# Normalized Full-Stack Development, LLC

# Table of Contents

# Table of Contents
(continued)

# Speech Details

Speaker: Samuel Berger

Topic: Rock Solid Security Methods for Projects with Critical Data

Description:

Protecting your important data has become an urgent issue in today's world of cyber criminals. Twitter, Carnival Cruises, SANS, Adobe, eBay, Equifax, Heartland Payment Systems, LinkedIn, Marriott International, Yahoo and many more have all been victims of major cybercrime. Ransomware is particularly sinister holding your data as hostage by cyber criminals. Even when paid your business is often destroyed and your critical data disseminated throughout the world. The advanced techniques covered in this lecture will show the data specialist as well as the security administrator how to lock down their server environment completely eliminating these threats to your mission critical data.

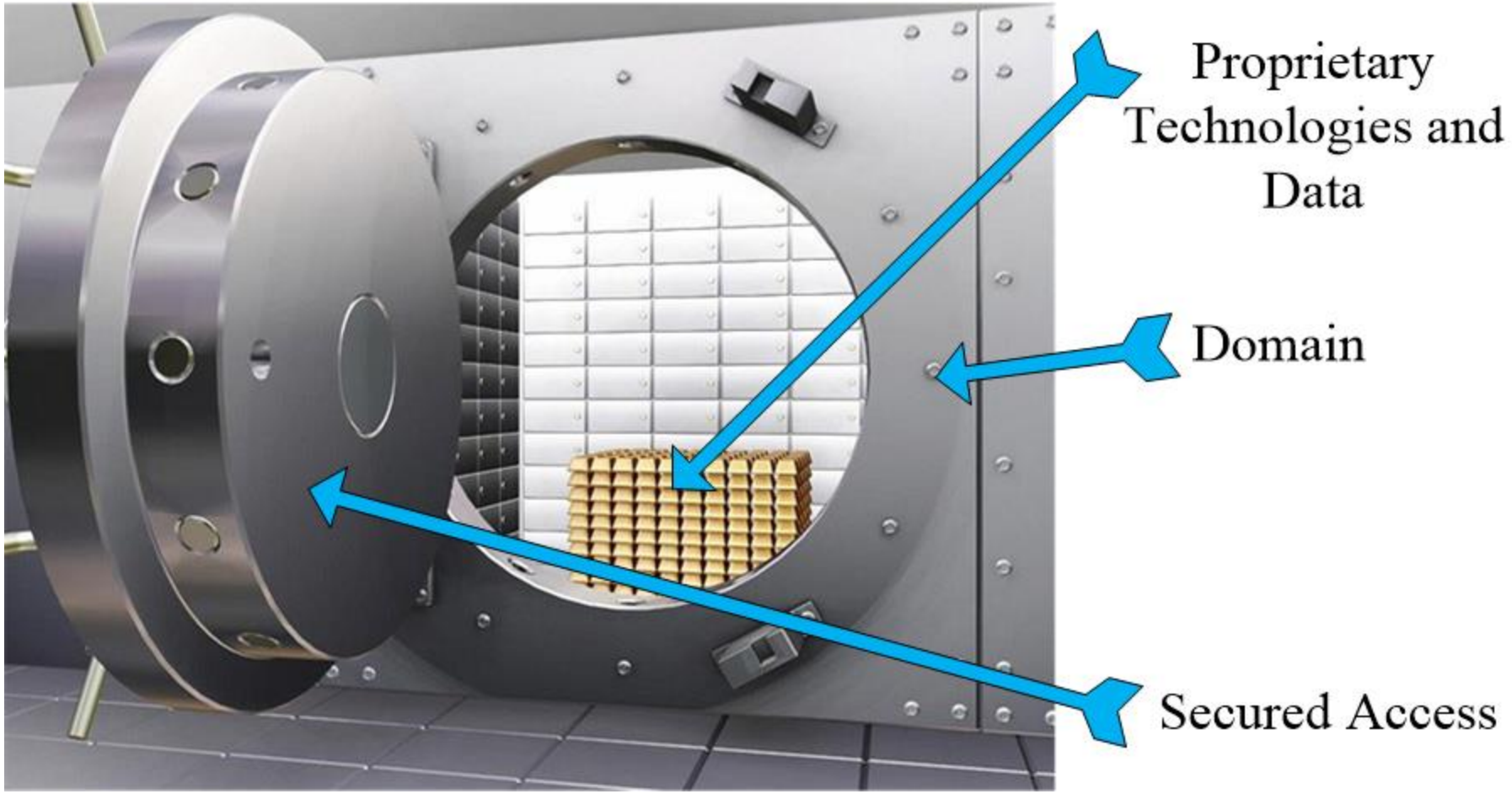
# Background

Samuel is an entrepreneur and developer as well as a data scientist and data security expert working with mass data projects since 1989. He used advanced mathematics, technology and massive amounts of data to predict the world's largest financial market – FOREX. His systems earned clients such as Daiwa Securities, Bank of Montreal, Julius Bär Group Ltd., Société Générale, royalty and national treasuries and many more, returns of over 18% per annum non-compounded over five and a half years. At peak these systems traded over one billion dollars in a day. Samuel has consulted for companies such as E*Trade, Enterprise Architect for Capital Group Companies ($1.3 trillion under management), Verisign, Walt Disney Company, SGI (another Fortune 500) and two industry founding VOIP unified messaging companies where he led database architecture and development. He is an expert in database architecture (both transaction and warehousing) using relational algebra and programming (wrote millions of lines of custom code), as well as on fault tolerance, load balancing, locking architecture, blockchain architecture, Big Data architecture, and database optimization.

Special thank you to the following for reviews and comments as well as added contributions (in alphabetical order):

Robert Abate
Jay Davey
Alec Doughty
Craig Ford
Jason Hodges
Jeff Hodges
Robert Hodges
Erick Peterson

# Company Production Cyber Vault

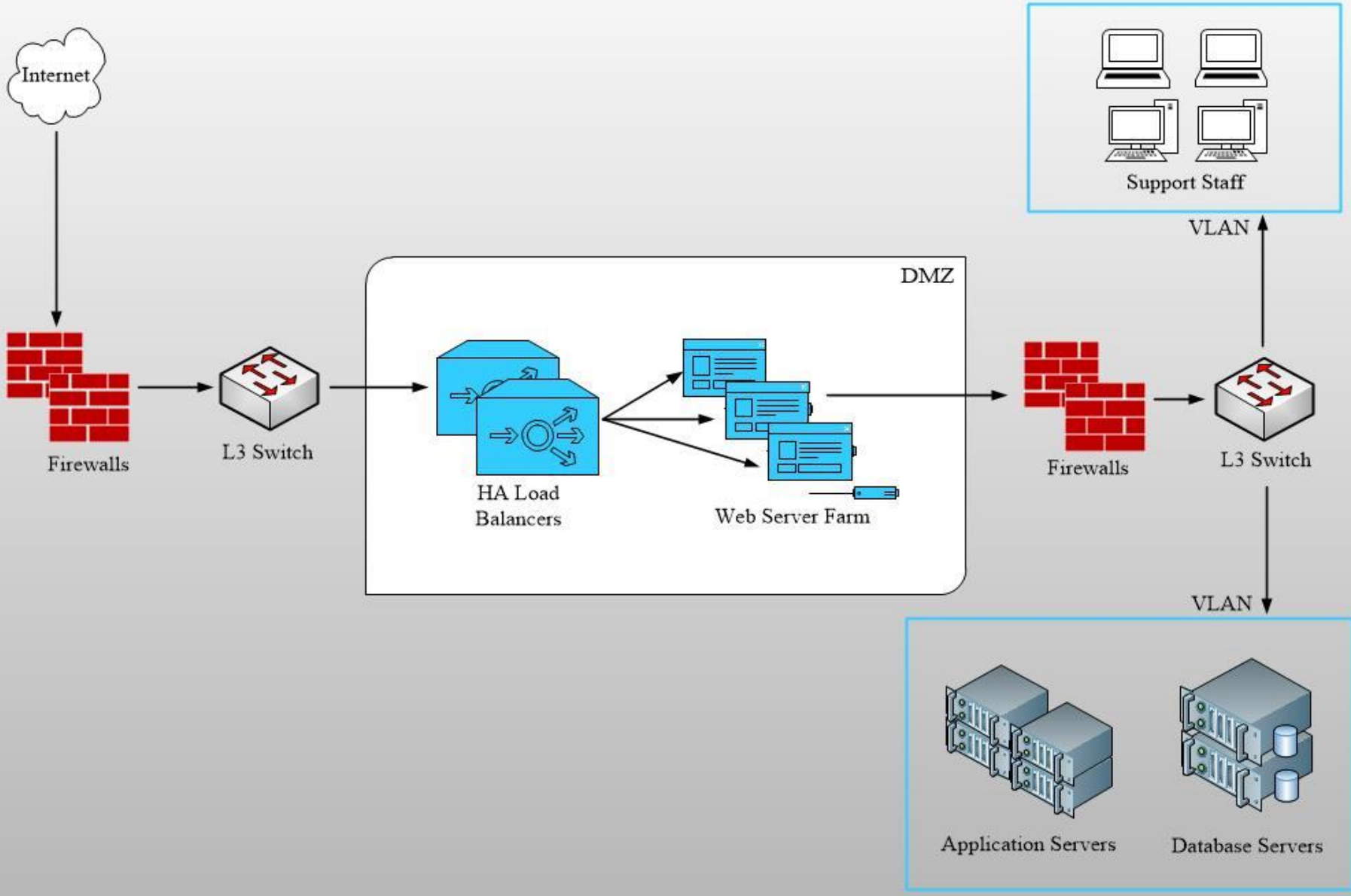


# Introduction

Although this lecture does cover many corporate-wide security measures that should be followed, it is primarily focused on securing intellectual property and data stored and maintained in a professional production domain environment.

As this topic is vast, I have put a great amount of detail in the slides themselves; however, time will not permit me to cover all of these points. So the slides will be made available and can be used for future reference when securing your production domain and, to a lesser extent, in setting up corporate security policies. My hope is that this information will be widely disseminated and contribute to substantially reducing damage to, and theft of, hard-earned intellectual property and data.

# Typical Production Domain



# There are Three Ways to Attack Production Domains

Cyber predators are always after your hard earned valuable information

# 1st Method: Get Control of the IIS or OS Layers and Seize Control of the Domain

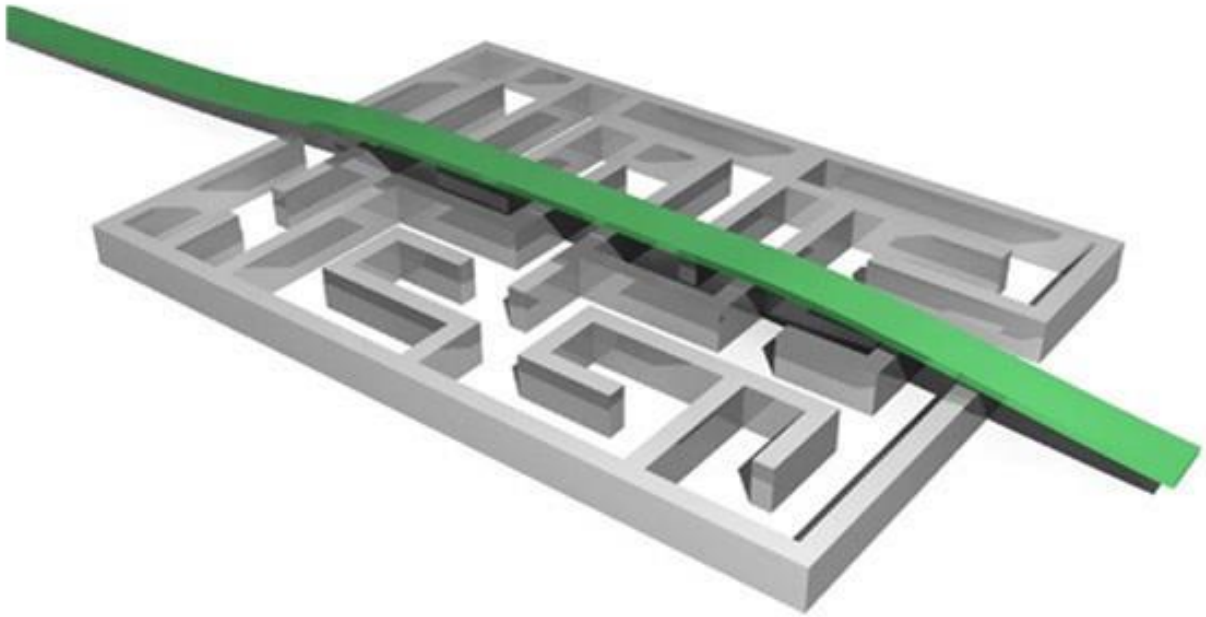
1) Distributed denial of service (DDoS) attacks:

   In simplest terms a DDoS attack is a bombardment of requests being made to a specific URL in order to overwhelm its maximum capacity to serve and thus making the URL unavailable to legitimate users. This method of attack is often preceded or followed by a demand message for Bitcoin in order to stop the disruption of the organization's services.

2) Direct attacks via compromising IIS or OS security:

   Attackers are basically looking for open doors left by not properly closing all non-functional access points and unnecessarily leaving rights active for admins that are not currently accessing a given server. Using default settings is in most cases a security risk as well.

# Bypassing Your Secured Access



Although secured access is critical, criminals do not like following rules

# Let’s Start at the Front Gate

Firewall

Options and Strategies:

1) Real-time Blacklist (RBL) Services (also known as Domain Name System-based Blacklist or DNSBL): Use Dynamic Block Lists and Malicious IP feeds with Active Threat Prevention services.
2) Monitor traffic to identify malicious activity and block IP addresses.
3) Lock down ports not required for ingress/egress for services provided.
4) Lockdown Intrusion Prevention configurations. Restrict based on protocols and services.
5) Lockdown based on GEO-IP filtering where possible.
6) Block connections to/from Botnet Servers (All, Dynamic Lists or Custom Lists).

# IIS Layer

Note:  I used IIS and Microsoft software for this example.  Time being limited Apache, Unix, Oracle, etc. will not be demonstrated today.

IIS Dynamic IP Address Restrictions: [1]

- Dynamic IP address filtering, which allows administrators to configure their server to block access for IP addresses that exceed the specified number of requests.
- The IP address filtering features now allow administrators to specify the behavior when IIS blocks an IP address, so requests from malicious clients can be aborted by the server instead of returning HTTP 403.6 responses to the client.
- IP filtering now features a proxy mode, which allows IP addresses to be blocked not only by the client IP that is seen by IIS but also by the values that are received in the x-forwarded-for HTTP header.

When configuring restriction settings, Deny Action Type can be set as:

- **Unauthorized**: IIS returns an HTTP 401 response.
- **Forbidden**: IIS returns an HTTP 403 response.
- **Not Found**: IIS returns an HTTP 404 response.
- **Abort**: IIS terminates the HTTP connection.

[1] IIS Dynamic IP Address Restrictions section quoted from Microsoft Documentation
https://docs.microsoft.com/en-us/iis/manage/configuring-security/using-dynamic-ip-restrictions

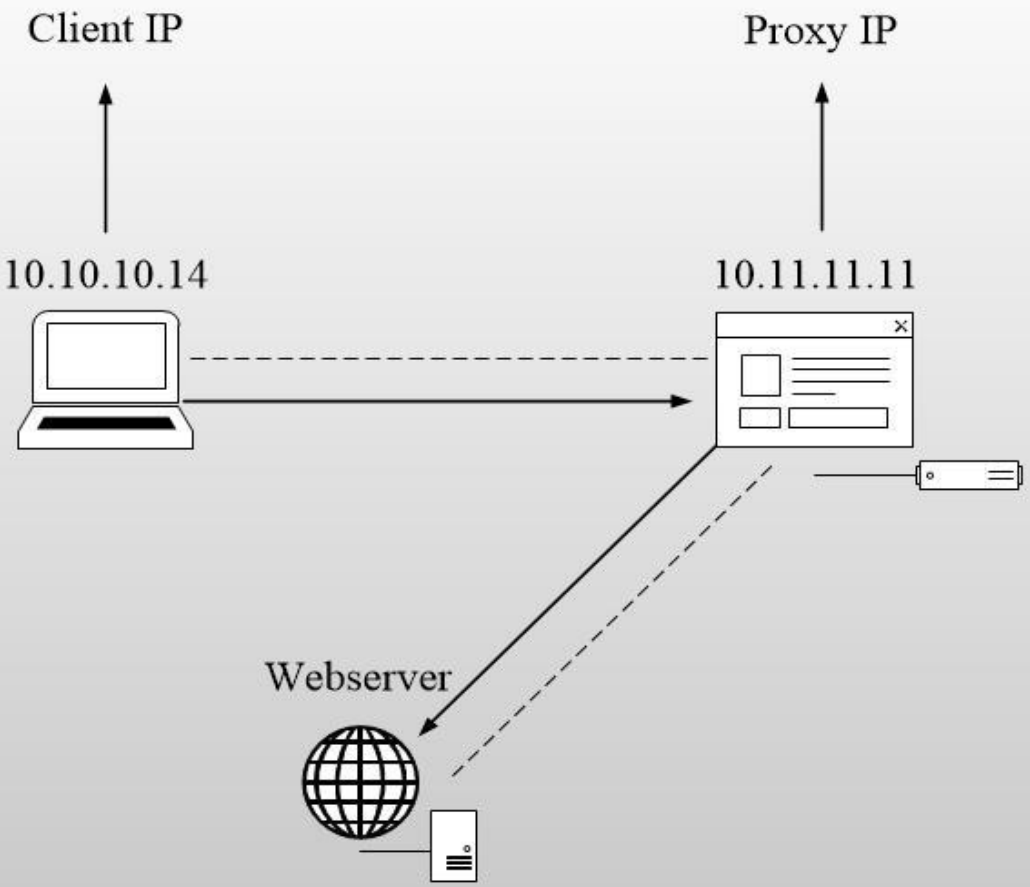
# IIS Layer
## (continued)

Proxy Mode: [2]

One of the challenges to IP filtering is that many clients access IIS through one or more firewalls, load-balancing, or proxy servers; so the IP address may always appear as the server in the request path that is nearest to the IIS server. In IIS 8.0, administrators can configure their server to examine the *x-forwarded-for* HTTP header in addition to the client IP address in order to determine which requests to block. This behavior is called "Proxy Mode."

At NFSD we are working on extending our proactive "**Cyber Falcon** ™" software system to analyze the *x-forwarded-for*, *x-forwarded-host*, *x-forwarded-proto* (or *x-proxyuser-ip* if the client is with Google), *etc.* not only for system protection, but also to inform IT so that they can pass the information when collected to the Internet Provider of the attacker and possibly the FBI or appropriate foreign police agencies. Note: the US and EU block most of this information by their privacy laws; however, many attackers come from countries outside of these jurisdictions and thus at times the information is not blocked.

[2] IIS Dynamic IP Address Restrictions section quoted from Microsoft Documentation
https://docs.microsoft.com/en-us/iis/manage/configuring-security/using-dynamic-ip-restrictions




The "x-forwarded-for" HTTP header, this is a de facto standard that allows proxies and load balancers to properly handle client-server traffic. The client (10.10.10.14) wishes to navigate to the web-server but the web-server has a reverse proxy or load balancer (combination) in the path of communication, therefor proxy (10.11.11.11) Will handle the HTTP request on behalf of the client and append the "X-forwarded-for" header to the HTTP headers.

# IIS and Server General Settings

IIS Customizations:

1) Remove default website from IIS server.
2) Configure IIS Directory Security. Monitor IP addresses (subnets, etc.) that are trying to access files.
3) Utilize IIS lockdown tools cautiously to assist with the lockdown process and to ensure that connectivity with backend services is not lost.

Server OS Customizations:

1) Disable/remove all services not required to run base application. Remove FTP & SMTP if not used. Use TCP/IP port blocking. (More to follow)
2) If running FTP on a server; configure to prevent Brute-force attacks by disabling account based on login attempts.
3) Regular maintenance schedule to apply updates.
4) Monitor all local logins and immediately shut down any unauthorized access. (More to follow)

# OS Layer
## (continued)

5) Monitor Local Admin groups on servers – no additions! All admins must use local individualized admin accounts, never a domain or general admin account. (More to follow)
6) Restrict write access to servers to Admins only (i.e., no user write access). (More to follow)
7) Disable NetBIOS over TCP/IP.
8) Server non-user login passwords should be 50 characters long using multi-Unicode unique characters.
9) All applications such as SQL Server that have to run on an admin account use their own login account with the 50 character multi-Unicode password.
10) Every application requiring to run as an admin given its own account following the password requirements.
11) No two custom application accounts have the same password.
12) For each of these server non-user admin accounts an application will auto-generate a new 50-character multi-Unicode password based on the server name, purpose and MAC address. The application passwords will remain static, however, will use obscure usernames so as to be difficult to identify.

# OS Layer
## (continued)

13) All servers roles and features removed that are not necessary for its very specific functionality (i.e., Print & Document services, IIS, BITs, SMTP, SNMP, etc.).
14) The “Administrator” account for each server is not an administrator and given minimal rights in every server in the domain. As such, they act as Trojan horses.
15) The Administrator account has its password changed to a new randomly created multi-Unicode password every minute for every unit in the domain.
16) Actual individual administrators each have their own admin account on each server; however, the accounts are to be disabled. These accounts should only be enabled when needed and only for that precise time. (More to follow)
17) Individual admin account for humans must have a different password per server. Cumbersome, but effective. Adding a portion of the computer name to an admin’s usual password should suffice. This prevents an intruder from being able to instantly access the network using whatever method they used to enable the login in the first place (unlikely to do the first part, but the second piece buys some time).

# OS Layer
## (continued)

18) Database servers deny data access to local admins and domain admins. Only database accounts allowed access.
19) An admin account usable by an application running on the DB server to manage enabling and disabling/disconnecting admin accounts by force. This account will have an auto-generated password that is changed every minute. The password is calculated based on time, server name and server MAC address.
20) No domain admin accounts outside of what is required by Active Directory.
21) Domain admin accounts to be accessed only from a computer that is isolated for that purpose only (i.e., no email, thumb drive access, file access, etc.). This unit should be accessed remotely only when needed.
22) Remote desktop support is turned off by default in all domain servers.
23) An application will run on the administrator's computer that connects to a database. The stored procedure will verify the administrator's user name, password and MAC address for verification and once verified their specific admin account will become enabled as well as Remote Desktop access and the Remote Desktop port number will be opened.

# OS Layer
## (continued)

23) Every minute the database will check to verify that the login is active. If it becomes idle for over one minute then it is forced to disconnect, the login is disabled, the port is shut and the Remote Desktop function is disabled. Longer access that is not idle sensitive can only be achieved from the console at the datacenter.
24) Any time an admin account is made active a text is sent to the admin asking for verification that they are the one making the request. Without a text response back the application will fail to enable the account.
25) Attempts to access via the console will also require this process to activate the admin account. The only difference is that the system will allow console access for longer periods as Remote Desktop and opening up ports is not necessary making the access significantly less exposed.
26) Ideally, the IT admins should have two computers. One where they are a standard user from which they do all emailing, standard work, etc., and the other accessed via a simple desktop switch whenever they must do administrative work and require additional permissions. This virtually eliminates all SSH tunneling, keylogger Trojans, ARP poison/spoofing (essentially man-in-the-middle attacks), etc.

# Domain Layer

1) The data and the access portal (IIS or Apache) must reside in its own domain.
2) All access to your data must come through the access portal with all direct access removed.
3) IP of your data server(s) cannot be exposed outside of the domain. This is true of all of the servers within your domain.
4) Remote desktop, and other methods, should all be disabled.
5) All ports, outside of the ones precisely needed for functionality, shut off within the domain.
6) SQL uses ports 1432 and 1433 by default. No reason to keep using the default. Either use an obscure port designated for another application not being used in your domain, or use one of the many available ports never assigned (tens of thousands to choose from). In my experience SQL runs perfectly on other ports; however, you must test any application when you change its default port address to make sure that this address is not hard coded somewhere in the application. Also, the functionality that I have used may be different from yours, necessitating testing even SQL itself. That being said, all major database systems should run on almost all ports.

# 2nd Method: Penetrate the Access Layer via Weaknesses in the Security Protocols

Direct data attacks via compromising web/application security:

Compromising the web/application security in order to gain control of the data directly via the web/application portal is a very common method, and security configurations in most cases are amazingly weak. Most websites and applications have very poor security and common mistakes are made allowing SQL injection and other methods to not only gain access to the data, but also even take control of the entire domain. Further, human mistakes make up the majority of opportunities to gain key credentials and production domain security information.

# Data Access Layer

1) All databases accessed only by user logins, no system administration access.
2) Database access logins have database access only (i.e., specifically deny “datawriter” and deny “datareader” access and give no other database level rights).
3) Database access login only given permissions to access specific stored procedures. All data access must be via stored procedures.
4) All stored procedure names are descriptive, however, have three characters or more added to the name using non-English Unicode range language specific unique characters. Russian, Icelandic, Spanish, etc. have unique characters that are IIS and software compatible for naming conventions. Note: many unique characters are not compatible with various necessary software components. Some are necessary for base functionality and others for support software. An example is that MS SQL allows some characters that its own query analyzer does not properly support.

# Data Access Layer

(continued)

5) No database on the server to allow dynamic SQL to run for any reason! Dynamic SQL allows for SQL injection, which is where you can pass commands with user rights of the process. In the case of dynamic SQL the user rights must be at an administrator level giving the SQL injection statement maximum rights to the data. Worse is the fact that SQL is by default a local admin and often set up as a domain admin. This gives the SQL injection statements the ability to completely compromise your entire domain – making it the single worst mistake you can make outside of making your OS admin password "password". The most common use for developers of dynamic SQL is doing database searches, making it easy to find and attack. All databases must follow the same protocols, even admin databases used for load balancing and administration functions.
6) SQL server user account login has the password changed frequently.
7) Create two application logins actually to be used with passwords changing every minute two minutes staggered (i.e., every minute change one of them) so that every minute the application has a login available all of the time with no chance that the login will occur at the same time as a password change potentially causing a failed attempt. Based on the exact time, the application can calculate which login to use.

# Data Access Layer
## (continued)

8) The passwords are maximum length multi-Unicode using unique characters.
9) A routine in the application can calculate the new password given the login purpose, date and time of day, making certain that a password will always be both unique and calculable for both the application and data layers simultaneously.
10) Update regularly to address known published vulnerabilities.
11) Beware of open source code that may contain custom attacks causing zero-day exploits (vulnerabilities or holes opened by the custom code that allow attackers to gain control undetected).
12) Do not allow 5G! Stay away from any 5G implementations either for personal use or corporate use. Forbid all employees from using a 5G device of any kind. 5G by design is a massive security disaster. Our government should not allow 5G to be used anywhere in the US under any circumstances.

# Secured Socket Layer (SSL)

1) SSL is an important encryption layer between the servers within your domain. It does not affect traffic going between the client and the web/application layer, only traffic being sent from the web/application layer to the rest of your domain servers that have the SSL certificates installed.
2) The main goal of SSL is to prevent a hardware or software sniffer from capturing information from your network traffic and disseminating that information to the controller of the sniffer. Sniffers can usually only pass a small amount of the network traffic back to their controller without being detected or causing the network to become slow and unresponsive, leading the admins to search and find the cause. As such they are often programmed to detect logins and passwords carried over the network, allowing the controller to gain access to your servers and data.
3) Make sure all of your domain servers have SSL (preferably the EV edition) for greater security and transparency to your client. SSL is relatively inexpensive.

# Secured Socket Layer (SSL)
(continued)

4) Important to note that foreign countries with potentially bad intentions have been purchasing some of the SSL makers and vendors. Take the time to research the ownership and make sure they are entirely US or Western European owned as the intellectual property laws are much stricter in those jurisdictions. Keep verifying ownership every six months.
5) SQL Server, as most advanced relational database management systems (RDBMS) do, create their own sub-network within your domain that is not commingled with any other network traffic and is not subject to SSL. As such, SQL Server in 2000 came up with their own internal SSL that comes included in SQL Server. In order to reach Zero Trust Architecture you must apply this layer of encryption as well. Note: due to this network architecture, database traffic is much harder to sniff, however, this is no reason to allow this hole to persist.

# The Human Factor

1) Ego is the #1 killer of companies and the cyber criminals' best friend!
2) Egress surveyed 500 IT leaders from the US, UK, Belgium, the Netherlands, and Luxembourg. 97% of those surveyed put insider breach as a significant concern.
3) In my personal professional opinion over 85% of cybercrime is committed with the intentional or unwitting assistance of at least one employee of the company. Kroll, in a report to the Information Commissioner's Office (ICO) in 2017 had it higher, listing the company's employees responsible, primarily due to human error, for 90% of all data breaches.
4) As normal users do not have the super user rights or decision making capabilities within an organization that cyber criminals are looking for, the employees that these IT leaders fear most are, ironically, executives and the IT professionals themselves.
5) First mistake is to give those not really needing an admin account that level of access. Even "local admin only" access is more than sufficient to compromise the entire company.
6) Computers only require admin rights to install software. Once installed simple user rights suffice.

# The Human Factor
## (continued)

7) IT professionals rarely require admin permissions and can designate a specific computer that they can remote in to for any domain management activity. From their local computer they should always be logged in as a standard domain user. This will protect them from being compromised by emails and work files.
8) IT professionals should only use a domain admin account on local user computers, including their own, when installing, and possibly configuring, software.
9) Sales and other executives often want a copy of the production database on their local computer. Very big mistake and a common key source for information by a cyber criminal as the database system tables, even when separated from the master DB, still have the login names and their permission levels. This tells the cyber criminal surgically where to strike once they are in the domain. Also, those databases themselves that are on the exec's machine can be a treasure trove and are NOT protected.

# The Human Factor

## (continued)

10) The most dangerous threat is a custom attack targeting a single company. These are tailored to exploit known specific company information and vulnerabilities and have a greater level of effectiveness. With some basic open-source intelligence (OSINT), which the DOD defines as information obtained "from publicly available information that is collected, exploited, and disseminated" [3], criminals can gain a tremendous amount of information about companies and their key employees that make trust and familiarity far easier for the attacker to achieve, and often leads to well-crafted spear phishing attacks and other social engineering. The only way to prevent such an attack reasonably is to not to allow users to log in as admins. Any user.
11) The desire for executives to save money is another key opportunity for the cyber criminal. In the attempt to economize the choice is often made to give up control of domain hardware access either by using Cloud services hosted by other entities or by outsourcing IT completely. The opportunities for physical data theft are covered in the next section dealing with the third way to attack a production domain.

[3] Public Law 109-163, Sec. 931 titled "National Defense Authorization Act for Fiscal Year 2006"

# The Human Factor

## (continued)

12) In the case of outsourcing IT, outsource IT companies often outsource **themselves** for profitability enhancement to inexpensive providers frequently located in politically aggressive nation states – especially for all work that can be done remotely. The person that is given full uncontrolled and unmonitored access to your intellectual property and data remains unvetted. Even in a case where the primary provider does not outsource, you still may have no idea of whom they have hired or contracted with to work on your critical systems. You are completely at their mercy.
13) In my professional opinion far more intellectual property is stolen by foreign entities and persons – owing to forfeiting control to save money – than all of the other methods of gaining access combined. This is the single biggest way that an employee (executive in this case) can ensure dissemination to competitors and bad actors of key company technologies, trade secrets, intellectual properties and data.

# Ransomware Attacks Companies Across the Globe

# 3rd Method: Get physical control of the data via a backup drive or the production servers directly

The physical control of the hardware on which the intellectual property and data reside is no longer commonly controlled by the owners. Monetary considerations have prevailed, allowing access for cyber criminals to directly, often with little effort, gain great value.

# Physical Control - Backups

1) Backup drives are rarely encrypted. Enterprise Strategy Group polled nearly 400 companies and found that over 60% did zero encryption of their backups. Many others encrypted some and only 7% encrypted all of their backups. Interesting to note is that their study found no relative difference based on size with 56% of the companies with revenues over $5 billion never having encrypted a single backup. All backups should be encrypted as a company standard.
2) Most backups are stored in the company IT section (usually a locked room where the IT staff work part-time or full-time), usually on an open shelf for convenience, which typically has significant foot traffic from developers, IT staff, staff visitors, executives and others. As such, they are open to theft. All backups should be secured in a locked segregated location preferably with cameras and a security system. If disabled, the system should be set up, via remote uptime monitoring, to immediately alert the IT staff.

# Physical Control - Backups
## (continued)

3) Large companies often pay for offsite storage. Security problems arise as no one vets those that are maintaining or delivery the backups (usually low wage employees). Virtually never is a backup tape required for use and as such no one audits to verify that all of the tapes are accounted for and actually contain the original data placed on the tapes at the time they are sent (i.e., a tape can be stolen, or replaced with a blank, and no one will ever know). For security, verification procedures need to be implemented for all off site tapes.

Best Practices for Backup Maintenance:

1) Backups maintained offsite using another server(s) in the same domain, yet another secure co-location. Being in the same domain can allow full use of SSL EV encryption during the transfer.
2) For fault tolerance two locations carrying the identical data are preferable.
3) Backups are all encrypted.
4) No backups are kept on site.
5) The server(s) that the backups are sent to have the same security methods and protocols as the rest of the domain.

# Physical Control - Backups
## (continued)

6) A dedicated one megabyte line is used for the transfer of the data and communication between the datacenters so as to avoid exposed traffic (i.e., off the net).

7) Make sure the backups fully cover all disaster recovery requirements. In SQL, for example, the “msdb” database is required to restore jobs running on the SQL server as all jobs are stored in this DB. The master DB controls all of the logins. Just having the login and password is not sufficient as the exact encrypted ID is required to match the login in the master DB with that of the user DB. If they do not match then the user DB will be inaccessible by the application layer, making the site or application useless. It can take significant time and be error prone to recreate these permissions by hand. Only the most experienced DBA will know how to write scripts on the fly to recreate these permissions.

# Physical Control - Backups
(continued)

8) It has been over 20 years since I last worked at a company that required a full disaster recovery test of a given system once per month. The last company that I saw do this was the Walt Disney Company. Very important to do as it is not only a great way to prepare, but you will also never know what you are missing if you do not follow this practice. I worked at one company that paid $380,000 to a consulting group to spend 9 months setting up all of their servers with full backup systems using Veritas NetBackup. All of their work was useless and heavily flawed. As the company never tested their work they did not know the extent of their mistakes until I wrote a report on all of their errors. Everything had to be redone. It took me only one week on my spare time. Not only did the provider massively overcharge, but they could not even accomplish the simple task for which they were paid. Do not trust, test and test regularly…

# Physical Control – Servers

Production servers are usually maintained off site in a large co-location also referred to as a data center. The facilities are usually independently owned and operated. Large Fortune 500 companies often have their own facilities for greater security and their extensive space requirements.

1) I have seen many companies, including multiple Fortune 500 companies, maintain these servers on site. Very bad idea as the most likely security threat to your actual hardware is your own staff. You increase the access and do so with those that know the value of the information. Developers are also more likely to take it upon themselves to modify production directly that invariably has adverse consequences, often serious.
2) Direct access to the hardware can also increase the systems exposure to malware via thumb drives that have been multi-purposed. Another risk is that of developers and IT personnel using the Internet from the console to look up something on an unknown site as they do not want to walk back to their own workstation.

# Physical Control – Servers
(continued)

3) Temperature control and monitoring as well as power is also less than professional in a standard office building. Specialized facilities cost tens of millions to build and have significant redundant power, temperature control systems and multiple bandwidth vendors with high throughput and minimal hops to the main International Internet backbone.

4) Typically companies that use off-site professional facilities rely on those facilities to vet their staff and to monitor security from personnel of other facility clients with respect to your caged equipment. It is important to put your own security monitoring equipment in the cage so you can detect intrusion as it is being attempted as well as ensure equipment functionality. Motion detectors, cameras and lasers can be used to help secure your valuable intellectual property and data.

# Physical Control – Servers
(continued)

5) It is important to note that I have not seen or heard of hardware being tampered with, or stolen, from a professional facility. Between their cameras and monitoring equipment combined with IT personnel frequently being present working on their systems it does not attract cyber criminals. The mere fact that they will have difficulty in finding the professional facility makes this very unlikely and another reason why storing the servers in your own building is a security flaw. However, that being said, if you have something of enough value then this is an element that typically can be made more secure. Better a little extra precaution as it is relatively inexpensive and simple to add. It also provides an audit tool for who on your staff have accessed production and when.

6) No outsourced IT for production under any circumstances. If your company's executives really feel the necessity to outsource IT, which I highly advise against, do not apply that decision to the production servers for any reason whatsoever.

# Physical Control – Servers
## (continued)

7) Side-Channel Attacks (SCA) – These are both very sophisticated attacks that can be imbedded in the hardware itself prior to purchase and emitting monitoring devices as well as simply gaining control of footage of cameras that observe admins logging in, etc. The sophisticated methods include imbedded hardware reverse encryption technologies. These methods are rare, but new cases have been recently reported in hardware as such hardware security updates should be deployed frequently to all production servers as a standard practice. The less sophisticated methods are becoming common and can be surprisingly effective.
8) Keep in mind any cameras that are watching when admin credentials are used either in a corporation or at the data center. This is an example of a low sophistication side-channel opportunity.
9) Your security is only as good as you maintain it. The Epstein case a perfect example of missing tapes and camera “not working”. Do not endanger your company by ignoring your security infrastructure.

# Physical Control – Servers
(continued)

10) Another side-channel attack that is targeting cloud computing in particular is an attack on CPU caches. By design CPUs work on multiple processes with different threads; however, the same common cache. This flaw is so fundamental by design that it can take more than a decade for hardware manufactures to resolve.
11) Enforce company-wide, and without exception all servers, encrypted USB ports. Also, disable out of sight (i.e., behind the unit) USB ports.

# Physical Control – Shared Hardware

1) No cloud deployments!  Physical security governance.
2) Shared hardware means that you cannot control who else has admin on the server that controls your data and what they install.
3) Backups are made of the entire server by individuals you never know or vet. No clue as to where those backups are stored or who has access to them.
4) Co-location access cannot be controlled or monitored.
5) Hardware side-channel attacks, such as the CPU cache method mentioned, cannot be secured.
6) Some company data is not critical with respect to dissemination, yet may be critical to maintain availability.  As such, each company must make its own determinations regarding the protective levels that they it wishes to deploy.

# Be Proactive Not Reactive and Put Your Ransom Money Away

Be proactive and secure your data to avoid paying off criminals

# Proactive Seek and Intrusion Elimination Software – NFSD “Cyber Falcon ™”

1) Cyber Falcon ™ uses a low level scripts to check system information for early intrusion detection. The method that Cyber Falcon ™ uses is so thin that memory and CPU usage are both virtually undetectable.
2) If an intrusion is detected Cyber Falcon ™ attacks the intrusion to regain company control.
3) Cyber Falcon ™ keeps full logs of all checks and in the event of even a suspected intrusion, it alerts IT personnel immediately.
4) Once IT has configured the production security protocols Cyber Falcon ™ continuously monitors to ensure that all protocols are maintained. In the event they have been altered it immediately notifies IT and any other personnel on its notification list.
5) Cyber Falcon ™ even monitors for DDoS attacks and if the OS configurations have been altered.
6) We are working on capturing information real-time that will help identify and locate the attacker.
7) Cyber Falcon ™ is designed to be a real-time production security force.

# Forensic Analysis – How To Find What Weakness Was Exploited and Possibly the Perpetrator

Just like in a murder case clues are often left behind

# Forensic Analysis

In the event that your data is published:

1) You can potentially tell a lot from the published data about how you were attacked. I gave an example in my IDEAS data security speech in 2017 of the DNC data theft. At that time the Director of the Federal Bureau of Investigation, the Director of the Central Intelligence Agency and the United States Director of National Intelligence were all going on national TV and stating that they had personally viewed intelligence information indicating that Russia stole the data from the DNC servers and provided that information to Julian Assange and WikiLeaks. Assange denied the claims as did the Russians, but the public believed our intelligence leaders as they appeared far more credible as should be the case.
2) People give false information for many reasons, however, math and data tell the truth. When a cyber criminal penetrates domain security, whether true or not, they assume they have a very limited amount of time. As such they are looking for key information only. The information taken was heavily mixed, with the bulk of the information being useless.
3) Profiling the publisher can also be extremely useful. Assange only wanted data in its entirety. He focuses on company backed-up information that can be copied and published for a full picture and let the readers do the filtering.

# Forensic Analysis
## (continued)

4) The DNC information was all from one source, emails, and was dispersed without filtering between the President to the most unimportant staff member. Tens of thousands of emails often with attachments. The evidence published pointed to a very high probability of a data backup tape being taken by an internal staff member, which fit Assange's profile as well.

5) I got a lot of heat for my statements; however, in the last two months the interviews of these officials made in the top secret Intelligence Committee secure skiff were just released with key documents. All three claimed under oath that they had seen no intelligence indicating that Russia had anything to do with the DNC data theft.

7) In the end, you can only trust the data. The data published also told us the date the data was stolen due to the dated emails. From there you can check the specific backup tapes of that day and the next day for fingerprints and DNA. High probability that both are available.

8) Trust the evidence not the people presenting their story with possible motive.

# Forensic Analysis

## (continued)

In the event that your data is not published, but you see confidential elements being used elsewhere:

1) The first point is to go to the source and work backwards till you find the exact method of the breach. So you start with the data and work back to the initial entry point.
2) When the data is not yet published but you still know information was stolen, you must start with the location at which it was stored. There are three ways that data can be breached from a database:
    a) A backup was taken.
    b) The data was taken from the database directly.
    c) The database server OS was compromised allowing for the data files to be taken.
3) It can be very time consuming to review each backup tape independently looking for clues if the data resided for years in your systems. As such, this should be your last effort. It is very important to handle the backup containers with gloves and use the minimum contact as if the tape has been stolen; the only clues may reside on the container or blank drive itself in the way of fingerprints and DNA.

# Forensic Analysis
## (continued)

4) The easiest, time wise, is to see if the OS was compromised as the OS system logs will display unusual activity unless they have been wiped, which could have happened, but rarely occurs. One method is to write code that catalogs each unique entry and records the times that entry was made. You can remove numerical data prior to making the comparisons and have a reasonably good chance to spot the unusual entries fairly rapidly.
5) When analyzing the OS security logs you can also see all of the unique applications running over time. This can catch a rogue application that entered your OS.
6) If the data breach is caught in a timely manner the database system logs keep track of the exact system calls and the user account that made them. This can be extremely informative if a direct approach using something like SQL injection or a compromised login.

Once you have identified the method then you can backtrack to the initial source of the breach.

1) As mentioned, with a backup tape it is more standard investigatory work with fingerprints and DNA.

# Forensic Analysis
## (continued)

2) In the case of an unauthorized application you can search the administrators computers and thumb drives as well as the local server Internet history (as admins often download directly to the server, which is a security mistake already covered). At some point that application was installed, so you will have the infection date, which is valuable.

3) If no source for the application exists then you must pursue the logic that your domain may have been compromised via an unsecured point of entry. Those points are far to numerous to cover here, but include applications allowed to be enabled that are unnecessary, ports not being closed, poor app security (i.e., app running on an admin account, etc.), side-channel attacks, software sniffers, etc. The more holes your company has allowed, the more holes you have to check.

Once you know how and when you have a much better chance to figure out who. Also, in the process you can properly secure each hole as you research them and once done do not forget to finish securing your environment so that this disaster will never reoccur.

# Thank You!

# NFSD

## US Coders - Transparent Method
## Guaranteed Results - Fixed Cost

Samuel Berger
Vice President of Technology
sberger@relalg.com
www.linkedin.com/in/samueleberger