GOVSEC 2006 Government Security Expo & Conference

April 26, 2006 Washington, D.C.

Consideration of Targeted Hostile Software in Homeland Security

by

Peter V. Radatti, Ph.D.

Distinguished Visiting Research Professor
Florida Institute of Technology
University Collage 150 West University Blvd., Melbourne, Fl 32901 USA
www.fit.edu

CEO / President
CyberSoft Operating Corporation
1508 Butler Pike Conshohocken, PA. 19428 USA
+1 610-825-4748 * radatti@cyber.com
www.cybersoft.com

This document is Mr. Radatti's notes from his presentation at GovSec. It is not intended as a standalone white paper.

Computer security today is a failure. It gets in the way of getting the job done; therefore, it is normally bypassed or weakened. Even if security systems are implemented correctly, they are often bypassed for mission or operational reasons or are inadequate.

The **largest single hole in computer security** exists in all systems. It exists even in systems that are of very good design. There is no way to eliminate this problem but you can reduce its scope. That problem **is the "unforeseen."**

For the purpose of this presentation, I will **use the word "virus" loosely** to mean all forms of attack software including viruses, worms, bots, Trojans, bombs and all other forms, both known and unknown.

Most people feel that the installation of an anti-virus/anti-spyware program will protect them from virus attacks. This assumption is false.

Virus, spyware and all other **scanners can only detect attacks that they already know** about. They do not directly protect against zero day attacks, nor do they protect against stealthy, low profile attacks in which a sample was never collected for analysis. **A virus can be designed that makes collection a low probability.**

There are methods in place that attempt to address the zero day problem in the form of **behavior blockers** and **code emulators**. These methods are **only partly successful** and can not address viruses that were designed specifically to avoid this type of detection.

Heuristics also plays a part in zero day defense but the word "heuristics" has become a **marketing buzz word** and no longer has any real meaning. When it has meaning, it is because a manufacturer has defined it with a specific meaning.

CyberSoft has a specific definition for heuristics. We perform heuristics using a human analyst. I feel that automated heuristic systems are not appropriate because they can do unexpected things. At CyberSoft, heuristics are performed on a case by case basis by an analyst who is skilled in the process. A good example is virus family detection. Computer viruses normally come in families of related, but distinct programs. Each virus has a detection algorithm assigned to it. If there are enough viruses, an analyst will review all of the algorithms and design one new algorithm to replace them. This new algorithm will normally detect all of the known, plus any new viruses created in that same family.

CyberSoft heuristics work well because history has shown that viruses writers like to piggyback onto successful attacks. That is the entire reason families exist.

Computers are normally the guards that protect themselves. That is, they run their own protection software such as anti-virus. This is the same as putting the **chickens in charge of guarding the chicken coop**. There have been viruses that specifically target and corrupt anti-virus programs to help spread the virus.

There are solutions to this problem!

If none of these methods is 100% effective, is there an answer? Yes. In addition to using a combination of a scanner and any of the other methods already mentioned, there is **one tool that can detect almost all forms of attack** including software and human attacks, **baseline integrity**. CyberSoft addresses this using a tool called the Cryptographic Integrity Tool. **(CIT)**

The CyberSoft CIT tool provides a report that provides a list of files that have been added, deleted, modified or duplicated. It also reports any files whose cryptographic signature has been flagged as dangerous. A tool of this type does not rely upon signatures, emulation, behavior or heuristics. The only way to hide from this tool is not to touch the file system. Any attack which is only memory resident can be removed by rebooting the system. In addition, this report provides the most import of all computer security reports, **Aggregate Data**. Using aggregate data, you can detect insider attacks from spies, outsider attacks from hackers, workers who are not doing what they should and all forms of software attack that touch the file system. The only drawback is that the report must be read by an administrator.

There is a special version of baseline integrity that CyberSoft also addresses, **self-healing**. The CyberSoft self-healing system is called Avatar. It was designed for a "lights out" battlefield operation, with the assumption of proactive hostile intent. Avatar will **put the baseline back** where it belongs once an unauthorized modification is detected. This **preserves the operations capacity** of the system and disables the attack.

Another method of protection that is often used against both hackers and viruses is **physical separation**. This is also known as **compartmentalization**. It helps against outsider attacks. **It will not stop a virus attack**. I have seen fully compartmentalized computers in which access is restricted to a few people, no external network connections exist, and movement of media is strictly controlled that have full-blown virus infections. In fact, one of these incidences is why my company CyberSoft exists. Bricks and fences do not provide protection against virus attacks.

There is an argument that a **high-security computer system** such as Trusted Unix or SELinux will provide **protection against virus attacks. This is false**. Dr. Fred Cohen proved that the fastest way to gain privilege in any system, including trusted systems, is the use of viruses. Tom Duff also proved this, in addition to being the first person to observe a virus infection originating from backup media. In fact, only common computer operations are needed for attack software to be successful. A correctly configured, hardened, isolated, trusted computer system is not immune.

In Britain, there was a concerted effort to produce educational systems that could not be subverted by attack software. They invented and fielded the **Acorn computer**. This computer held the entire operating **system and all programs in ROM** memory. They were **still infected**. Today very few devices use ROM memory, but use Flash RAM. This would make it even easier to attack. Properly designed secure hardware is not protection against software attacks.

There is a classification of software attack that is not often discussed but occasionally seen. It is **a targeted attack**. A targeted attack is a program that is targeted against a specific entity or role. For example, a virus could be released that looks for all systems whose name end in .gov or .mil, or for all systems in a specific IP block for which public information identifies the owner. They can also be programmed to keep a very low and slow profile. In effect, a **sleeper virus.**

Sleeper viruses can be programmed to be **opportunistic.** They can exist for long periods of time before becoming resident in their target objectives. They can also become more active upon the receipt of a trigger. Consider a virus that attacks cell phones. They have existed for years. Such a virus could be programmed to wait for a specific date before triggering a payload. What would happen if all cell phones in an area were knocked out during a terrorist attack? At minimum, it would increase panic. At worst, it could also get into the emergency cell phone systems.

An attack can be targeted to a specific system but exist in the wild! This was projected by **Sung Moo Yang in 1997**. http://www.cybersoft.com/whitepapers/featured/cruise.shtml

Is there a solution to software attacks? I believe **there are answers** and have been working since 1988 on solutions at CyberSoft (www.cybersoft.com). These solutions have been turned into common off the shelf (COTS) products called the **VFind Security Tool Kit.** There are three versions of this product depending upon the problem set that needs to be solved.

In addition to software attacks, there are **basic computer security paradigms** that my colleagues and I have been working on. We feel that **significant changes need to be made** in computer security implementation. We are not alone in this feeling. The NSA, USAF, USN, US Army and other organizations have all pointed to new ways of thinking about computer security. We have pulled together answers from all of these organizations to point to a new paradigm. This new way will not only provide more protection against software attacks but also against hackers, spies and unauthorized actions by authorized users. I think these **changes need to be made** now.

Computers allow us to do things that were impossible or not cost effective in the past. This has caused an explosion in the use of computers in everything from door locks to ovens to air traffic and port control systems. They have roles in places that are unexpected to the average person. Computer systems are **layered in extremely complex structures** that are hard to comprehend by your average person. In fact, your average person may not even realize that computers have been embedded in many traditionally

analog devices. An example is cell phones. Computers are important in almost all disaster response programs. Computer security breaches can cause disasters. Consider the **power failure** in North America that occurred August 14, 2003 that was attributed to a transmission **circuit protection device that functioned correctly**. The remainder of the **power grid did not function properly**.

This places computers at the center of priority for large scale attacks. They are both the **linchpin to success and a central point of failure**. They are a **high target objective**. There are lots of smart people in the world and it is much easier to break into a computer system than to build a good one.

Compartmentalization does not solve the problem and may **make solving problems**, once an attack is noticed, even **harder.** The target becomes too great of a value to ignore.

Computers will do what they are told.
Computers can be told to do things that you don't want.
Computers can be subverted in ways that make it hard to detect.

Biometrics do not work unless layered. **Biometrics are easy to capture and replay**. Once inputted, a biometric signature is only a binary stream.

PKI is a failure, which means the way most people do online banking and stock trading is not safe. It also means that secure **networks** that reply on **PKI/PKE are not secure**.

Once the make and model of **any device is known**, then **specific targeted attacks** can be devised. For example, an access control panel. Micro-drill a hole in the panel and **inject signals** to the PC card. It is the same with all electronics including wall mounted panels. Wires are normally just below the drywall, and are easy to detect and perhaps inject signals. False faces on keypads can capture key codes, known as **skimmers**.

The **number one problem** for security is the **human.** Most security is dependent upon the user. How do we get users not to share passwords or be vulnerable to human engineering (spoofs, etc.)? A survey by SonicWALL indicates that 44% of users do not memorize their passwords. Humans can give up the system unknowingly and bypass security.

User education is not the solution. We can not make users experts.

Factors of Identification

1. Something you know (user ID and password)
2. Something you know (user ID) and something you have (security token)
3. Something you know (user ID) and something you have (security token) and who you are (one of your biometric identification factors – finger print, iris scan, facial characteristics)

The priority of factors is important. You should use what you have (security token) and who you are (biometric). All tokens must match.

The primary purpose of **something known** is to allow the user to **signal** between **normal** and **duress** situations.

A **complete solution** must be designed **from the top down** for security! All signals on cables must be encrypted or playback can happen anywhere.

Security systems should be transparent and passive, not requiring overt actions by users for security to accomplish its job.

As a user approaches a closed area, the security system should sense the user's security token and verify the user's biometric identity, unlock the door and possibly even open the door. After the user passes, the security system should lock the door behind the user. Computer systems should sense the user's security token and verify the user's biometric identity and unlock the system when approached by the user, and the system should lock as the user departs. **All security interactions should be passive** – without overt user action associated with security as he accomplishes his job.

The paradigm of **unique user IDs for login/logoff does not match real life** situations. For example, multiple users will use the same terminal without logon/logoff sequences between users; an operator may work a position and pass it to the next shift without logoff sequences to preserve the desktop's state. The NSA has separated identity from role in SELinux. With this construct, a person in the **role** can start the role's session and it can be continued by someone else in the same role. Actions are accounted to both the role and the person in the role. Moving from the user ID and password paradigm to a two or three factor identity security system causes this mismatch to stop. In many current systems with the paradigm of unique user IDs for login/logoff, security is circumvented by the sharing of user IDs and passwords to accomplish the mission. To implement real security with two or three factor identification, **separation of role from identity** becomes critical.

Security should help people perform their jobs, not be in their way. Users should never have to memorize user ID/password combinations.

The DoD vision of the **Global Information Grid** represents a completely networked vision for the battle space where everything and everyone is networked and interoperating from the front lines, to the air and sea, to headquarters.

In the **SciFi Channel Battlestar Galactica** (2005) Episode 1, due to a security breach caused by one man, **the enemy walks through the entire defense network**. In fact, the defense had no idea what was happening as errors avalanched, resulting in a total shutdown. While this is a TV show, if the Global Information Grid is not designed with very strong security, it **could happen to us**. This show is available on DVD. If you know what to look for, it will scare you silly.

Time is growing short. In the past, virus and hacker attacks were performed mostly by antisocial teenagers. Today they are in a minority. **Organized crime** is a significant factor. All major nations, including **hostile nations,** have cyber-attack and defense units. We already know that some **terrorist groups have the technology for cyber warfare**.

All of this can be implemented today. Most of it has been already implemented as **separate parts and only needs system integration**. CyberSoft has been working on this for several years and **we are very close.** I believe that the **time for an integrated security system is today.**

Radatti's Rules of Computer Security

1. The minimum cost of computer security failure is directly related to the value of the information lost, the loss of the system's operational capability, operational corruption and the cost of cleanup.

2. The outcome and financial costs associated with a computer security breach is always higher than predicted and unknowable by all parties involved.

3. If a computer security solution is common enough that an attacker can study it, then its value may be greatly diminished.

4. The more interconnected a system or network, the greater the opportunity for a security breach.

5. Nothing is completely secure, but you can make it difficult.

6. The job of computer security is to allow only the good guys to do their jobs effectively and efficiently.

7. Computer security is a never-ending task that requires constant vigilance.

8. There is no relationship between the cost of security and its effectiveness.

9. Great computer security is a result of design.

10. When designing great computer security, it is critical not to forget things like wires, simple physics and most important of all, the users.

11. A $1,000,000 computer security system protected by a $20 door lock is only worth $20.

12. Intelligent people are always available, on both sides.

Radatti's factors of the Rules of Computer Security

1. Open networks invite attacks.

2. Formally prepare for a security breach before it happens.

3. Authorized users can and will perform unauthorized actions.

4. An exhaustively tested system is already obsolete and insecure. Constant vigilance requires a method of constant improvement without months of testing.

5. Expect attackers to try easy but unexpected ways first. That includes physical entry and social engineering.