CASE STUDY : PONY ENTERTAINMENT DESIGN A NEW MONITORING SYSTEM

Braden Simpson
braden@uvic.ca
V00685500

January 31, 2013

1 INTRODUCTION

This report will describe analyze and investigate the components of Pony Entertainment, and then describe how a new electronic monitoring system should be created. The monitoring system will cover all aspects of Pony, and will have multiple components: *Real-time Monitoring*, and *Retrospective Monitoring*.

2 BENEFITS

This monitoring system will provide many benefits to the organization, including:

- Stronger sense of security – Public advertisement of the monitoring system's existence will boost the sense of security among users of Pony's systems
- Certification – Once the company has fully implemented the monitoring system, it can be certified for security standards compliance, once again, boosting public image
- Compliance with law – When dealing with as much user information as Pony does, it requires them by law to be able to give some information about users that would not be possible without a monitoring system
- Accountability – Monitoring provides the users with accountability and non-repudiation, making people more responsible for their actions

3 Prior Requirements

Creating a large system such as this will require a large amount of sponsorship and support to get it going and keep the project from being put behind in priority. Firstly, the project must have a **High level sponsor**, such as a C-level executive or someone with power in the company. As well, there must be an accomodating amount of physical security and resources for the teams to implement the project.

Then there must be a small team assembled, with experts from all aspects of Pony's systems. This team would then perform a *Gap analysis*[1], this creates the roadmap for the project over it's lifecycle. The project uses the goals described in the analysis to evaluate it's performance.

Next, there needs to be a team assembled to implement the monitoring system, which would scale in size with the amount of systems and work that has to be done.

4 Implementation

4.1 Previous System

First, any existing monitoring systems must have their performance evaluated. If they are easily to integrate into the new monitoring system, they should be kept, otherwise discarded. The information from the existing systems is still valuable, because it would potentially reveal mistakes that should be addressed in the new system.

4.2 Environments

The different environments that Pony's systems run in all have to be monitored, this will require specific experts from those environments. The systems should have their systems collecting data, which is then relayed to a central data manager, which is then piped back to the main system. This is an example of the multi-tiered system shown in the presentation slides[1].

The environments should have rigorous firewall protection and network monitoring, using routing technologies to detect intruders and malicious users. As well these networks shouldn't only be watching the traffic from outside, they should be tracking internal users as well, since they are also potential suspects.

4.3 Data Collection

There are many methods of collecting data for monitoring systems, but some of the ones that can be useful is simply logging all of the traffic routed between all the

[1] Gap analysis is a tool that companies use to compare their actual performance with their potential performance. http://www.ahrq.gov/qual/qitoolkit/d5_gapanalysis.pdf

servers and systems that Pony runs. That information would be organized and inserted into databases for further analysis later on. Data collection is a very delicate process, and all the laws of privacy must be tightly followed, more about this is talked about in Section 4.4

4.4 Storage and Law

Internet privacy is a very prevalent area right now in law courts because of the rapidly changing ideas of user's rights to privacy on the internet, as well as the changing methods of communication which are really easy to log. The attributes stored must be carefully chosen, and if there are any sensitive attributes (identifiers, credit cards, etc.), they must be encrypted.

4.5 Analysis

Analysis would be done on the data after it has passed through the collectors and into the databases. Different types of analysis can be done using programs like netflow[2]

4.6 Real-Time

Real-time monitoring is a way to instantly detect problems or intruders with Pony's systems, as they happen. To implement a system such as this, there are a few considerations to make:

- Choose a smaller set of attributes to monitor, as there will be much more data to parse through as it comes in. There can be no downtime to parse data, the system must keep up to the input.
- Use programs that automatically parse data and check for inconsistencies (large amounts of credit cards being accessed, user info being copied, etc.)
- Use network access tools to detect unwanted connections

Once all of this is in place, the real time monitoring can run separately from the other monitoring systems, parsing though data as it comes.

5 Conficker

Conficker is a computer worm that was first noticed in November 2008. It infected millions of computers and was a very large issue for many organizations, including the University of Victoria.

[2] Program which allows easy filtering and textual analysis of network traffic.

5.1 What is Conficker

Conficker was a very advanced computer worm, which used over 20 different vulnerabilities to infect as many computers as possible. It targeting Windows operating systems, and used exploits such as Dictionary Attacks[**?**] on password. The ultimate goal of Conficker was to create a really large Botnet[3], and it was very successful in doing so. Conficker infected millions of computers in over 200 countries[**?**]

Conficker's infection vectors were very large, the sheer number of exploits used, made it very difficult to remove. Some ways infection was propagated was through HTTP downloads, LAN networks, removable media, unsecured shared folders.

Some of the consequences included, locking users out, killing their malware detection programs, blocking certain DNS lookups, stopping windows from updating, and more.

5.2 Other Prior Worms

5.2.1 Blaster

Blastia was a worm in 2003 that came from a chinese cracking collective named XFocus took a patch that was released by windows, and reverse engineered it, therefore revealing the exploit that was fixed in that patch.

Blaster's intentions were to attack Microsoft itself by causing a DDoS attack on windowsupdate.com. Luckily for microsoft windowsupdate.com simply redirected to windowsupdate.microsoft.com, so the damage was not all that severe. Other outputs of the worm were messages "Billy Gates why do you make this possible? Stop making money and fix your software!!!" Also, users were faced with unwanted restarts of their computers due to the Remote Procedure Call service crashing due to the exploit.

5.2.2 Welchia

This was a worm that is described as a *helpful* worm. It exploited a vulnerability in the Remote Procedure Call service in Microsoft Windows, then downloaded patches to fix the vulnerability. Finally Welchia would automatically remove itself.

5.2.3 Sasser

The Sasser worm was noticed on April 2004, which used an exploit in Microsoft Windows Local Security Authority Subsystem Service(LSASS), which gave it the name Sasser. It was created by a German hacker named Sven Jaschan.

[3] Botnets are computer networks where infected computers can be given commands from a master computer. A common use of botnets is to perform Distributed Denial of Service attacks.

Security analysts speculated that the worm was reverse engineered from a Microsoft security patch, similar to Blaster.

The consequences of Sasser were not as bad as most worms, it only propagated itself and shutdown the user machines.
This worm was made to fix the vulnerability of the worm 'Blaster', described in Section 5.2.1.

5.3 After Conficker

Computer worms may have not gotten quite as much press as they used to, but they continue to be a big problem. The Botnets still exist, just the symptoms are not seen as often. Hacking is more difficult for small time hackers like Sven Jaschan, but it has become more of an option for organizations and governments.

Stuxnet was probably the most powerful worm after Conficker, which targeted nuclear silos in Iran. It was a weapon, a worm with a purpose. The hackers aren't gone, they are just more professional now.

References

[1] S. Radin, "Presentation Slides, UVic Software Engineering," Victoria, BC, Canada. [Online]. Available: http://www.ece.uvic.ca/ henrylee/2013/04/04Monitoring.pdf

[2] S. international, "SRI international, Technical Report." [Online]. Available: http://mtc.sri.com/Conficker/