How Implementing Five Security Controls Can Reduce Your AML/CFT Attack Surface and Help Defend Your Bank's Anti-Money Laundering Software Against Threats Related to the Posting of "Red Flag Warnings".

Executive Overview:

The financial sector continues to be a prime target for highly sophisticated threats against automated and semi-automated systems. Recently, North Korea was linked to a SWIFT system attack where over $100 million was stolen from the Bangladesh Bank. In another infiltration, an estimated $1 billion was gained from over 100 banks worldwide by the Carbanak Group. In the U.S., a Trojan named Odinaff was used against the financial industry by individuals whose work resembled that of nation-state actors. As methods used by global terrorists and money launderers are continually being defined and redefined, efforts concerning software development that help facilitate the prevention and detection of the topologies used by such organization have come to the forefront of the industry. In the infamous Bangladesh Bank heist, there is no doubt that additional funds would have been funneled through the system if not for a typo, a benign sanctions hit and an exorbitant amount of luck. In its current state, how secure is your AML/CFT software and how are your vendors dealing with advanced persistent threats against their applications? What can you as an AML manager do to ensure your solution is safe? This paper will walk you through what is at risk when an AML/CFT solution is gamed and covers five security controls that can reduce your solution's attack surface and help defend against threats related to the posting of "red flag warnings."

Kent Stern
CISSP, CAMS, CTT, MCT, MCSE+I,
MCDBA, MCSD, MCITP, MCTP

Table of Contents

Introduction

When we distill an ML/FT issue down to a single case that involves a transaction or group of transactions over time, we will still have an opportunity to have an almost unlimited number of transactions and related attributes that can or will fall under the case. When we include these issues and others, such as the separation of process, application and data responsibility, which is commonly involved in enterprise environments, we may see patterns that materialize as vulnerabilities. When these same patterns are analyzed by an adversary and gamed as a method or attack vector, it may produce an advantage over the system. This paper will at a high level cover methods specific to reducing your AML/CFT solution's attack surface that can help defend against threats related to the posting of "Red Flag Warnings." It will also topically discuss how divergent services work behind the scenes, what threats may be mounted against them and look at the risk associated with their use.

How forcefully are software development companies securing the AML/CFT detection processes employed by their software? What attack methods have they identified and documented as needing direct corrective or deterrent measures? Have they accepted the risk or have they rejected the prospect of their software being attacked and played for profit as a non-issue?

Multiple 'Recent Cyber Incidents Against Financial Institutions'

Attacks on financial institutions and the automated software that integrates with back-end processes are not as rare as one may be led to believe. Here is a statement as reported to Reuters pertaining to the Bangladesh incident,

"SWIFT is aware of a number of recent cyber incidents in which malicious insiders or external attackers have managed to submit SWIFT messages from financial institutions' back-offices, PCs or workstations connected to their local interface to the SWIFT network."

"We cannot comment on the details of any particular customer or incident, but confirm that the commonality in what we have seen is that - internal or external - attackers have successfully compromised banks' own environments and thereby obtained valid operator credentials with the authority to create, approve and submit messages from those entities' interfaces."

Understanding technologies, processes and the associated threats

To understand how these technologies and processes are used and the threats associated with them, we'll first need to understand why they are used and some history behind the issue. In 2004, the International Narcotics Control Strategy Report was produced by the Bureau of International Narcotics and Law Enforcement Affairs. It was titled "Money Laundering Methods, Trends and Typologies". In the report, Table 1 displayed the "Frequency Distribution of SAR Filings by Characterization of Suspicious Activity." It covers SAR filings from 1997 through 2003. When reading down the "Violation Type" column, one method stands out from the rest since it has zeros for the annual totals beginning in 1997 through 1999. A notation on the line states, "The violation of Computer Intrusion was added to Form TD F 90-22.47 in June 2000," hence the null values. The fact that a computer intrusion wasn't seen as a viable AML/CFT method on the form until the 2000 revision was made may be an issue, but if you look at the considerable increase in the methodology's use in the report's remaining four years, it might have been a tell that it was quickly becoming a technique of choice. If you look at the other violation types listed, you may also recognize that until recently, many of the processes were paper-based. Today, almost all have become core digital processes in their own subsystems, which have a direct relationship with the primary back-end banking solution. The ability to take a picture of a check and deposit it with a phone-based application is now the norm. These new processes and applications can, for our purposes, be seen as attack vectors when developing offensive or defensive solutions for AML/CFT application processes.

Table 1: Frequency Distribution of SAR Filings by Characterization of Suspicious Activity

April 1, 1997 through June 30, 2003

Violation Type	1997	1998	1999	2000	2001	2002	2003
BSA/Structuring/Money Laundering	35,625	47,223	60,983	90,606	108,925	154,000	72,462
Bribery/Gratuity	109	92	101	150	201	411	261
Check Fraud	13,245	13,767	16,232	19,637	26,012	32,954	16,803
Check Kiting	4,294	4,032	4,058	6,163	7,350	9,561	5,333
Commercial Loan Fraud	960	905	1,080	1,320	1348	1,879	934
Computer Intrusion[1]	0	0	0	65	419	2,484	3,605
Consumer Loan Fraud	2,048	2,183	2,548	3,432	4,143	4,435	2,271
Counterfeit Check	4,226	5,897	7,392	9,033	10,139	12,575	6,445
Counterfeit Credit/Debit Card	387	182	351	664	1,100	1,246	659
Counterfeit Instrument (Other)	294	263	320	474	769	791	615
Credit Card Fraud	5,075	4,377	4,936	6,275	8,393	12,780	6,037
Debit Card Fraud	612	565	721	1,210	1,437	3,741	4,575
Defalcation/Embezzlement	5,284	5,252	5,178	6,117	6,182	6,151	2,887
False Statement	2,200	1,970	2,376	3,051	3,232	3,685	2,316
Misuse of Position or Self-Dealing	1,532	1,640	2,064	2,186	2,325	2,763	1,564
Mortgage Loan Fraud	1,720	2,269	2,934	3,515	4,696	5,387	3,649
Mysterious Disappearance	1,765	1,855	1,854	2,225	2,179	2,330	1,264
Wire Transfer Fraud	509	593	771	972	1,527	4,747	4,317
Other	6,675	8,583	8,739	11,148	18,318	31,109	15,854
Unknown/Blank	2,317	2,691	6,961	6,971	11,908	7,704	2,290
Totals	88,877	104,339	129,599	175,214	220,603	300,733	154,141

[1]The violation of Computer Intrusion was added to Form TD F 90-22.47 in June 2000. Statistics date from this period.

Understanding AML/CFT System Requirements and the Threats Against Them

We'll start with three short narratives. Our first can be thought of as a sales pitch from an AML/CFT software vendor. So our point of departure is a topical overview as to how their software will meet the AML/CFT analysis challenge and produce information on various transactional activities. If specific criteria are met, their solution will initiate a new investigative case or raise a "Red Flag" notification. The second narrative is more specific as to how the machine learning technology may be employed by those same vendors to implement the finding of suspicious activity and raise the red flag. The third narrative is our point of arrival and looks at the issue from an adversary's perspective. It's important to note that these narratives are not rating or scoring the methods for their ability to find suspicious activity. They are only to show a sampling of the many methodologies that are being employed by software vendors to identify activity and accounts that may be suspicious and what the adversary must deal with when mounting a project against that same system. Keep in mind that in many cases, the AML manager's focus is regulation and reputation-based and is aligned with the software vendor to ensure that if audited prior to, during or after an event, the bank will show compliance and a high level of due diligence.

Narrative One - Our AML/CFT Software:
"Our Software uses configurable rules, profiles and behavioral analysis to detect suspicious activity for all segments of the customer base. Each transaction is monitored and when a questionable activity is found, a "red flag" notification is raised and it is presented as a case detailing the transactions that are suspect. As a start, rules are used to detect known money laundering and fraud scenarios. Each rule is produced to detect specific behavior based on a number of parameters. All rules and some behavioral models are configurable by the bank administration, as they will determine what is considered suspicious. Our Software also comes with many standard templates that can be configured to monitor for AML-related suspicious activity. Rules can be configured based on different attributes such as a customer's risk classification (i.e. low, medium, high, etc.), customer type (individual, business, bank, etc.) and other attributes to ensure the generated cases are relevant and require further investigation. The advanced reporting features use multiple charts and graphs to help visualize the enterprise environment and our data import-export feature allows you to merge external data for in-depth analysis which makes detecting behavior related to your bank's overall customer base easier.

Profiles are used to detect unexpected behaviors or anomalies in behavior. Our software utilizes three types of profiles to flag suspicious behavior:

- User, account and transaction-level type profiles automatically detect a variation in behavior of a customer. The profile defines the expected behavior of a customer based on past activity and compares the current period to the past. The risk class of the customer determines the tolerances allowed above the expected behavior (i.e. low: 100% above customer's average, medium: 50%, and high: 20%). Any activity that exceeds the average behavior plus the tolerance will result in a suspicious activity case.

- Country or geographic profiles determine countries that a customer is expected to transact with. Any transaction to or from a country not defined in the country profile will trigger a case.

- Peer profiles compare expected behavior of a customer to that of the customer's peers. A case is created for any customer whose behavior is statistically significant outside the normal deviation of the group behavior. Groups can be defined by one or more attributes and the tolerances set for each set of groups.

Narrative Two – Machine Learning Templates:
Machine learning (ML) is becoming the go-to solution for analyzing financial and system actions in order to detect fraud. Currently, there are numerous vendors stating that their software and implementation offers a better probability-scoring solution than their competitors. Today, one can use machine-learning templates which come complete with pre-built modules using R, Python and other languages. These are used by many vendors to help data scientists and application providers, build, test and deploy, both on-premise and cloud-based fraud detection solutions quickly. This in turn can reduce the go-to-market time for initial builds and changes. Over the years, the methodologies used by software vendors and data scientists have changed to reflect the growth in ML analysis. Some see it as a cardinality and selectivity problem, others as a binary classification problem and even topological data analysis (TDA) solutions have been implemented. If you elect to use binary classification as the solution method, you may run into a few issues. One is that because the instances of fraud within the transaction set as a whole are usually very rare, the class population can be unbalanced. Moreover, when fraudulent transactions are discovered, the offending accounts are in many cases blocked from creating future transactions to prevent further losses. Therefore, it is common for model performance to be measured using account-level metrics.

An excellent Machine Learning example that could be used by financial software vendors is the "Microsoft Cortana (Azure) Machine Learning Experiment - Online Fraud Detection"

Good Guys Toolbox for AML/CFT
As an AML manager, you probably have a thorough understanding as to the rules that will or won't generate an STR red flag, but how privy are you to the underlying behavioral and machine-learning aspects of the software?

If you are not an expert in R or Python, or any of the languages being used deep within the solution, or if it is cloud or service-based, it may be out of your control. These software-based processes are considered to be a major tool in the "Good Guys Toolbox for AML/CFT."

So what is the output from the analysis? What does a "Red Flag Warning" look like? Just to make sure we are all on the same page, here is a sample of activities and typologies that were deemed suspicious in a FINTRAC Typologies and Trends Report:

Sampling of Case "Red Flags"

- Customer used multiple names/identities, in conjunction with providing multiple addresses, making it difficult to ascertain the true identity of the customer.
- Subjects used various spellings of their names and/or misled reporting entities regarding their names.
- The frequency of the customer's visits was excessive, and also involved the use of a wide range of MSB agent locations.
- The purpose of the transactions, and the relationship between the beneficiary and the ordering clients of the wires, does not appear to make business sense.
- The customer made frequent visits to conduct currency exchanges, sometimes two or three times in a given week, and sometimes in the same day.
- The individual converted small denominations of U.S. cash into larger denominations of Canadian cash.

In a nutshell, you do your homework, run it through management, make it fit into the budget, interview a few vendors, see what your peers are using, do a trial and then buy into a solution that will, with the help of your analysts, find the bad guys and their transactions.

Now, for our point of arrival, let's look at a case study of the same issue from the other side of the fence. Just as before, the case study narrative is not to be rated as the method or process which produces a final solution in the form of success or failure, or even one that gives us a calculation of the same. It is only to show one of many methods in the form of a simplified solicitation that could be initiated by groups of individuals, state or non-state actors to investigate activity on accounts that may cause certain behaviors of interest.

Note: It is important to understand that the ability to obtain a set of rules or situations and "game" that same set to a global audience of mathematicians, code writers and game analysts is an option when testing solutions for offensive actions against a target.

Narrative Three – Solicitations for Gaming:

For some, the norm for peer review is not always executed with the intent of finding the ways a rule, process or calculation will hold true when compared to solution norms, but why it will not. The debate as to whether it is more advantageous to look for why something is true verses false as a method for coming to an answer is beyond this solicitation, so for our purposes, either method can be employed at your leisure. Let's start with an example of the work to be reviewed. We know that the algorithms developed by many AML software providers have different iterations, typologies and scenarios that can be applied to the following:

- Sequence matching
- Outlier detection
- Behavior analysis
- Topological Data Analysis (TDA)
- Link analysis
- Rule matching
- Machine Learning Analysis
- Action Analysis

So let's look at both the Currency Transaction Report (CTR) and the Suspicious Transaction Report (STR). Both are well-known BSA "rules." Here is a small portion of the common CTR rule:

"A bank must electronically file a Currency Transaction Report (CTR) for each transaction in currency (deposit, withdrawal, exchange, or other payment or transfer) of more than $10,000 by, through or to the bank. Certain types of currency transactions need not be reported, such as those involving "exempt persons", a group which can include retail or commercial customers meeting specific criteria for exemption."

When this rule is converted to a calculation and then implemented on the back-end AML/CFT software, the main intent of the rule and work of the software and calculation can easily be met with a simple account, time and amount rule. "One account to many currency transactions over time rule." But this as well as most other rules come with conditions. We must add the "Aggregation of Currency Transactions" portion, the "Businesses are Independent" variable and then of course we need to add a little drama with the "Types of currency transactions subject to reporting requirements, individually or by aggregation" rule. But wait, what about the "exempt persons" portion of the rule? Yes, there are usually governmental loopholes in place for rules that will circumvent a system, which is also always a potential solution parameter or method. This is where we have a problem. So let's go back and look at the gaming side of this issue. Together, we could call all of these CTR rules the "public input rules" because they are known by everyone and are straightforward when implemented as a whole. When we look at an STR solution it's a little different, it's more complex. The STR profile has more flexibility as to what profiles and parameters are used, how they are used and how they will generate the STR. So to that point and depending on how you decide to mount the problem, we might want to change the name that we use for all of these STR rules from the "Public input rules" to the "Random IV" (randomly generated initialization vector). This may make more sense from the perspective of the person solving the problem with zero knowledge of the actual rules applied. Remember, the calculation is not at the application level where you would be audited. In many audits, one or more transactions are placed into the AML/CFT system at the application layer (at the front end) and then we see them flow through the system. Based on the input transactions and the STR rule calculation(s), it may or may not generate an "output" which is the STR "Red Flag" alert. For gaming, the output can be seen in two ways, as the Secret key or as a Random IV. With STR's, we should not know the exact input or the perspective of the trained model and more importantly, we should not know if any output is or was generated. So overall, an STR solution will have semi-unknown bank transactions, semi-unknown random "public input rules" that process on those transactions and a totally unknown output (zero knowledge Red Flag output). So once again, how will we attack the problem and what is it that we are actually gaming?

Two simple (and extreme) examples with tangible return values come to mind.

1. Based on some known "public input rules" and some known input transactions over time and location, can we give a probability score for transactions that **will and will not** produce "Red Flag" output (and maybe a future STR)?
    a. A person receives bi-weekly $2,531.27 direct deposits. The person has a high credit score, has worked at the same government job for 10 years, has the same house, has a mortgage with your bank, has a consistent balance and has been a customer for 15 years.
    b. A person from Venezuela opens a new account and immediately makes an $8,000 deposit followed by daily incoming wire transfers of $7,500. They also make daily outbound wire transfers of $7,000 to a U.S. dollar account in Panama. The person is in the U.S. on a visitor visa with the cosigner being a U.S. citizen with poor credit and a low monthly balance.

When we look at how to answer these questions, we'll need to look at the business and processes as an enterprise with a holistic view that includes all areas which the AML solution touches and how the exemptions are and can be used. That leads us to the backbone of the solution, the AML/CFT professionals, analysts and the software they maintain. Most AML/CFT professionals work at the application level and the remaining layers of the OSI model are outsourced, delegated to general IT or left to the company or department that writes and maintains the AML software. So just as there is an "exempt persons" CTR rule, there are also loopholes and methods in most AML/CFT software that may give us a door into the "zero knowledge Red Flag output" portion

of the equation. An example would be that many vendors have front-end, application-level interfaces that allow privileged end users to manually input changes to the lower level calculations based on their bank's needs. A working example would be when a bank's AML team assembles annually to review and if needed, update the bank's monitoring thresholds to mitigate risks associated with their current "Red Flag" STR reporting solution. In other words, they can manually change the calculation's "Public input rule" or "Random IV", which once again has a different meaning depending on how you choose to mount and solve the issue at hand and solicitation as a whole.

If under review an insider was to supply, or one was able to ascertain by packet capture or other method any of these "input rules or "IV's," you could attack and game the calculation as you would an encryption problem. From there, it may be possible to produce a method capable of providing quality geography-time-amount-NAICS-TransCodeType-based bank or client-level probability and confidence scores for a transaction or group of transactions. Remember this "Gamed" peer review is not conducted to ascertain if the calculation will raise a red flag; a red flag will be raised if the calculation is run and the requirements are met. It is gamed to determine what are the methods or more importantly what behaviors will tune the "IV's" to make true or false, in order to circumvent, mitigate or initiate the raising of the red flag long term, and that is the work of this gaming solicitation.

Defending Against an Unseen Threat

So after reading all three narratives, it may give rise to the question of not how does my software find and report suspicious AML/CFT activity, but how does my AML/CFT software defend against a global threat audience. We'll start with a little background on gaming solutions and then dive into some of the methods needed to mitigate the threat risk that may be used to game the system for the benefit of others.

When a gaming solicitation is read, numerous valid solution methods and vectors will inevitably come to mind and be raised as being the best way to solve the problem. For brevity and without going into a topic that is best saved for another day, how would one rate answers such as "the best solution to the problem" or "the least problematic method with the highest reward?" In an enterprise solution, there may be thousands of branches, franchises and access points around the globe. Some accounts may be accessed from cash exclusive countries or foreign financial institutions that use PTAs, money brokers and other covered financial businesses who provide their account holders with access to your banking system or pass through to another system. With so many entities touching so many other entities over time, there can be an almost unlimited quantity of methods that one could propose and subsequently use to meet the gaming requirements listed in the case study. Each one of those methods involve processes that have attributes associated with them. A solution metric could be defined as the "probability of mission success" and time, action and/or other variables could influence the score of the metric to the point that when all factors are weighed, it falls below or above other solutions and is dropped or becomes a finalist for actual implementation.

There are many types of mission process variables and categories that are evaluated; here are a few of the most basic from the Action Category:

- **Action Accuracy:**
  Extent to which actions executed are directed to the intended purpose.
- **Action Completeness:**
  Extent to which actions executed encompass the full scope of the task.
- **Action Consistency:**
  Extent to which actions executed are consistent with actions in an earlier time frame.
- **Action Correctness:**
  Extent to which actions are executed without error.
- **Action Appropriateness**:
  Extent to which actions executed are the appropriate ones to achieve the intended purpose.
- **Action Efficiency:**
  Extent to which actions executed are efficient in the use of resources.
- **Action Synchronization:**
  Purposeful arrangement of actions in time, space and function.

As you can see, when compared to a more thought-out solution prototype that takes into account all of the many categories and variables associated with well-defined projects, some of the standard AML/CFT methods that today's software are designed to catch may not stand up against peer or simulation review.

Hardening Your Solution:
What processes must we secure or implement to defend against an adversary that can use an almost unlimited amount of data, which can be fed into a machine-learning model, be implemented from almost anywhere in the world and is using a distributed system that is entirely digital? Remember, the software vendor is advertising that their software catches more ML/FT activity and will stand up to an audit better than their competitor. They are not advertising how their solution implements the three pillars of information security. In most cases, that is still left to the OS, DB or in general, the downstream layers.

The latest SWIFT system attacks and others that are similar have one thing in common; they elevate privileges, they move money and then they must stop before the model kicks in or the model kicks in, sends a red flag and they must stop. So, based on historically successful attack methods and contrary to popular belief, your adversary's threat analysis and the methods that may produce the highest probability of **"long-term"** mission success may not be one that tries to directly outsmart the AML/CFT model or calculation. Transactional or behavioral attacks using elevated privileges alone while still missing one of the unknown variables will certainly be caught. The hit-and-run attack method is not a solution consistent with a long-term presence. For a long-term adversarial implementation, one will need a method of "owning" the software or related processes such that it gives them an insight as to the unknown or Random IV portions of the calculation. The method may use any number of vectors from distant relationships such as private keys, service accounts and/or network traffic.

In the Bangladesh SWIFT heist, the following was reported by Reuters:
"The malware used in the Bangladesh Bank hack was cunning, in that it allowed attackers to steal money by surreptitiously altering the Oracle database used by SWIFT's software, and then send the appropriate SWIFT messages to other institutions to facilitate money transfers, according to BAE Systems. The malware also ensured that related messages that would normally have been sent to a printer at the initiating bank to create a paper trail were suppressed to help hide attackers' tracks." "The malware is designed to hide the traces of fraudulent payments from customers' local database applications and can only be installed on users' local systems by attackers that have successfully identified and exploited weaknesses in their local security," the SWIFT spokesman says.

Stating that the malware was "cunning" may be a little theatrical, but yes, disabling a printer during a holiday will postpone the notification and money will continue to flow out of the system but as stated before, it is still following a basic "hit-and-run" methodology.

To that end, even an insider or the reporting software that holds case data can be an attack vector. Because if you are able to obtain a sample of transaction data from any pending or historical cases that produced a red flag, attacking the calculation would be a downstream follow-on step which would then have a much higher probability of long-term mission success.

Since many AML managers and analysts don't have any back-end responsibilities for their AML/CFT software other than trusting in the fact that the company you purchased it from and your IT department do everything possible to implement a viable "Defense in depth" posture, we won't cover all of the physical, technical and administrative controls that should be in place. We'll review a few high-level items that can be inspected and audited by the AML manager. These stand out as "must implement" risk reduction tasks, as they tend to be a sticking point in our infrastructure audits and are always on the top of our penetration methods list.

Reducing your AML/CFT Solution Attack Surface
What is your AML/CFT Solution Attack Surface? From a risk-based approach, it is the sum of the application's security risk exposure. It is the aggregate of all known, unknown and potential vulnerabilities and controls across all software, hardware, firmware and networks. A smaller attack surface can help reduce vulnerabilities and make your AML/CFT solution less exploitable, thus, reducing risk.

In real-world terms relating to an AML/CFT software implementation, this means hardening the operating system, network, application and related services by disabling functionality, services and access that is not required while maintaining the minimum functionality that is required. In many larger financial institutions, this work has been completed and full-time teams of IT security employees watch over their flocks. But remember, larger banks may have downstream banks in cash-based countries and some countries like Bangladesh (and its neighbors) don't have the same "data security" as those in the EU and U.S.

Implementation Change 1: Implement Security Zones
Implement a security zone to ensure the separation and restriction of users and data.

One simple metric that is commonly used in financial and web account security is to know the IP address, MAC, country and provider of the account holder when the online account is created and additionally for successful and unsuccessful logons. This same methodology must be implemented and exceptionally enhanced in your AML/CFT installation. For most environments, there's usually logging and restrictions for clients that access your software (domain access, laptop access, software logon and IP restrictions). But I have found that in many cases, there are no hard restrictions for traffic and access to and from the network, VM and software other than simple VLAN separation, IP and account restrictions and in some cases, allowable remote VPN access.

As a standard and at a minimum, a security zone should be defined to isolate the entire AML/CFT server (or if VM, parent) and any part of the solution, to restrict all authorized network traffic by time, port, protocol, IP, MAC and custom ID Key. In instances where "a case of high value" has been created, there should be no shared connection medium or at least a movement from direct or VPN to secure IPSec direct or secure IPSec direct with terminal server connections. All traffic connecting to the subnet that the AML/CFT server lives should be restricted whether the traffic's origination or destination is local or remote, client-based or cloud-based. In many cloud-based solutions, this is the default but some may only have restrictions based on IP across the switch or VLAN.

You may say "we do that" and I agree most EU and U.S. banks do, but in many cases, not all downstream objects are covered. It is very important that all clients be under the same restrictions. Clients (computers or any objects) that access the application receive red flag warnings or use case management software that contains the data or similar data in any format should be isolated from other clients. No other client in the environment should be able to connect to map drives (to or from) or remote onto any client that connects to or from the AML/CFT software application. Additionally, if the solution is cloud-based, all back-end Admin, SA and access accounts from the service company should be known and the ability to receive immediate notifications should be in place when they log on interactively, access any portion of the production application or access the database environment. Cloud solutions are a standard, but we find that companies are not vetting the service providers who have access to your database to the same level as their own.

If available, IPS/IDS solutions must be implemented to augment zone security. There are many network analysis packages available to audit how, which and from where clients access your AML/CFT solution. One such package is SolarWinds. If you do a two to three week capture with the subject being the server that runs your software, you should have a good list of computers and users that have been authorized to access the physical network. You may find others and if you allow remote access, it can get quite entertaining as you may see addresses you weren't expecting.

Implementation Change 2: Understand "Privilege-Attached" Accounts
Audit and Secure Critical "Privilege-Attached" Service Accounts

One of the primary tasks of a threat that has infiltrated your network is to initiate the escalation of privileges. Gaining more access and expanding their reach in order to complete a project task is in many cases, a step associated with reaching a mission objective. It is also an important project task associated with obtaining system ownership. Many different mechanisms can be used to achieve an escalation of privileges, though primarily they involve compromising existing accounts that already have elevated privileges. This is opposed to the more difficult task of setting or granting privileges and/or rights on an object to complete work, since changing and granting permissions is generally a tricky proposition in an IPS/IDS monitored environment.

When I ask students “Generally, which accounts do you think attackers go after”, they usually answer administrator accounts or accounts of people that have rights over the system, like the AML manager. Yes, they are good accounts and they are generally the accounts that generate notoriety when compromised, but in a system that uses service and service-type accounts, it can be said that, “Service accounts are gold.” In a Microsoft system, these are called "Privilege-Attached" Active Directory Accounts.

"Privilege-Attached" Active Directory accounts or “Service Accounts” are domain accounts that have not been made members of any of the groups that have the highest levels of privilege in Active Directory, but have instead been granted high levels of privilege on many servers and work stations in the environment. For our uses, they are most often local or domain-based accounts that are configured to run services on servers, typically for applications running on large sections of the infrastructure (SQL Server, SSIS, SSAS, DNS, DHCP etc.). Although these accounts may have no privileges in Active Directory, if they are granted high privilege on large numbers of systems, they can be used to compromise or even destroy large segments of the infrastructure, achieving the same effect as compromise of a privileged Active Directory account. For example, an ETL application extracts, transforms and loads data. In order to do its work, it must have been granted privileges such as the ability to read data in the source and write data to the destination. A simple solution may be to pull data from one system such as a transactional input system, then transform it by applying business logic, calculations or data type changes to conform the data and as an output (if there is one), load it into another table or send a notification. If you were to obtain the service account for that particular instance, you will have obtained control over the processes, calculations and data involved directly with that account.

Most major database providers have some implementation of the “Service Account” model, so no matter which solution you implement, the need to secure these accounts is paramount. If your AML/CFT implementation uses a Microsoft database, it is no different. It will need a service account to run the SQL Server Service, SSIS, SSAS, DNS, DHCP and many other related services. The issue is not that the service account security model is inherently insecure. It’s that in many cases, the service account password is not changed for long periods. In other cases, the same service account is used for multiple services, or in some cases, multiple services on multiple servers. Implementing a password management system and interactive versus non-interactive accounts is a vital step to overcoming this security issue. I have seen managers trying to manually reset service account passwords every 90 days on large server implementations and I can tell you, it is not feasible long term. Microsoft’s best practices for enforcing password policies states - Where security is a concern, good values are 30, 60 or 90 days. Where security is less important, good values are 120, 150 or 180 days.

Implementation Change 3: Implement a “Policy of Least Privilege”

A “policy of least privilege” must be implemented for all AML/CFT software, work stations and devices that connect to the software.

This may seem obvious, but it’s still a major security issue and threat method as it was years ago. So what can the AML Manager do to reduce their solution’s risk to this known problem? There are two major issues that you as the solution owner should be concerned with and they are at the application and work station level.

Application Least Privilege

When an audit is conducted, they usually check application privileges inside the application. That is, when a user logs in what can they see and/or do once access has been gained into the software solution. The issue is not there, it's what access to resources and subsystems can that same user gain from outside that application. For example, once a person has logged into the application, they may want to run a report or work with an alert that has been generated. If they have been granted privileges to that area, all is well. But what other related systems does the application "touch" when running the report or querying for the latest alerts? Does the application natively generate reports or does it use another subsystem such as Microsoft SSRS, SAP Crystal, IBM Cognos or others to generate the reports? If it does, can that same user log on directly to that reporting system and access the data from there? If so, are you auditing and securing that system to the same level as your AML/CFT solution or are you denying them the rights to access it directly? The same can be said about the data storage method. Can you access the databases or database management system from outside the application using common SQL query and code tools like Excel or Notepad++, or has the application implemented "Application Role" security? Implementing least privilege on related systems and auditing for least privilege in this case would ensure a reduction in risk due to a reduced attack surface area.

Workstation Least Privilege

The paragraph that started this section was alluding to the fact that the local line of business end users (the AML analyst sitting behind the computer) should not have administrative rights over their computer. That statement in itself is enough to fill a library with noteworthy papers on domain and data security, but what we're interested in is the following:

When you implement security zones to ensure the separation and restriction of users and data and you ensure your service account infrastructure is safe, there are only a handful of accounts and objects that can make changes to your AML/CFT solution as a whole. At this point, we're not looking at the ability to make changes to the AML/CFT software or any of the related subsystems. We're at the computer level and with local administrator rights, we're allowing the end user to make operating system configuration changes, registry changes and/or install application software on her or his own computer, which has access to the AML/CFT solution. This alone is not a problem since most large infrastructures use image-based software installations and they have group policy or restrictions on the software that can be installed. The problem is based on the probability and history associated with end user computers being the target of attack via malware, ransomware, spyware and almost every other method available. The threat only needs to be invoked or run to be effective in its mission. If the computer is situated in a secure zone, yet still has Internet access and the local user has the right to make changes to their OS, that may be the threat vector of choice and may produce the highest probability of mission success. An example would be a spam/phishing campaign that could include a compromised attachment, which when clicked could result in the running of malicious code. But wait, they should get the User Account Control (UAC) pop-up, they click no and all is well, right? Sorry, I wish it were that easy for a million different reasons. Sometimes it's turned off and sometimes the end user clicks "yes." Maybe the malware starts at the target's phone, where the same password is used for a "placed" phone app and their domain account… and on and on… So what can the AML manager do to resolve this? The first question is "can you (as the AML manager) and members of your team connect to the internet from your company computers?" Would you be ready to give it up entirely? Talk to your IT department and see how they can reduce or restrict rights for end users that access your AML/CFT software. They should also restrict end users from being able to make OS changes and disable downloading from untrusted or remote sites. There is always a happy medium, which aligns with the allowable risk.

Implementation Change 4: Audit and Manage the Security of .dll's

Understand the use and security of .dll's and all versions of native access client software.

Three issues fall under this category. In layman terms, there is a bridge between the end user software (like Excel) and the back-end database (like Oracle). In this context, the native access client is a class library (DLL) that allows you to connect remotely to the underlying database of an application. The first issue is that over the years, many malware-laden versions of these drivers have been downloaded and put into production. If you do a quick internet search, you will still find a host of download sites offering compromised clients. Unless there is

a very good reason, only download and/or use native client software for database connectivity that has been downloaded from the respective vendor's site and has been vetted by your IT security team. If you install a new application that connects to your back-end database and you need a new "driver" because the old one was 32 bit and the new one is 64 bit or some other reason, proceed with caution.

Secondly, you should have the latest version of each specific driver for each database that you connect to. For example, currently, a Microsoft environment will only support the SQL Server 2005 version of SNAC or later. If an earlier client that does not understand encrypted authentication tries to connect, by default the SQL Server will allow the connection, but the username and password will be transmitted in plain text. You can use SQL Server Configuration Manager to disallow unencrypted authentication from down-level clients by setting Force Encryption to yes, but then you would need to upgrade the clients with older software before they can connect to the database. With that being said, there are still many companies running older versions of software where this is an issue.

The third issues relates to both native drivers and ODBC or (Open Database Connectivity). This issue is a combination of the prior two items. ODBC is a standard application programming interface (API) for accessing database management systems (databases). Since most all major database providers offer an ODBC solution, it has the opportunity to be a high-impact attack vector.

ODBC Specific Issues Identified:

- Using older versions of ODBC that does not support high encryption or store login names and passwords securely when coupled with the client software.
- Using an Administrative account with multiple clients for the ODBC connections to the database.
- Hardcoding a single user account in multiple clients for ODBC connections to the database.
- Misconfiguring the ODBC software applet, such as not selecting the "Use encryption" checkbox.
- Downloading ODBC drivers from shareware or non-approved sites.

Example: If your AML/CFT software uses a Microsoft database back-end and application roles are used for end-user permissions within the application, an ODBC issue that needs to be addressed within the environment would be the following:

"If calls to the sp_setapprole stored procedure will be made across a network, you must ensure that the connection is encrypted—for example, using SSL or IPSec—to avoid exposing the application role password, because the password is effectively sent in plain text. sp_setapprole accepts an optional @encrypt = 'odbc' parameter, but this uses an ODBC function that obfuscates the password rather than truly encrypting it. The ODBC encrypt function is not supported by the SqlClient library."

Old versions and configuration issues aside, another issue with database connectivity clients and ODBC is that there may be quite a few versions floating around on network shares, servers and clients. In many cases, each version of each vendor's RDBMS loads and uses its own set of drivers (SQL 2012, 2016, Oracle, etc.) and within those, are again differences between 32 and 64 bit versions. So as time goes on and new RDBMS's and clients are loaded, there are opportunities to download or shift a "placed" version into production (Excel) or into a corporate share where the company approved software is stored.

What can you as an AML Manager do to address this issue?

First, no local admin rights. Second, use only tested and approved image-based clients. Third, in some environments, hash values are managed for all system files. This ensures no system files are changed and/or modified and then moved onto a production machine. At a minimum, the servers that run the AML/CFT software and all of the clients that access the software should be "hash-managed." If this is out of the realm of what your IT department can handle, a simple DeNISTing audit against the native database drivers and ODBC files on those same severs and clients should be completed. If the files are not on the NIST list, a checksum for the original files should be obtained from the vendor.

Implementation Change 5: Consolidate your AML/CFT Assets

Implement a Secure - Enterprise Class Case Management System

Implementing a case management system (CMS) reduces the AML/CFT's security footprint and enables you to audit all actions within the AML/CFT process. This is vital to a secure solution. In the past, I've had the opportunity to watch as crime analysts and case managers open a new copy of Excel for each new incident or case. If the case grew larger or had data that exceeded Excel's row limit, they would use Microsoft Access to hold the data. They would keep everything nice and tidy by mapping a drive to a shared folder and storing the data on the network or in some cases, a separate folder on their computer. It was very simple and cost effective, alas not the best solution for high value data that may be part of an adversary's gaming solution. In other cases, I've seen departments purchase group software that used shareware databases that had not gone through any security evaluation. When I ask to see the documentation for the protection profiles, blank stares abound. Let's see how this may look if you were to conduct a basic threat vector analysis. Generally, most investigative cases have three areas that can be used as a starting point for threat vector analysis.

Threat Vectors in the Case-Building Process Where "Red Flag"-Related Data May Live.

1. **The system used to store the case notification and the initial information related to case.**
   Depending on how you work new cases, this could be a dashboard in the AML/CFT software, reporting solution or even a workbook where you keep a list of "working" cases for processing and reporting. In law enforcement, it may be the first report filed by the officer. This would live in the case management system (CMS) data repository.

2. **The data that is used to investigate the issue.**
   This is working data, not the final result of the investigation and analysis. This may include data scraped from the Internet, 314(a) request, wire transfer analysis, OFAC check information, link analysis data and visualizations, background checks on account and individuals related to the issue, investigative documents and transcripts, recordings and more. This data may be stored in disparate secure systems or as workbooks, and/or documents and images that are stored on the network. They will all need to be securely stored and may be used as evidence.

3. **The final output of the analysis and case.**
   This may be similar to the investigation data, but will be in its final form with the who, what, where, when, why and how it all happened spelled out in its entirety. Where this lives may vary.

The first question that comes to mind is "Where is this data stored and does any of it contain the elusive 'Random IV' that is needed to solve for our gamed calculation?" My past experience has shown that in many cases, over time, the data gets out and is stored outside of a tagged, secure environment. So a disparate case analysis environment should be avoided. To the extent possible, you must use a secure-enterprise class case management system to effectively manage all aspects of all cases in a single environment. Note: Include all areas where the data may be held or sent. Don't forget about case data that may be sent to attorneys or disclosed to other official departments.

First and foremost, enterprise case management solutions must have the ability to natively secure, manage and audit all objects and assets of a case at rest and in-flight. This environment should be treated exactly the same as the production AML/CFT software as it has much of the same data that can be used to solve the gaming puzzle.

Just as in the AML/CFT solution, you must secure all related environments that the software touches:

- Secure the physical environment
- Secure the server environments
- Implement network auditing and security
- Application updates and patching

- Minimize service surface attack area
- Standardized authentication methods
- Secure access to database resources
- Encrypting data and data paths
- Secure coding practices
- Secure and encrypted audit log consolidation and transfers for databases and related applications

To that point, what should the AML manager look for in a case management solution when trying to prevent or mitigate the effects of threats to the system?

Mandatory Aspects of a Secure CMS Solution

- **The Case Management System must able to scale to a large number of assets and objects.**
  Enterprise cases contain large amounts of investigative information such as case detail, images, entity attributes, events, evidence information, tasks, incidents and contacts. The backend database must be scalable and allow for the networking of users on multiple device types. It must support multiple languages on the front-end application and back-end storage. It must also allow for image and text storage along with advanced search capabilities for those objects. Oracle, IBM and Microsoft are common database vendors that have stood the test of time. Additional features should include workload management, report generation (case reports and advanced metrics), system administration and again, multilingual support.

- **All work and communications completed within the system must be securely logged and encrypted at rest and in flight.**
  The ability to audit the information in real or semi-real time is necessary. As many systems now use log consolidators and log analysis software, the transfer of log data across networks and systems must be encrypted with the ability to securely audit the process of log consolidation and reconciliation. This is extremely important because when challenged in court, the statement “she sent an email and it had the AML software calculations” may not hold up as a valid statement or even an actual action when pounced upon by a team of data experts who have been employed by a savvy defense team. Without a well-defined and clearly documented log solution in place for all data actions relating to the fact that an action has occurred and why and how it could not have been accomplished by another with administrative rights over the system must be clearly set out. Once again, if given a chance to testify in court, an expert in this area will challenge the validity of what actually happened.

- **The system employs a proven high availability and disaster recovery solution.**
  Losing months of hard work or all work in a system can be devastating to a case and the organization’s credibility as a whole. This issue can be seen as a vulnerability or weakness that makes the enterprise solution susceptible to numerous attack methods. Everything from the loss of a power panel in the server room to a corrupt memory chip in a server should be addressed to combat HA and DR attack methods. High availability is not always if the system is up or down; it is also the availability of a service to perform work at a certain level of performance. A 99.9 percent uptime SLA would be a minimum for an enterprise system.

A Note on AML/CFT Software Evaluation, Assurance and Risk Assessments

In the context of AML/CFT and CMS application security related to threats, all processes, operating systems, databases and related touched systems as a whole should have completed a risk and threat evaluation. Each software vendor has their own view as to what secure means and it is up to them to implement application security as a whole. Before you buy into a case management solution, you as the “Authorizing Official” should have the evaluation results of a “Risk Assessment Process for Internal and External Threats.” At a minimum, this process should define each application process that presents a threat risk, which in turn will define your applications’ targets of evaluation (TOE) where standardized “Protection Profiles” would be applied and investigated. Outside of the software, the same threat and risk framework must be applied to all aspects of the infrastructure, both upstream and downstream, which your case management software interacts. This includes interaction your software may have with vendors and their vendors.

NIST SP 800-37 Rev 1 states the mitigation of the risk issue as such:

“Ultimately, the responsibility for adequately mitigating unacceptable risks arising from the use of external information system services remains with the authorizing official. Organizations require that an appropriate chain of trust be established with external service providers when dealing with the many issues associated with information system security. A chain of trust requires that the organization establish and retain a level of confidence that each participating service provider in the potentially complex consumer-provider relationship provides adequate protection for the services rendered to the organization. External service providers may also in turn outsource the services to other external entities, making the chain of trust even more complicated and difficult to manage. Depending on the nature of the service, it may simply be unwise for the organization to place significant trust in the provider—not due to any inherent untrustworthiness on the provider's part, but due to the intrinsic level of risk in the service. Where a sufficient level of trust cannot be established in the external services and/or service providers, the organization: (i) employs compensating controls; (ii) accepts a greater degree of risk; or (iii) does not obtain the service (i.e., performs missions or business operations with reduced levels of functionality or possibly no functionality at all).”

Conclusion

To properly secure an AML/CFT solution, we must continue on an approach that includes a “holistic view of all areas the solution touches.” For our definition of “evaluated secure workspace”, this includes any environment the transaction “action” touches, from kernel and memory space through security API, Internet router and related cloud systems. Evaluations are mandatory for many installations and as such we can ask, “As a standard, does the independent or internal AML auditor evaluate for specific external threats to your software at that level?” Probably not. These are all attack vectors that can be used to manipulate, circumvent or skew the calculations, models and processes in order to game the solution and produce a high method confidence score. This in turn may allow the insider or adversary to yield a higher probability of success when laundering or extracting funds or when selling the method for a fee or percentage of the profit. As the industry continues to harden its landscape, it must still deal with both sides of this equation. Low bar rules like “exempt persons” will continue to give threat agents opportunities on the political business side and application layer audits will continue to give data and packet agents access to vulnerabilities on the bank-business side. With standardized case management solutions and advances in behavior models, many of these past incident issues are now being overcome. The threat and evaluation solutions that are being produced are quickly evolving to take on the next generation of AML/CFT threat agents and they are having an impact. But as always, we only need to watch the headlines to see how successful they are.

About the Author

Kent Stern is the Director and Lead Data Mining Security Architect for CodeCenters International. For over 30 years, his work has focused on data architecture, pre and post incident/attack reconciliation and data auditing. He teaches data mining architecture, MDX and R, along with data security courses and NIST SP 800-xx compliance for governments and the global 500. His work has taken him to every corner of the globe and into niche areas ranging from anti-money laundering to ship-based container tracking. He lectures on how to mount and/or defend legal cases which deal in data security and specializes in countering an opposing “expert’s” testimony. He has been a Microsoft Certified Trainer (MCT) since 1999 and along with his many other credentials, he holds the ACAMS Certified Anti-Money Laundering Specialists and ISC2 CISSP certifications.

He can be contacted Kent@CodeCenters.com

Notations:

Bangladesh Bank SWIFT System Theft and FBI probe into North Korea

http://www.npr.org/sections/thetwo-way/2016/05/27/479760450/north-korea-linked-to-81-million-bangladesh-bank-heist

https://en.wikipedia.org/wiki/Bangladesh_Bank_robbery

http://www.reuters.com/investigates/special-report/cyber-heist-federal/

https://www.nytimes.com/2016/05/27/business/dealbook/north-korea-linked-to-digital-thefts-from-global-banks.html

Symantec October 2016 Odinaff and Carbanak threats https://www.symantec.com/connect/blogs/odinaff-new-trojan-used-high-level-financial-attacks

Odinaff NJ Cybersecurity & Communications Integration Cell
https://www.cyber.nj.gov/threat-profiles/trojan-variants/odinaff

March 2004 Money Laundering Methods, Trends and Typologies - International Narcotics Control Strategy Report
Bureau of International Narcotics and Law Enforcement Affairs
https://www.state.gov/j/inl/rls/nrcrpt/2003/vol2/html/29910.htm

Online Fraud Detection: Step 1 of 5: Generate tagged data
https://gallery.cortanaintelligence.com/Experiment/Online-Fraud-Detection-Step-1-of-5-Generate-tagged-data-2

2012 Oracle Best Practices for Anti Money Laundering (AML) System - Selection and Implementation
http://www.oracle.com/us/industries/financial-services/062008.pdf

FINTRAC Typologies and Trends Reports–July 2010
Money Laundering and Terrorist Financing (ML/TF) Typologies and Trends for Canadian Money Services Businesses (MSBs)
ISBN: 978-1-100-16310-9
http://www.fintrac-canafe.gc.ca/publications/typologies/2010-07-eng.pdf

2002 SANS An Overview of Threat and Risk Assessment James Bayne
https://www.sans.org/reading-room/whitepapers/auditing/overview-threat-risk-assessment-76

2017 Defense Acquisition University Object Action and System Engineering
https://www.dau.mil/tools/dag/Pages/DAG-Page-Viewer.aspx?source=https://www.dau.mil/guidebooks/Shared%20Documents%20HTML/Chapter%203%20Systems%20Engineering.aspx

24 October 2014 Joint Publication 3-26 Counterterrorism Actions
http://www.dtic.mil/doctrine/new_pubs/jp3_26.pdf

13 July 2006 Joint Publication 3-13.4 Detection of Adversary Deception
https://fas.org/irp/doddir/dod/jp3_13_4.pdf

Cybersecurity and Software Assurance Minitrack 5-8 Jan. 2016
ISBN: 978-0-7695-5670-3

http://www.cert.org/cybersecurity-engineering/research/cybersecurity-and-sw-assurance-measurement-and-analysis.cfm?

2006 Microsoft Corporation Best practices for Securing Critical and Service Accounts
https://msdn.microsoft.com/en-us/library/cc875826.aspx

2017 Microsoft Corporation Attractive accounts for Credential theft
https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/attractive-accounts-for-credential-theft

2008 Microsoft Corporation TechNet Best Practices for Enforcing Password Policies
https://technet.microsoft.com/en-us/library/ff741764.aspx

Dynamic-Link Library Security ODBC and Drivers
https://msdn.microsoft.com/en-us/library/windows/desktop/ff919712(v=vs.85).aspx

2016 Microsoft Corporation SQL Data Auditing Best Practices
https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine

2017 Microsoft Corporation Implementing Least-Privilege Administrative Models
https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models

1999 Microsoft Corporation - The Administrator Accounts Security Planning Guide
https://technet.microsoft.com/library/cc162797.aspx