Geospatial Anomaly Detection in Internet Usage

Chad M.S. Steel, *Member, IEEE*

***Abstract*—Geospatial anomaly detection makes use of the spatial clustering present in web requests to identify requests which are physically outside of the expected location values. These outlier requests may indicate malicious usage, misconfigured systems, unusual foreign activity, or spyware infections. Our system uses a dynamic, cell-based outlier detection approach to highlight anomalies in real time, even on large volume systems. With multiple modes of operation, the system can be easily deployed to just about any environment that uses a proxy server for Internet access.**

***Index Terms*—Geospatial data, anomaly detection, proxy log**

I. Introduction

GEOSPATIAL anomaly detection is used to identify outliers in geographic usage data. This paper examines geospatial anomaly detection as a useful tool in identifying Internet misconduct. The proxy log data from 229 users comprising 6.2 million hits is utilized as a basic dataset for analysis.

To perform the detection, each hit is geolocated by the address of the server. The results are then examined to identify outliers using spatial outlier detection, and manually examined to determine their cause.

By performing geospatial outlier detection, users who visit irregular sites will be highlighted. Specifically, users visiting foreign sites not generally associated with an institution can be identified without any a priori knowledge of the institution's expected usage patterns.

Overall, geospatial anomaly detection can be added to general intrusion detection as a mechanism for finding inappropriate usage. In addition to heuristic and statistical methods, geospatial anomaly detection can easily be integrated into a comprehensive intrusion detection capability. Because traditional models have focused on volumes, timing, and content of usage, the addition of geoanomaly detection brings a completely new mechanism to the arsenal.

II. Prior Art

A. Outlier Detection

General outlier detection algorithms come from the world of statistics. Models of outlier detection have been used for auditing, process improvement, and anomaly detection in for both general and applied statistics [1, 2].

B. Spatial Outlier Detection

Lu et al apply statistical outlier detection to the field of spatial data, and compare four algorithms for effectiveness. They identify the z, iterative z, iterative r, and Median algorithms as being relevant to the successful detection of spatial outliers[3]. Because the outliers in individual web usage have associated quantities, malware or misconfiguration may be detectable using these techniques.

For general spatial outlier detection without associated values, direct statistical application using distance has been used. k-Nearest-Neighbor and cell-based approaches have been successfully used in detection[4, 5]. For very large datasets, additional approximations have been successfully employed[6]. These techniques are used for geospatial anomaly detection to identify overall distance and density-based (as opposed to value-based) outliers which represent unusual global usage.

C. IP Address Mapping

IP address information has been successfully mapped by IP2Location[7] as well as GeoLite[8]. Additionally, information on network address translation was presented in the original proposal for removal of non-public IP addressing[9].

D. Intrusion Detection

General intrusion detection has primarily been concerned with network-based intrusions and attacks. Axelsson provides an excellent survey of the field[10]. While general web-centric intrusion detection is a major topic of research[11-13], user activity driven intrusion detection is still in its infancy in terms of prior art.

III. Dataset

The dataset used represents the HTTP and HTTPS usage data for 229 users over the course of thirty days. The data were collected from a small business organization which made use of a proxy server through which all web requests were routed. The proxy server recorded each request, a corresponding timestamp, and the user name making the request. 6.2 million total requests were logged for all of the users over the thirty day period.

The requests represent a homogenous group of users from the perspective of their employment and geospatial location. The users at the organization were permitted some personal use of the Internet, as long as it was not considered excessive.

Specific uses were prohibited, however, including eBay trading, viewing pornography, and Internet gambling.

Because of the cultural differences present in the workforce and the allotment for some personal use, some diversity in usage patterns is expected. Given a large enough dataset, however, the aggregate personal usage should be homogenous enough to eliminate personal interests as a sole factor in outlier generation.

IV. METHODOLOGY

A. Overview

The methodology of the proxy-based geoanomaly detection system is broken into two steps. First, the web requests from each user are geocoded by the location of the web server – this is done statically for the initial dataset, but can be performed dynamically for the live system. Second, the geocoded requests are grouped into a cell structure which is used for rapid outlier detection. The overall methodology is displayed in Figure 1 below.

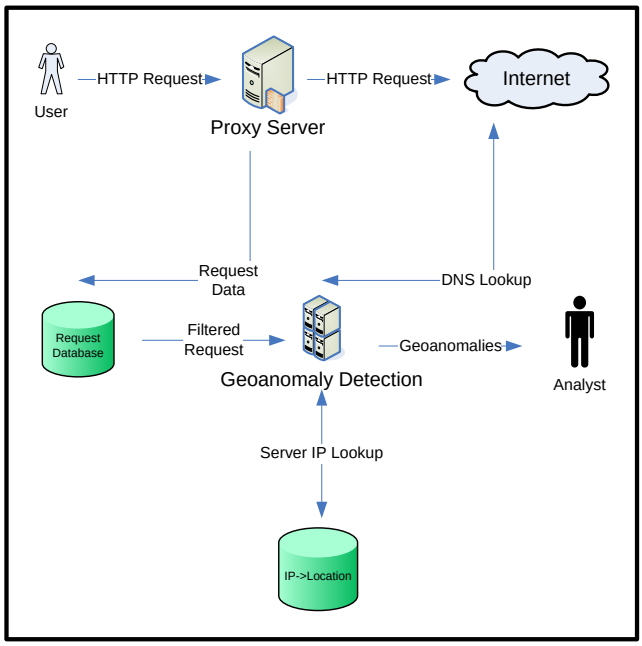

Fig.1. HTTP Request Geoanomaly Detection

B. Architecture

The proxy log geoanomaly detection system utilizes a two tiered architecture with a thick client, in addition to an external API for visualization. The API integrates with Google Earth to allow for robust mapping of results and greater visualization of the data.

The presentation tier is written in C# and consists of a GUI with user-changeable parameters. The presentation allows for the connection to a backend datastore through ODBC, and displays both the pre-analysis raw data and the results of the analysis. C# was chosen for its ease of integration with numerous backend data sources.

The data layer is a SQL Server 2003 database. Because the geographic operations are performed on a cell-basis, the ease of integration and the scalability of SQL Server were chose over MySQL or similar packages with integrated geospatial structures. Additionally, the SQL Server 2003 implemented allowed for easy downstorage to Access 2003 for remote, standalone operation of the tool.

In addition to the main tool suite, there are two external applications which are utilized. First, the Microsoft API for external image viewing is used to resize and view the cell output visually. Second, the Google Earth API is used with KML, the Google Earth geographic markup language, to send results directly to a Google Earth application.

C. Geocoding

Geocoding was performed on the web requests to identify the web server the request was made to for each request. First, the data was cleansed to remove any personally identifiable information – for an actual, live implementation this step would be undesirable, but it was done with the sample data as part of the usage agreement. Second, a reverse DNS lookup is performed on each request and that IP address is geocoded into longitude and latitude coordinates on the map. The resultant data is then passed on to the geoanomaly detection engine for processing.

The original dataset was de-identified to remove any personally identifiable information. All query strings were removed completely, and unique domain names associated with the organization were altered to be generic. This was done to mask any individual information present in requests, and to mask the organization represented. Each individual user name was replaced with a unique, sequential identifier to maintain grouping information.

After performing the initial cleansing, the dataset was parsed into fixed records and bulk loaded into a Microsoft SQL Server database. SQL server was used as a high performance choice given the specific algorithm implemented for outlier detection, as it did not require specialized geospatial extensions.

After loading, a separate aggregated table was created based on usage groupings per-site. This allowed for quicker lookups (one lookup per domain name) in the following steps. Additionally, data not used in the geolocation algorithm (such as response time for the request) was discarded. Finally, the data was cleansed of requests that did not result in successful transfers and only 200-level HTTP responses were considered. This had a positive side effect of removing duplicate requests where HTTP code 302 redirects were issued.

Each of the domain names listed was mapped to an IP address using a reverse lookup. The reverse lookup was implemented using a domain server local to the original organization. This ensured local DNS persistent lookup locations remained consistent for subsequent requests[14]. This is not expected to be an issue as users of large services like Yahoo or Google are not choosing the service based on

location, and because of their high usage volumes geolocating all requests to the same location removes them as outliers as would be expected.

Once the IP locations for each request were obtained, the sites were geolocated into latitude and longitude point locations. The point locations were determined by a mapping of IP address ranges to actual physical coordinates on a map. In addition to the actual ranges, a set of normalized ranges were calculated for each site to allow for cleaner calculations in the other steps. The normalized range mapped from values between 0 and 100, based on the highest and lowest longitudes and latitudes present[1]. This allowed for the full range of space to be used, instead of having compressed space based on unused locations (very few web servers are within the Artic Circle, for example).

D. Geoanomaly Detection

After geocoding, a spatial outlier algorithm was applied to the full dataset to find abnormal usage. The outliers were identified using the cell based approach described by Knorr et al[4]. This approach was used for its speed in implementation for a live system when combined with a simple most recently accessed queue structure, but worked well for the static data also.

Following the geocoding work, each of the points was mapped to a grid of cells. Each cell in the grid was of size $l = \frac{D}{2\sqrt{2}}$ [4]. D was used as part of the distance-based outlier definition DB(p,D) from Knorr et al which states:

> An object 0 in a dataset T is a DB(p, D)-outlier if at least fraction p of the objects in T lies greater than distance D from 0.[4]

D was permitted to be variable, but a value of 1 appeared to work well for refining the number of anomalies. With a D value that is too low, there are too many anomalies detected to adequately investigate. Additionally, too many grid entries cause update problems when new entries are added or removed. With a D too high, interesting outliers are missed. Given a range of 0 to 100 and a D value of 1, 80000 possible cell locations are generated. Given approximately 6.2 million datapoints, an average cell density of 77.5 entries would be expected for a flat distribution.

Given the large dataset used and the desire to have a smaller number of outliers (trading a larger recall in return for a higher precision), a p value of .999999 was used. This value, given a normal distribution, should identify as outliers any items that don't have at least 6 neighbors within a distance of D.

After mapping the data to cells, each of the cells was coded as to the presence of outliers or not. First, a threshold of $m = t(1-p)$ is calculated where t is the total number of items, resulting in an m of 6.2. Next, each of the cells is scanned to find cells that cannot contain outliers in addition to those that probably contain outliers. L1 and L2 neighbors are shown in Figure 2. Specifically, from Knorr:

1. For each cell with more than m objects, label that cell red and all L1 neighbors of that cell pink (unless already red). Neither the cell nor it's neighbors will contain outliers.
2. For each white cell, sum all of the L1 neighbor values plus the cell value. If they are greater than m, label the cell pink.
3. If the cell is still white, sum all of the L1 and L2 neighbors plus the cell value. If that total is less than m, all objects in the original cell are outliers.
4. If the total in 3 is greater than m, evaluate the distance of each object in the cell to each L2 object and add each L2 object less than distance D plus the L1 and cell object counts. If the total is less than m, mark the object an outlier.[4]

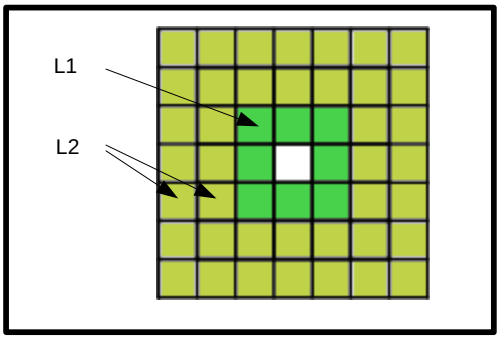

Fig 2. L1 and L2 outliers.

The system is designed to run in two specific modes – weighted and unweighted. The weighted mode inserts each individual site visit into a cell. The unweighted mode takes each unique site as a single visit and ignores the total number of visits to that site.

The weighted mode is designed to identify outliers which have a significantly lower number of hits than expected for their neighborhood. Individuals visiting a single site or two that are outside of the norm will be highlighted by the weighted results.

The unweighted mode is purely designed to identify outliers which are geographically distant from their neighbors but which may represent a large number of hits. Misconfigured systems and malware may result in a large number of hits generated to the same location, but with an unweighted algorithm the results will still appear as outliers.

E. Visualization

Following the cell assignment and coding, the resultant data is visualized in three ways. Initial visualization is through the general highlighting of outliers. Secondary visualization is

[1] This assumes a static dataset. For a dynamic dataset, the range would need to be fixed at the full range for normalization as it could not be known ahead of time where future requests would be located.

performed by color coding a normalized grid based on the algorithm results. Tertiary visualization is achieved by using an interface to Google Earth to display the results in an interactive fashion.

The primary visualization tool is the GUI, which consists of tuning boxes for each of the parameters and a selectable datagrid for displaying results. The use of a datagrid allows for near-instant sorting and for quick export of the selected data to other applications for analysis. The outliers in the testing were exported into Excel for manual review directly from the datagrid.

In addition to the standard GUI, a secondary area which shows clusters visibly is integrated into the tool. Using the color coding proposed in [4], the tool expands and separately highlights individual cells which contain outliers separately from non-outlier white cells. Additionally, the cell visualization can be magnified for enhanced viewing of individual areas in an external application. A screenshot of the GUI is shown in Figure 3 below.

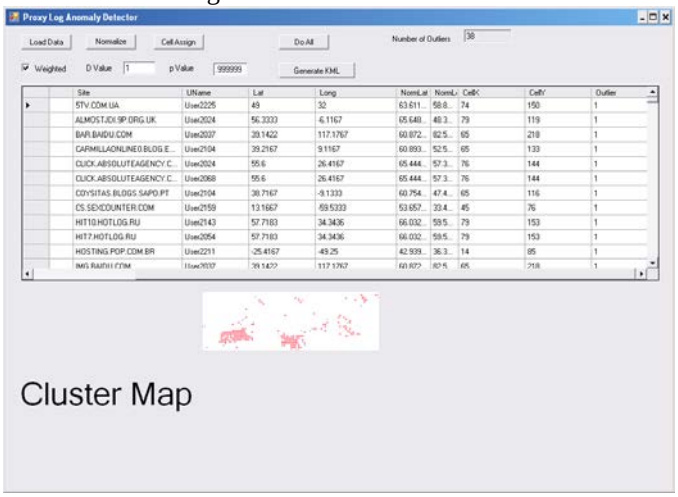

Fig 3. Screenshot of proxy analysis GUI

Finally, the application integrates with Google Earth for interactive viewing. The outlier results can be translated into KML for specific Google Earth plotting. The plotting can include the outliers and the original data, so clustering can be viewed on a map and the distances can be shown between outliers and clusters in a direct fashion. The generated KML can be ported easily to other applications for plotting or transformed through XSLT to be added automatically to a database or XML format which would integrate with other alerting systems. Figure 4 shows the Google Earth results of outlier analysis, while Figure 5 shows the overall mapping. The sparse outliers in Figure 4 are in stark contrast to the clear clusters visible in Figure 5. Figure 5 has two clear clusters on the east and west coast corresponding to common hosting areas. These can be seen later in the cluster maps for the results below.

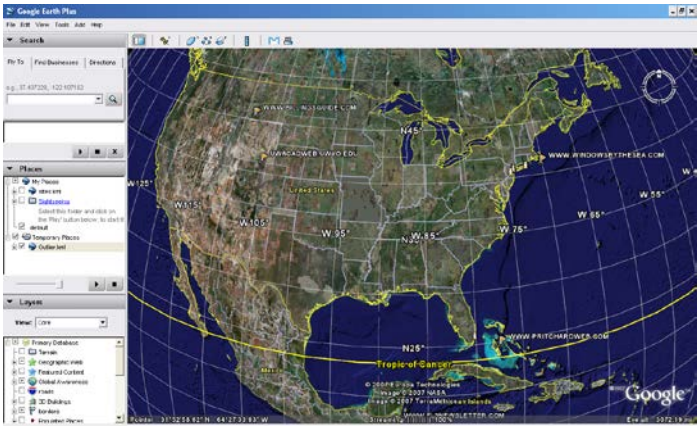

Fig 4. Outliers detected.

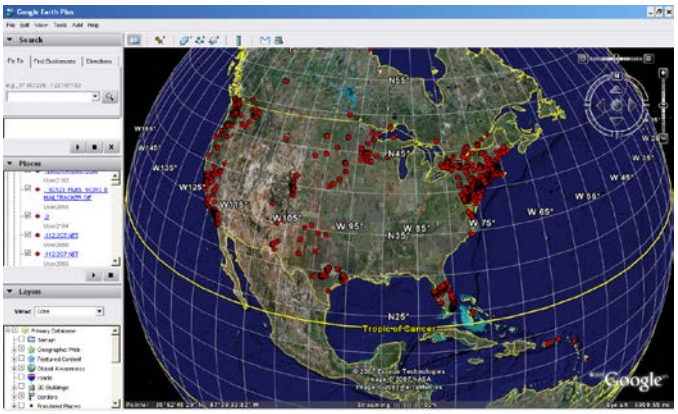

Fig 5. Full site map.

V. Live Operations

A. *Overview*

The design of the system was intended for three modes of operation - batch mode for analysis of longer-term trending as well as real time and pseudo-real time modes for detecting anomalies as they occur. The details of batch mode are described above, and both real time and pseudo real time modes are described below.

B. *Real Time Mode*

Real Time mode (RTM) detects misuse as it occurs. Each individual site request is plotted and checked for outlier status based on a sliding window of the last N results.

The window size is mostly irrelevant to calculations, as the definition of an outlier is relative to the total number of objects. It may stand to reason that a larger window may provide better results, however this is contrary to the temporal nature of requests. In a large-window scenario, outdated accesses may hide future outliers.

As an example, a sudden interest in news stories from Beijing in anticipation of the Olympics being held there is not unusual. With a small window, there may be an initial outlier for the first access, but it may be expected that more regular accesses to that geographic location were valid for a specific timeframe around the games. Several months after the games, hits to sites that are located there geographically would be

expected to decline sharply, and may become outliers. If a large window is used, those outliers will never be detected because of a strong, albeit short, signal that occurred much earlier for a known reason.

The individual items used in the window are stored in a simple most-recently accessed queue structure which includes a pointer to the object itself. As a new hit is requested and added to the end of the queue, the least recently used item is removed from the queue as well as the cell in which it resides, resulting in two separate operations. Any new outliers generated are then output to an analyst for further review.

The cell structure limits the number of checks that must be performed for each update to detect outliers. Each update may result in a new outlier detected for itself or for other entries (due to the removal of the least recently used items). If the added item is a newly detected outlier, it is output as such. If an outlier or outliers result from the removal of the oldest item, they are not output as anomalies but the cell structure is still updated.

For each addition, there are a maximum of 49 cells which must be checked, plus an additional 49 for the subtraction (assuming no overlap between the cells). The possible changes are as follows:

- Red Cell. Adding an item to a red cell introduces no changes to the cell itself. Subtracting an item only changes the value if the subtraction reduces the number below m for the cluster.
- Pink Cell. A pink cell may be turned red if adding an item increases it's total m value beyond the threshold. Reducing the m value for the cell will have no impact on calculation for that cell (though it may impact adjacent cells).
- White Cell. Removing an entry from a white cell will have no impact. Adding an entry to a white cell may change that cell to red (or effect adjacent cells).

Because the cell checks are independent of each other, the addition and/or subtraction is an ideal candidate for recent multi-core CPU's as part of a multithreaded application.

C. Pseudo-Real Time Mode

Pseudo-Real Time Mode (PRTM) uses two windows – the overall window used in RTM as well as a smaller window which dictates batches to pre-process. Batch pre-processing works on the principle of temporal locality – that a visit to a page on Amazon.com is more likely to be followed by another visit to the same page than to Bestbuy.com.

Because of the temporal locality of hits, there are wasted operations in RTM that can be pre-processed for bundling. As an example, two additional hits on a "red" square won't change the value of that square any more than one hit did. By batching groups of hits together, only those cells that need updating are recalculated. There is a point of diminishing returns – as the smaller window size approaches N, a complete recalculation becomes necessary.

The second feature of value in PRTM is the cancellation effect. If a value is added to a cell, and that same value is subtracted from a cell, there are no updates necessary for that cell at all, nor for it's L1 neighbors (there may be distances checks for L2 neighbors which are effected). When adding or subtracting large numbers of entries, a net-effect can be calculated for each cell and that value used for application to the cell table.

The PRTM works best when there is some consistency to the web accesses. Specifically, if the distribution consists of a Gaussian pattern, the number of items removed will tend to cancel out the number added for any particular period (with more cancellations the larger the period). For more temporally skewed web accesses (for example, a large rush of checking morning scores on ESPN.com followed by a dropoff in the afternoon), the PRTM would perform worse than a straight RTM implementation.

VI. DISCUSSION

A. Overview

Running the anomaly detection in batch mode given the parameters above and the full dataset resulted in two sets of returns – one for the weighted and one for the unweighted values. The weighted values returned a greater number of outliers, while the unweighted needed the p value reduced to .99995 to return enough outliers to be useful

Both operations resulted in the return of outliers based on the principle of locality. The principle of locality provides that web accesses will be clustered closer to the point of the user for both the LDNS reason and the likelihood of the user visiting local sites more often than remote. Secondarily, usage outside of this region would be expected to be either clustered around a common, shared service (such as the aforementioned ESPN.com) or an outlier. The results below show this.

One optimization used after an initial run was the removal of 404 error results for pages not found. This removal was done to eliminate hits on sites which did not exist. Though in this example we are looking for usage patterns that find sites that were actually visited, inclusion of 404 errors might be useful for future efforts to include the detection of misconfigured software.

There are two limitations present in the existing implementation, both caused by the flattening and normalization of the geographic coordinates.

First, there is a limitation in RTM/PRTM in adding new entries – the new entries must be within the latitude/longitude range originally chosen for the cell grid. This can be overcome by dynamically expanding the grid, which would not be cost effective from a real time performance standpoint. Alternatively, a larger than needed range can be used initially which includes all of the effective land area globally. Finally, a reasonable initial range can be chosen and anything that falls outside that range automatically considered an outlier, or alternatively added to the closest existing cell.

Second, the system does not take into account the circular nature of the globe. Cell outliers at the edges are not "wrapped" to the other side of the grid for true geographic

detection. This only impacts a small number of outliers, and if this is believed to be an issue (based on the results maps for the initial data it wasn't) the edges can be treated as special cases.

B. Weighted

The weighted method identified 38 outliers out of the entire corpus. The outliers represented the usage of 15 distinct users, or just under 10% of the user population. Given the large number of individual hits, the overrepresentation of specific users probabilistically indicates a pattern to their overall usage leading to them being identified as outliers.

The cluster map for the weighted results is shown in Figure 6. The red areas surrounded by pink indicate smaller clusters, and these appear to be organized into two meta-clusters. The individual outliers are shown in the blue color and are clearly delineated from the clusters themselves.

After reviewing the findings from the weighted outlier patterns, the overall results shown in Figure 7 were obtained. The results were verified manually by reviewing the specific usage in the raw data that corresponded to the outliers as well as other usage from the same users to determine the root cause of the outliers.

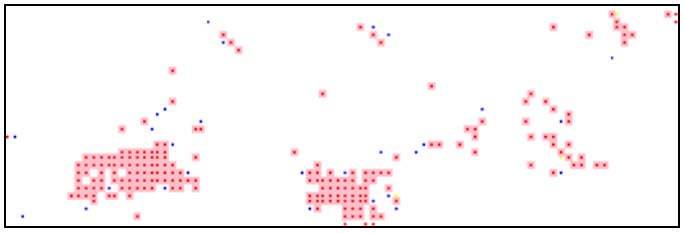

Fig 6. Weighted cluster map.

The weighted results proved to be effective at detecting foreign use – users that were located at the particular facility but were on a temporary assignment from overseas. These individuals, based on their usage patterns, tended to visit sites that were in either their native language or were local to their native country, including newspapers and local portals. These sites were expected to be located in their native countries and therefore outside of the normal, expected geographical usage. Because these users had large numbers of hits outside the norm, they tended to have multiple associated outliers for their activity. Sixty percent of the total outlier usage identified was related to foreign users.

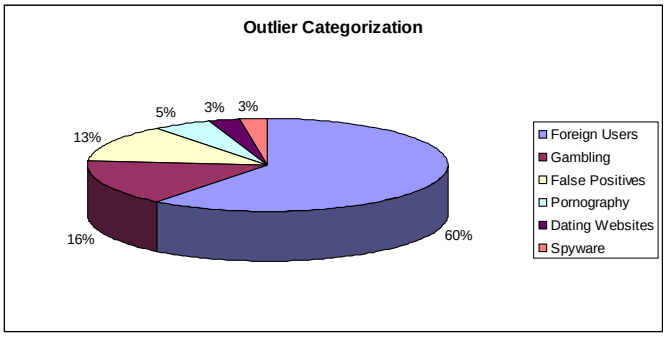

Fig 7 Weighted outlier categorization

Secondarily, sixteen percent of the usage was found to be gambling related. Given the legal problems with hosting online gambling in the United States and most European Union countries[15], gambling websites have primarily moved outside the country. Most organizations ban online gambling, so the presence of gambling sites as outliers properly identifies inappropriate usage.

False positives were the third most populous outlier type at thirteen percent. While significant, many intrusion detection systems have significantly higher false positive rates (many well above 90%). The false positives represented actual websites outside the standard usage that were not inappropriate or otherwise unauthorized. Specifically, the sites identified appear to be standard users looking for information on remote vacation locations. Geographically this is anomalous, but would be quickly dismissed by a human analyst.

Like gambling, pornography is more likely to be hosted offshore and in remote locations than other websites. Also like gambling, pornography viewing is prohibited in most organizations which makes the usage inappropriate. Pornography usage represented 5 percent of the total outlier findings.

Three percent of the total findings were identified as dating websites. This was anomalous, and is also inappropriate in many organizations, but is unusual as most dating websites are local (at least the same country) as the users accessing them. These may be considered false positives, but have been broken out here for future review.

A final three percent of the findings were found to be spyware. This is an important finding as well – anomalous spyware programs will frequently hit offshore locations and upload user information to countries with less stringent privacy laws.

C. Unweighted

The overall cluster map of results for the unweighted run is shown in Figure 8 below. The results indicate two large "metaclusters", as well as several smaller clusters of usage. The resultant map is similar to the weighted, with a few differences in outliers identified as described below.

The unweighted results were an almost exact subset of the weighted results. Both the unweighted and weighted results provided almost identical percentages in each of the categories. The unweighted results are shown in Figure 9 below.

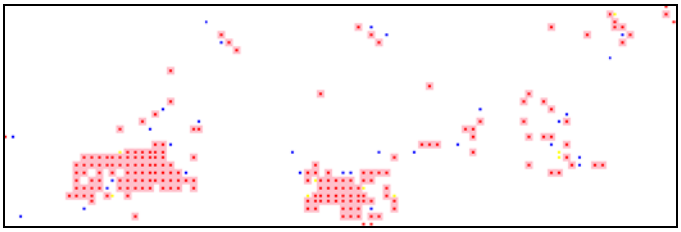

Fig 8. Unweighted cluster results.

It is expected that the weighted results would find more spyware in the case of larger incidences of infection. In the

case of these results, the organization in question filtered all spyware and viruses in two ways – at the desktop and through egress filtering in the proxy logs. Because filtered egress items were tracked separately, this differentiator was not present in the findings.

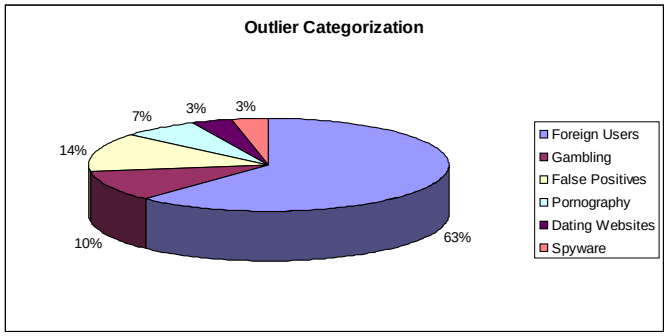

Fig 9. Unweighted outlier categorization.

VII. Future Work

A. Overview

The proxy log anomaly system provided excellent results without major tuning. Both the unweighted and weighted methods presented similar results, but more work with each could be beneficial. Specific areas of future work are live metrics, user-centered detection, and alternative method implementation.

B. Live Metrics

Though the test system used real data, it was not put through in RTM or PRTM. To fully test the live system, a high usage operation would need to be examined. In the test system, approximately 90 seconds are needed to process six million hits. Assuming an overlap factor of 10 for hits, the system still has the potential for 6500 hits per second with pre-cached DNS entries and associated IP ranges.

For IP ranges outside the cache, the time consuming effort is the IP->Latitude/Longitude mapping. This mapping took several *days* in brute force mode, but could be reduced to log(n) speed for live operation. This live mapping would require a large amount of memory to function quickly – two to four gigabytes at a minimum. Alternatively, an effective, custom swapping mechanism could be employed based on a priori cluster knowledge (high probability clusters would have their ranges placed into main memory).

To fully test live metrics, two tests could be implemented. First, the dataset used could be run through in sequential mode (with a rotation scheme). The window sizes could be varied and the output outliers examined. Second, random selections from the existing set (or just random sites) could be inserted to determine the effect on the outliers detected. Ultimately, live detection would need to be optimized to provide a quick-response mechanism similar to other, network-based intrusion detection systems.

C. User-centered Detection

The initial implementation of the anomaly detection system treats all hits as belonging to an anonymous user. One feature that may be useful in detecting malware infections and/or hijacked accounts would be user-centric anomaly detection.

Unlike general anomaly detection, user-centered detection looks at each user account as it's own corpus. As such, any activity that is not consistent geographically with a specific user would be highlighted.

Because spyware/malware site hits are more likely to be geographically outside the areas visited by a particular user, this would be more easily identified on a user-specific basis. Additionally, an account that was hijacked from a user and was being utilized by another individual may show up as a geoanomaly because of the outliers which would be expected to appear. Finally, an individual user that abruptly changes geographic usage patterns (e.g. develops a sudden interest in the geographic region where a competitor is located) would be detected by these methods.

Because an individual user has orders of magnitude fewer requests, the use of a PRTM operation does not make sense. Additionally, even with a RTM operation, it would be expected and recommended that the overall window size be as large as possible.

D. Alternative Method Implementation

The implementation used was chosen for the ability to limit updates to a specific, easily identifiable subset of entries. This makes logical sense, but may or may not be the best overall approach.

There have been numerous approaches ranging from density-based outlier detection to iterative approaches to alternative distance-based approaches in other disciplines. Each of these is expected to have it's high points and low points.

Iterative approaches would be expected to find more outliers on the weighted side, especially if weighting were used as another factor. Evaluating the number of weighted hits as part of the criteria limits the total number of points which need to evaluated (multiples are rolled up as one) but would still be expected to be slower than cell based approaches. Whereas the cell approach won't detect large usage (just small outliers), a modified iterative approach could.

Density-based approaches would be expected to perform similar to the cell approach. Essentially, though it is a distance-based algorithm, the cell approach ends up evaluating the density of individual cells as part of its outlier detection. A more traditional density-based approach may be more sensitive to loose clustering, but would be expected to have a longer update time.

Alternative distance-based approaches such as the nearest neighbor methods would also have potential for returning unique values. Specifically, the cell approach ends up quantizing data and an intracellular skew would not show up. On other distance-based approaches this might show up and alter the results.

VIII. Conclusions

The geoanomaly detection system shows promise for the detection of spatial anomalies in the form of outliers. The system was able to detect multiple instances of unusual usage that would be expected to be turned over to a human analyst in the categories of unusual foreign usage, pornography, and gambling. The overall false positive rate was also found to be very low compared with other activity-based misuse detection systems. Coupled with the ability to detect installed spyware through usage, the system is definitely viable.

In addition to the detection performance, the cell-based approach appears to have the capability to operate in real-time mode, providing analysts a way of detecting incidents as they occur. Coupled with visualization provided by Google Earth, the tool could easily be integrated with other, similar systems.

References

[1] V. Barnett and T. Lewis, *Outliers in statistical data*, 3rd ed. Chichester ; New York: Wiley & Sons, 1994.

[2] D. M. Hawkins, *Identification of outliers*. London ; New York: Chapman and Hall, 1980.

[3] C. T. Lu, D. Chen, and Y. Kou, "Algorithms for spatial outlier detection," pp. 597-600, 2003.

[4] E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets," in *Proceedings of the 24rd International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc., 1998.

[5] R. Sridhar, R. Rajeev, and S. Kyuseok, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* Dallas, Texas, United States: ACM Press, 2000.

[6] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold, "Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Data Sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 1170-1187, 2003.

[7] "IP2Location™ IP-Country Database," Accessed On:

[8] "GeoLite IP-City Database," Accessed On:

[9] IANA, "RFC 3330: Special-Use IPv4 Addresses," Network Working Group 2002.

[10] Stefan Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," *Chalmers University Tech Report*, 2000.

[11] K. Borders and A. Prakash, "Web tap: detecting covert web traffic," in *Proceedings of the 11th ACM conference on Computer and communications security* Washington DC, USA: ACM Press, 2004.

[12] A. Saidane, Y. Deswarte, and V. Nicomette, "An intrusion tolerant architecture for dynamic content internet servers," in *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security* Fairfax, VA: ACM Press, 2003.

[13] F. Valeur, G. Vigna, C. Kruegel, and E. Kirda, "An anomaly-driven reverse proxy for web applications," in *Proceedings of the 2006 ACM symposium on Applied computing* Dijon, France: ACM Press, 2006.

[14] K. Park, Z. Wang, V. Pai, and L. Peterson, "CoDNS : Masking DNS Delays via Cooperative Lookups," *Princeton University Computer Science Technical Report TR-690-04*, 2004.

[15] " Unlawful Internet Gambling Enforcement Act of 2006 " http://www.gambling-law-us.com/Federal-Laws/internet-gambling-ban.htm. Accessed On: 28 April 2007