Security Use Case:
Digital Marketer

> "Finding the 'bad guy' wasn't going to be a winning battle if I could only use 'known bad' searches and rules. What I really needed was something that could assist me in automatically finding those users that categorically act different than most 'normal' users."
>
> – Craig Merchant, Sr. Security Architect at Responsys

COMPANY TYPE

Digital marketing and advertising

USE TYPE

IT Security

KEY BENEFITS

- Detect rogue users using netstat data
- Identify data exfiltrations
- Find value in IDS log noise

THE COMPANY

Founded in 1998, **Responsys** is a leader in relationship-based digital marketing, allowing their clients to build automated "customized interactions" with their customers via email, display ads, mobile, and social media based on their activity. Responsys was acquired by Oracle in 2014 for $1.5 billion.

THE PROBLEM

With an infrastructure capable of sending out massive amounts of emails in a short period of time, and with a full list of valid customer email addresses as an opt-in only service, Responsys is a huge target for spammers. Many major competitors (**Return Path, Silverpop, Epsilon** and **CheetahMail**) had experienced major security breaches in 2010 and 2011 – which led to massive losses of large enterprise customers, many of who came to Responsys.

Responsys wanted to be sure to protect their client's sensitive customer email information and quickly identify rogue users who were abusing the system, but traditional techniques (aggregating and searching IDS, gateway and internal service logs) were not working. Responsys struggled to effectively monitor all their systems for a variety of reasons:

System was too complex to accurately monitor manually

- No realistic way to manually write a model and thresholds to monitor the behavior of their 1,600+ unique servers and thousands of connections each day
- Pattern matching behavioral analysis would be almost impossible to build by hand
- Their unstructured data (like Linux Syslog or F5 data) was difficult to manually profile behavior for
- Too complex and labor intensive to create a baseline of "normal" behavior on a per-host, per-application, per-network basis

Could only be reactive and monitor for known threats – no way to find zero-day threats

- Could only monitor and write alerts for what they knew to look for, and only knew what behavior should be flagged based on previous experiences
- A lot of times you only find out that data is "interesting" after there is a problem

Responsys was monitoring each server's typical network behavior as a ratio of inbound to outbound traffic. Monitoring servers for a significant change in that ratio would be a good early warning system of a successful exfiltration attempt. Unfortunately, standard search or statistical techniques were not always useful since standard deviations of average traffic ratios across servers would not account for the huge differences in normal behavior between servers. And since searching for anomalies in servers' traffic signatures required understanding their typical behaviors, the approach is increasingly unrealistic as the number of servers increased.

The bottom line: Responsys couldn't accurately write rules for every possible anomalous behavior that should be flagged, especially the ones they wouldn't know to watch out for.

THE SOLUTION

Since 2013 Responsys has teamed with Prelert and been using Anomaly Detective to detect rogue users with automated population analysis and identify unusual internal activity and traffic to identify employees' computers affected with malware.

THE RESULTS

Anomaly Detective helped Responsys:

- **Detect Rogue Users Using Netstat data:** Identify unusual external connections by monitoring Netstat data such as a user FTPing large quantities of data
- **Find data exfiltrations:** Monitor NetFlow logs with real-time behavioral analysis
- **Make Sense of IDS Noise:** Transform the number of severe IDS alerts from 2K-6K daily to an accurate and manageable dozen per week

Operational benefits:

- Increased mean time between failures

Security benefits:

- Shorter mean time to detection
- Greater visibility into complex and blended threats

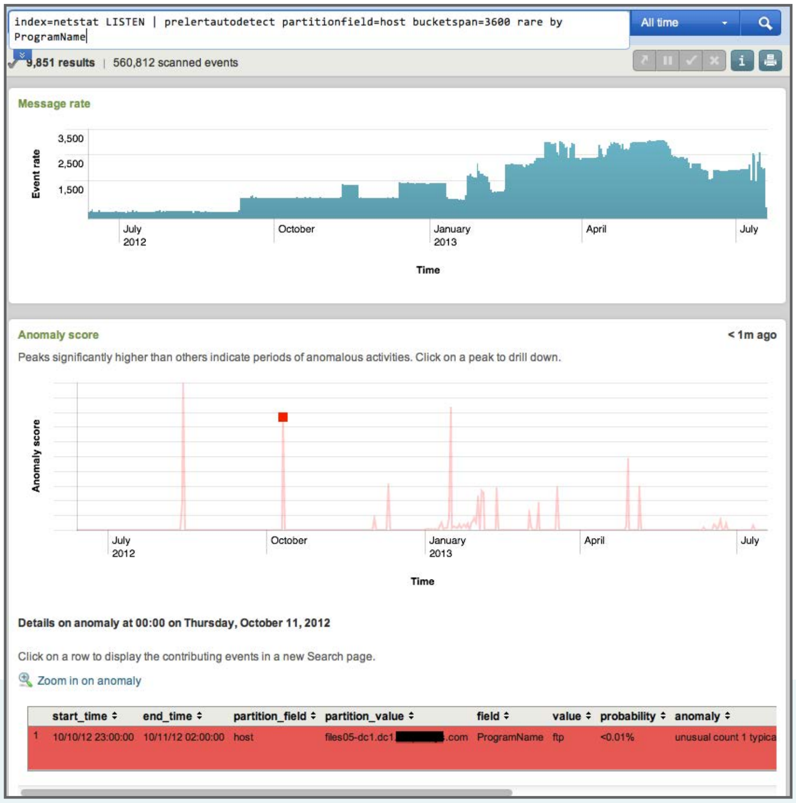
Anomaly detection software provided exactly the tool Responsys was looking for. Using network traffic data, thousands of servers are automatically baselined to determine their normal network I/O profiles. This is monitored in real-time and alerts are issued when a change occurs in a server's traffic profile. Since anomaly detection software ranks the severity of alerts based on the probability (severity or rarity) of the anomaly, they can easily filter out minor fluctuations and zero right in on significant changes that could signal attacks or exfiltration attempts.

USE EXAMPLE - DETECT ROGUE USERS USING NETSTAT DATA

Identify potential rogue users through unusual external connections in Netstat data

Craig Merchant, Sr. Security Architect at Responsys, was able to identify suspicious FTP behavior with a quick autodetect command.

Merchant wanted to look for any anomalies in their Netstat data. He aggregated a year and a half's worth of machine data from multiple machines and searched for any rare applications by host. Anomaly Detective flagged one severe anomaly of an unusual outbound connection back in 2012.



```
index=netstat LISTEN | prelertautodetect partitionfield=host bucketspan=3600 rare by
ProgramName
```

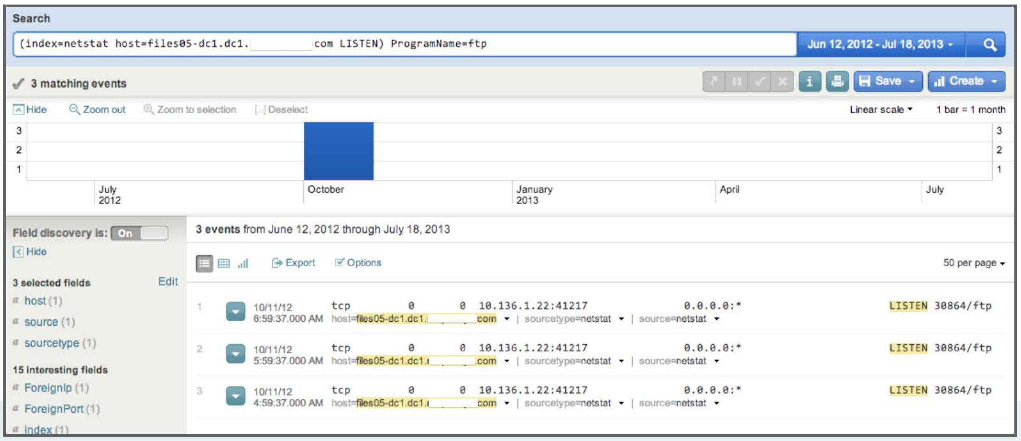
Details on anomaly at 00:00 on Thursday, October 11, 2012

Click on a row to display the contributing events in a new Search page.

Zoom in on anomaly

	start_time ⇕	end_time ⇕	partition_field ⇕	partition_value ⇕	field ⇕	value ⇕	probability ⇕	anomaly ⇕
1	10/10/12 23:00:00	10/11/12 02:00:00	host	files05-dc1.dc1[redacted].com	ProgramName	ftp	<0.01%	unusual count 1 typica

By clicking on the anomaly, Merchant was able to see more related information, and was startled to see that this was coming from a security box he himself managed.



Search

```
(index=netstat host=files05-dc1.dc1.          com LISTEN) ProgramName=ftp
```

✓ 3 matching events

Field discovery is: On

3 events from June 12, 2012 through July 18, 2013

3 selected fields Edit
- host (1)
- source (1)
- sourcetype (1)

15 interesting fields
- ForeignIp (1)
- ForeignPort (1)
- index (1)

```
1  10/11/12          tcp      0      0  10.136.1.22:41217        0.0.0.0:*              LISTEN 30864/ftp
   6:59:37.000 AM    host=files05-dc1.dc1.        .com  | sourcetype=netstat  | source=netstat
2  10/11/12          tcp      0      0  10.136.1.22:41217        0.0.0.0:*              LISTEN 30864/ftp
   5:59:37.000 AM    host=files05-dc1.dc1.        .com  | sourcetype=netstat  | source=netstat
3  10/11/12          tcp      0      0  10.136.1.22:41217        0.0.0.0:*              LISTEN 30864/ftp
   4:59:37.000 AM    host=files05-dc1.dc1.        .com  | sourcetype=netstat  | source=netstat
```

Merchant drilled down to look at the list of related events and identified that it was an outbound FTP call that was flagged. He copied the destination IP address displayed on the dashboard, did a quick lookup, and was relieved to see that it belonged to a vendor of their own IDS solutions and this anomaly was from when Merchant had FTPed them over some very large log files. While this was not actually a security threat, it assured Merchant that should a hacker or rogue user try to FTP data out of the network, it would be easy to spot.

This type of analysis can easily be applied to different types of data people might be using, like PS input for Unix app for running processes, or the equivalent using WMI for windows. You can ask the software to, for example, "show the processes that are running on one of my 10 web servers that are supposed to be identical," or "take the output of LS mod and show me an output that has a kernel module loaded that none of my other web servers do" – you can do this for file integrity events, RPM database, etc. Put simply, **anything that is a process, application, or action that one machine might take that is outside the norm is really easy to detect with this tool.**

USE EXAMPLE - FIND DATA EXFILTRATIONS

Identify advanced persistent threats and data exfiltration attempts by monitoring NetFlow data

In addition to other logs, Responsys uses NetFlow as a security tool to monitor IP traffic. While it may be easy to obscure an attack on an IDS system – forge a source address, trigger thousands of alerts, and do what you want hidden in all the noise – it's nearly impossible to hide from NetFlow.

Since for the most part, Responsys' employees are "consumers" of data (very few upload or export), they can easily be monitored by looking for changes in the **app_byte_ratio.**

app_byte_ratio = (src_app_bytes – dest_app_bytes) / (src_app_bytes + dest_app_bytes)

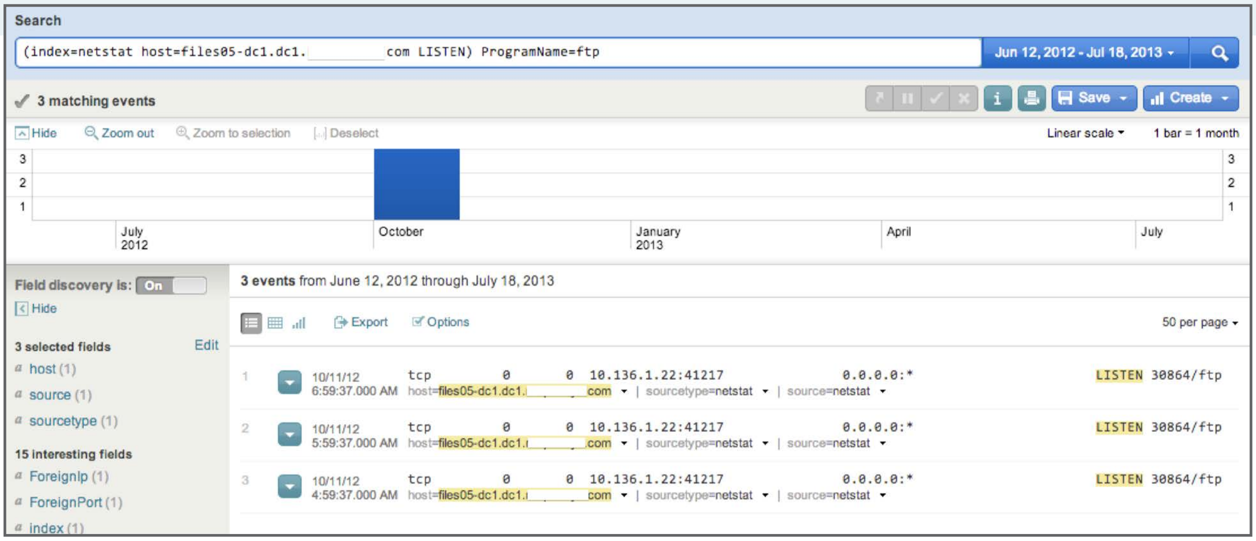
An app_byte_ratio of 1 would be a "pure data producer" and an app_byte_ratio of -1 would be a "pure data consumer." Any changes from the norm in this ratio for a host port could signal that a host has been compromised. This is a useful metric to use because it is not volume dependent – any deviations over time will be flagged, whether it is a small amount or a large amount of data being uploaded.

To test this, Merchant has set up several test scenarios to ensure this monitoring is effective. In one instance, Merchant uploaded a large chunk of data on a port to see if it would be flagged.

In Anomaly Detective, Merchant set up a command to look at the app_byte_ratios for all 65,000 unique ports to see if there had been any significant changes.

```
sourcetype=”netflow” dest!=10.* | eval dp=dest +” : “dest_port | prelertautodetect
bucketspan=300 partitionfield=src app_byte_ratio by dp
```

The command used above says to use NetFlow data as the source, look at destinations that are not on the internal network, create a field that is a combination of the destination and destination port, then look for the app_byte_ratio value by the destination and destination port, partitioned by source, and identify any anomalies (significant changes). This is the output Anomaly Detective generated:



```
(index=netstat host=files05-dc1.dc1.[redacted]com LISTEN) ProgramName=ftp
```

3 matching events

3 events from June 12, 2012 through July 18, 2013

3 selected fields
- host (1)
- source (1)
- sourcetype (1)

15 interesting fields
- ForeignIp (1)
- ForeignPort (1)
- index (1)

```
1  10/11/12            tcp    0    0  10.136.1.22:41217    0.0.0.0:*    LISTEN 30864/ftp
   6:59:37.000 AM      host=files05-dc1.dc1.[redacted]com | sourcetype=netstat | source=netstat
2  10/11/12            tcp    0    0  10.136.1.22:41217    0.0.0.0:*    LISTEN 30864/ftp
   5:59:37.000 AM      host=files05-dc1.dc1.[redacted]com | sourcetype=netstat | source=netstat
3  10/11/12            tcp    0    0  10.136.1.22:41217    0.0.0.0:*    LISTEN 30864/ftp
   4:59:37.000 AM      host=files05-dc1.dc1.[redacted]com | sourcetype=netstat | source=netstat
```

Anomaly Detective clearly flagged the deviation caused by the data upload (which altered the **app_byte_ratio**).

USE EXAMPLE - MAKE SENSE OF IDS NOISE

Reduce the number of severe alerts in IDS logs to an accurate and manageable few

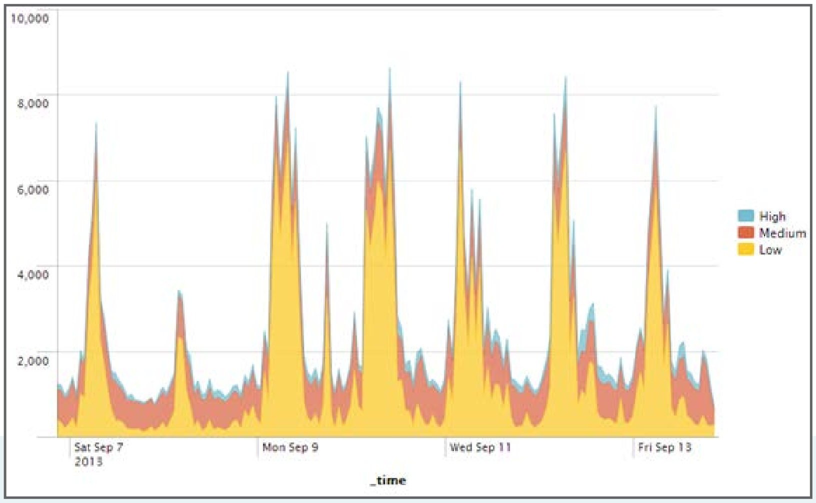
Before Anomaly Detective, Responsys' IDS logs would register between 100K-150K events per day, and would generate 2K-6K high severity alerts – an impossibly large quantity for their security team to sift through. The high signal-to-noise ratio and sheer number of false positives was creating unnecessary work tickets, and the admins were worried about attackers taking advantage of this noise.

Instead of relying on writing rules and thresholds (which despite their best efforts resulted in thousands of false alerts), Anomaly Detective helped the team easily monitor for red flags in their IDS logs with a single command:

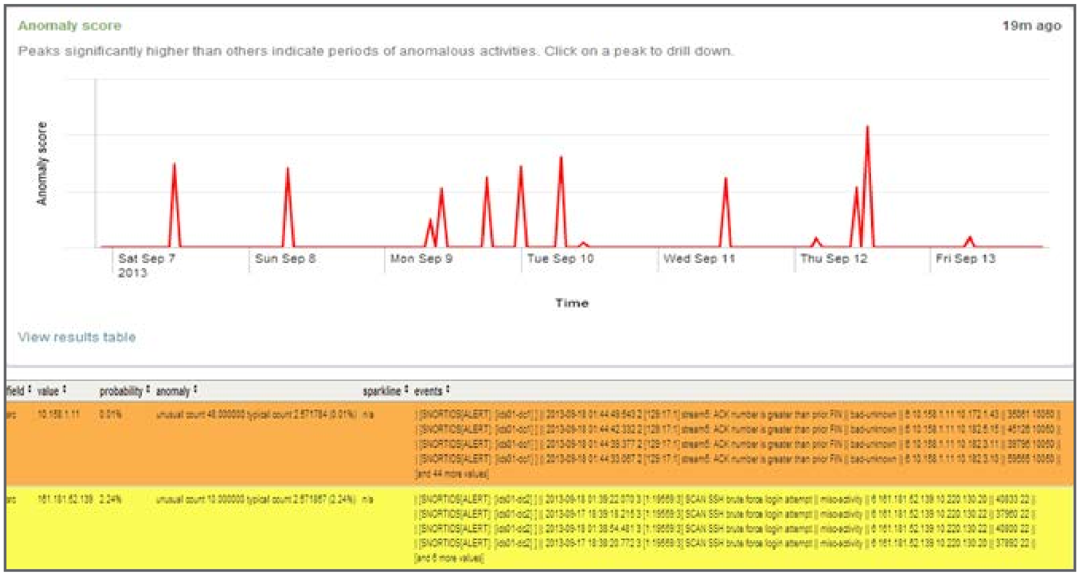
```
index=ids sourcetype=snort | eval dest:protocl:port=dest+”:”+dest_port | prelertautodetect
population_rare(signature_name), population_distinct_count(category), population_rare_
count(dest:protocol:port) by src
```

This command is saying to look at the IDS index from Snort (the software they use to aggregate their data), and create a new field that is the destination protocol and destination port. Run this through Prelertautodetect and identify the source as the population, and for each individual source that Snort sees, look for rare signature names for that particular source. Then look for an unusual distinct count of categories for that source, and when that source talks to a destination protocol and port look for connections that are rare that the host has never triggered before, or where that host is generating an unusual amount of events for that particular destination.

Anomaly Detective helped Responsys turn their daily IDS alert dashboard from showing thousands of alerts each day…



Into 1-3 IDS alerts per day:



Peaks significantly higher than others indicate periods of anomalous activities. Click on a peak to drill down.

field	value	probability	anomaly	sparkline	events
src	10.158.1.11	0.01%	unusual count 48.000000 typical count 2.571784 (0.01%)	n/a	| [SNORTIDS[ALERT] [ids01-dc1] | | 2013-09-18 01:44:49.543 2 [129:17:1] stream5: ACK number is greater than prior FIN | bad-unknown | 6 10.158.1.11 10.172.1.43 | 35861 10050 | | [SNORTIDS[ALERT] [ids01-dc1] | | 2013-09-18 01:44:42.332 2 [129:17:1] stream5: ACK number is greater than prior FIN | bad-unknown | 6 10.158.1.11 10.182.5.15 | 45126 10050 | | [SNORTIDS[ALERT] [ids01-dc1] | | 2013-09-18 01:44:39.377 2 [129:17:1] stream5: ACK number is greater than prior FIN | bad-unknown | 6 10.158.1.11 10.182.3.11 | 39796 10050 | | [SNORTIDS[ALERT] [ids01-dc1] | | 2013-09-18 01:44:33.567 2 [129:17:1] stream5: ACK number is greater than prior FIN | bad-unknown | 6 10.158.1.11 10.182.3.10 | 59666 10050 | [and 44 more values]
src	161.181.52.139	2.24%	unusual count 10.000000 typical count 2.571867 (2.24%)	n/a	| [SNORTIDS[ALERT] [ids01-dc2] | | 2013-09-18 01:39:22.070 3 [1:19559:3] SCAN SSH brute force login attempt | misc-activity | 6 161.181.52.139 10.220.130.20 | 40833 22 | | [SNORTIDS[ALERT] [ids01-dc2] | | 2013-09-17 18:39:18.215 3 [1:19559:3] SCAN SSH brute force login attempt | misc-activity | 6 161.181.52.139 10.220.130.22 | 37960 22 | | [SNORTIDS[ALERT] [ids01-dc2] | | 2013-09-18 01:38:54.481 3 [1:19559:3] SCAN SSH brute force login attempt | misc-activity | 6 161.181.52.139 10.220.130.22 | 40800 22 | | [SNORTIDS[ALERT] [ids01-dc2] | | 2013-09-17 18:38:20.772 3 [1:19559:3] SCAN SSH brute force login attempt | misc-activity | 6 161.181.52.139 10.220.130.20 | 37892 22 | [and 6 more values]

Above is a dashboard for a week's worth of data, showing a small handful of alerts, as well as their severity. Anomaly Detective slashed the number of alerts into a manageable handful. Viewers can click on each anomaly flagged to easily drill down and see related events as well as more details to determine which alerts require further investigation.

TRY IT FOR YOURSELF

See what Anomaly Detective can do to make sense of your IDS alerts, Netstat logs, NetFlow, and more. **Download it for free** and you can have it set up and automatically monitoring your data in the background in real-time within minutes. No more manual alert thresholds. No more sifting through mountains of false alerts.