Chad R. Johnson (2015)
Seminar Research Paper

GOOD FENCES, GOOD NEIGHBORS: BEST PRACTICES FOR ENDPOINT SECURITY

Approved: _____Dr. Sabina Burton_____ Date: ____05/05/2015_____

Paper/Project Advisor

GOOD FENCES, GOOD NEIGHBORS: BEST PRACTICES FOR ENDPOINT SECURITY

A Seminar Paper

Presented to

The Graduate Faculty

University of Wisconsin – Platteville

In Partial Fulfillment

Of the requirement for the Degree

Master of Science in Criminal Justice

Criminal Justice Theory

By

Chad R. Johnson

2015

ACKNOWLEDGMENTS

My most sincere appreciation, eternal gratitude, and undying love go to my late wife Sara, who lost her battle with cancer at the age of 29 in the summer of 2014, the year before the composition of this document and my attainment of this Master's degree. We fell in love when I was a high school drop-out, and her unerring belief in me and unwavering support was an inspiration and source of strength. I miss her every moment of every day. I wish that she could have lived long enough to see our dreams realized, and hope that I am able to impart even a fraction of that strength to our daughter, Vera. Sara deserves far more than the petty words I have to offer here.

I would also like to thank every person that made this document possible. It would not have been done without a host of people that have played a part in my life. Every teacher that took an interested, every boss that gave me a chance, and every person of authority that cut me a break and wouldn't give up on me, all made the difference between being the subject of a seminar paper in the criminal justice field and the author of one.

GOOD FENCES, GOOD NEIGHBORS: BEST PRACTICES FOR ENDPOINT SECURITY

Chad R. Johnson

Under the Supervision of Dr. Sabina Burton

Statement of the Problem

Because of the sensitive nature of police work, the information collected, and the tools used, law enforcement agencies present a very special case in terms of information security. Regardless of the jurisdiction the agency covers, there are challenges that are inherent to the tasks. This is for two primary reasons. One is that law enforcement organizations make particularly attractive targets, due to the perception that proving the agency is vulnerable sends a message that everyone else is vulnerable as well. The second reason that these organizations make particularly attractive targets is that the risk is often worth the reward, because law enforcement agencies contain a trove of sensitive data. Every single day, a police department handles copious amounts of sensitive data, internal and external communications, retains digital evidence, and has privileged access to even more attractive networks such as national criminal databases or the Department of Motor Vehicles network.

Regardless of the protections and countermeasures in place to prevent intrusion, the primary point where security is most critical is directly in front of the user: their own workstation. The topic of this capstone paper focuses on identifying the best practices for securing those endpoints and ensuring safe operation by the end user. Many of the principles

explained would be universally applicable to any endpoint, but special attention was paid to the special challenges of securing law enforcement endpoints.

Methods and Procedures

Primary data was gathered in part from crime statistic sources, such as the Bureau of Justice Statistics, which tracks crime rates in a separate cyber-crime category. In addition to this, information was gathered from expert sources confirming both, the frequency and scope of attack on law enforcement networks, as well as the special challenges they face. These include trade journals, experts, and white papers of a well-known and respected reputation within the Information Security community. These include the three main Information Security trade publications – SC Magazine, SecurityFocus, and SecurityNewsPortal. Finally, some first-hand accounts from the author on how these best practices described have proven to be effective at mitigating threats. Furthermore, this work builds upon previous research on this topic, including previous work submitted and accepted by the university.

Summary of Results

Utilizing existing technologies in a statistically typical environment can yield effective security measures at little to no cost. However, an investment still must be made not only for security professionals within the organization, but also at a user and management level, to make security and adherence to sound practices a priority for everyone involved in the use of technology.

TABLE OF CONTENTS

INTRODUCTION

Manifest Destiny pushed our forbearers west. Out there, so the promise went, was a world of possibilities and vast swathes of verdant virgin lands rife with opportunity and potential. All anyone needed to do was to have the fortitude to go out and take it. So many did, and soon our nation's borders touched each ocean. In between those coasts was a great expanse of sparely populated land, a small handful of communities, and very little in the way of law and order. No law enforcement agency could possibly hope to secure and enforce the laws enacted in the name of our people. Unfortunately, an oft byproduct of ambition is desperation, and so for those that went west and failed there became a new mission in surviving at any cost. The result was a state of lawlessness that in our age has become the stuff of legends, now romanticized in our western movies and literature. Eventually, the blank spaces on the map filled in. Brave men and women fought to bring order to the chaos, and did what was necessary to prevail. In a very short amount of time, historically speaking, the frontier days were over as we turned our eyes to new horizons.

We now live in an age of a new frontier. The birth of the Information Age was not unlike the opening of the gates westward. A vast expanse lay before us, and we were told to claim it and make it ours. It is a landscape so large that we cannot perceive its borders, and we cannot count its acres. Like those pioneering days, it's a place where there has been little law enforcement. Even today, decades after the first chat room was created in 1973, there is virtually no sense of a true presence of law among the endless global collective consciousness of the online world.

Truly, who could blame law enforcement for being slow to claim their place in this new world? It's a topic that baffles many people. Even the leaders of our society, whom we have

charged with passing legislation to protect our freedom, have virtually no understanding of technology. This is demonstrated when lawmakers attempt to explain the Internet as "a series of tubes," or describe a delayed email as "An Internet getting tangled up with all these things going on the Internet" as Senator Ted Stevens said in 2006. However, as technology becomes increasingly ubiquitous in our society it is becoming more difficult to ignore. A poll conducted in 2014 showed that while adoption of new technologies by those sixty-five years old and older still lags far behind the rate of younger individuals, it has increased exponentially from the rates previous years (Smith, 2014.)

This signals that the day has long since passed for law enforcement adoption of this technology. In many ways, there has been some progress here, but to a large extent the average officer has no special knowledge of how this technology works, and how it can be utilized for good or bad (Ashford, 2013). This occurs, while simultaneously that same officer finds their daily police work occurring more and more in digital format and using online communications. This means that the knowledge of these systems can no longer be the domain of a handful of specially trained officers. Each officer must possess a basic understanding of these tools, just as they must demonstrate a competency with their sidearm or issued vehicle.

What follows is a guide for guides. It's an accounting of the best ways to secure endpoints under the protection of a law enforcement agency, and an explanation of the layers of security that should be considered both by an administrator as well as an end user. The goal is not to create a document to train an officer, but to give the reader the tools and background that they would need to create and effective security policy and communicate that to the officers that become a part of that system.

Security from a Law Enforcement Perspective

Writing for The Police Chief magazine in 2007, First Sergeant Charles L. Cohen of the Indiana State Police wrote a compelling article on the troubles that arise when uninformed officers meet with digital evidence, giving the theoretically but all-too plausible scenario below:

> "**Defense Attorney:** 'Did you recover and preserve the volatile memory from my client's computer?'
> **Examiner:** 'No.'
> **Defense Attorney:** 'Is there a way to recover and preserve the volatile memory in a forensically sound manner?'
> **Examiner:** 'Yes.'
> **Defense Attorney:** 'Why did you not recover and preserve the volatile memory?'
> **Examiner:** 'It was lost when the computer was unplugged.'
> **Defense Attorney:** 'Who unplugged the computer?'
> **Examiner:** 'The detective.'
> **Defense Attorney:** 'Is it true that about one thousand books' worth of information can be stored in one gigabyte and that there were three gigabytes of information lost when this computer was unplugged?'
> **Examiner:** 'Yes.'
> **Defense Attorney:** 'Is it possible that the evidence of my client's innocence was among those three thousand books' worth of information?'
> **Examiner:** 'Yes.'
> **Defense Attorney:** 'So the detective destroyed the evidence of my client's innocence.'
> **Prosecutor:** 'I object.'" (Cohen, 2007.)

Even though the scenario above is fictional, it illustrates the importance for the average officer to be familiar, or at least comfortable with, technology. It also shows how a lack of understanding can be used to manipulate the facts. The fictional attorney presses the examiner on whether or not there could have been "three-thousand books' worth of information" lost. This is a phrase that could easily be said and accepted in real life. However, the assertion itself relies

upon an oversimplification of the truth. What is "three-thousand books' worth of information" anyway?

A book is an object finite dimension. Ten-thousand words, eight-hundred pages, one and a half kilos, a blue cover, smells like mold…. All of these are definitive and quantifiable, but which of those attributes translates to a single unit of "data?" How about the color blue – each pixel is 8-bits, but the number of pixels depends upon many factors. For the sake of argument, we will make it very simple, and say those three-thousand copies of a digital photo of a book (about 5mb each) requires 15 gigabytes of storage. This was only possible fairly recently (in chronological terms, not technical) with the introduction of x86 chip sets, and it's not entirely unusual in gaming or professional computers, but the average user is more likely to be somewhere in the 4gb to 12gb range. If we are only talking about the text, as raw as possible, with no formatting at all, then three-thousand text files of about book length would be only about 3gb. Well within the range of an average home user. However, if we wanted to store all quantifiable data to the point where we have all of the information we can possible have on those books, it would be far beyond the current physical limits of volatile memory storage.

This is the issue when it comes to the law enforcement perspective of cyber-crime. The law demands facts. While digital crime has no shortage of facts, they are often more complicated and less identifiable that traditional crimes. Every concept of most current law enforcement training programs is turned completely on its ear by modern digital crimes (Ashford, 2013.) For example, there isn't evidence to find, but traces of it. Information does not stop moving. If illegal data is discovered, the officer must understand and assume that it is constantly in motion, and could exist simultaneously in locations all over the globe, and in places that are nearly impossible to detect upon initial inspection. It may be encrypted. It may be obfuscated. It may

not be there with the owner's knowledge. For an experienced officer with years of traditional detective work under their belt, cyber-crime is like trying to catch ghosts. No one knows if they exist, and even if you could prove they do you wouldn't know what to do with them.

To combat this, law enforcement agencies are hiring computer scientists in an effort to on-board people with the familiarity and expertise necessary to fight cyber-crime (Ashford, 2013). This is a great step forward for law enforcement, but it still does not help the essential problem of law enforcement – Quis custidiet ipsos custodies? Who watches the watchmen? While police move out to create a presence on the Internet, who is protecting the police, who are most vulnerable to attack? A security system is only a strong as its weakest link. One of the most common and immutable attack vectors in any network is the person sitting behind the computer. Having all officers educated in safe technology use, and putting protecting in place on those endpoints is essential to covering that weakness in the armor.

Scope of the Problem

Cyber-crime is only relatively new in terms of law, but it has been around for decades. Despite this, crime statistics tend to evaluate crime more slowly. The Bureau of Justice Statistics, for example, has a publication available from 2005. Anyone could tell you that technology has exploded in the ten years since, and the number of headline-grabbing data breaches no longer sporadically dots the front pages, but now seems a ubiquitous addition to any news source. Cyber-crime against law enforcement agencies has become more common as well.

Between June and February of 2012, there were numerous attacks on law enforcement agencies. Agencies as high profile as the FBI all the way down to the Arizona Department of

Public Safety and the County Sheriff's office of Baldwin County, Alabama (DataBreaches, 2012). There's no question that law enforcement agencies, despite their role in society, as just as vulnerable as anyone. Perhaps even more vulnerable given the potential treasure trove of data that law enforcement networks could provide.

That is the main consideration when one considers the scope of the problem at hand. When determining risk, one considers not only the vulnerability of the asset, but it's importance as well. In terms of importance, even the lowest level of law enforcement agencies is many times more risky than most businesses. Police hold digital evidence, public and private records, and usually have access to other law enforcement networks as well. These connections could go to the highest levels, theoretically making the Baldwin's County Sheriff's Department the back door to the FBI. Just imagine the damage that could be done if a single determined attacker was able to infiltrate a law enforcement computer network, and just quietly listen, collecting all of the data that was transmitted to a single computer terminal.

For law enforcement, the scope is long and breadth is wide for data security. One journal article discovered during the course of researching this topic compared digital evidence to the collection of physical evidence of a special nature, such as DNA samples, in that "[t]hey have no obvious use or meaning until a criminalistics expert analyzes them and consequently determines their forensic significance. […] evidence which on the surface may appear meaningless but upon further analyses by computer forensic examiner might prove crucial in clearing a case." (Hiinduja, S. 2007.) The critical difference being that every human can generally recognize a hair sample or blood, not every person can necessarily recognize digital evidence. This is why basic training in technology is so important.

Security from an Attackers Perspective

Hackers are unique individuals. While modern times have brought about the professional hacker working within organized crime syndicates, for the most part hackers are primarily motivated by the challenge, while the monetary reward is secondary. These are bright individuals that generally do not relate well to people. As such, they have a proclivity toward idealistic political affiliations such as anarchy or libertarianism. They justify their activities, figuring that they are only helping to reveal holes in the security system, or are champions of right bringing justice to a vilified target. They see themselves as romanticized outlaws, and even use terms like "black hat" and "white hat" to denote their affinity for chaos or order.

Understanding the psychology of a hacker can be difficult. There have been entire volumes dedicated to the subject. However, for the uninitiated, there is little better place to start than with the document created by a hacker known as "The Mentor" in 1986 for the purpose of explaining exactly what motivated him. The so-called "Hacker Manifesto" is, in its entirety,

> *"Another one got caught today, it's all over the papers. 'Teenager Arrested in Computer Crime Scandal', 'Hacker Arrested after Bank Tampering'...*
>
> *Damn kids. They're all alike.*
>
> *But did you, in your three-piece psychology and 1950's technobrain, ever take a look behind the eyes of the hacker? Did you ever wonder what made him tick, what forces shaped him, what may have molded him?*
>
> *I am a hacker, enter my world...*
>
> *Mine is a world that begins with school... I'm smarter than most of the other kids, this crap they teach us bores me...*
>
> *Damn underachiever. They're all alike.*
>
> *I'm in junior high or high school. I've listened to teachers explain for the fifteenth time how to reduce a fraction. I understand it. 'No, Ms. Smith, I didn't show my*

work. I did it in my head...'

Damn kid. Probably copied it. They're all alike.

I made a discovery today. I found a computer. Wait a second, this is cool. It does what I want it to. If it makes a mistake, it's because I screwed it up. Not because it doesn't like me... Or feels threatened by me... Or thinks I'm a smart ass.. Or doesn't like teaching and shouldn't be here...

Damn kid. All he does is play games. They're all alike.

And then it happened... a door opened to a world... rushing through the phone line like heroin through an addict's veins, an electronic pulse is sent out, a refuge from the day-to-day incompetencies is sought... a board is found. 'This is it... this is where I belong...' I know everyone here... even if I've never met them, never talked to them, may never hear from them again... I know you all...

Damn kid. Tying up the phone line again. They're all alike...

You bet your ass we're all alike... we've been spoon-fed baby food at school when we hungered for steak... the bits of meat that you did let slip through were pre-chewed and tasteless. We've been dominated by sadists, or ignored by the apathetic. The few that had something to teach found us willing pupils, but those few are like drops of water in the desert.

This is our world now... the world of the electron and the switch, the beauty of the baud. We make use of a service already existing without paying for what could be dirt-cheap if it wasn't run by profiteering gluttons, and you call us criminals. We explore... and you call us criminals. We seek after knowledge... and you call us criminals. We exist without skin color, without nationality, without religious bias... and you call us criminals. You build atomic bombs, you wage wars, you murder, cheat, and lie to us and try to make us believe it's for our own good, yet we're the criminals.

Yes, I am a criminal. My crime is that of curiosity. My crime is that of judging people by what they say and think, not what they look like. My crime is that of outsmarting you, something that you will never forgive me for.

I am a hacker, and this is my manifesto. You may stop this individual, but you can't stop us all... after all, we're all alike."(The Mentor, 1986)

To an attacker, a law enforcement network presents a very attractive target. A soft victim, loaded with valuable data, and the fact that they are police makes it all the sweeter. Often, attacks on police departments occur in response to a police action, and the attack is meant to be a protest against the conduct of the department, or a single officer, or even just authority in general. In general, there doesn't need to be much cheese in the trap for criminals to go after law enforcement networks.

From the perspective of a defender, many security professionals worry about the ramparts and drawbridge. They put their time and effort into protections at the network level, to control and monitor the flow of traffic and an out of their network. However, attackers have the same general knowledge as any information security person. They are away of the technologies and techniques and they know that the weakest and most reliably exploited part of a security system is the only part of the system that can't be upgraded, hardened, mitigated or remediated: The human. In terms of data collection and processing, humans are strong. Our minds are capable of things computers will never be able to do. However, in terms of security, humans are weak. We do not handle randomness well, and so we are predictable. We feel we cannot rely on our memory, so we pick easy passwords. Finally, since attackers are human as well, they know humans just as well as they know computer systems.

The human element is the one an attacker will most often exploit. Whether that's taking advantage of predictable patrol routes, causing confusion over the phone and tricking someone into giving away important information or just sending out phishing emails and tricking someone into entering their credentials into a fake website, the target is the human sitting behind the screen. This is known as "social engineering" due in part to how human fallibility and ability to be manipulated is as consistent as math.

The route an attacker takes to infiltrate a computer system is known as the "attack kill chain" and it is essentially the seven generic steps that every attack follows. The idea is that each of these steps represents a potential turning point, a decision branch where the defender has an opportunity to detect and halt and attack or where the attack moves on. The first step in the kill chain is reconnaissance. It's the initial step where the attacker selects a target, and begins combing through social media, news articles, public records, or begins social engineering to get as much information about the system as they possibly can. At this point, the attack could be stopped by have employees that are savvy enough to not releasing anything potentially useful by spotting someone probing for information.

The second step is the lure. It's the point at which the attacker attempts to get the users of the system, in this case officers at end points, to make a mistake or otherwise unwittingly allow the attacker in. This can be done with spear phishing, redirects, watering holes, or any number of ways an unwary user could be enticed to do the wrong thing. The attack can be stopped here by educating users on safe technology usage.

Step three is the redirect, where the user has fallen for the initial lure. The user has erred, and at this point all attempts to educate the user and trust in human intuition has failed. The next lines of defense are entirely technological in nature. The attack has not yet infiltrated the network, because it has not infected the endpoint. The only hope is to have the endpoint hardened, and defenses in place directly on that attack vector. We must assume that all external-facing defenses have failed, because the user whom has been empowered and entrusted with their workstation has opened the door for them.

This document is meant to address steps two and three, or, how users can be educated and endpoint hardened to stop an attack dead in its tracks before it even infects that single computer.

We will discuss methods for this, including user entitlement, group policy settings, disk encryption, web content filtering, endpoint firewalls, and application virtualization.

Steps four through seven are, in order: The exploit where the attacker has gained enough insight into the system to check for vulnerabilities, such as a weak browser extension, and outdated version of an office program, or a zero-day available on a PDF reader. This document will discuss available technologies that detect exploit attempts before they can be utilized.

The next step is the dropper, where a malicious payload infects the machine. This is traditionally stopped with anti-virus software, but that single layer of security is often woefully inadequate. We will discuss how methods, such as application control, can often work more effectively than anti-virus at preventing infection, and how utilizing both is ideal security.

The next step is the call home, where the infected machine reaches out to the command and control servers, and reports to the attacker that a foothold has been established and further instructions are given. Here, we will discuss how endpoint firewalls and application control policies can be used to prevent this from happening.

The final step is the theft, where there has been a complete security failure thus far and the defenses have one last chance to prevent the attacker from walking off with the loot. In this document, we will discuss safe data handling practices, group policy usage, disc encryption, and entitlement and access control methods to demonstrate how an ounce of prevention could save a pound of cure.

PROBLEM OVERVIEW

Common Attack Vectors

An attack vector is a term that describes the point of ingress for an attacker. If we were talking about physical security (and, in part, we are) then the attack vector would be the upstairs window, or back door, or the old chimney. Likewise, a pivot point would be an attack vector that an attacker can compromise specifically to gain access to other systems, or to obfuscate the source of the attacks.

Law enforcement endpoints make ideal attack vectors. The end users, being the weakest part of security, unwittingly assist the attacker in gaining access to their system. It doesn't matter much how, but in this case we will say that the user was tricked into clicking a link. Since the agency has weak security, the end users had an inordinate level of permission on their workstation, as they have administrator privileges. The link leads to a rootkit, which infects the machine and gives the attacker ingress into the computer system. At this point, it's trivial to collect the user's credentials, which the attacker uses on the user's own workstation to do reconnaissance on the network, or launch additional attacks on other systems. Even other common attack vectors still utilize social engineering, or rely upon it. A type of software vulnerability attack vector commonly launched against websites is known as cross site scripting, which can use social engineering to trick one of the site's users into executing a malicious script upon the site, and doing their dirty work for them.

Just about every attack vector comes back to the human element. The only question is where in the long chain of events the error was made. For example, injection attacks are generally the result of developers not taking the time to properly test their code and validate responses. Brute force attacks, whereby an attacker simply keeps trying over and over again to

guess the correct username and password, are only successful due to the user not using strong passwords and the administrator's failure to limit login attempts. In the end, a computer is a tool. It only does or does not do what a human instructs it to do or not do. Any failure is ours, and my sheer law of averages the odds are good that a successful attack will come from the workstation of an end user.

Special Challenges

Law enforcement agencies face special challenges when it comes to preparing their personnel and infrastructure to protect against a potential attack. For example, law enforcement agencies are often in possession of information that they must, by law, retain for a certain period of time. Depending on the nature of the data, they may have to adhere to a retentions policy in excess of six years. That leaves agencies with a stockpile of valuable data. To make matters worse, while other organizations can demarcate their data into different categories in order of importance, law enforcement agencies must assume that all data connected to their operations are sensitive, since even something as seemingly banal as shift schedules or officer's personal contact information can be extremely useful for the recon stages of an attack.

Another special challenge that law enforcement agencies face is the need for portability. Officers don't always spend all of their time in the station, where security persons can set up multiple defenses and more easily control the flow of traffic in their network. Security solutions must be as mobile as the officers are, and this means thinking about security in completely different terms, with different tools and skills in play. Mobiles devices also mean they are more

susceptible to damage or theft. The answer for this in standard environments is usually cable locks or tracking devices. This isn't always possible for law enforcement.

This is a good point to introduce another information security term, because it is a component in another special challenge law enforcement agencies face. The "attack surface" is a term to describe the amount of non-standard or potential security risk software and hardware attached to an endpoint. Most organizations have a narrow attack surface that is broadened by permissive computer use policies. For example, a business that has endpoints that only officially runs the Office 2010 suite, but because of lax end user entitlement control, the users themselves have installed Spotify, Weatherbug, OpenOffice, Evernote, etc. These unauthorized installations broaden the attack surface, because now both sides of the security game have even more variables in play.

Law enforcement agencies may or may not have the same problem with entitlement control, but even in the best of cases there is still a much more breadth in the attack surface of any agency's endpoint, due to the use of specialized computer programs and databases that are necessary for the functions of law enforcement. It has been my personal experience with implementing endpoint protections to the campus Protective Services department at UWSP that these programs can be quite old, and are often so highly specialized that they are written by a handful of developers working specifically for the field of law enforcement. They may even be developed in-house. The problem with this is that this means the programs are often not often as rigorously tested for vulnerabilities. The age of a program can be highly detrimental to its security. To make matters even more complicated, the aforementioned data retention requirements might even mean that a vulnerable program cannot be removed because it is

necessary to retrieve that data. This presents a weakened position for defenses, but not insurmountable.

One last example of a special challenge is the importance of interconnectivity between agencies. Law enforcement relies upon sharing information. For example, if the Portage County Sheriff's Department shares information with the Marathon County Sheriff's Department, then they are able to enforce the law that much better. Furthermore, there are different levels of law enforcement, and each must communicate vertically as well as horizontally. Some agencies may have a repository of data that other agencies rely upon for information, such as DNA databases, most wanted lists, demographic data, DVM records, etc. Without access to these systems, a law enforcement officer cannot do their job. Unfortunately, that very access is extremely attractive to an attacker. Attacking the DMV database directly might be a challenge. An attacker would be facing all of the security and defenses in place around that network. However, they know that the DMV has made an exception in their security rules to allow law enforcement agencies access to it, and the law enforcement agency's security is much weaker. This is all the attacker needs to begin recon on the law enforcement network.

ENDPOINT SECURITY

It's time to begin discussing the best practices for securing an endpoint from attack. However, the real challenge with describing the best way to go about such a topic is knowing where to begin. For the ease of the reader, this section will be cordoned off into logical topics. Keep in mind that this is not to imply that any one element is more important than another. A security system is successful because it has layers. Without one of the elements, then the system as a whole is weaker.

Dr. David Carter of the School of Criminal Justice for Michigan State University authored a document for the U.S. Department of Justice that laid out a guide for law enforcement intelligence. In that document, he identified the key components to information technology management. This document details effective endpoint security practices that cover all of them. Those elements are:

1. Manual Assurance of Data Handling: Virtually all data is handled manually at some point. There must be security standards and quality control of data that is entered into the system.
2. Physical Security: There must be effective measures in place to ensure the security of the facility housing computer(s), servers, and any other related hardware (e.g., PDAs) and peripherals (e.g., printers) that have access to the system.
3. Operations Security: Processes for quality control of personnel who are system operators/managers as well as security monitoring of people who have access to the secure area where computers and servers are housed. This includes maintenance and custodial personal, clerical personnel, and others who may have access to the secure area.
4. Management-Initiated Controls: This includes…
    - Management oversight of system operations.
    - Administrative policy for computer access and use.
    - Fair use policies if a public website is provided.
    - Establishment of data security management policies.
    - Establishing data classification protocols for control and access.
5. Computer System Control: Strict access to the system should be controlled by:
    - Authorization of personnel. Defining policies and standards as to who may have access to the system, for what purposes access is granted, and defined standards of acceptable use of the system.
    - Software access controls. 81 Beyond standard user name and password controls, and all the well-known security precautions associated with

these, the system should be protected by a Virtual Private Network (VPN) for access control by authorized users.

• System protection and inoculation. All networked systems should have a multistage firewall and constantly updated virus definitions.

6. Encryption for wireless devices: Network encryption should be enabled if wireless devices are used with an intelligence records system or intelligence-related communications.

7. Access audit controls: A real-time auditing system should monitor all accesses to the system, user identification, and activity during the user period, length of time, and IP number of the computer accessing the system.

8. Control of remote storage media: Policies need to be established and technological controls instituted to monitor the use and control of restricted data related to remote storage media (e.g., disks, CDs, thumb drives, etc.). (Carter, 2004, pp 90-91.)

Description of a Common Sample Environment

As part of his duties working for the University of Wisconsin – Stevens Point, the author of this paper worked closely with end users and the Information Security department to develop what would eventually become known as the "Protected Campus Load." This was a standard machine – Window 7 Enterprise x64, connected to a standard campus domain utilizing Active Directory management. The base image of the machine, the "Standard Campus Load" contained common productivity software, such as the Office 2010 suite, an AppV client, the Alertus desktop alerting client, and the necessary enterprise tie-ins for SCCM and other centralized management tools. No third party browsers were deployed or supported, due to the inherent nativity of Internet Explorer. For security, browser updates occurred, generally, thirty to sixty days after release. However, there was no enterprise level policy against third party browsers,

and a version of Mozilla Firefox was even made available through the user self-service portal, known as the UWSP Application Center. Mobile devices were iOS and Android, and contained no special ROMs or configurations beyond OEM. Mobile device management was facilitated by AirWatch, but the basic rules in place are available across any competitive Mobile Device Management platform. Network assets at the server level were Windows platforms, running 2003 to 2012 r2, with supporting hardware from Cisco.

The initial purpose of the project was to secure the endpoints for "business areas" of the university, like the bursar's office or accounting. Ostensibly, this was to protect the university's financial information and to maintain PCI compliance. It proved to be very effective, and protected these endpoints form a series of zero-day attacks and malware outbreaks. In general, it was very easy to deploy, operate, and maintain. This was the case for every department we deployed the project in, until we entered the Protective Services department. This department was the campus police force, which from a technology standpoint was fairly modern since they had the same standard equipment as any department. The challenge then was not a technical one, but a practical one based in the special challenges that come with protecting law enforcement machines. My experiences here were the impetus for creating this document.

This is the most standard configuration of any other environment. According to Net Market Share, Windows operating system versions make up over 91% of the current market share for desktop platforms.

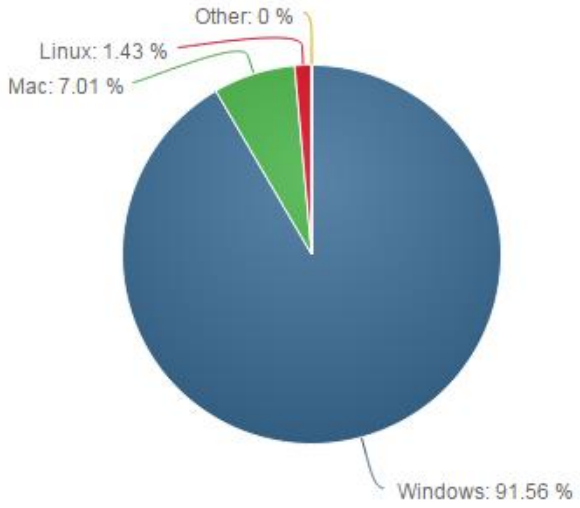

Figure 1 - Current trends among desktop operating systems (www.netmarketshare.com)

Among mobile devices, Android is the current leader with over 46% of the market share, with iOS relatively close behind with over 42%.

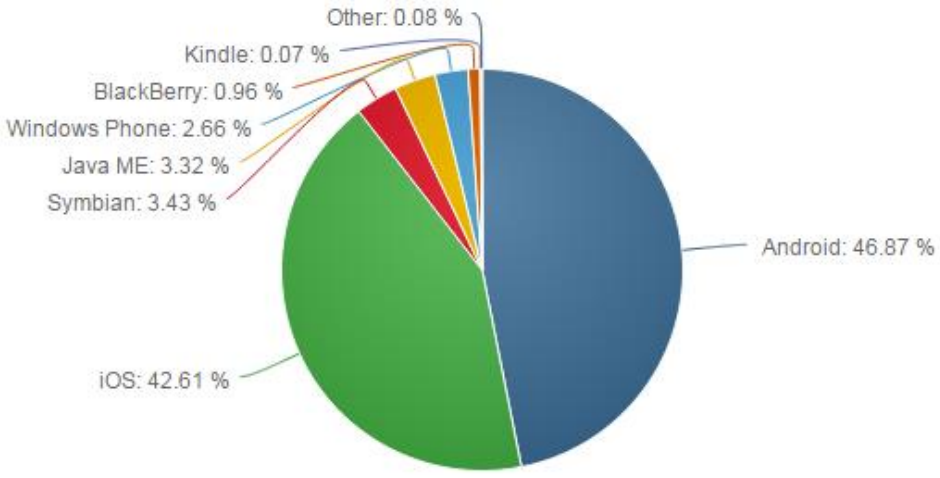

Figure 2 - Current trends among mobile operating systems (www.netmarketshare.com)

For server assets, about 30% of webservers are Windows-based, but the network infrastructure based upon the Windows platform such as DHCP and Active Directory controllers is as standard as the Windows operating system on desktops, with studies from the Enterprise Research group estimating that as many as 71% or more of all production servers that can be measures are based upon a version of Windows. Further studies show that the fewer server assets an organization has, the more likely they are to be Windows-based, with larger organizations simply increasing the number of Linux/Unix systems.

You'll find domains like this in every field from finance to government, and from retail to non-profit. While larger organizations tend to use more Linux/Unix systems, and the most powerful computers on the planet run Linux/Unix based systems, the reliability and familiarity of the Windows environment, which is tailored to integrate so easily with other Microsoft products, is still generally the OS of choice for server assets for organizations of any stripe. Chances are the environment of a law enforcement agency is very much like this. As the following chart shows, the majority of environments for governmental organizations (including law enforcement agencies) are Windows based.

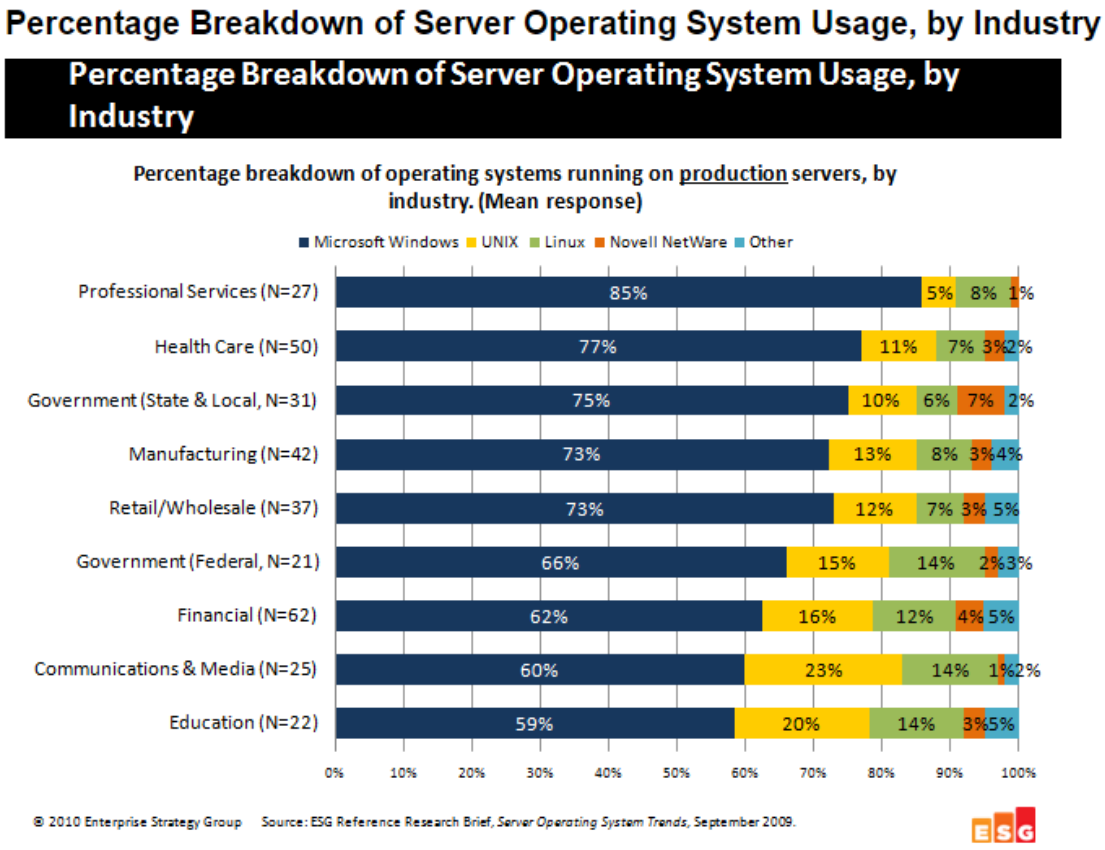

Figure 3 - Server operating system trends by organization type (Enterprise Strategy Group, 2010.)

For that reason, the slant of this document is geared toward endpoint protection for Windows machines in a Windows environment. However, the principles put forth below are not limited to any operating system. The actual specifics of your environment are less important than the lessons the author learned while developing that project. In essence, regardless of what you have available, there are still the same concerns and considerations, and the same general approach to security. These considerations boil down to what the author refers to as the five pillars of endpoint security. They represent layers of security that protect the machine from attack, and prevent an end user from making a costly mistake.

The five elements of effective endpoint security are: First, exploit mitigation, utilizing products such as firewalls, anti-virus software, application control policies, and the Enhanced Mitigation Experience Toolkit from Microsoft. Second, content filtering, utilizing web content filters, internet proxies, application virtualization, and endpoint firewalls. Third is physical security, with disk encryption, access control devices, and sound access security policies. Fourth is entitlement control, using sound security practice when granting rights, handling administrator access, and using group policy to secure endpoints. Finally, user participation, where the most important task is to educate the user and provide them with the tools they will need to do their part to maintain a secure environment.

Non-Interfacing Protections

A good security system is one that doesn't require a lot of intervention. A person might forget to bolt a door, leave their access device at home, or neglect to select a password strong enough to prevent a brute force attack. Unfortunately, since computers are primarily tools for people to use, there are few things that don't require any intervention. However, there are a few very effective measures that a security professional can put in place that will harden an endpoint without the user even being aware that they are present.

Disk Encryption

One of the biggest problems law enforcement agencies face is the issue of physical security (Anderson, 2007.) Laptops, smart phones, mobile devices are all carried into the field when an officer leaves their headquarters. Even the best security is generally rendered moot

when the device is physically stolen, and at that point the device is considered entirely lost. However, that does not necessarily mean that the data is vulnerable.

Disk encryption makes the data on the device unreadable unless the operator has the corresponding key to decrypt it. These days, with encryption methods like AES 256, the hopes of a hacker brute forcing the device to decrypt, or cracking the encryption are virtually impossible. In the case of Windows devices, there is an encryption technique available built into the operating system, known as BitLocker. While any encryption technique of sufficient complexity is sufficient, what sets BitLocker apart is the fact that it is disk encryption with hardware as well as a software component (Microsoft, 2015.) Encryption systems that are purely software based all will suffer from the drawbacks inherent in software-only encryption – they can be beaten by reverse engineering or exploitation. The hardware component makes that more difficult, because an attacker would have to beat the hardware as well as the software.

BitLocker is a disk encryption feature built into the Windows operating system, and is thus fully supported by Microsoft. This feature is fully integrated into all modern Windows platforms, including server and mobile. However, this does not extend to all versions or the product. Only the Ultimate and Enterprise versions of Vista and Windows 7 (Server 2008+), and the Pro and Enterprise editions of Windows 8/8.1 (Server 2012+) contain BitLocker integration (Microsoft, 2015.)

Cryptologically, it uses AES (128/256) by default, with a cypher-block-chaining algorithm combines with an elephant diffuser in a method developed by Microsoft technicians. It has been actively in use since 2006 (Microsoft, 2015.) In addition to this, the technology is more robust than typical encryption measures because it is not only software-based, but also has a hardware component in the form of a TPM (Trusted Platform Module) security chip built into to

every BitLocker-capable device. The TPM chip can both bind a private key to prevent any access to the device at start-up, and it can also seal the key. Sealing works by comparing hashes on critical OS files (known as PCRs – Platform Configuration Registers) and unlocking the drive at start up. Note that BitLocker CAN be configured to encrypt a device without utilizing the TPM chip, but this method makes it vulnerable to software-based attacks just as any other software-only solution.

In terms of general security, BitLocker is a sound solution, but it is not entirely unbeatable. The inclusion of the hardware component and flexibility of implementation bring about inherent risks to security. For a start, TPM chips do contain storage root keys that can be extracted from the chip in a difficult process that has been in use since 2010. However, this process requires physical possession of the device. Also, configuration of the technology can allow the user to enter a passcode to unlock the device at boot. This solution requires a person to make the necessary adjustments and therefore it is one that can be exploited. However, these issues are minimal, considering that they both deal directly with human interaction with the security protocol, and hence will be an issue with any solution. In the case of BitLocker, Windows devices set up for BitLocker use (generally any device with Windows Vista to 8.11 installed) will come with a TPM chip installed and a BIOS that is TCG (Trusted Computing Group) 1.2 compliant.

As mentioned, BitLocker as well as any other effective encryption system makes use of a diffuser. This is essentially a protocol that “mixes up” the data prior to encryption. The encryption algorithm ensures the data is unreadable without the key, but the diffuser ensures the data is unusable even if decrypted (Microsoft, 2015.) An effective encryption technique should

use a diffuser with a separate key from the encryption component, as any use of a shared key can be a concern for security as well as implementation.

Speaking to the administrative aspects of the technology, Bit Locker's integration with the operating system makes it a highly controllable solution. Utilizing administrative rights on an individual per-machine basis, or organization-wide rules set forth by group policy are both viable options. The technology is flexible enough that it can be configured in multiple ways to meet demand and to fit into an organizations workflow. The technology is flexible enough that there is very little excuse not to use it unless you have another method available, since it can be seamlessly integrated into the user experience.

Any other encryption method in place should use BitLocker as a template for an acceptable alternative. Disk encryption protections should, whenever practically possible, confirm to the following standards: First, at minimum, it should us AES-256 bit encryption. It should have a hardware as well as software component. It should have secure storage of decryption keys and PINs, and both should be set to minimum complexity requirements consistent with a strong password policy, which will be detailed later in this document. Fourth, it should be centrally manageable to ensure consistent application. Finally, from a technical perspective, the encryption method should have effective tamper-protections inherent in its implementation, such as diffusers that utilize a separate encryption key than the AES, platform configuration registers, and an encryption method that contains a self-destruct protocol upon tampering is highly desirable for mobile devices or endpoints that handle (but do not warehouse) particularly sensitive information.

Application Control

There are few parts of endpoint security more critical, useful, and delicate than application control or software restriction policies. Not only is it the absolute last line of defense again malware, but it is also an important way to ensure that the endpoint does not run any non-standard software that would broaden the attack surface.

Application control is, simply, the use of technologies that prevent an executable from running under a certain set of conditions. These conditions vary. It could be a rule that states that no binary may execute out of a specific directory. It could be the banning of binaries of a certain name, type, from a specific publisher, or with a specific hash. This is a powerful feature, because if one assumes proper entitlement control, an end user will not be able to install programs. Application control also ensures that the end user can only run programs that have already been installed and are approved by the system administrators. For example, an endpoint that is primarily for visiting a web page to enter in time sheet information could be restricted to only Internet Explorer x86, and with web content filtering any other website could be blocked. The administrator has effectively turned this endpoint into an appliance with as single purpose, and any attempt to run another program (even one as simple as notepad) will be denied.

Application control policies are also a powerful tool against malware. Generally, malware executes under the user's context, and usually from the user's temporary directory under application data. By writing a clever set of rules, a security professional can lock down this directory to prevent the running of unauthorized programs, and prevent the initial infection entirely, without sacrificing any usability for the end user.

The project the author of this document developed utilized another tool that is integrated into the Windows enterprise-level platforms known as App Locker, but software control policies

have been common in Windows environments since long before Vista, and there are plenty of similar products available for other operating systems. Again, whichever product you are utilizing should meet the minimum requirements, and the benchmark for that is App Locker.

Its strength is that it, like Bit Locker, is centrally manageable through group policy. It is capable of explicit and implicit denials and allows, and supports a wide range of rule sets based upon the meta data associated with the binary, including exceptions to the rules (Microsoft, 2015.) This data includes not only publisher, but version, location of origin, and date modified to make a rule to circumvent with false data to such a point that it would be much easier to attempt to compromise the legitimate application. App Locker is also especially hard to work around. Disabling the protection requires more than just stopping the application control service running, as the service needs only to be started to initiate the rules (Microsoft, 2015.) If the service is stopped, the rules are still well in place. This means that for a rule set delivered through group policy, an attacker aiming to disable the application control policy would have to stop the service, but would also have to override a domain-level policy.

Another hallmark of a good application control policy is the ability to audit. The main problem with restricting applications is that eventually a legitimate program will be blocked. Good audit controls ensure that you can properly trace the source of an error, as well as examine binaries laughed as secondary services to other applications, like for Citrix GoTo Meeting has a single binary which itself executes a litany of other binaries to create the virtual meeting space. This is, of course, also quite nice for documenting malware modalities, as there is a visual record of the efficacy of your protection. With this auditing, you can ensure that the policy is acting effectively, and when properly implemented should be blocking up to over ninety percent of all malware type found in the wild.

An effective rule set is deceptively simple. App Locker even comes with a default rule set that has three simple rules – 1) Allow everything in the system directory (C:\Windows, for example) to allow the user to run system-critical functions, such as a manual policy update, or PowerShell (for scripts executing under the user's context.)This is ok, because a standard user should not have access to save anything to the system directory. 2) Allow everything in %Program Files% because a standard user does not have the ability to install anything here, and so it is assumed the programs here were put in place by an administrator. 3) Allow administrators to run anything. These are the default rules, but in addition, an effective application control policy would add: 4) Disallow any binary from removable media, such as a flash drive. This means that if the user needs to run anything, it must be manually copied (Microsoft, 2015.) This simple inconvenience can stop a lot of malware that will auto-run, and prevents the endpoint form being infected just because the user has being taking work home with them back-and-forth. 5) Disallow user executed binaries in their own profile. By default, a user pretty much has total control over everything in their own profile, but this includes a lot of locations that most users aren't even aware exist and pose potential security risks.

As mentioned above, the primary location in question is the user's %AppData% folders. However, a blanket disallow won't work because there are legitimate programs that make use of this location, which amounts to about any that wills tore personalized application data. The key then is to principally disallow the execution of binaries at the root of these directories. For example, in the case of App Locker, there would be four rules. First, disallow all binaries from the root of the application data directory: C:\Users\user\AppData\ but create an exception for subdirectories with wild cards, making the exception path C:\Users\user\AppData**. Then three more rules are created to cover the root of the three primary location directories, Local,

LocalLow, and Roaming. We do this by making the same rule and exception as above, but for each of them specifically. Since App Locker processes rules in an order of precedence that prioritizes explicit denials over explicit allows, and implicit denials of implicit allows, that gives us a rule set that locks down AppData, AppData\Local, AppData\LocalLow, and AppData\Roaming without interfering with any application storing or using the application storage directory.

This rule set can be made even simpler if your environment is strictly controlled. If you have a standard set of software, then you don't need to make room for the unknown, and you can completely disallow all binaries form AppData except for those you specifically know to be acceptable and required. Whichever route you go, these five simple rules are your primary minimum for controlling applications on the end point.

Web Content Filtering

Whenever possible, the best thing to do is to keep your endpoints from reaching the Internet. If every workstation is only connected to an intranet, then your ability to protect the machine is made much easier. However, this is scarcely practically possible in this age, and so eventually at least a few of your end points will need to be protected from the Internet. The other methods documented here are effective ways to do that, but even if you can't keep the endpoint off the Internet entirely, it's still prudent to control the traffic between the computer and the world by restricting the content the end user has access to.

Despite the fact that Internet Explorer has a reputation for being one of the least secure browsers available, it does have one important feature that the competition does not. Namely, it

can be centrally managed with group policy. While other browsers will support various content filtering methods, on the integration of Internet Explorer into the operating system gives administrators the ability to truly enforce the content filter policy while also ensuring the end user (or anyone else) cannot simply turn it off as soon as they log in.

In addition to the obvious benefit of enforcing the acceptable use policy of your organization, web content filter solution should also contain the important benefit of logging, which monitors the web traffic of users, so that you can use that information to establish trends if there is a possibility that an attacker has set up a watering hole. A watering hole, being a euphemism for a popular web site or location that many of your users visit for work or social activities, is a method an attacker will use to trick many of your users into compromising the security of the network.

There are a lot of options when it comes to web content filters. There are numerous third-party applications, and there are even multiple methods built into the browser or operating systems, such as the RATs system (a voluntary content-rating system for websites.) However, for a robust enterprise solution that requires the smallest monetary investment for the best return on security there are few better options than an internal proxy filter.

A proxy filter is a separate server that handled and evaluated web requests. A short script, known as a "proxy auto-config" (PAC) file is created. It's a very simple script that essentially takes the web request, evaluates it, and then either allows the user directly through, or sends it to the proxy server for filtering. A very simple version would look something like the following:

```
//*** Sample Proxy Auto-Configuration Script
function FindProxyForURL(url, host){
//Set Proxy Servers
var myproxy="LEproxy.FBI.gov";
//Loopback and localhost are always direct
if ((host == "localhost") ||
   (shExpMatch(host, "localhost.*")) ||
```

```
    (host == "127.0.0.1")) {
       return "DIRECT";
}
//Exceptions for Direct connect
if (isPlainHostName(host) ||
    dnsDomainIs(host, ".FBI.gov") ||
    dnsDomainIs(host, ".whitehouse.gov")||
    dnsDomainIs(host, ".nyc.gov/NYPD")){
         return "DIRECT";
}
// Default return condition is the proxy assume everything
else is on the internet
return proxy;
} //End of proxy config
```

The above sample can be broken down essentially as thus: We simply begin the script and set a variable for our proxy server, which in this case is LEProxy.FBI.gov. We make exceptions for localhost and loopback, because we don't want those handled as normal web requests or evaluated by the remote server. We then make exceptions for the websites our agency deals with a lot, and whom we trust implicitly to be safe and secure. In this case, those include three domains, FBI.gov, whitehouse.gov, and the NYPD website. Finally, anything not explicitly excepted is treated as a request to the Internet, and is sent to the proxy server. In the event that a proxy server is not available, a local proxy can be set up on the endpoint with the use of programs like Squid.

There is a caveat with the usage of proxies for web content filtering, in that they all suffer from one inherent issue. Every website that is returned "direct" (which does not go through the proxy server) is treated as a trusted website in terms of internet security zones. This means that you must be highly restrictive of the content that you select to be trusted, because if any of them are compromised, it will eliminate any inherent security in your endpoint's browser, and render the proxy filter moot.

End Point Firewalls

Many security professionals consider a firewall at the network level sufficient. They concentrate on preventing things form getting into the network, and completely overlook the fact that there is a back door at every single workstation connected to it. Endpoint level firewalls are important tools, and if one is so inclined can also be used to support your web content filter by blacklist or whitelists blocks of IPs that you know your end users will never need access to.

There are many third-party firewall products out there. Many of them perform much better than the native Windows version, Windows Firewall with Advanced Security. However, the native firewall application wins out in my opinion because it is both free and integrated into the operating system, but also because it is centrally manageable through group policy. This keeps all of your rules in one place, making management much easier. Since the primary drawback of Windows Firewall with Advanced Security is the inability to utilize wildcard characters in any path rules, your usage of it is really dependent upon how important that feature would be for your environment. Typically, this would not be a major drawback for organizations, unless you are leveraging the firewall to perform user application control, or you have standard software that executes from the user's profile directory. In those cases, since the firewall executes and runs under a System context, you would need the wildcard to ensure that the rule applies to other users' contexts.

Since a firewall is principally a networking appliance, whatever your firewall product, the final product should be about the same. By default, the Windows Firewall is set to automatically disallow any inbound traffic not specifically allowed and to automatically allow and outbound traffic not specifically denied. A better security practice is to automatically deny both, and to create program exceptions for the standard software you know the user will need to be running.

This ensures that traffic flow is controlled on the endpoint, and would also ensure (along with application control) that an unapproved browsers or malware are unable to communicate across the Internet. In most standardized environments, this will be a perfectly acceptable and unobtrusive way to greatly enhance the security of your workstations.

Another way the end point firewall can be employed is as a second layer of defense to your web content filtering policy. The firewall can be set, as above, to block all inbound and outbound traffic, and then an exception made for the IP block owned by a certain domain. Some firewall products will even allow domain name whitelisting or blacklisting. This ensures that even if the web content filter fails to prevent traffic to an unacceptable website (via tampering or error) then the firewall still would not allow traffic to that IP address or domain name.

Removable Device Restriction

Removable media, such as flash drives and DVDs, present a difficult conundrum in system security. On the one hand, these products are important tools for productivity, and allow an agency to safely share information in a way that keeps the data from being transmitted across your network. On the other hand, if a flash drive becomes infected and is shared around to different workstations; it could quickly spread an infection. For this reason removable devices should be strictly controlled.

As mentioned above, application control policies can be used to disallow the execution of any program running from any drive other than the system drive, but it is always better to add another layer of protection whenever possible. In our sample environment, for example, group policy can be used to only allow removable drives if they have been encrypted with BitLocker, or even disallow any external device completely including unauthorized input devices.

As a technology itself, removable devices can actually be effectively used for good security. For example, if redirected or roaming profiles are enabled, then the user's profile never sits on the endpoint, but rather in a better secured server location. Then by using flash drives to save any local work, the device can simply be unplugged form the machine and locked in a safe at the end of the workday, and there will be no data left on the workstation itself. In fact, the entire operating system can be loaded from a flash drive, and with the use of those same redirected profiles, all data and the entire operating system can be removed and secured, and the workstation itself is nothing more than the hardware that it temporarily uses. While this makes for excellent security, it's not practical for organizations of any appreciable size, as it also carries a high risk that the flash drives could be cloned or stolen by an internal source.

Application Virtualization / Isolation

The use of application virtualization as a security tool is not one that is often discussed. Ostensibly, application virtualization products are meant to be leveraged to make the job of the workstation and endpoint teams easier. For example, Microsoft's AppV product makes it possible to offer applications to users as virtual packages, which the user can then run as if it were installed on their local machine, but which does not have all of the overhead of an actual installation. The application runs in a virtualized space, which uses the resources of the computer, but runs in a different space commonly referred to as a "sandbox"." When the application ends, some data is retained, but the space the program ran in is destroyed and the resources returned to the local machine.

The reason that these products are not generally considered for security applications is because while products like AppV offer virtualization they do not offer isolation, which is the

key to using it as a security product. For all intents and purposes, the programs that run through product like AppV act very much as if they were actually installed, meaning that even through the program is running in the sandbox, it can still access or be accessed from the machine. There is nothing preventing data from leaking into or out of the sandbox. However, with some products that do offer isolation as well as virtualization, such as Invincia or Cameyo, programs can be virtualized and run in an isolated session. To the user, it seems normal, but behind the scenes at the operating system level the program is running in a completely separate space. At the first sign of trouble, the session is destroyed, the information in the sandbox dumped, and the program restarted.

Overall this makes for a very effective way to give users the flexibility they need, while providing a measure of security. For example, malware infections commonly originate in web browsers like Internet Explorer or Firefox. We are using web content filtering to try to limit that, but for users that need to surf the web without restriction, a virtualized version of Firefox could be offered. With the proper configuration and other security measures in place, the risk inherent in unfiltered web browsing can be mitigated to an acceptable degree.

Exploit Mitigation

If your environment makes use of virtual applications, then an exploit mitigation tool is a very nice way to compliment that security layer. If you do not make use of virtualized applications, then an exploit mitigation tool is an absolutely essential part of an effective security platform. Though our modern computers run many different applications that serve many different purposes, the methods that an attacker uses to exploit those vulnerabilities are no necessarily novel from application to application. Techniques that detect and mitigate a potential exploit before it has a chance to compromise a program have been around for more than a

decade. Over that time, those techniques have become very effective at preventing attacks and have become so indispensable that they are integrated into many operating systems. These systems include those of the Windows platform.

However, since these techniques can occasionally cause problems running legitimate programs, these protections are turned off by default. To make use of them on Windows machines, one needs to make a series of registry changes or install the management tool form Microsoft. That tool is known as EMET (Enhanced Mitigation Experience Toolkit) and it is essentially a small client that monitors running programs for a number of exploit methods. The administrator can configure the client to determine which programs are protected or not, and which methods of exploit are monitored or not. It is also centrally manageable through group policy, like all other protections in this document.

EMET, or any other effective exploit mitigation tool, monitors for exploits and offers at least two types of protection. Those two protections are DEP (Data Execution Prevention) and ASLR (Address Space Layout Randomization.) The way these work can be a highly technical explanation, but DEP essentially monitors sections of memory that do not usually contain code, such as the stack or heap, and if an attack that attempts to inject and execute code in those areas is detects, the tool kills the applications process immediately before it has a change to execute. ASLR is a protection that makes it more difficult for an attacker to predict which address an application's module will be loaded in. Being able to predict that address space means that an attacker could inject code into that location, and have the application launch it for them under the user or system context, and ASLR does not stop an attacker form doing this, but makes it much more difficult for them to do so. Neither methods are unbeatable, but they do make it more difficult and more costly for an attacker to persist. At a minimum, DEP and ASLR should be

used together in an exploit mitigation tool for all critical or potentially vulnerable programs, such as web browsers, word processors, PDF readers, and Java applications.

Microsoft's EMET provides a number of other possible protections, which make it a more effective tool. Any other tool considered may include these. These protections include buffer overrun detection, and SEH (Structured Exception Handling), which are two of the more common vulnerabilities. Buffer protection means that the temporary memory an application is using is periodically checked for validity by measuring a known quantity of space or checking generated numbers.) If the check fails, the application is terminated. Exception handling is more straight-forward, which enables contingency plans for when an application is instructed to perform an operation that was never accounted for, or is impossible (like a divide-by-zero.) Generally, the contingency is to terminate the application, as it is the safest way to handle a potential exploit.

Of course, all of these protections are meant to support poor application programing. Microsoft has a list of exploit protections a developer can include when creating their application, which they have put in order of importance. As a security professional, you should not assume that a developer has enabled these protections, and make sure that your exploit mitigation security layer accounts for the first two protections after the C++ Compiler Version on the following list:

Defense	Priority
C++ Compiler Version	Critical
Address space layout randomization opt-in	Critical
DEP opt-in	Critical
/GS stack-based buffer overrun detection	Critical
/SAFESEH exception handler protection	Critical
Function migration	High
SEHOP opt-in	High
Pointer Encoding	Moderate
Heap corruption detection	Moderate

Figure 4 - Windows ISV Software Security Defenses (Howard, et al., 2010)

Sensitive Data Protection

You cannot protect data if you don't know where it is. Data has a tendency to spread. When digital data can be saved, copied, cut and pasted, duplicated, and archived it becomes very important to keep track of where that data is stored. It's not at all uncommon for an end user to access sensitive data from a secured location, but then inadvertently or intentionally save it to another location such as their local drive or to a removable device. Over time and without proper controls, this means that sensitive data can end up in unlikely and insecure places. The best way to protect this extraneous data is to not have it sitting around in the first place, and that requires a sensitive data protection policy with a tool known as a DLP (Data Loss Prevention.) This tool

can come in several forms depending on your goal, but in general DLP scans for sensitive data in three states – when it is in use (when someone access the data,) when it is in transit across your network, and when it is at rest (stored on a device in scanning scope.)

Technically, the term "Data Loss Prevention" is a fairly generic term for any technology or device that protects sensitive data in those three states. For the purposes of this document, the term is much narrower and is being used specifically to identify products that scan for sensitive data by checking the file's contents and matching it to known regular expressions to determine if it contains data that the administrator wishes to protect. For example, the free software Spider4 is such a product. It scans for credit card numbers, passwords, social security numbers, and allows the administrator to add custom expressions to look for specific strings. It works quite well, but is really more of a detective tool because it does not offer features that would be critical in a broad environment.

The Data Loss Prevention systems can also be used simply for monitoring. Even when an officer or other authorized individual accesses confidential data, it is wise to keep a record of it, and these systems can be configured to audit who, when, and from where data was accessed or altered. This feature alone can make your DLP solution one of the most important aspects fo your security system.

Identity Finder is a product very much like Spider4, but offers those missing features. It allows for the creation of organizational units of endpoints, upon which the administrator may apply separate policies that scan be configured to scan for different data. For example, an administrator could put all of the computers in the payroll department in a group that scans for bank account and credit card numbers, but put the rest of the department's computers in a group that scans for social security numbers, dates of birth, or case numbers. Any product that offers

these features, as well as the ability to schedule scan, deliver reports, and create automated workflows as Identity Finder does should be considered the benchmark for a worthy sensitive data protection system.

Another issue to consider in the area of sensitive data protection is the increasing reliance on mobile devices. In an age where everyone is carrying a cell phone or tablet, and content that includes sensitive data can be delivered wirelessly on command, the need to manage the data on these devices is stronger than ever. The fact that they are mobile devices and that the two most popular mobile operating systems are not generally easy to integrate with Windows environments presents a different sort of challenge for the security administrator. If the device cannot be scanned for sensitive data, then a system needs to be put in place to remove that data in the event that it is lost, stolen, or compromised. This is facilitated with a good mobile device management platform.

If your organization uses any mobile devices, then it should have a good mobile device management (MDM) platform. It is the best way to manage the devices, active like an Active Directory for cell phones and tablets. A good MDM system will include several very important features, like the ability to track the device using GPS or WiFi, the ability to monitor it for application installation, the ability to create compliance policies, and most critical to the current subject at hand it should also include the ability to do an “enterprise wipe.” That wipe is a term for removing any content from the device that was delivered through the MDM platform. Ideally, it would also include the option to do a complete factory reset remotely. A good mobile device management platform like AirWatch, combined with secure operating systems like SAFE or KNOX devices from Samsung, can make for a very effective mobile device security system, though details on that topic are outside the scope of this document.

Anti-Virus

The last line of defense for an endpoint is the anti-virus program. Many organizations rely entirely on the anti-virus being effective enough to mitigate any potential risk to a machine, but the truth is that any anti-virus program on its own is not very effective at protecting a computer. These programs rely on "definitions" which is essentially a small file that contains information on a particular type of malware, which the anti-virus uses to check files and determine if it is a legitimate file or potentially harmful. In a way, it is much like an inoculation, and just like a physical viral infection an inoculation will often only protect you from that particular strain of virus. Obviously, the main weakness is that a malicious payload must be detected, a definition created, and then distributed to each endpoint in order to effectively block malware. This means that there is an average of 4-6 weeks between when a virus is seen active "in the wild" and when anti-virus programs will be able to protect a machine from it.

Modern anti-virus solutions so have some limited ability to detect "unknown" threats, or viruses that do not yet have an available definition. These types of protection are known as "heuristic" or "behavioral" protection. While some product handle this better than others, there is again no one company that does this much better than another. In any event, relying on these protections against 0-day attacks (i.e. attacks that are new, or which the software's programmers have zero days to fix the issue) is not a sound plan, as even the products that tend to handle these events best will miss a few. On average, most products will catch around 80% of unknown malware. The current industry leader in heuristic protection, Bitdefender, still only caught 98% of attempted attacks in a recent test. This sounds impressive, and it certainly is, but the need for additional layers of protection is still apparent when you see that 98% translates to the program

failing 20 times over the course of the test, and that the antivirus would not have had to protect against 1200 attacks with other proper security measures in place.

There is also much debate over which anti-virus program is "the best." There is so much debate that entire organizations devote themselves to measuring the efficacy of each one, and post their results in the form of a report. Looking through the historical record of one such website like AVComparatives, it becomes clear that there is no real obvious "best" as definitions generally come out at the same time and are obtained by each equally. There are some anti-virus programs that are certainly worse, but generally speaking the data shows that as long as you pick a solution with a good reputation, the result is about the same. The main issue with making sure your anti-virus solution is effective is with you as the administrator. Most importantly, it is critical to make sure that definitions are loaded as soon as they are available, and that the program itself remains updated and patched. An anti-virus program is an application like any other, and therefore can be compromised if there is any vulnerability.

For a centrally manageable solution, there are many to choose from. Nearly every well-known security organizations, like Symantec, Kaspersky or McAfee offer some product or products that provide virus protection that integrates with a server appliance to provide a console with metrics and reporting. Generally speaking, these features are only necessary if there is a need to keep statistics on the effectiveness of the system, because ideally there will be no special configurations for any endpoints. The solution will merely be to ensure that the anti-virus solution is installed on all end points, that there is an exception in the firewall to allow it to automatically update, and then to just monitor to make sure that the program is updating to the latest versions, which can be done with other free tools in a Windows environment, like WSUS.

Interfacing Protections

All of the preceding protections are put in place on a workstation, or a single computer that is destined to eventually be the work tool of at least one user. Therefore the security system should be implemented to be mindful of the fact that, at some point, the user will take possession of the computer for the purposes of doing their work. As the user of the machine, they are an integral part of the security system because no security is useful unless there is willing participation. Thinking in terms of physical security, an organization could spend thousands of dollars on door locks, card readers, and biometric scanners. All of those are useless if the employees find participating in the system such a daily hassle that they put a heavy box in front of the door and leave it open.

Securing the Human

The security technician must approach the system as an end user. They should assume that the end user will be a lay person with the only goal of completing their work in a way that is efficient and as pleasant as possible. They should also assume that they have no knowledge of technology beyond that necessary to do their work. If the environment is mixed between the tech savvy and initiate, then the system should cater to those with the least knowledge. This is essential to overcome a number of cognitive quirks in the human psyche. These biases, well noted in psychology, are the Dunning-Kruger, the "curse of knowledge" phenomenon, and the unwarranted assumption of mutual knowledge.

To expand on that topic for a moment, it is imperative to approach security from the proper perspective. Security is a human problem. Technologies will change and evolve. They will become obsolete and be decommissioned and newer technologies will take their place. The

element of any security system that does not change is the human element. Despite out capacity for imagination and inventiveness, we are particularly consistent creatures. This is the primary reason the human is the weakest part of a security effort. However, we have the advantage of having knowledge of the nature of ourselves, and we can, with some effort, empathize with other humans. It's from this position that the terms above and many other applicable insights originate. As such, any effective professional in the field of information security should have a working knowledge of basic human psychology, and especially the principle of the stage of competence from which the aforementioned phenomenon extend.

The Dunning-Kruger effect resides in the first and second stages of competence, and it is essentially the end-user's overestimation of their knowledge and ability, and therefore underestimating the value of experts. It is likely more commonly expressed with the idiom of "a little bit of knowledge is a dangerous thing." Users, overconfident in their skill, will often make errors in judgment and might even attempt to disable or circumvent security measures due to their faith in their level of knowledge. Keeping this in mind, the security professional should anticipate security breaches not only from the position of an attacker, but from the position of a user. For example, a protection that is a particular nuisance to the end user is likely to be tampered with by a confident, if not competent, user.

The other principles mentioned are the "curse of knowledge" and the unwarranted assumption of mutual knowledge. These two principles apply to the information security professionals within the final stages of competence. The "curse of knowledge," simply put, is the problem at hand. It explains the inability of those well-versed in a topic to think of problems from the perspective of lay people. The assumption of mutual knowledge on the other hand, is the fallacy that an expert in a field will suffer from assuming that everyone has an understanding

of the topic as they do. For example, it might be common knowledge in data security that passwords should be more than fifteen characters long to prevent the generation of a vulnerable LM hash, but it would be an error to assume that any part of that statement is mutual knowledge with anyone outside of the field. This principle is frustrating for both parties, as the user constantly feels like security is "talking over their head" or making them feel stupid, while security feels as if they are explaining until they are blue in the face and no one is listening.

Generally, there will be very little compromise between security and usability in the practices thus far laid out in this document. As, in any security system, there is always the possibility that a user will intentionally or accidentally run into the protections in place having an understanding of how a lay person approaches security, and how security interfaces with a user's workday, is essential to ensuring that the user experience runs smoothly. A security system is only functional if the people within it all participate.

Participation in security protocols can be compulsory, through policing for compliance and punitive measures for failures, or it can be willing. Attaining willing participation is more difficult, but far more effective. Compelling a person to participate in anything, security system or not, will only secure their involvement enough to meet the minimum standard to avoid punishment. A willing participant, on the other hand, becomes a partner in the system. They will go out of their way to report security issues, and will understand their role and the importance of security.

Social Engineering

A user participating in security is important, but equally important is the user's role as a security measure. The routine activity theory of crime of Felson and Cohen examines the

circumstances that provide an opportunity for a crime to be committed. Under that theory, three things are necessary to provide an environment wherein a crime may occur (Cohen & Felson, 1979.) First, a likely offender is assumed, because a repository of data is much like a vault full of valuables, in that a defender must assume a criminal will attempt to acquire it just as a bank must assume a criminal will attempt a theft (even if there is no clear and present danger of it.) Second, the absence of a capable guardian is also assumed, because the goods are digital, the invasion invisible, and there can be no guardian capable of intercepting every attack before it reaches the outer perimeter of the system. Such a thing would require heavier security at an ISP level, which would definitively change the role the Internet plays in our lives. Essentially, effective defense assumes a failure.

The last element in routine activities theory is the one thing we can improve in our chances, and that is a suitable target (Cohen & Felson, 1979.) The end users are that target. They are the first line of defense against an attack, because they (or their credentials) are the primary target of most attacks. Educating end users on the existence of social engineering and importance of security can have a tremendous impact on the environment, because successful social engineering relies upon the target not being wary enough to not disclose sensitive information. This is even more critical in our age, where social media has become an important tool for both business and personal life. Call it savvy or paranoia, but it's this lack of understanding and caution that makes the user a suitable target.

The key to success is striking an appropriate balance between reckless and paranoid that is both practical and sustainable given the resources available in your law enforcement agency. Social engineering can take many forms, and could happen any time. Any time the phone rings and someone claiming to be a third-party contractor your company works with, it could be a

potential attacker probing for information about the organization's structure or security habits. Every email requesting the contact information for the employee's manager could be a customer wishing to make a legitimate complaint, or it could be a careful crafted message intent on gathering enough information about the manager to spoof an email from them. The key to finding that balance lies in providing each individual using the technology with the tools and knowledge they will need to make sound judgments on every potential interaction with an attacker. For example, phishing emails may contain misspellings, links should be checked before clicking to ensure they appear accurate and are not obfuscated, the caller's information should be requested and recorded on every call, access devices or employee badges should be scrutinized for authenticity, and other such practices of a caution. While the danger persists that social engineering will take place via phone, email, or even in person, more effective and more subtle is that which takes place over social media. "Catfishing" is a term for the practice of creating false or misleading social media profiles, and then masquerading as these individuals to gain the confidence and cooperation of another person. Often, this is used to establish an online romantic relationship in an effort to enlist the direct aid of a user.

By way of an example, consider the following scenario: There are several employees of H&R Block that are mutual friends on Facebook (for personal or professional reasons.) An attacker goes onto the H&R Block website, and locates an office in their city, Atlanta. The website lists the associates in that office, but if it didn't all he would have to do is go in there and collect business cards. The attacker then finds the name of another H&R Block associate at another office in Brookhaven, GA. We will use the name Linda for that person. If Linda already has a Facebook account, the attacker takes whatever information they can find from Linda's account – pictures, demographic information, everything. If they don't have a Facebook account,

they can still find out whatever they can from the Internet or by making inquiries. Either way, the attacker crates their catfish account to look as much like Linda's account as possible. It won't look perfect, but it doesn't have to. The attacker can then begin sending out private messages to the employees in the Atlanta office. All they need is a convincing lure, and for the sake of this example let's say that he has been able to identify the manager of the Atlanta branch as Mike Hanson. He creates a message along the lines of "Hi. I'm Linda from the Brookhaven office. I'm sorry for contacting you like this, but I just fund something about our boss, Mike, and I didn't want to risk sending it through company email. Check this out: http://www.thisisnotmalwareipromise.com"

Facebook's privacy preferences might make this difficult, but unless everyone on the list has configured their accounts to be more secure, the messages will get through. He doesn't need all of them to click on the link, but just one. Once their computer is infected, it's that much easier to get into the data they need. Perhaps the employee works from home and either accesses their work computer remotely, or uses a flash drive to carry work back and forth. Maybe they just happen to use the same password for their personal and work accounts. Either way, the attacker now has a way in because the first line of defense has been breached.

Avoiding falling for social engineering does not take a lot of technical expertise, nor does it take a paranoid hermit-like view. All it really requires is taking basic precautions and staying on your guard, as you would when dealing with a stranger in person. If a stranger approached knocks on your door and asks to be let in, chances are that you would not let them in unless they gave a reason. If the stranger approached and said, "I work for the phone company. Your phone is not working. Please let me in." Chances are the average person still wouldn't let them in, at least not without verifying the problem and their credentials. This is precisely what people do

when they fall for social engineering tactics, because we are less on our guard. The key is empowering the users to follow their own instincts. When a person calls on the phone, gather their name and callback information. Verify their identity of possible. If not, then don't give them any information that isn't publicly available. If the conversation sounds suspicious, report it to the proper security authority. Never click on links in an email without verifying they are legitimate. If you are unsure, do not click on them and report the email. If the email is from a known sender, and the message seems uncharacteristic, contact them over the phone and by sending a new email (not replying to the suspicious one) and ask them if it is correct. In general, the key is make users mindful of the fact that they have something that criminals want – Access.

Password Policy

Passwords are another area where security interfaces directly with the user. While it's often possible to enforce basic security standards, it's still up to the user to follow them, and our tendency to take the path of least resistance leads to predictable behaviors, which creates vulnerability. The popular tech website SplashData (2015) performs a survey of common passwords every year. Each year, the same passwords tend to appear over and over again. In 2014, the top five most common passwords were, in order: *123456*, *password*, *12345*, *12345678*, and *qwerty* (the first six letters on the top row of the common Latin script keyboard.) The remainder of the list is equally predictable, being a combination of number sequences (1 through 0, 696969, and abc123, for example) and common words from the dictionary (baseball, football, monkey, shadow, mustang, dragon, master, superman, batman, love, secret, sex.) There is also a list of common password choices that tend to be widely made, with the most common being these or variations of: *letmein*, *trustno1*, *access*, *iamgod*.

This is not just limited to basic users, either. Many IT professionals are guilty of this as well. In April of 2015, Cisco released a security bulletin about a persistent threat from the hacker group SSHPsychos. They were able to exploit numerous assets using basic passwords set up by IT staff. The list is very much the same as those for standard users, with *123456*, *qwertyuiop*, *qwe123asd*, and *operator* being the top four (TALOS group, 2015.)

A policy might make it possible to enforce minimum security standards for passwords. For example, there may be a rule that all passwords must be at least eight characters long and contain a number. Taking an example from the list SplashData (2015) created, we will use "*12345678*" as an example. It meets out minimum standards. A website, http://howsecureismypassword.net , contains an application that checks to see how secure a password is. Essentially, it works the way a brute force attack would. First, the attempts to match it to the ten-thousand most commonly used passwords. After that, it calculates the number of possible passwords based upon the character sets presented, and estimates how long a common desktop PC would take to crack the password (Small Hadron Collider, 2014.)

Our example above, "*1234678*," would be cracked instantly, even though it is acceptable according to the standards security has put forth. If security mandates a combination of letters and numbers, then our user can select, "*password1*" which would also be cracked instantly. If we mandate at least one uppercase letter, "*Password1*" is also instantly cracked, as is "*PassWord1*" and many other variations. All of these would still be accepted by a password validator. People do this because we are good when it comes to creation and imagination, but bad when it comes to storing and recalling information. Some of us can remember a truly random password. "*Rp7ty6ff5dd4%!*" would take 2 billion years to crack (Small Hadron Collider, 2014), but many of us would never be able to recall it.

The good news is that we don't have to. It can be proven that people can have relatable passwords that are highly secure. The key is using a passphrase that does three things. First, includes a number. Second, includes a special character. Finally, includes a misspelled word. Let's use a phrase as an example now. Using "*Water the tree*" as a password, it would take 655 million years to crack. Adding a special character (in this case one organic to the phrase) we get "*Water the tree.*" which takes 46 billion years to crack (Small Hadron Collider, 2014.) That's even more secure than our random password above. If we add a number, "*Water the 1 tree.*" then we end up with a highly secure password that would take 2 quadrillion years to brute force. The only problem we have now is that it is possibly susceptible to social engineering. We need to make the password highly personal. We do this by adding in an intentional misspelling, and change it to "*Woter the 1 tree.*" and we are left with a unique and easily remembered password that is about as secure as we can make it.

Another note on passwords regards the practice of writing them down. Many users and security professionals alike have been repeatedly hounded on the practice of writing their passwords down, and told about what a dangerous thing it is to do. While this is true if the password is left near the computer or in a place where it could easily be found, but with the way modern technology works that is not always the case. A greater danger is the practice of caching credentials in web browsers (the familiar pop up "Would you like me to remember this password for you?") (Ollmann, 2015.) This has reached a point where the greater danger is actually found in not writing the password down and instead utilizing these features. In practice, there is nothing wrong with keeping a written list of passwords, so long as they are treated as sensitive information. As such, they should be kept secure, and luckily most people already have such a secure location with them at all times, which we all call a wallet. In the event that a wallet is lost

or stolen, the user will already likely be contacting their banks and credit card companies and cancelling accounts. As a practice, they should be changing their passwords at that point anyway, so adding on their workplace accounts to that is a relatively harmless burden.

Physical Security

The practice of keeping written passwords in a physically secure location brings up the topic of physical security. This is often overlooked by security professionals, since the need to keep people out is generally more pronounced than the need to protect assets from attacks originating within. Often, physical security isn't even a part of the data security team's responsibilities. In any event, it should be part of their concern, as it is just as easy (or even easier) to compromise a system from within, and to steal data when the attacker is physically sitting at the terminal. An attacker within a building can set up rogue access points, or plant listener devices on workstations. Consider how often anyone actually looks behind the computer. Someone with enough access and a working knowledge of physical computing could build a device to plug into the network port of the computer and connect to the wall jack and relay traffic via WiFi to an attacker with a laptop sitting in their car across the street. A device like this could run for months off of just a nine volt battery.

Generally speaking, physical security is very good when it comes to law enforcement agencies. This should be obvious, but one can never discount the potential threat of a particularly brazen individual. Access devices (swipe cards, ID badges, hard tokens, etc.) should be a particular concern, since the access they provide can give access to less secure entrances to a location, such as parking garages or employee entrances. Because these are issues to the users,

they are primarily responsible for their safety. However, it is the responsibility of security to ensure they are used correctly, and by the authorized individual.

Ensuring is a task that requires monitoring the proper use of access devices and providing the infrastructure to monitor them. This not only included logging of things like card swipes and token usage, but also with traditional physical security measures like security cameras, door locks, biometric sensors, and other anti-theft or anti-tampering technologies. While the entire building is important, particular attention should be paid to particularly sensitive areas, such as server rooms which house assets with sensitive data, and jump boxes that allow access to them. Most importantly, attention should be paid to internal threats as well. If an asset that hold sensitive data does not need to be used by every host in the network, then access should be limited to only those that do need access. Internal access control lists should be tailored to provide access only to authorized individuals. Policies should be written to enforce the amount of time a computer can be idle before locking, and users should be trained to lock their workstation whenever they leave their desk.

A Secure Environment

The job of maintaining a secure environment becomes much easier once you have a functioning security system, but it does not eliminate the need to provide routine maintenance on your system to keep it secure. Upkeep is a large part of keeping one step ahead of attackers. At this point, the work now turns to monitoring the system, checking for failures, and doing penetration testing to find weaknesses in your system. In a broad sense, this is a separate topic from this document, and one that would likely fill several manuals on its own. However, there are a few key elements that must be considered that apply to end point security, and hence fall into the scope of this document.

Entitlement Review

One of the most obvious and yet overlooked aspects of security is restriction. Simply put, no one, user or administrator, should have access to resources that they do not need to perform their duties. The use of access control lists (ACLs) to prevent users form access unauthorized resources is a basic as putting up fences and locking doors around a building. This is simple on a smaller scale, but before a major bureaucracy as an organization develops. People's duties may change via promotion or reorganization. Access might be needed on a temporary basis. Any number of situations develops to the point where, unless properly managed, access requests and changes can become a full-time concern for security professionals. However, it's one that is critical to successful security.

The elements of successful entitlement review boils down to one rule, as pointed out in the CISSSP text book for the professional certification: Default to no access. When a person enters your organization, access should never be assumed. If templates are used for access, they should be reviewed frequently and with a conservative attitude. If a request comes in to expand access, it should be questioned. As important as it is for individuals to be able to access the systems and network shares they require, the change should always be justified.

This extends to the user's own workstation as well. It's often easier to take the path of least resistance when it comes to endpoint maintenance, and it's much easier to make the user an administrator over their own workstation. This is often seen as harmless, since it's just one system. Unfortunately, that provides a security vulnerability for an attack vector that is very exploitable for attackers.

Vulnerability Scanning

The use of products like Nessus or Nexpose provides excellent feedback for administrators with minimal effort. They are products that, like anti-virus programs, contain definitions for known vulnerabilities. It then scans your end points for those vulnerabilities, and reports any potential problems. While not a completely effective replacement for actual penetration testing, ongoing vulnerability scanning is a good way to locate vulnerabilities in a passive fashion, giving you the information you need to stay on top of your environment. This makes active penetration testing more effective, as you can then focus on the unknown vulnerabilities rather than the ones known by the vulnerability scanners.

The key to utilizing these tools is not in running them. Relatively speaking, they are simple to set up, maintain, and use. The trouble is that, since they are essentially programmatically checking for known vulnerabilities, that you are limited to those that are known well enough to be written into an existing definition. The second problem is that an automated system will only be able to tell you if a vulnerability matches the definition, but not if the issue is a false positive. For example, a tool like Nessus may be able to tell you that an asset may be susceptible to DNC cache snooping (meaning an attacker could user it to see who has contacted the DNS server, providing a list of known business contacts.) Now, this is only really an issue if it is reachable by untrusted individuals. The program cannot know that, and so it's left to the security professionals to investigate a perceived vulnerability to confirm it as a flaw.

End Point Health

End point security is the concern of the security department, and everything else is generally left to the various help desk and technical services departments. The fact is that

security touches every aspect of user interaction with technology, and so there are some areas of concern for information security when it comes to the specifics of end point deployments. An outdated version of Adobe Reader that the technical services team did not consider important, for example, could be an attack vector. A Windows security patch that the OS development team did not consider a critical update could provide an attacker an opportunity to disable or circumvent security features. Whatever the case, the point is that a security professional has a vested interest in maintaining an environment that is healthy and efficient, and that definitely includes the workstations.

Version control and OS and browser plugin patching can be accomplished by a number of products, depending on what you have available. Third party programs like Ninite , or Secunia are industry leaders in version reporting, and Microsoft's own WSUS or SCCM makes compliance monitoring quite easy. However, these are not typical tools provided to information security, but rather are part of endpoint technologies. This only illustrates the need for effective communication within the IT department, as well as underscoring the importance that all users (even other individuals in IT) be willing participants in the security system.

One of the final topics in this area is one of the most important. When it comes to end point security in the environment we have described there are few tool more critical than group policy. Group policy, a part of a Windows enterprise environment, provided the security technician the ability to centrally manage all of the protections thus far listed. In addition to that, it also gives the administrator the ability to regulate each computer, including the ability to allow or deny the use of external devices, access to the various control functions on the machine, and it does all of this with a level of authority higher than a local administrator. This means that even if

a local administrator's account is compromised, the attacker cannot directly override your configuration without somehow breaking group policy.

The strength of group policy is its modular nature. Policies are created in various "objects" – essentially different rules. These policies can be applied as broadly or as narrowly as your organizations active directory structure will allow. Further targeting can be applied to the object itself, having the object only apply if the conditions of a filter are true. For example, and object applied to an organization unit for "laptop users" but filtered to only apply to machines that have an operating system of "Windows XP." Within the policy, even further targeting can be done at item-level. For example, a rule that says "all users will have an icon for Wordpad on their desktop" but item-level targeted to apply only to users in "Wordpad users" group in Active Directory.

What's better, group policy can be pushed at a domain level, and then disjoined, meaning it continues to be in effect until the machine is again joined to the domain and the policy updates from the central domain controllers. This means even mobile machines can be locked down and controlled with group policy. An even better option is investing in a technology such as Direct Access, which allows group policy to update on remote machines as if they were still connected to the domain locally. Regardless of where group policy resides in your organization, it is critical that the security department or the managers within be given access to create and edit a series of security-focused group policy objects.

EFFICACY OF METHODS

Real-World Examples

There are numerous examples of the effectiveness of these strategies. By way of first-hand examples, there are fewer more compelling than was has becomes, in my experience, the quintessential endpoint protection problem: Crypto-variant outbreaks. They represent the perfect endpoint security problem, wherein proper controls ensure a complete mitigation of the threat and failure presents absolute catastrophe, with the attack vector entirely in the end point realm. This malware, once it gains a foothold on the machine, will begin encrypting the contents of the computer and any network resources it is able to reach. It enters via the endpoint, and requires end user interaction to infect. An example scenario would be the end user visiting a link and downloading the malware, and running it under the assumption that it is a desired program. Once infected, the virus encrypts the files, and leaves instructions to pay a ransom to unlock the data. These crypto-variants have become extremely common, and many law enforcement agencies have fallen victim to them (Greenberg, 2015. Colon, 2015. Khandelwal, 2013. Wei, 2015.)

In the summer and fall of 2014, crypto-variants seemed to enjoy and upsurge in usage. The University of Wisconsin – Stevens Point, being an organization with ties to other academic institutions, was under threat from this malware, and an assessment of the risk involved made it clear that proactive measures needed to be taken. A plan was developed utilizing application control policies taken from this author's Protected Campus Load project, and testing began. During the process of testing, one machine became infected by the Cryptolocker virus. Advancement was halted by the quick reporting of the user and subsequent removal of the device from the network, but the data on the machine was irrevocably encrypted and had to be restored from backups. Following this, the site-wide security enhancement was pressed into service and

auditing began. Three hundred more machines over the course of the next two weeks reported attempted infection. None were able to gain a foothold and no more files were encrypted. Service calls from end users as a result of the protection itself were zero. This is a 100% effective solution, resulting in no downtime or service interruption to the users.

In the fall of 2014, the author began working for Footlocker.com. The business did not have any end point protections in place, and they also faced a crypto-variant outbreak. Their entitlement control policies gave the user group "Everyone" full access to the network shares. A single multi-user workstation became infected, and as a result hundreds of gigabytes of data was irrevocably encrypted and had to be restored from backup. This is a 100% failure, resulting in hundreds of thousands of dollars to repair. Truly a situation where an ounce of prevention would have been better than a pound of cure, the remediation of this outbreak involved forensic work to develop a PowerShell script to even identify the scope of the problem. The failures in this scenario were several, all violations of the principals laid out in this document. First, the lack of entitlement review meant the user (a simple call center employee) had far more access than was required, and was an administrator over their own system. Second, the lack of web content filtering made the retrieval of the malware possible. Third, no application control policies mean that the virus could be run, and the lack of a reliable anti-virus did nothing to stop it.

Internet Explorer is a frequent target of zero-day exploits, because it is the native browser included with the world's most popular operating system, Windows. In the fall of 2013, the Information Security department of the University of Wisconsin – Stevens Point became aware of a zero-day that impacted the version of Internet Explorer most common on workstations in the environment. Again, it was discovered that an element of the Protected Campus Load, Microsoft's Enhanced Mitigation Experience Toolkit (EMET) prevented any attempted exploit

of the zero-day bug. Within 90 minutes of the news breaking, all machines were covered by EMET, and with no intervention or interference with the end user.

Examples of effective usage of these techniques are found elsewhere as well. In July of 2014, police and private white-hat hacker groups were able to effectively leverage endpoint security features to put a stop to the Shylock banking Trojan (Walker, 2014.) This virus utilized several known attack vectors, including the end user. An article written by Danielle Walker for SC Magazine in quotes a member of the Dell SecureWorks team in describing the tactics attackers would use. He said, "[Attackers] try to trick the users into believing they are communicating with a bank [representative] when, in fact, they are communicating with the criminal. They were getting information they needed to impersonate the victim [when] logging in." (Walker, 2014.)

CONCLUSION

Summary

In our sample environment, which is a primarily Windows-based enterprise organized with Active Directory and making use of Group Policy, with operating systems on servers of Windows 2008 R2 and endpoints with Windows 7 Ultimate or above, we have created a security systems with a cost (excluding the licensing cost of operating systems and infrastructure, as those are organizational costs and not security costs) of between zero and ten dollars per machine. This is well within the budget of even the poorest of law enforcement agencies, and it only proves that good security is more of a matter of having the appropriate attitude and posture.

As any law enforcement officer will know, security is a human problem, not a technological or financial one. This document has, by design and necessity, provided a look at security from a technical perspective. These are only tools to address a problem with people.

This cost, of course, is variable depending upon the third-party solutions or financial decisions that your law enforcement agency chooses to make. What's more important, however, is that the principals outlined here above will make each and every workstation attached to your network hardened against attack with a minimal ongoing investment in maintenance and material resources. At this juncture, attention can truly be shifted to network-level or gateway-level security concerns, with much less emphasis places upon the end-user technologies. In short, you have ensured that every window is as secure as they can be, and now you can truly focus on the doors. It would be wrong to say that you can henceforth ignore endpoint security. There will always be the ongoing task of monitoring for threats and new technologies, and making adjustments as the threat landscape changes.

Future Research

For those that have a continued interest in end point security or in end user interaction, there is still an ample amount of information to uncover. Though technology use and advancement has exploded over the course of a single generation, in terms of how humans relate to it, the discipline is in its infancy. End point security itself has much more to explore than the simple overview this document provides can allow. This is becoming even more of a critical field as the use of biometric scanners and wearable technology are currently on the cusp of being

the primary technologies we interact with. It quite common for mobile devices available for sale today to contain biometric security features, like fingerprint scanners, or voice and facial recognition software. These are often in tandem with more traditional security functions, like access PINs, passwords, and now the ubiquitous "pattern" password.

For wearable technology, the invention of flexible OLED displays now makes it possible to create a garment that has displays sewn into it. Google has release Glass, and very soon will be starting production on Glass 2.0. This wearable tech is fairly rare at this point, but other wearable technologies, like smart watches, daily activity and sleep trackers, and sports training devices with built in sensors that record altitude, barometric pressure, heart rate, and GPS position are all quite common. What's more amazing, is that nearly every one of us carries a device that does all of these things nearly every day. Our mobile phones have become indispensable parts of our lives.

As their importance has grown, so has their vulnerability. Attacks on mobile devices were once quite rare. Now, they seem to be nearly as common as attacks on server or desktop systems. Since the trend toward mobile devices and wearable tech does not show any signs of disappearing, these areas will be of critical focus in the future of the field of information security.

Recommendations

This document is not meant to be a highly technical, in-depth look at endpoint security. It is not even intended to be a technical document at all. Instead, the purpose of these words has been to put workstation security into perspective for the law enforcement agency, and security

professionals therein. If successful, this document will not have necessarily taught the reader about any new tools or technologies, as nothing outlined herein is new or groundbreaking subject matter. Instead, it will have inspired a new, more endpoint-mindful perspective on your law enforcement agency's environment. In that spirit, the primary recommendation is to take that inspiration and concoct a workable solution for the end users and workstations under the umbrella of your organization.

By far, this document has favored Windows-based environments, specifically with those that take full advantage of enterprise features through the use of group policy. It is highly recommended, if you have the option available to you, that this be the way your agency is organized. While other environments, like those based upon Linux, are not necessarily less secure they are also not more secure, despite the insistence of some. Security patches are being released continuously for both systems, and the severity of security alerts for the operating systems is comparable. In short, vulnerabilities are being discovered in both on a continual basis. In terms of volume of known malware, Windows may be the leader, but that it only because they are far and away the most common operating system for both private and commercial applications.

The ongoing work of endpoint security is critical to maintaining the integrity of your entire security structure. Logs and service calls form end users must be carefully watched and handled in a timely fashion. The threat landscape must continually be monitored through keeping on top of the latest news and trends in security. The author of this document accomplishes that through several means. First, multiple RSS feeds from reputable security news sources, such as the SANS Internet Storm Center, SC Magazine, Dark Reading, and many others. Second, by keeping an eye on the latest broadcast CVE (Common Vulnerabilities and Exposures), which are

official alerts for known active or potential exploits. However, keep in mind that these broadcasts are often many days old. Generally speaking, an exploit has been known and disseminated long before these alerts, so be aware that if you learn of it once the CVE alert is released, you are already behind on mitigating the risk. Because of this, the third way this author stays informed of security situations is by frequenting forums and chat rooms where hackers gather. There are many such places with large, active communities where white, grey, and black hat hackers gather to discuss methods and theory. These places are generally found in the "deep web." If you are unfamiliar with that term, it refers to places on the Internet that are not index by search services, and so are not so easily found by the uninitiated. Finally, the best way to keep on top of new threats is to think like an attacker. When your infrastructure team brings up a new proposed change, work through how you might exploit it. Work out how you might break your own security systems. If there are vulnerabilities to be found it is much better for you to find it than to learn about it after an attack.

To this end, it's important to develop and maintain the skills of an attacker. Learning what an attack is can be important, but it's often equally or more important to be able to identify what an attack does. Learning scripting languages like PowerShell, Python, and Ruby can be helpful in understanding how exploits can be written, how to stop them, or how they are developed. Learning programming languages like C# or C++ can help you to understand how RATs (Remote administration tools) can be used to control zombies in a botnet. In the world of computer security, the lines between good and bad, white and black hat, do not lie in the tools used or the methods learned. They lie entirely in how that information and those skills are utilized, and nothing more. It is a fundamental philosophical difference.

REFERENCES

Anderson, M. (2007). Internet security – Firewalls & encryption the cyber cop's perspective. *Forensic Restitution*. Retrieved from: http://www.restitution.co.za/news.php?id=17

Ashford, W. (2013). Infosec 2013: Cyber-crime challenges law enforcement. *Computer Weekly*. Retrieved From: http://www.computerweekly.com/news/2240182575/Infosec-2013-Cyber-crime-challenges-law-enforcement

Bureau of Justice Statistics. (2005). Cybercrime. Retrieved from: http://www.bjs.gov/index.cfm?ty=tp&tid=41

Carter, D. (2004). *Law enforcement intelligence: A guide for state, local, and tribal law enforcement agencies.* U.S. Department of Justice Office of Community Oriented Policing Services. Retrieved from: http://www.cops.usdoj.gov/pdf/e09042536.pdf

Cohen, C. (2007). Growing challenge of computer forensics. *The Police Chief*. Retrieved from: http://www.policechiefmagazine.org/magazine/index.cfm?fuseaction=display_arch&article_id=1136&issue_id=32007#2

Cohen, L. & Felson, M. (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review*, 44. Pp 588-608.

Colon, M. (2015). Illinois police department pays ransom after Cryptoware infection. *SC Magazine*. Retrieved from: http://www.scmagazine.com/illinois-police-department-pays-ransom-after-cryptoware-infection/article/399677/

Cyber Crime Statistics. (2015). Cyber-crime statistics. Retrieved from: http://cybercrimestatistics.com/

DataBreaches.net. (2012). Law enforcement targeted by hackers. Retrieved from: http://www.databreaches.net/law-enforcement-targeted-by-hackers/

Diaz, J. (2015). Cyber criminals use hybrid tactics of ransomware and catfishing to extort mobile users. Android Headlines. Retrieved from: http://www.androidheadlines.com/2015/03/cyber-criminals-use-hybrid-tactics-ransomware-catfishing-extort-mobile-users.html

Enterprise Strategy Group. (2010). Percentage breakdown of server operating system usage, by industry. Retrieved from: http://www.esg-global.com/blogs/data-points-and-truths/percentage-breakdown-of-server-operating-system-usage-by-industry/

Federal Bureau of Investigations. (2015). Cyber Crime. Retrieved from: http://www.fbi.gov/about-us/investigate/cyber/cyber

Greenberg, A. (2015). Massachusetts police department pays $500 following ransomware infection. *SC Magazine*. Retrieved from: http://www.scmagazine.com/massachusetts-police-department-pays-500-following-ransomware-infection/article/407584/

Hinduja, S. (2007). Computer crime investigation in the United States: Leveraging knowledge from the past to address the future. *International Journal of Cyber Criminology, 1(1).* Retrieved from: http://cybercrimejournal.com/sameer.pdf

Howard, et al. (2010). Windows ISV software security defenses. Retrieved from: https://msdn.microsoft.com/en-us/library/bb430720.aspx

Khandelwal, S. (2013). U.S. police department pays $750 ransom to retrieve their files from CryptoLocker malware. *The Hacker News.* Retrieved from: http://thehackernews.com/2013/11/us-police-department-pays-750-ransom-to.html

Malenkovich, S. (2013). 10 arrests that shook the cybercrime underworld. Retrieved from: http://blog.kaspersky.com/10-arrests-that-shook-the-cybercrime-underworld/

Martin, K. (2005). Bavarian police predict new viruses. *SecurityFocus*. Retrieved from: http://www.securityfocus.com/brief/49

Martin, K. (2006). US police using data brokers. *SecurityFocus*. Retrieved from: http://www.securityfocus.com/brief/233

The Mentor. (1986). The Hacker Manifesto. Retrieved from:

http://www.mithral.com/~beberg/manifesto.html

Microsoft. (2015). AppLocker Technical Reference. Retrieved from:

https://technet.microsoft.com/en-us/library/ee844115.aspx

Microsoft. (2015). BitLocker drive encryption overview. Retrieved from:

http://windows.microsoft.com/en-us/windows-vista/bitlocker-drive-encryption-overview

NetMarketShare.com. (2015). Market Share Reports. Retrieved from:

http://www.netmarketshare.com/

Ollmann, G. (2015). It's safer to write your password down. *The Day Before Zero*. Retrieved from: https://www.damballa.com/its-safer-to-write-your-password-down/

Robinson, T. (2015). Chinese police department purchased spyware. SC magazine. Retrieved from: http://www.scmagazine.com/chinese-police-department-purchased-spyware/article/392187/

Small Hadron Collider. (2014). How secure is my password? Retrieved from:

https://howsecureismypassword.net/

Smith, A. (2014). Older Adults and Technology Use. Pew Research Center. Retrieved from: http://www.pewinternet.org/2014/04/03/older-adults-and-technology-use/

SplashData. (2015). "123456" maintains the top spot on SplashData's annual "Worst Passwords" list. Retrieved from: http://splashdata.com/press/worst-passwords-of-2014.htm

TALOS Group. (2015). Threat Spotlight: SSHPsychos. Retrieved from: http://blogs.cisco.com/security/talos/sshpsychos

Walker, D. (2014). Police, security firms abate Shylock malware threat. *SC Magazine*. Retrieved from: http://www.scmagazine.com/police-security-firms-abate-shylock-malware-threat/article/360350/

Websense. (2013). The seven stages of advanced threats: Understanding the cyber-attack kill chain. Retrieved from: http://www.webssnse.com/sevenstages

Wei, W. (2015). Chicago police department pays $600 cryptoware ransom to cybercriminals. *The Hacker News.* Retrieved from: http://thehackernews.com/2015/02/cryptoware-ramsomware-bitcoin.html