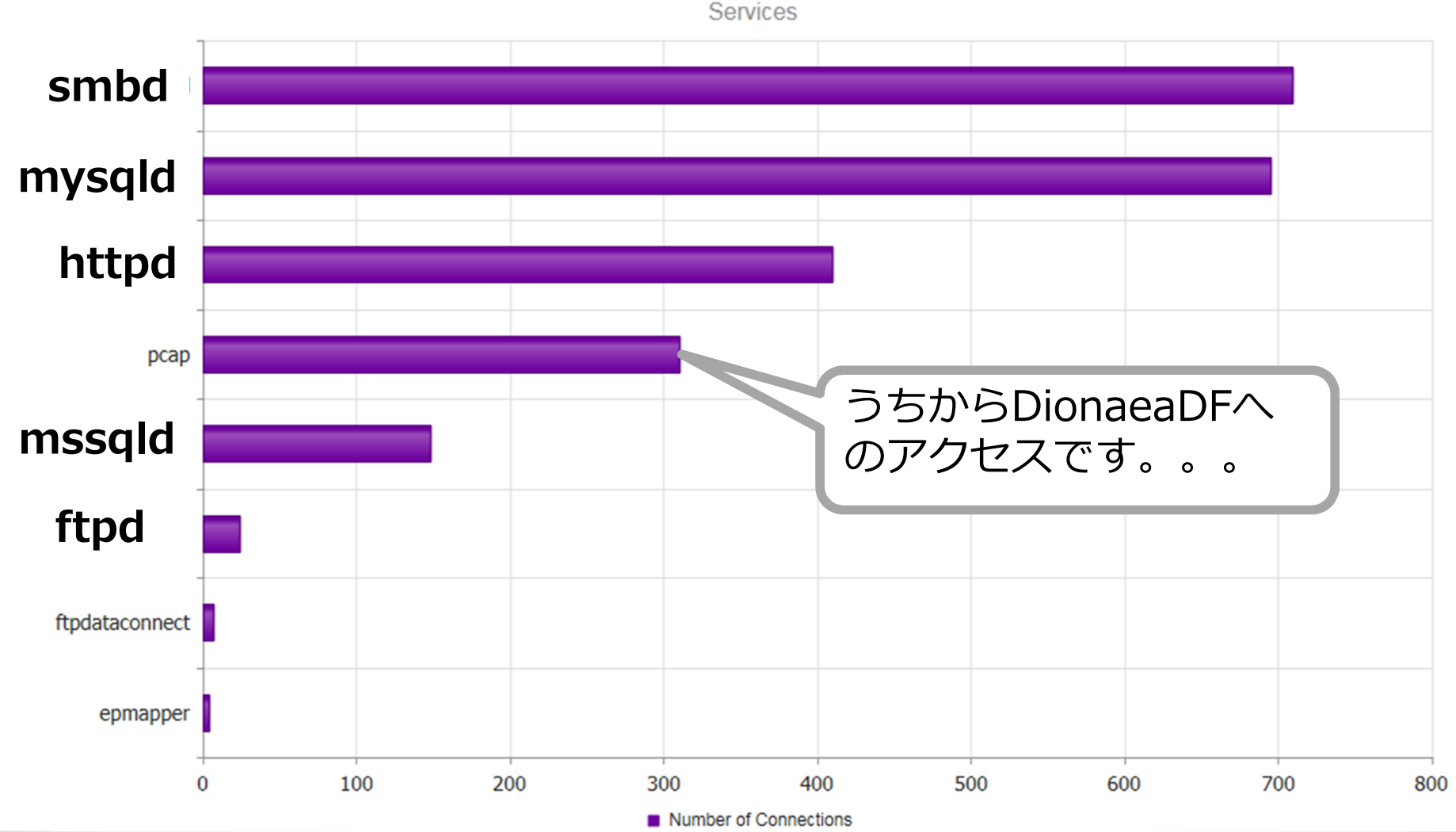
順位	送信元地域	前四半期の順位
1	中国	1
2	米国	2
3	日本	3
4	台湾	5
5	韓国	6



https://www.jpcert.or.jp/tsubame/report/report201501-03.html

- **圧倒的な中国と米国の脅威！**

人気のサービス

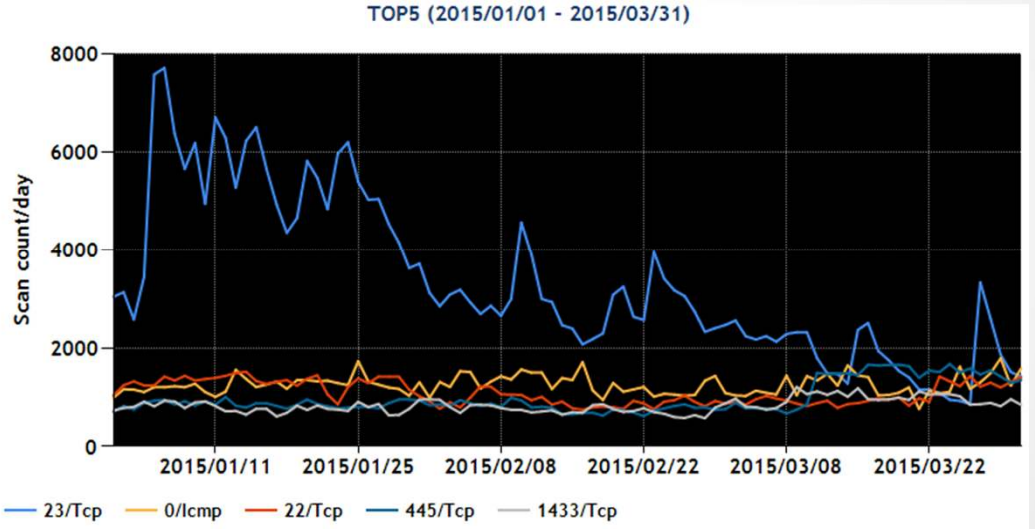

Services
smbd
mysqld
httpd
pcap
mssqld
ftpd
ftpdataconnect
epmapper
うちからDionaeaDFへのアクセスです。。。
0
100
200
300
400
500
600
700
800
Number of Connections

インターネット定点観測(宛先ポート)

- JPCERTで公開されているインターネット定点観測レポート
- 2015年1月~3月に観測された日本宛のパケット分析結果

[表1：宛先ポート番号トップ5]

順位	宛先ポート番号	前四半期の順位
1	23/TCP (telnet)	1
2	0/ICMP	4
3	22/TCP (ssh)	2
4	445/TCP (microsoft-ds)	3
5	1433/TCP (ms-sql-s)	5



https://www.jpcert.or.jp/tsubame/report/report201501-03.html

HTTPアクセス

- /epgrec~/templates/envSetting.htmlに対するアクセスが一番多かった

```
GET /epgrec/templates/envSetting.html HTTP/1.1
GET /templates/envSetting.html HTTP/1.1
GET /epgrec_old/templates/envSetting.html HTTP/1.1
GET /tv/templates/envSetting.html HTTP/1.1
GET /rec/templates/envSetting.html HTTP/1.1
GET /epg/templates/envSetting.html HTTP/1.1
GET /epgrecUNA/templates/envSetting.html HTTP/1.1
GET /una/templates/envSetting.html HTTP/1.1
GET /epgrec_una/templates/envSetting.html HTTP/1.1
…
```

- EPGrecはWebベースの録画予約システム
- 昔は任意のPHPファイルを置いて実行できる脆弱性があり、流行ってたみたい

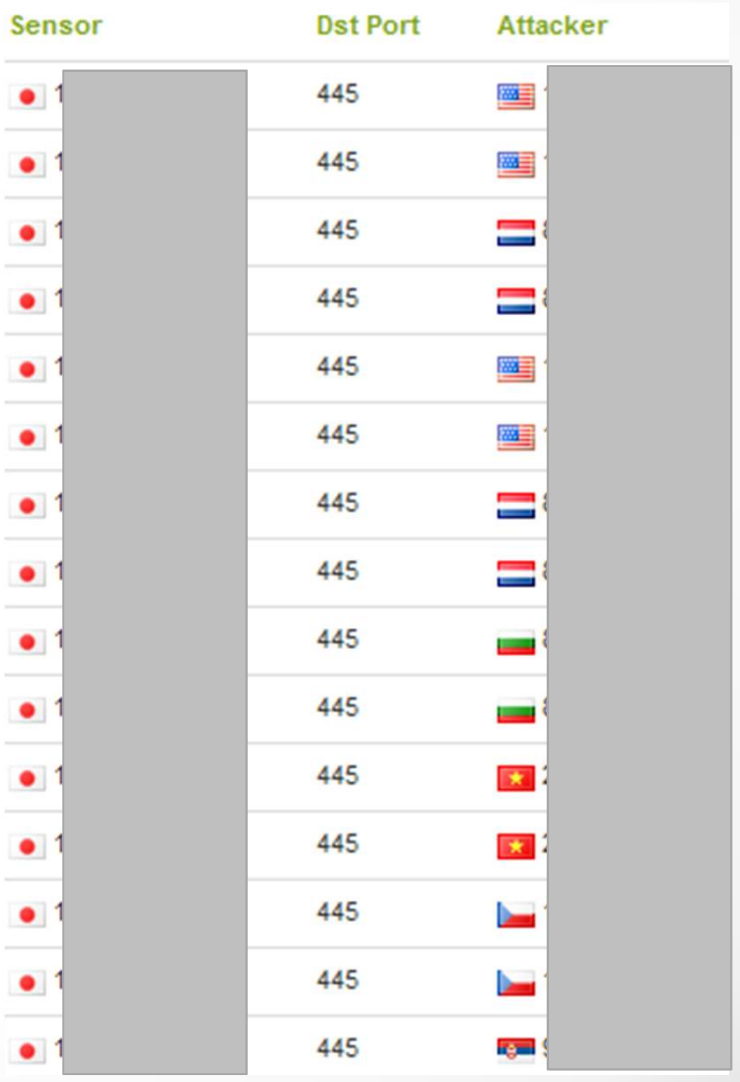
その他にもチラホラ

- phpMyAdmin
  - ✓ GET /phpmyadmin/scripts/setup.php HTTP/1.0
  - ✓ phpMyAdminのsetup.phpにおける任意のPHPコードを挿入される脆弱性
- WordPress
  - ✓ GET /wp-admin/admin-ajax.php HTTP/1.1
  - ✓ テーマやプラグインでAjaxを使用している場合、admin-ajax.phpはアクセス制限から除外する必要がある
    - ➢ http://wpdocs.osdn.jp/ブルートフォース攻撃
    - ➢ WordPressが使われているか確認しているのかな？
- TomcatのWeb管理
  - ✓ GET /manager/html HTTP/1.1

SMBへのアクセス

- いろんな国から標的にされてました

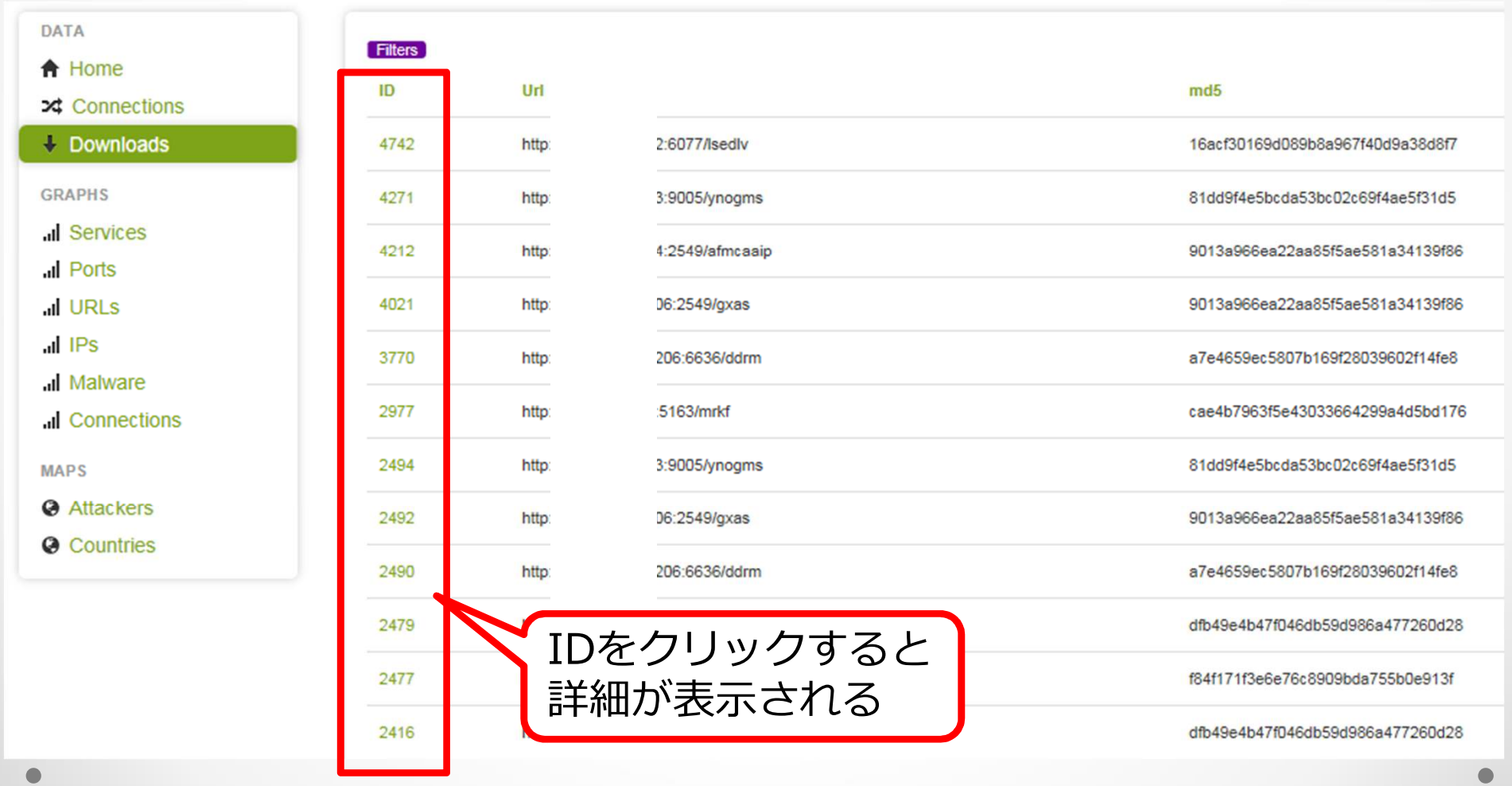
Sensor	Dst Port	Attacker
[redacted]	445	[redacted]
	445	
	445	
	445	?
	445	?
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	



Sensor	Dst Port	Attacker
[redacted]	445	[redacted]
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	
	445	

ダウンロード

- SMBを悪用されて何かをダウンロードさせられちゃってます。。。



ダウンロードの詳細

Info

State	Protocol	Service	Timestamp	Root	Parent	CC	IP_SRC	Port_SRC	IP_DST	Port_
accept	tcp	smbd	01-07-2015 14:31:11	2344	None		[redacted]	2061	[redacted]	445

TCP 445

RPC Binds

ID	UUID	Transfer Syntax
1136	4b324fc8-1670-01d3-1278-5a47bf6ee188	8a885d04-1ceb-11c9-9fe8-08002b104860
1135	4b324fc8-1670-01d3-1278-5a47bf6ee188	8a885d04-1ceb-11c9-9fe8-08002b1048

脆弱性情報

RPC Request

ID	UUID	OpNum	Service	Name	Vuln
688	4b324fc8-1670-01d3-1278-5a47bf6ee188	31	SRVSVC	NetPathCanonicalize	MS08-067
687	4b324fc8-1670-01d3-1278-5a47bf6ee188	32	SRVSVC	NetPathCompare	MS08-067

MS08-067

- 特別な細工がされた RPC リクエストを受け取った場合、リモートでコードが実行される脆弱性をつかれていた

こんな攻撃を受けたみたい

ワームに感染してHTTPサーバを
立てられた人

俺のDionaea

http://自分のGloablIP:[ランダムポート]/[ランダム文字列]
をダウンロードしにこい!!

Listen:
TCP445

http://自分のGloablIP:[ランダムポート]/[ランダムストリング]
をください!

Listen:
ランダム

マルウェア
やるよ!!!

こんな攻撃を受けたみたい

ワームに感染してHTTPサーバを
立てられた人

俺のDionaea

これがConfickerの動き

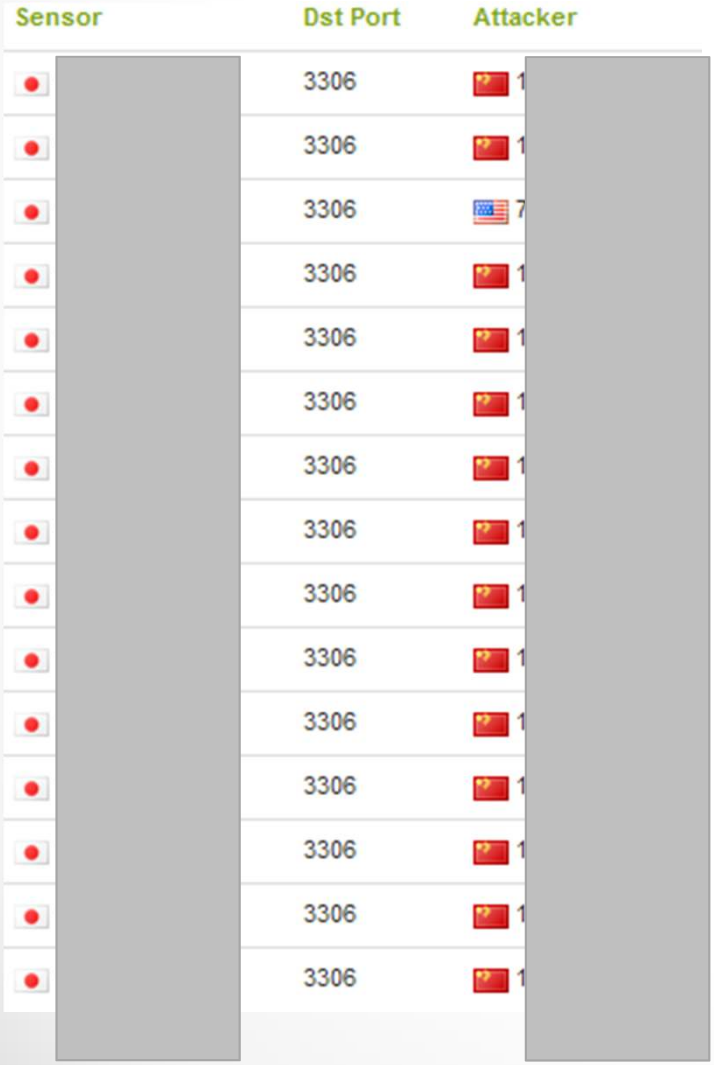
Liste
ランダム

マルウェア
やるよ!!!

まじで
さんきゅー！

MySQLへのアクセス

- パッと見で９０％以上のアクセス元は中国！



MS-SQLへのアクセスは？

- まぁ予想はついたけど、、、90%以上が中国！！

なぜ中国はDBを狙うのか？

- DBには機密情報がいっぱい？
- 機密情報を盗んだらブラックマーケットへLet’s Go？
- 中国の目的はお金？

まとめ

- 昔の脆弱性、攻撃と思って甘くみてはいけない
  - ✓ 当たり前ですが、パッチの適用、アクセス制限はしっかりと！
- すぐに攻撃はやってくる！
  - ✓ テストだからと言って油断してはいけない！
- ご協力いただいた皆様ありがとうございました！