University of Victoria, Software Engineering

Pony Entertainment and Conficker Case Study

Jordan Ell
V00660306

January 28, 2013

1 Pony Entertainment

This case study will focus on the scenario presented in class of Pony Entertainment. Pony has identified that security compromises have been attacked and identified by anonymous sources online. The intrusion has targeted areas such as DMZ, database servers, and sensitive user data. I have been tasked with improving security through a monitoring and logging system. The following subsections describe the preparation I will take, the logging systems which will be used, and the monitoring processes which will be put into place.

1.1 Preparation

The first step of preparation is to analyze the weaknesses of the current infrastructure as well as where previous attacks or exploits have been found to occur. This will create a type of baseline for what needs to be solved. This analysis also includes ensuring that the internal infrastructure of networks and company policy are secure. While protecting against external threats, internal threats remain if this is not looked at or taken care of. The second step of the preparation phase involves people inside the company. Here we are looking for 3 targets. One, ensure that we have employees who can handle any of the new procedures we throw in. This includes both technical people (perhaps a dedicated security IT team), as well as HR people for policy training if the technical items we wish to put in involve all employees at a procedural level. (This could be logging of individual machines or self reporting of suspicious activities.) Two, we need to ensure the company has sufficient funding as this procedure may involve large scale time effort of

employees (which will effect salaries). Without funding this project is dead in the water. Finally, we need management buy in from the managerial and corporate levels. Security is ultimately a business strategy and without the consent of the business operators, upgrading the monitoring process will ultimately fail.

1.1.1 Logging

For logging I have identified several new procedures which should help Pony Entertainment notice intrusions before being notified, notice these activities in real time, and be able to assist law enforcement in legal prosecuting actions. My first step is to log network activity. This can be accomplished by installing either Netflow or Wireshark at all external gateways that control traffic from external to internal facilities that Pony may have. Netflow will allow suspicious activity over the network to be logged in a concise manor of data download and upload to external computers. This allows monitors to check suspicious ports being accessed or if certain IPs are overloading the systems, etc.
The next step in logging is to log internal servers to Pony and their activities. This can be accomplished through the use of Syslog (or Windows equivalent given Windows servers). Syslog logs activity on a per server basis, tracking items such as log in attempts, failed commands etc. This will allow Pony to check for suspicious activity for persons who have direct access to the internal servers. This will also improve accountability for actions taken in case of emergency or in case of errors on said servers.
Pony also identified databases as a critical point of attack for intruders. I suggest a combination of the previously mention Syslog as well as a separate database log. A database log keeps track of connections to the database as well as specific actions taken while inside of the database. This can allow the tracking of suspicious activity inside any given database such as editing user accounts, changing passwords, or tampering with monetary values. Aside from the database logs, I also suggest some form of safe data storage both on and offsite. Onsite may be as simple as installing CCTVs inside of server rooms while offsite may be using companies such as Iron Mountain and performing audits on a regular basis.
Finally, if employees of Pony entertainment as given personal laptops which are used outside the office (chance of being lost or stolen), I suggest local event logging software on each laptop. The local event logs should be sent to a server inside the company. These logs will allow monitoring of stolen or misplaced laptops to ensure that they are not being used in a malicious way.

1.1.2 Monitoring

Through the tools mention in the section above, we have created an environment which will be tracked in almost ever important action. We now need policies and tools for allowing Pony employees to monitor the information that is being collected in an efficient manor. For monitoring of logs being generated, I suggest two items. The first is email monitoring. In most Unix environments, logs can be emailed to technical staff (IT or security team) for further analysis. This can be easily set up through the use of the cron

tab. The cron tab can be configured to email these employees on a regular (I suggest daily) basis for analysis. However, these logs can be enormous and time consuming to go through. This is why I suggest my second item with the use of log parsing tools such as awk or grep or other commercial tools. These tools will allow the emailing of only certain parts of a log file. This may be only security notifications of warning or higher. These tools allow for employee analysis of only critical notifications that are coming from the log.

The second monitoring process involves the network logs from Netflow. Again the procedure above of emailing and parsing should be used, however, and additional step should be taken by Pony employees. The team in charge of monitoring should keep an updated black list of dangerous IPs which should not be allowed to access the network. This will allow blocking some malicious attackers from repeated attempts at breaking into Pony systems.

Finally, some steps should be taken to ensure these new logging and monitoring systems are preforming as expected. One, an initial baseline should be taken (as previously mentioned) as well as regular audits to create a sense of then and now. See where our security measures have come from as well as where they are heading to ensure that we are meeting goals as well as obtaining results. Lastly is to ensure accurate time is being kept between all logs and servers. The time of servers is critical to establishing a chain of events and may be of critical importance to any legal prosecution which may be caused by attackers.

2 Conficker

This section of the case study will focus on the worm known as Conficker. Here, I will outline the background of the worm (what is it, how does it work, etc.), describe 3 major worms prior to Conficker, and any new worms.

2.1 Background

Conficker is a computer worm which targets the Windows operating system. Conficker uses flaws in the Windows security architecture to gain access to administrator passwords which are then used to add the infected machine to a botnet. Conficker was detected in November of 2008 and infected millions of computers, some of which were government, business, and home machine in over 200 countries.

Conficker used malware techniques in order to become such a successful worm. While most malware techniques and procedures are known to anti-virus researchers, Conficker's advantage was that is used so many malware techniques which made is hard to stop as all would need to be dealt with. Conficker's developers are also thought to be monitoring anti-virus patches and efforts in order to consistently update the worm in order to propagate it further. Conficker propagates itself by taking advantage of a network service (MS08-067) in the Windows platform. Once this weakness is exploited, the infected machine becomes apart of the Conficker botnet and can be used and controller by the creators of Conficker.

Once s security breach has been accomplished, Conficker (one of its variants A,B,C,D,E) uses tools such as bugger overflows to execute shellcode, HTTP servers, attaching DLLs to running processes, sharing itself by infecting removable media or over network connections. Through these procedures, the computer's virus can be updated as well as controller by the creators. Conficker also takes measure to protect itself such as encrypting its network payloads, re-configuring system restore points, and disabling services such as automatic updates and security centers.
Direct consequences of the virus include: account lockout policies being reset, user accounts being locked out, anti virus websites being locked out, domain controllers, slow response, slow local network, Microsoft services not working. Some of the known damage caused by Conficker include: causing the French Navy to not be able to download flight plans, The United Kingdom Ministry of Defense being infected, the armed forces of Germany having infected machines, and the Greater Manchester Police being disconnected from their internal police network.

2.2 Prior Worms

Here, I will describe three computer worms that were prior to Conficker: Badtrans, Code Red, and the Kak worm. I will describe how these worms spread as well as what damage each worm was known to inflict.
Badtrans was a computer worm which attacked Microsoft Windows machines. BadTrans was primarily distributed by email because of a known issues with older email programs. The problem was that these email clients would oftentimes run attachments as soon as the email message was viewed as opposed to waiting for the user to execute an attachment. Once Batrans has been run, it propagated itself by emailing itself out to all email addresses it could fine. Badtrans main attack was the installation of key logging software. This software logs the keystrokes of individuals and then emails this information out to the creators.
Code Red was a computer worm which attacked machines running Microsoft's IIS web server. Code Red exploited a security vulnerability known as a buffer overflow. The worm essentially just used a very long string in order to allow the worm to execute any code it wished on the machine. The damage caused by Code Red was a defacement of the server's web sites. The website would display the text "HELLO! Welcome to http://www.worm.com! Hacked By Chinese!". Infected machines were also know to launch DOS attacks.
The Kak worm was a computer worm which attacked machines using JavaScript and propagates itself by using Outlook Express. The worm was most known for causing an automatic shutdown of the infected computer on the first day of every month at 5:00 pm. The machine also often displayed fake alert messages to the user.

2.3 Post Conficker

Computer worms continue to be an issue post Conficker. More and more sophisticated malware techniques are being introduced and taken advantage of on a daily basis. Ar-

guably a bigger worm than Conficker, Stuxnet, has been even more damaging by taking control of PLC units in industrial settings such as power plants. Computer viruses will always continue to be a problem as it is near, if not completely impossible to write always secure code.