

# Security Incident Discovery and Correlation on .Gov Networks

Cory Mazzola, MSIA, CISSP  
US-CERT Surface Analysis Group

Timothy Tragesser  
US-CERT Fusion Analysis & Development



Homeland  
Security

# Agenda

- Overview
- Data Collection
- Malware Activity Sets:
  - Beaconsing
  - Redirection
  - Suspicious Activity
- Findings/Analysis
- Samples/Examples
- Recommendations
- Takeaways



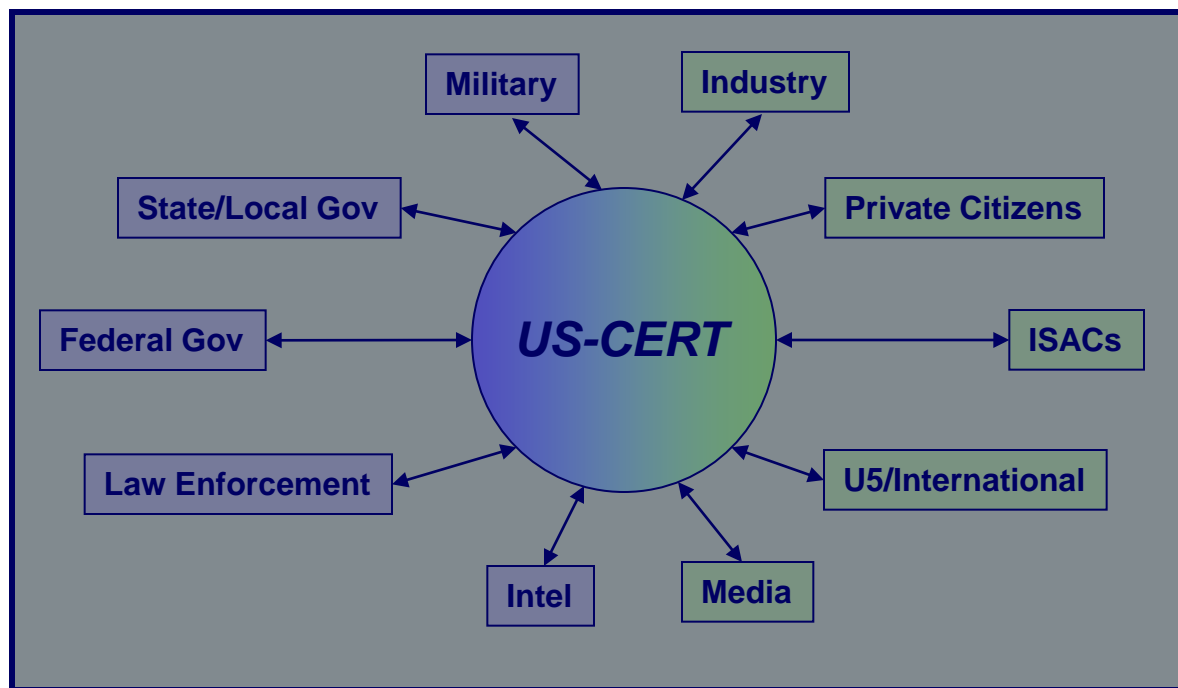
# Who we are...

- ***US-CERT is the operational arm for cyber security under the Department of Homeland Security***
- ***Analysis Branch uses flow data from Einstein sensors deployed across .gov networks***



**Homeland  
Security**

# Information Correlation...



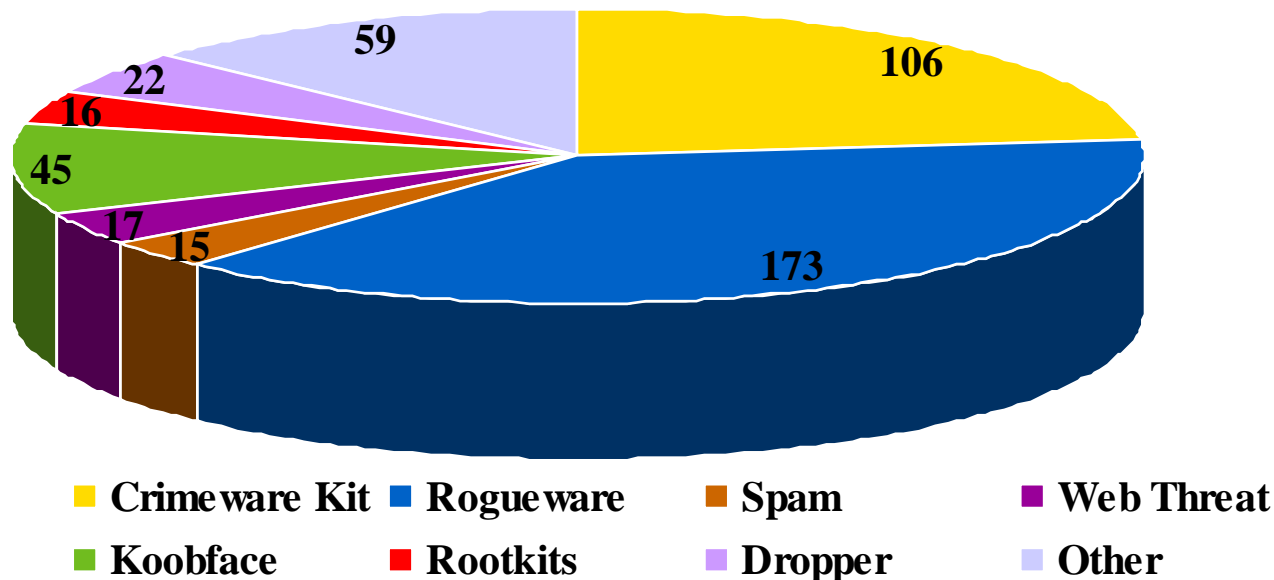
*Facilitating collective analysis of cyber threats through partnerships.*



Homeland  
Security

# Threat Summary

- Security incidents reported to/by US-CERT since 1 January
  - ~108,000 total incidents reported YTD
  - 13,000 Malicious Code Incidents YTD
- Malicious Logic Incidents comprise primary focus area



# Context

- What we have:
  - Repository of federal/state/local govt, private/foreign sector security incidents
    - *~108K so far this year*
- What we needed:
  - Automated method to detect and identify security incidents/events using netflow
- What we devised:
  - Queries to mine database, correlate information and positively identify security incidents



# Prep: Data Collection

## Initial Data Pull/RW Binary Creator

- Creates bin file to prep and execute queries:

```
#!/bin/sh

perl -pi -e "s/ \\|/g" hosts.txt
perl -pi -e "s/ /|/g" hosts.txt
perl -pi -e "s/ //g" hosts.txt

BINFILE=`date "+%Y-%m-%d-%T.bin"`

day=`date +"%a"`

if [ "$day" = "Mon" ];
then
    STARTDATE=`date -d '-4 days' +"%Y/%m/%d"`
    ENDDATE=`date "+%Y/%m/%d"`
elif [ "$day" = "Sun" ];
then
    STARTDATE=`date -d '-7 days' +"%Y/%m/%d"`
    ENDDATE=`date "+%Y/%m/%d"`
elif [ "$day" = "Sat" ];
then
    STARTDATE=`date -d '-8 days' +"%Y/%m/%d"`
    ENDDATE=`date "+%Y/%m/%d"`
else
    STARTDATE=`date -d '-3 days' +"%Y/%m/%d"`
    ENDDATE=`date "+%Y/%m/%d"`
fi

if [ -f $BINFILE ];
then

echo "$BINFILE already exists !!!"
echo "Please insure rwprocessor.sh is not already running and then move or remove $BINFILE"
else
    if [ -f temphosts.txt ];
    then
        rm -f temphosts.txt
    fi

    if [ -f temphosts.set ];
    then
        rm -f temphosts.set
    fi
```



# Initial data pull: RW Binary Creator

- Creates bin file to execute queries against (cont.)

```
for i in `cat hosts.txt | cut -d "|" -f1 | sort | uniq`  
  
do  
  
echo $i >> temphosts.txt  
done  
  
rwsetbuild temphosts.txt temphosts.set  
echo "Einstein query from $STARTDATE to $ENDDATE"  
echo "Created $BINFILE"  
  
rwfilter --anyset=temphosts.set --type=all --start-date=$STARTDATE --end-date=$ENDDATE --pass=$BINFILE &  
  
if [ -f temphosts.txt ];  
then  
    rm -f temphosts.txt  
fi  
  
if [ -f temphosts.set ];  
then  
    rm -f temphosts.set  
fi  
Fi
```





# Malware Activity Patterns

- ***Main Focus Areas:***
  - ***Beaconing***
  - ***Redirect***
  - ***Suspicious***



Image from procalme.com



**Homeland  
Security**

# Beaconing

- Goal is to detect and identify beaconing activity to/from constituent systems
  - Regular and irregular patterns
  - High and low volume connections
  - Known malicious IPs/domains
  - Investigate to identify data exfiltration / low-and-slow actions
- Triggers when victim IP address sends requests on the same dest port with a consistent packet size and at a specific time interval or pattern (i.e., 60 secs., 60 mins., etc.)
- Beaconing is a symptom



Image from Wellroundedsquare.com



U.S. DEPARTMENT OF  
**Homeland  
Security**

# Beaconing

- *Personal favorite*
- 'Quick and easy' to vet true positives
- Good indicator of compromise/infection

*Sample Output (beaconing occurring at 1 hour / 10 minute intervals):*

<i>sTime </i>	<i>sIP </i>	<i>dIP </i>	<i>sPort </i>	<i>dPort </i>	<i>bytes </i>	<i>sensor </i>	<i>InitFlag</i>
2010/10/04T13:06:38	199.9.9.9	195.161.112.6	1315	80	1623	USGA	S
2010/10/04T14:16:40	199.9.9.9	195.161.112.6	1366	80	1623	USGA	S
2010/10/04T15:26:42	199.9.9.9	195.161.112.6	1418	80	1623	USGA	S
2010/10/04T16:36:44	199.9.9.9	195.161.112.6	1515	80	1623	USGA	S
2010/10/04T17:46:45	199.9.9.9	195.161.112.6	1600	80	1623	USGA	S
2010/10/04T18:56:48	199.9.9.9	195.161.112.6	1721	80	1623	USGA	S

Automated  
Timestamps



Byte Sizes



Initial Flags



Homeland  
Security

# Beaconing Script

- The beaconing script uses several commands, as sampled below, to filter by flows for indications of hourly/daily/weekly beaconing activity:

```
for bytes in `rfilter --saddress=$victimip --daddress=$badip --type=all  
bin/$i.bin --pass=stdout | rwuniq --fi=bytes --flows=5 --no-titles --no-final-delimiter --no-columns  
| cut -d "/" -f1`  
do  
    daycount=`rfilter bin/$i.bin --type=all --saddress=$victimip --  
daddress=$badip --bytes=$bytes --pass=stdout | rwcut --fi=9 --no-titles | cut -d "/" -f3 | cut -d "T"  
-f1 | sort -u | wc -l`
```



# Findings Analysis: Beaconing

- Using seconds/milliseconds to build timeline
  - Helps dispel irregularities
  - Common traffic obfuscation technique for FakeAV and Rootkits

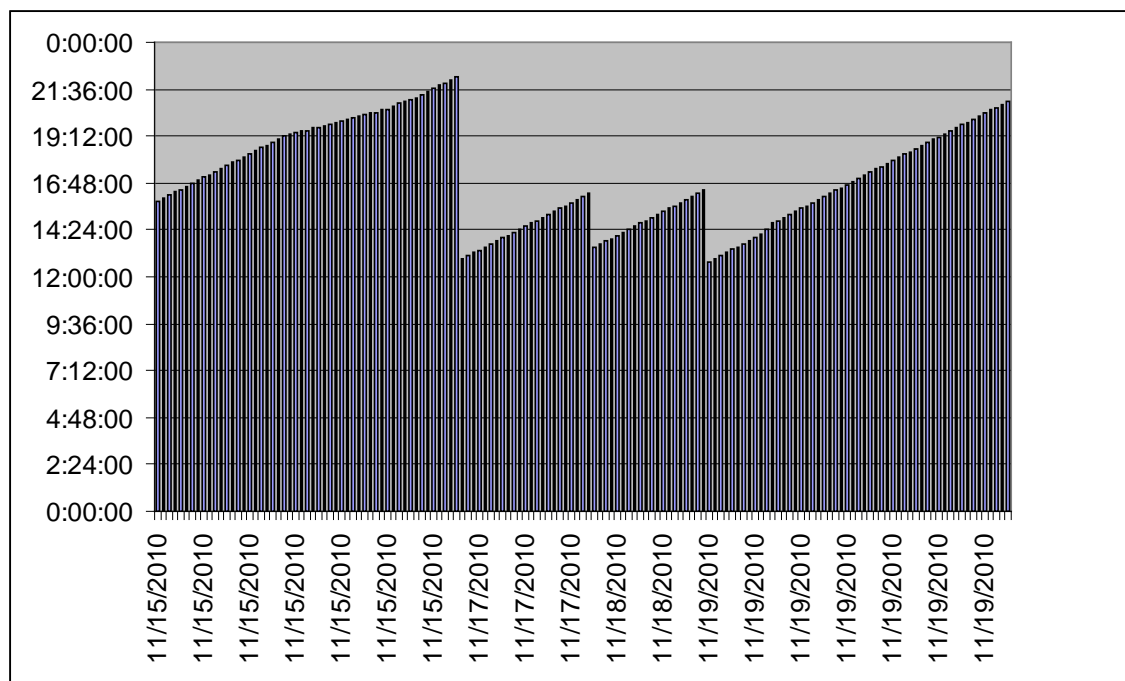
Sample Output (note the second count):

sTime	sIP	dIP sPort dPort	bytes	sensor	initialF	R	Records
2010/08/17T11:25:23	199.9.9.9	94.228.209.200  1529  80	549	USGA1	S		1
2010/08/17T14:21:23	199.9.9.9	94.228.209.200  1989  80	549	USGA1	S		1
2010/08/17T21:26:24	199.9.9.9	94.228.209.200  2346  80	549	USGA1	S		1
2010/08/17T22:32:24	199.9.9.9	94.228.209.200  2602  80	549	USGA1	S		1
2010/08/18T02:09:24	199.9.9.9	94.228.209.200  3103  80	549	USGA1	S		1
2010/08/18T05:43:24	199.9.9.9	94.228.209.200  3607  80	549	USGA1	S		1
2010/08/18T14:10:25	199.9.9.9	94.228.209.200  3996  80	549	USGA1	S		1
2010/08/18T16:18:25	199.9.9.9	94.228.209.200  4295  80	549	USGA1	S		1
2010/08/18T18:51:24	199.9.9.9	94.228.209.200  4640  80	549	USGA1	S		1
2010/08/19T05:22:24	199.9.9.9	94.228.209.200  1229  80	549	USGA1	S		1
2010/08/19T09:56:24	199.9.9.9	94.228.209.200  1341  80	549	USGA1	S		1
2010/08/19T15:42:24	199.9.9.9	94.228.209.200  1806  80	549	USGA1	S		1
2010/08/20T06:24:24	199.9.9.9	94.228.209.200  2186  80	549	USGA1	S		1
2010/08/20T09:37:25	199.9.9.9	94.228.209.200  2321  80	549	USGA1	S		1
2010/08/20T12:04:25	199.9.9.9	94.228.209.200  2871  80	549	USGA1	S		1
2010/08/21T15:22:25	199.9.9.9	94.228.209.200  3439  80	549	USGA1	S		1
2010/08/21T17:34:25	199.9.9.9	94.228.209.200  3532  80	549	USGA1	S		1



# Findings Analysis: Beaconing

- Graphical Representation
  - Easy-to-read synopsis of activity
  - Helpful handout/reference for constituency



- Victim IP observed beaoning every 8 minutes and 55 seconds



# Beaconing Script: Excel Charting

Beaconing excel macro is used to give pattern charts:

```
Sub Patterns()
'Patterns Macro
'Macro recorded 12/3/2010 by ttragess
'Keyboard Shortcut: Ctrl+Shift+T

Columns("B:B").Select
Selection.Insert Shift:=xlToRight
Columns("B:B").Select
Selection.Insert Shift:=xlToRight

Columns("A:A").Select
'Range("A549").Activate
Selection.TextToColumns Destination:=Range("A1"), DataType:=xlDelimited, _
    TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, _
    Semicolon:=False, Comma:=False, Space:=False, Other:=True, OtherChar _
    :="|", FieldInfo:=Array(1, 1), TrailingMinusNumbers:=True
Columns("A:A").EntireColumn.AutoFit

Columns("A:A").Select
Selection.TextToColumns Destination:=Range("A1"), DataType:=xlFixedWidth, _
    OtherChar:="|", FieldInfo:=Array(Array(0, 1), Array(10, 1), Array(11, 1)), _
    TrailingMinusNumbers:=True

totalrows = ActiveSheet.UsedRange.Rows.Count totalrows = Int(totalrows) beginRange = 1 loopcount = 1

For i = 1 To totalrows
Range("A" & i).End(xlDown).Select
'patterns Macro
'Macro recorded 11/26/2010 by ttragess
'Test contents of active cell; if active cell is empty, exit loop.
Do Until IsEmpty(ActiveCell)
```



# Beaconing: Excel Charting (cont.)

```
ActiveCell.Offset(1, 0).Select
endRange = ActiveCell.Address(False, False)
' myCell = ActiveCell.AddressLocal

endRange = Right(endRange, Len(endRange) - 1)

If loopcount = 1 Then
beginRange = 1
Else
beginRange = i - 1
End If
loopcount = loopcount + 1
i = endRange + 1
endRange = endRange - 1
goodguy = Range("D" & beginRange).Value
badguy = Range("E" & beginRange).Value
bytecount = Range("F" & beginRange).Value
Loop

Range("E" & beginRange).Select

Charts.Add
ActiveChart.ChartType = xlColumnClustered
ActiveChart.SetSourceData Source:=Sheets("Sheet2").Range("G" & beginRange)
ActiveChart.SeriesCollection.NewSeries

ActiveChart.SeriesCollection(1).XValues = "=Sheet2!R" & beginRange & "C1:R" & endRange & "C1"
ActiveChart.SeriesCollection(1).Values = "=Sheet2!R" & beginRange & "C3:R" & endRange & "C3"

ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet2"
With ActiveChart
.HasAxis(xlCategory, xlPrimary) = True
.HasAxis(xlValue, xlPrimary) = True
.HasTitle = True
.ChartTitle.Characters.Text = goodguy & " beaconing to " & badguy & "with a byte count of " & bytecount
End With
ActiveChart.Axes(xlCategory, xlPrimary).CategoryType = xlCategoryScale
ActiveChart.HasLegend = False

Next
End Sub
```





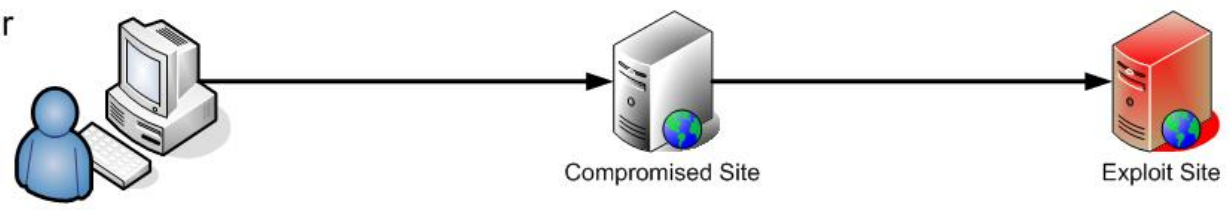
# Redirect Activity

- *Victim IP Address communicates with first mal IP/domain and is immediately redirected to a secondary mal IP/domain*
- Identifies malicious and anomalous activity
  - Tracks connections/patterns to IPs/domains of interest
  - Correlates activity with incident database information
  - Can help to:
    - Identify post infection beaconing such as pattern is seen every half hour before victim tries again.
    - Identify new types of malicious activity or malware based off of pattern recognition from the victim IP
      - First and last/size of bytes downloaded from each
    - Provide more than two attacker sessions and identify malicious traffic such as Gumblar

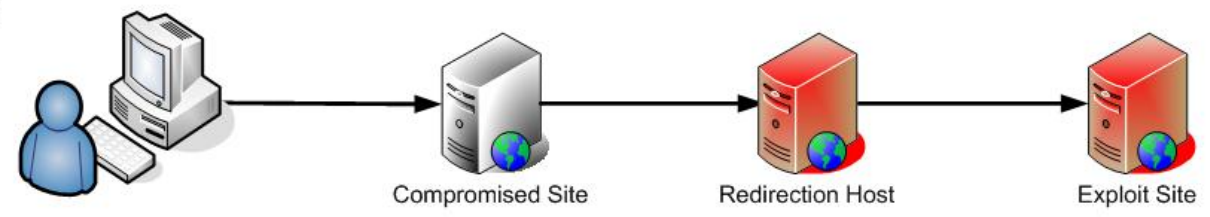


# Redirect Campaigns

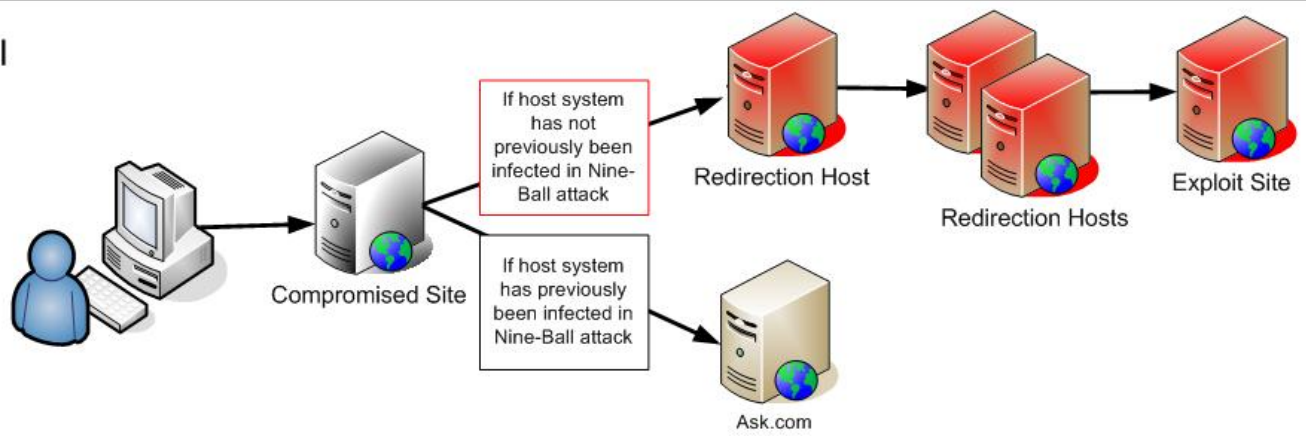
Gumblar



Beladen



Nine-Ball



# Redirect Criteria

- Victim initiates connection to first malicious IP address and then within milliseconds initiates connection to second malicious IP address. The victim then does the same activity 30 minutes later in a dual initiate connection to the malware IP address set.

- *VICTIM ----->> MAL1*
- *MAL1 -----> VICTIM*
- *VICTIM ----->> MAL2*
- *MAL2 -----> VICTIM*
  
- *VICTIM WAITS 30 MINUTES TO INITIATE NEXT SESSION*
- *VICTIM ----->> MAL1*
- *MAL1 -----> VICTIM*
- *VICTIM ----->> MAL2*
- *MAL2 -----> VICTIM*

- Alternate criteria:
  - Victim IP contacts several IP addresses/domains in sequence (and repeats activity). Examples include Gumblar or other fast flux activity.



# Redirect Code

- The snippet below creates the coupling between the victim and attacker IPs. Many more lines are used to accurately focus on back and forth communications, however this is the basis for pairing the attacker/victim:

```
# Check to make sure there was a ip.set for the pair of malicious IP addresses if so pull victim IP addresses  
and add them to one set.  
  
if [ -f $i.outweb.set ] || [ -f ${ip[$p]}.outweb.set ]; then  
    rwsetintersect --add-set=$i.outweb.set --add-set=${ip[$p]}.outweb.set --set=bothout.set if [ -f bothout.set ];  
then  
  
# Create the the flow data for the pair of malicious IP addresses.  
# from from the small binary files and place the results in a base.bin # Using the ip.set query of base.bin and  
place results in intersected.bin  
  
rwappend --create base.bin bin/$i.bin bin/${ip[$p]}.bin  
rwwfilter --anyset=bothout.set base.bin --pass=Intersected.bin  
count=`rwwfilter Intersected.bin --type=outweb --pass=stdout | rwsort --fi=22 | rwcut --fi=1-12,26 | grep -A 1  
$i | grep -B 1 ${ip[$p]} | wc -l`
```



# Findings Analysis: Redirect

- *Sample Output*

- *Quick second/millisecond session redirects*
- *Detected recent gbot activity w/ 2k+ infections*

sIP	dIP	sPort	dPort	packets	bytes	flags	sTime
attacker IP1	victim	80	1514	5	629	FS PA	2010/10/27T14:58:03.219
attacker IP1	victim	80	1519	5	629	FS PA	2010/10/27T14:58:05.072
attacker IP2	victim	80	1515	4	589	FS PA	2010/10/27T14:58:07.243
attacker IP2	victim	80	1515	1	40	A	2010/10/27T14:58:07.418
victim	attacker IP	1514	80	5	470	FS PA	2010/10/27T14:58:08.174
victim	attacker IP	1519	80	6	517	FS PA	2010/10/27T14:58:08.026
victim	attacker IP	1515	80	8	602	FSRPA	2010/10/27T14:58:11.159
victim	attacker IP	1515	80	1	40	R A	2010/10/27T14:58:14.418



# Suspicious

- Seeking to detect and identify 'suspicious activity' and outliers
  - Communicating with known mal IPs
  - Pattern matching/identification
  - Conjecture
- The query covers activity that may not be caught elsewhere
  - Low and Slow beaconing that may not be caught
  - High port to high port activity
  - Rootkit type activity with unique instructional patterns



Photo courtesy of CurrentTV



**Homeland  
Security**

# Data Exfil Criteria

- *Beaconing can potentially become data exfiltration when:*
  - *The victim IP address downloads a percentage of total packets exchanged (at least with web traffic).*



Image from huffingtonpost.com

*Noted false positives when the victim is a web server and normal web traffic exceeds downloaded data of 70-90% and uploads of 10-30%*



**Homeland  
Security**

# Suspicious Script/Code

- The suspicious script gets all possible victim IP addresses and then prints out traffic based on time (what the communication looked like back and forth) to help determine suspicious patterns. Simply put it is a straight rwcut filtered on time.

```
for j in `rwwfilter bin/$IP.bin --type=all --pass=stdout | rwuniq --fi=1 --no-titles --no-columns |  
grep -v $IP | cut -d "|" --fi=1 | sort -u`  
do  
sensor=`rwwfilter bin/$IP.bin --any-address=$j --pass=stdout | rwwcut --fi=12 --no-titles --no-  
columns --no-final-delimiter | head -1`  
sensor=`grep -w $sensor ../sensor.txt | head -1 | cut -d "|" -f2`
```





# Findings Analysis: Suspicious

- Heuristic detection techniques
- Rarely detects FakeAV

*Example Output: Victim IP uploaded 21360 bytes and downloaded 8142 bytes to malicious IP Address:*

sIP	dIP sPort dPort pro	packets	bytes	flags	sTime	dur	eTime	sensor initialF
victim	attacker 37688  80  6	6	288	S	2010/12/06T15:58:26.288	92.985	2010/12/06T15:59:59.273	
victim	attacker 41745  80  6	6	288	S	2010/12/06T15:58:35.282	92.985	2010/12/06T16:00:08.267	
victim	attacker 38283  80  6	6	288	S	2010/12/06T15:58:47.025	92.985	2010/12/06T16:00:20.010	
victim	attacker 23620  80  6	6	288	S	2010/12/06T15:59:02.375	92.982	2010/12/06T16:00:35.357	
victim	attacker 22906  80  6	6	288	S	2010/12/06T15:59:26.089	92.984	2010/12/06T16:00:59.073	
victim	attacker 48356  80  6	6	288	S	2010/12/06T16:00:05.258	92.984	2010/12/06T16:01:38.242	
victim	attacker 24169  80  6	6	288	S	2010/12/06T16:24:20.051	92.984	2010/12/06T16:25:53.035	



# Requirements

- **Commodity hardware and available storage capacity**
- **In-house development capability to create/tune/maintain scripts**
  - Update scripts based on new patterns and emerging threats
- **Process to coordinate actions/activities**
  - Standardization/certification of analytical process and background
- **Manpower to verify and/or vet findings for accuracy and action**



# Recommendations

- **Provide user-friendly portal/system to process findings**
  - **Hierarchical view for different users**
    - **Incident summary or overview for management**
      - Paraphrase activity and provide easy-to-understand format
      - HTML and Executive Summary reports
      - The report script is approximately 2500 lines of shell script and analyzez different parts of the above logs to give initial findings.
    - **Detailed view explaining specific query findings (e.g., beaconing, suspicious, etc.)**
    - **Detailed technical specifics for findings and incidents**
      - Incident findings
      - Department impacted
      - Associated activity
- **Provide automated methods and templates for processing**
  - **Vehicle and report template to disseminate validated findings**
    - i.e.- “Notify Accounting of virus identified on IP 1.1.1.1”



# Recommendations (cont.)

- **Standardize incident criteria, taxonomy, templates**
- **Normalize incident handling/analysis processes**
- **Standardize product and include incident information**
  - **Network Flow data**
    - Usual Stuff: Src/Dest IPs/Ports/Proto/Bytes/Time/etc.
  - **IP correlation / analyst notes / database entries**
  - **Include references (proprietary, open source, etc.)**
- **Trust but Verify**
  - **Ensure automated findings are checked for accuracy and properly vetted prior to dissemination, formal reporting and/or follow-up action**





# Considerations

- **Integrate into operations**
  - Ensure capability is properly integrated into operations commensurate with organizations priority and operational necessity
- **Maintenance and Functionality**
  - Be able to allocate support levels to add/modify as necessary
- **Eyes-on analysis/vetting**
  - What person/department and what level of granularity





# Benefits

- **Discover and detect security events and malicious activity**
  - Predicated on flow data
  - Expand incident discovery/detection capabilities
  - Timely and effective reporting of security incidents
  - Enables mitigation and remediation of findings
  - Scalable and especially useful for large/compartemented enterprises
- **Automated query process**
  - 2-click vetting and approval process optimal (depending)



# Takeaways

- **Harness flow data to identify security events and incidents of interest across the enterprise**
- **Develop automated queries to do work for you and vet results for accuracy**
  - **Tune appropriately**
- **Layered view to provide a user friendly view of information and data pertinent to different levels of org.**
  - **Customize different views across organization:**
    - **Leadership / Security Operations**
    - **Technicians / Responders**
    - **Constituents (if desired)**



# Contact

## ■ US-CERT

- US-CERT Security Operations Center  
Email: [soc@us-cert.gov](mailto:soc@us-cert.gov)  
Phone: +1 888-282-0870
  - US-CERT Information Request  
Email: [info@us-cert.gov](mailto:info@us-cert.gov)  
Phone: +1 888-282-0870
  - GFIRST: [gfirst@us-cert.gov](mailto:gfirst@us-cert.gov)
- Information available at <http://www.us-cert.gov>



**Homeland  
Security**





**Questions?**



# Homeland Security