The Jing An Telescope Factory (JATF): A Network Security Case Study

Doug White
CIS Department
Gabelli College of Business
Roger Williams University
Bristol, RI 02809, USA
dwhite@rwu.edu

Alan Rea
BIS Department, CIS Program
Haworth College of Business
Western Michigan University
Kalamazoo, MI 49008-5412
alan.rea@wmich.edu

ABSTRACT

This case—an examination of a real world break-in to a Web server—provides a forensic examination of what happened to the Jing An Telescope Factory (JATF) and a suggested model for preventing such attacks. The case specifically focuses on the "hack" break-in that is commonplace with Web servers and illustrates the well-known mistakes made in the security arrangements by JATF. Select hacking techniques and an overview of network vulnerabilities, as well as discussions about tools and techniques that security professionals use are discussed in this paper. The authors propose a set of techniques and models that business should follow to guard against similar attacks. Students are encouraged to assess and implement solutions using the tools and techniques presented in the case.

Keywords: Network security, network assessment, hacking techniques, system hardening, case studies

1. CASE SUMMARY

This case—an examination of a real world break-in to a Web server—provides a forensic examination of what happened to the Jing An Telescope Factory (JATF). Following a discussion of what happened to JATF's network, students are presented with common hacks and network vulnerabilities, as well as discussions about tools and techniques that security professionals use to prevent and analyze attacks.

In the classroom and networking lab, students are encouraged to explore how JATF's network was compromised. They must apply the tools and techniques discussed in the case to create a new network diagram that incorporates network security design and protocols to prevent additional attacks and protect data. This case specifically focuses on the "hack" break-in that is commonplace with Web servers and illustrates the common mistakes made in the security arrangements by JATF.

2. ABOUT JATF

The Jing An Telescope Factory (JATF) is a medium-sized business located in Ninjing, China. The factory employs about 250 people. Out of these 250, about 25 are directly responsible for Information Technology (IT) Operations in the areas of networking, Web development, database management, and other typical IT operations. The network security breach discussed in this case occurred during the summer of 2002.

In this case, we'll first discuss the existing network architecture before the security incident. Then, we'll discuss reasons why the incident might have occurred. We'll finish with sample consultant recommendations. It will be up to you or your team to write a recommendation as well as design new network architectures for increased security and data protection that JATF can implement.

3. JATF NETWORK ARCHITECTURE

JATF maintains a large network interlinking intra-building departments and inter-building operations systems. A wide variety of servers and workstations exist on the network and most employees have workstations on their desktops. The particular server of interest was running Windows NT 4.0 (Chinese), and was using the Internet Information Server (IIS) to serve Web pages to a private network. JATF's network was behind a firewall preventing all access from the Internet to the internal networks of the company. This included the Main Web Server (MWS) that was hacked. The MWS was connected to the primary intranet of the company via a Cisco switch. Employees of the company had access to the Web server pages via the intranet, but not the Internet.

3.1 JATF Network Services

When the break-in occurred, JATF's MWS was running a variety of services, including IIS as a primary Web HTTP daemon and FTP. Although other services were in use, they played no role in the break-in so are not discussed here.

The primary security issue was with the FTPD service. FTPD is an application layer daemon supporting the file transfer protocol which allows the exchange of files between two machines. FTPD is an old service but is still widely used. Along with TELNETD it is considered one of the more dangerous protocols in use because it can be easily misconfigured, can run for anonymous users, and sends packets that are unencrypted. At JATF, FTPD was run as an anonymous login type service where users could login to specific directory structures for uploading and downloading files without identifying themselves with a login or password. While this is not a safe practice, many companies with only internal users opt for this configuration.

3.2 JATF Firewall and Logging

However, JATF's intranet wasn't an open system. To protect its intranet from external traffic JATF used a standard firewalling approach that involved a CISCO IOS based access control list (ACL) to restrict all access to the internal networks from the Internet. Thus, a rule such as:

```
deny ip any any
```

was used on the inbound interface into the network. This rule denies all entrance to the network. The firewall also denied any sort of ICMP (Internet Control Message Protocol), SNMP (Simple Network Management Protocol), or other packets through the firewall by rule. The only rule allowing access from the outside was a TCP (Transfer Control Protocol) established rule:

```
allow tcp any any eq established
```

This rule would allow for the return of packets which had completed the TCP handshake successfully with an outside site. There was a restriction on outbound packets to allow only port 80 HTTP connections and HTTPS port 443 attempts through the firewall outbound.

This is a fairly restrictive set of rules that would not allow any access from the outside easily but would not preclude internal users from downloading attack products (scripts, viruses, etc.) from the Internet. Unfortunately, it was hard to track any downloads because the logging system in place for the CISCO firewall was not saved but merely allowed to stream to a computer screen. When the buffer limit was reached, log entries simply were purged automatically. Because the level of external activity being logged was quite high, the buffer life span was very short (roughly 1-2 hours in the daytime and 5-7 hours at night).

3.3 JATF Backup System

In order to protect its data JATF was using a mirroring approach that duplicated the MWS's hard drive on a regular basis. In this case, changes were noted and updates made hourly to the backup. This type of system can be secure, but must be ***unidirectional*** with the main access point isolated from the Web access point and other internal users. At JATF updates were made on the MWS which was also running FTP and a variety of other daemons that were not necessarily being used at JATF. In particular, TELNETD was also running.

The backup system then copied from the MWS when changes were detected with queries at regular intervals. There was no firewall between the two systems and access was equal on the two machines as they had duplicate systems running. The attackers might have attacked the backup server but their changes would have then been overwritten in the next mirror.

4. SPECULATION ON JATF ATTACK TYPE

4.1 Organizational Situation Influence

JATF decided to invite various IT constituencies within the company to develop their own versions of the Website on company time. At least three development teams were working on variants of the site that they kept to themselves. The prize was being transferred to the Webmaster group which constituted promotions and pay raises for the programmers.

A system was implemented so that the new developers could access resources on a regular basis. JATF did not anticipate the possible complications of allowing anonymous access to both the backup Web server and Internet sites. With anonymous access the possibility of sabotage resulting from the intense internal competition became a possibility.

4.2 Possible Attack Mechanism

It's thought that JATF's network breach and resulting data loss was the result of a ***script attack***. Script attacks are fairly commonplace as they are quite easy to develop. Essentially, the hacker must first compromise the machine to obtain a root shell (meta-user) on the machine to be compromised

(not all attacks require root privileges but this is a common assumption). Script attacks then use various languages to process destructive operations very quickly, typically after the hacker has departed the scene.

Unfortunately, in some systems, root privileges may be obtained without an actual compromise. For instance, a typical process may simply be to find a program that has root access and "export" the attack script into its path. This type of action can often be executed by users with less than root privileges on a system, particularly when a directory has been given permissions for anonymous use.

Script attacks may be executed by other scripts in this fashion where the entire attack is simply loaded into a scheduled job and run long after the attacker has left the scene. Typically, these types of scripts also include attacks on the log files to remove evidence of the attack.

ForeverHack is a fairly obscure (in the West) script attack that relies on a simple compromise of the system to allow access and then the script virus can be run. This virus, developed by a Chinese hacker (foreverhack.net, 2003), operates on any files it has access to that have ***.asp*** (Active Server Pages) extensions. This includes the entire tree of many IIS Websites which have been developed using ASP. The files are replaced with a single Web page that contains the address of the hacker who developed the script. Thus, an attack can reduce an entire Web structure to ashes in seconds.

At JATF, the attacker most likely used anonymous access to FTP into the MWS from some other node on the intranet. This access was then used to install the foreverhack program (script push) as a part of a larger script that attacked the logs. While the system administrators could not recall the settings of various directories, it was possible using FTP access to change directories to areas on the system where a script could be run if the attacker created a script, scheduled the script to run or ran it manually, and then departed. By the time the administrators were able to identify a problem, the damage was done and the logs of the attacker were gone.

4.3 Backup Failure

Compounding the loss of data was JATF's inability to recover data. Like many organizations, JATF assumed its backup system, in conjunction with the firewall protecting it from the Internet, provided adequate means to protect the time and personnel hours invested in the competitive site. Most of the development data was also stored on JATF's mirror/backup server located behind the firewall. JATF's backup model discussed earlier is commonly used because it is simple to configure but creates a false sense of security. Many networks use a mirroring mechanism to protect production data. Unfortunately, in JATF's case, the backup was automated to "mirror" the main Website on a fairly rapid turnaround with no built-in archival functionality.

When the MWS was compromised, the backup server was also overwritten in a timely fashion. This means that not only were the Web files lost, but the logs, which were also maintained on this server, were overwritten by the attacker's script. Moreover, the development data was erased. At the time of detection of the attack, the mirror/backup server was exactly that—an exact copy of the compromised machine.

5. COMMON NETWORK ATTACK TYPOLOGIES AND TECHNIQUES

Even if we suspect that it was a script attack, we must decipher how the hacker placed, planned, and invoked the script. Script attacks use various languages to process destructive operations very quickly, typically after the hacker has departed. Moreover, you must always look at the most common types of attacks when investigating a network break-in. Below we list the three most common attacks and techniques. (**Appendix A** in the **Teaching Notes** contains additional common hacking techniques.)

5.1 Attack 1: The Ftp Malformed Command Buffer Overflow

Although this is a common hacker technique, this type of attack would not likely work in the Windows environment as the shell system is not as vulnerable to this type of attack as in the Linux/Unix environment. Typically in the **FTP Malformed Command Buffer Overflow** (FTPMCBO) attack, the attacker will login anonymously and use the FTP command set to attempt to either send packets that are of odd sizes for the commands or directly manipulate the commands based on the well-known protocol to cause a buffer overflow and ultimately achieve a root shell.

5.2 Attack 2: The Script Push

Misconfigured FTP may allow an attacker to *push* a script onto the system under the radar of the firewall/virus scan. The script also may be pushed into a directory that has been unsecured such as a cron directory in Unix or a directory where permissions are improperly set (the FTP directory). It is often the case that administrators neglect the executable permissions even when controlling other reads and writes. If this is so, the script may install a root kit, run a program such as foreverhack, install a Trojan, delete logs, etc.

5.3 Attack 3: Registry/File Acquisition

Another well-known Windows NT exploit is to acquire the system registry and modify it. It is all too common that administrators focus their security to control FTP and other access directories but neglect to assume that FTP might be used to change directories to unprotected areas. One approach is to acquire the registry and attempt to extract system passwords from this area. It is also possible to *push* a modified registry into the Windows system. The former is possible even with read only access. The registry could be configured to perform a variety of harmful actions.

6. NETWORK SECURITY ASSESSMENT

In all cases the best means to combat network intrusions and prevent data loss is to stop it before it happens. To do this, you must develop a network security assessment tool. In the section that follows, we outline one that will help you develop rudimentary security to protect systems from obvious risk and hackers with malicious intent. We also apply this tool to JATF's original pre-hacked network to strengthen its network security.

6.1 Risk vs. Return

A key element of security must be a decision about ***risk versus return***. For example, consider a server welded inside a safe and then dropped in the ocean. It's very secure, but not very usable. There's no risk but also no return.

A company must decide how much time and effort it can spend on security. However, the amount of risk must drive this decision. Customer data and mission critical systems must be protected. Hackers target high-profile systems and information that may be revealed for value or personal gain. Low-profile systems are targeted as potential platforms for attacks on other systems. Although hackers won't want the information on the low-profile system, there is still a risk of data being gleaned, destroyed, or compromised.

At the very least, a company must focus on securing its network. However, if there is greater risk to a system, more focus needs to be on that particular system. For that reason, a company must rank systems to determine which one is a low- versus a high-risk system.

Ultimately, basic security consists of four main elements: identification, assessment, observation, and prevention. You must consider and address all four in order to remain secure.

6.2 Identification

To begin, identify two things: critical points of entry into the network and mission critical systems. The most-likely entry point is the connection to the Internet. It is also important to identify any systems in use that are absolutely critical to the business. You must protect these systems and identify all possible weaknesses in these machines. For instance, a desktop that is used to store your accounting system may be considered mission critical, while a system used only in the manufacturing shop for printing out orders is not.

6.2.1 System Rank Scale: A simple plan is to list all the uses of a system and then consider the loss of the system. How big will the impact be if the system is compromised or erased? Using two ten-point scales you can develop a rating of security need for all the systems in your network. The first scale is an analysis of the ***critical level of the system***. A score of ten (10) represents a system that is indispensable (e.g., an e-commerce transactions server) and a one (1) represents a system that is connected to the network but can easily be replaced/repaired in a failure. Reserve the zero score for machines that are not connected to the network.

The second scale is the ***risk***. Machines may represent different levels of risk in terms of the amount of access they grant. A ten (10) on the scale might be a machine that is connected to the Internet, allows anyone to access the Web pages, supports FTP transactions, and has remote access. A one (1) on this scale is a single user machine that allows only inbound transactions (e.g., Web browsing). Reserve a zero score for machines that are very low risk such as a single user workstation that is password protected and not connected to a network.

Obviously, the scale is subjective but it should be useful if applied consistently. The scale creates a basic guideline for analysis of systems for security. In both scales, the operating system (OS) being used should be considered as some OSs are weaker than others, particularly dated OS legacy systems.

6.2.2 System Rank Applied at JATF: JATF's network consists of seven machines located in the factory: a Web server, a backup Web server, two administrative workstations, and three user workstations. The systems are identified by their fourth octet IP addresses as .1, .2, .10, .11, .100, .101, and .102, respectively. All of the systems are operating under Windows NT (e.g., a legacy system). The Web server (.1) is used for providing sales data and other information to Web users. The Web server uses FTP to support uploading of Website information. The backup Web server (.2) maintains a copy of the JATF main Website and developmental files. The administrative workstations (.10 and .11) have additional network and systems privileges. The remaining systems (.100, .101, and .102) support only one-way Web transactions and are single user, password protected machines.

In this simplified system the first application of the scale is to identify the mission critical systems. The obvious is the Web server and the backup Web server. These systems are the only components that are necessary for the Web component to function. The administrative systems for the Web operation are not critical. Thus, a quick ranking of the systems might look like Table 1.

The ranking implies that only the Web server is very important and since it would not devastate the company if it failed, the score is not a ten.

In the case of ***risk identification***, the exposure level of the machine is to be considered. As all the machines are firewalled, only the server is a high risk machine. The scores of the machines in terms of risks are shown in Table 2.

Table 1
Mission Critical Scores

SYSTEM IP	SCORE
.1	7
.2	5
.10	3
.11	3
.100	1
.101	1
.102	1

Table 2
Risk Scores

SYSTEM IP	RISK SCORE
.1	9
.2	3
.10	3
.11	3
.100	1
.101	1
.102	1

These scores indicate which machines are of greatest concern and where the security analysis should be focused. It may seem obvious where the risks lie in such a small-scale situation, but formalizing the process will assist you in locating which IP addresses should be watched more closely, particularly when many machines emerge as risks. The administrative systems (.10 and .11) are ranked more highly because, even though they are not connected to the Internet directly, they have high-level privileges and if they are compromised, it could result in serious problems.

6.3 Assessment

In assessment, you analyze the data gathered and develop a risk diagram for the systems in the network. At this point you need to both review the scores created earlier and also proactively assess the situation, particularly for high-scoring systems. The first technique is to develop a total risk score by adding the mission critical score and the risk score together for each machine as shown in Table 3:

Table 3
Total Risk Score

SYSTEM IP	TOTAL RISK SCORE
.1	16
.2	8
.10	6
.11	6
.100	2
.101	2
.102	2

6.3.1 Use of NMAP (Network Mapper) for Assessment: NMAP (Fyoder, 2003) is a scanning tool that provides a serious, professional tool for administrators at no cost. This tool is best used on a Unix-based system, such as Linux. With the assumption that NMAP has been set up and the Linux machine firewalled off from the rest of the system, an initial scan can be run. While NMAP provides many stealth features for disguising scans, none of these features will be covered here as the assumption is the administrators are scanning themselves and have no need to hide their legitimate activities.

It is also worth noting that NMAP is available with add-ons, such as NMAPFE, which provide graphical interfaces for using NMAP. There also is a Windows version available. A simple self-scan from NMAP can be performed any number of ways, but a basic scan might look like this:

nmap –sT –vv –O localhost (assuming localhost is defined as the loopback address of 127.0.0.1)

Figure 1 provides the return of the scan for this basic machine. The scan reveals a great deal of information about this system. It illustrates two things: 1) what the system looks like when a would-be intruder scans the system; 2) any unusual or unneeded services that may be running. Even though this system is firewalled off, there are still ports open that may be attacked from users inside the firewall. (Appendix B in the Teaching Notes provides a list of well-known ports and what they are typically used for.) It's important to know each port's function so that you can identify which system services are running at each open port. Conversely, you should know when a port should not be open.

The most critical information provided by the scan is the examination of open ports. In this case, the machine has eight open ports that may be running services that are in use, or perhaps the administrator has simply failed to disable unused services that are set up by default.

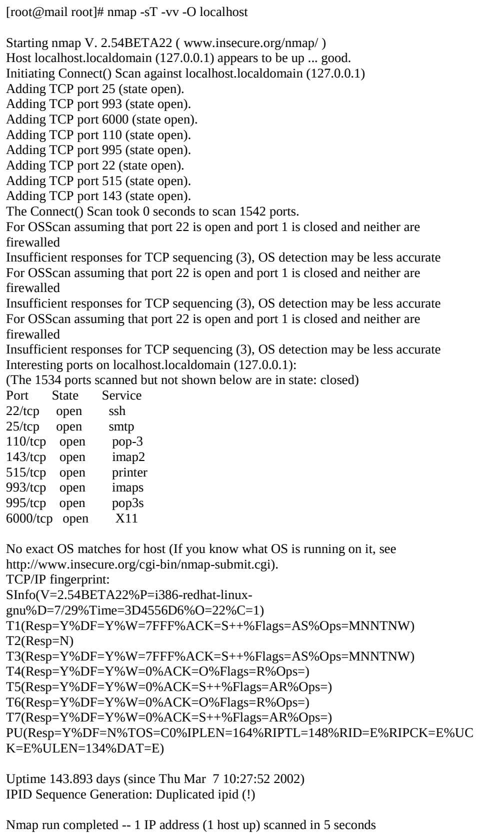
6.3.2 Working with the Ports: All of these ports can be Trojans or other hacking tools in disguise. NMAP simply reports the most common usage of the ports. The fact that NMAP says “printer” does not necessarily mean this is actually a printer port, it merely means that this is the most common usage of port 515. Many Trojans intentionally use common ports to avoid detection through misdirection. The best rule is to disable any service you are not using. If the corporate network administrator feels uncomfortable with this approach, the next best approach would be to log all activity on the port and see how and if the port is being used.

The remainder of the NMAP scan provides some information about the operating system. As Figure 1 illustrates, NMAP is always trying to collect fingerprint information to better discern which operating system is being run. This is useful only in regard to the failure of

NMAP to identify the operating system and the warning that IPID (Internet Protocol Identification) scanning is possible (this is a subtle form of systems probing for information).

Creating a script to automate the scan on a regular basis is a very good means of keeping an update on your servers. You can create scripts that email you a scan of all your servers once a week. You will quickly develop a “feel” for what your servers are running and a change should be obvious without a great deal of scanning of logs.

Figure 1
NMAP Scan



```
[root@mail root]# nmap -sT -vv -O localhost

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host localhost.localdomain (127.0.0.1) appears to be up ... good.
Initiating Connect() Scan against localhost.localdomain (127.0.0.1)
Adding TCP port 25 (state open).
Adding TCP port 993 (state open).
Adding TCP port 6000 (state open).
Adding TCP port 110 (state open).
Adding TCP port 995 (state open).
Adding TCP port 22 (state open).
Adding TCP port 515 (state open).
Adding TCP port 143 (state open).
The Connect() Scan took 0 seconds to scan 1542 ports.
For OSScan assuming that port 22 is open and port 1 is closed and neither are
firewalled
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
For OSScan assuming that port 22 is open and port 1 is closed and neither are
firewalled
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
For OSScan assuming that port 22 is open and port 1 is closed and neither are
firewalled
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1534 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
110/tcp    open        pop-3
143/tcp    open        imap2
515/tcp    open        printer
993/tcp    open        imaps
995/tcp    open        pop3s
6000/tcp   open        X11

No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=2.54BETA22%P=i386-redhat-linux-
gnu%D=7/29%Time=3D4556D6%O=22%C=1)
T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UC
K=E%ULEN=134%DAT=E)

Uptime 143.893 days (since Thu Mar  7 10:27:52 2002)
IPID Sequence Generation: Duplicated ipid (!)

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
```

6.3.3 Ethernet Sniffing: Ethernet sniffing has declined in popularity with the rise of switched as opposed to hubbed networks. Unlike hubs, which broadcast all packets across the network, switches *usually* filter broadcasts so sniffing is only a useful tactic for hackers if they can get close access to devices they wish to sniff.

Sniffing can be used to locate weaknesses, particularly weaknesses implemented by users on their own systems. System administrators can forward packets to a “sniffing server” or simply utilize a laptop to sniff any of their networks.

The most common problem areas for sniffing are services that send packets *in the clear*. The most notorious of these services, Telnet, FTP, SMTP, HTTP, etc., are extremely dangerous as anyone on the network can gather packets and sort them out to ascertain user names, passwords, account information, etc. The packets contain sequence numbers to allow them to be sorted out and many programs exist that are specifically designed for collection of this information (e.g., DSNIFF [Song, 2003]).

You should consider sniffing approaches to assess networks for weakness. This is a more time-consuming approach than an automated scan. However, it can quickly reveal services in use that may result in simple compromises by hackers and the use of unauthorized services across the network that may result in huge security risks. A common risk occurs when end users add a service—such as Telnet or FTP— to their workstation for their own convenience. Both of these services have known exploits. If a hacker can gain root access to a workstation by simply sniffing a password from an established connection, then there is no need for complex exploits. The hacker can simply log in as the legitimate user and internally access your network.

6.4 Observation

No matter what security you have implemented, you should still continually observe your systems. Observation is the means of logging the activity of both users and potential threats on the system. Logging, like much of security, is often neglected for a variety of reasons. The most common reason is that logs are often too voluminous to effectively review. Software is being developed to help interpret and analyze logs, but a tailored log is still the most effective method of observation.

6.4.1 Creating Logs: As noted above, logs must be tailored to each specific system. Typically, system administrators use the default logs in both Windows and Unix-based systems. This leads to ignoring the logs and/or logs that are too large to be useful. A more advisable approach is to create logs against the total risk scores of the machine as discussed earlier. First, identify what is “normal” for a system. Then create logging scripts that focus on atypical activity rather than common use.

It is certainly possible to have a difficult time describing what is “normal” and some servers may exceed the bounds of definition, but typically a server has only one or two functions assigned to it and the activity that may be normal is easy to define. Let’s use the example of an email and Web server. The email is SMTP and Secure POP-3. The Web server operates on port 80. This means that given no other needs on the server, all the activity would be on ports 25, 80, and 993. (See **Appendix B** in the **Teaching Notes** for additional port information.) One of the first things to log is any attempts to connect to other ports. Assuming you have a firewall, this should be minimal and any scans from internal systems should be a serious warning that the system is being examined. It is worth noting that some internal users may scan systems for fun or because they read a book like Meinel's ***Happy Hacker*** series (Meinel, 2003). This type of activity should be barred by policy and violations should generate a stern warning.

A second activity to be alert for is failed logins on servers. Normally servers should have few system logins (however, the admin will have to make this determination). Look for multiple attempts on machines where logins are uncommon. This is often a prelude to an attack or at least a hacker attempting to find a weak password using a cracking program such as John the Ripper (Openwall, 2003).

6.4.2 Securing Logs: Logging also requires a secure approach if logs are to be effective. A hacker who has root access to your system can easily delete or modify logs to remove any evidence of activity. Two basic approaches should be considered as a means of logging servers.

Use a logging server behind the firewall to receive log information via TCP/IP. If this server allows only connections from the machine it is logging and then only to receive packets, it will require that the hacker then break into the second machine. If the logging server does not accept any sort of connections except logging packets by some secure means, then this becomes another major exploit.

Secondly, an old-fashioned but effective strategy for logging is to simply log all the entries to a hard copy printer. Unless the hacker can get physical access to the printer, it will be impossible to change the logs. This is often the task of an outdated printer that has a built-in print-server.

6.4.3 Using Logs: Logs need to be usable. In other words, they should be to the point and clean. Eliminate all clutter and common, legitimate activity. If the log becomes too voluminous it will be unusable and ignored. The downside to cleaning up logs is the loss of usefulness if attacks occur due to missing information. It may be worth considering a staged logging where certain events trigger a more elaborate logging process of all activities of a certain connection.

6.5 Prevention

Observing system activity with customized logs can help you determine when a system is under attack or functioning outside of established parameters. However, preventing attacks before they reach a system is, of course, the best security measure. For this, you should rely on firewalls as a line of defense.

6.5.1 Firewalling Basics: Basic firewalling should be in place for all users connected to a network. Firewalls may be either localized system specific firewalls or network-based point-of-entry type firewalls. The discussion of firewalling is generalized to either type of firewall being used. It is recommended that all systems maintain some type of firewall regardless of the total risk score the system earns.

As with the assessment approach above, you must identify which services are in use on a given network (for the network-level firewall) and the system (for the client-level firewall). At the network level it is critical to identify common weaknesses that are well known and eliminate the ability for external interests to penetrate the firewall. The most common attempts are scans of known ports where the would-be attacker is looking for IP addresses to attack. Attackers will very commonly scan for port 21, 23, 25, and 80 searching for FTP, Telnet, SMTP, and HTTP servers, respectively. As opposed to an open system, it is recommended that the firewall block all attempts to connect to systems unless that service is being supported across the Internet. It is also useful to block all connection attempts to non-existent IP addresses as the lack of information (e.g., finding a system that is NOT there) is also informative.

6.5.2 Standard Approach: The standard approach is to block all packets at the firewall and only allow packets specifically identified in the firewall. This approach is standard in Cisco ACL (access control list) firewalling and all Cisco access lists contain an implicit deny as the final statement. This has the effect of excluding every connection unless a rule is written to allow it. As these rules create a choke-point on the network, this is an area where methods to prevent denial of service attacks also need to be implemented. In other words, a choke-point is a location where all packets must clear. When rule sets examine packets on a network, every single packet must be processed against the rule set (such as a firewall or IDS). As the number of rules in the choke-point grows, the processing power needed to process the packets grows as well. This is what makes it a likely target for denial of service attacks. In the case of small servers and enterprise networks, such as JATF's, the threat is much more real as low-end hackers often target these less defended networks.

The typical approach is to develop firewall rules that focus on rapid connection attempts to multiple ports and then either allow the source IP to penetrate the firewall for a brief period of time (as these addresses are usually spoofed, the attacker receives no return packets) or redirect the connection attempts to a "black hole" location which filters the packets. For any enterprise that relies on Internet connectivity for business purposes, it is critical to take some approach to identify DOS and deal with it before the attack can crash either the router or the server that the attack is directed against.

Firewall rules should allow only packets to specific services, not simply an open channel to the IP address. While this has the effect of creating additional rules, it prevents attackers from examining the system more thoroughly. It also eliminates the risk of Trojans being placed in the system and then being accessed from outside the network. A similar firewall on the local machine will further reduce this risk by controlling internal access as well.

6.5.3 Trust Building Approach: A second approach to firewalling involves building trust for users. This is not possible (at least not easily) for port 80 Web servers and other connections that require anonymous access, but it can be used to authenticate users from outside the network. Trusts may involve either VPN (Virtual Private Network) type identification or ticket-based authentication to allow access either on a per user basis or a per machine basis. Both types of connections create encryption between the client and the server as a means of further securing the operation.

Most routers and firewalls support a wide variety of methods for authentication. The current trend is toward using VPN software to allow users to authenticate their connection via a username and password before gaining admission to the network. Connectivity may still be managed on a user-by-user basis and users should be granted no greater access to ports than is needed by their job definition.

Kerberos systems (MIT, 2003) create encrypted “tickets” or keys on a given machine and then grant that machine access. This method is more seamless than the "per user" method but creates a greater risk if the machine is being shared across many users. Regardless, this approach allows users to login from their laptops without having to provide their username or other information across the Internet.

The more secure approach is to authenticate each user as this requires entry of the information on each different connection. This allows for better logging as the user is identified and can be managed on a need basis. None of these methods protect against social engineering and/or theft of a laptop with logins saved. (The Windows OS is notorious for this type of security risk since it saves all passwords and IDs in the registry if the user so chooses.)

6.6 Secure Backup And Failsafe Approaches

No matter which firewall configuration you chose to use, remember that it's not a complete solution. You must also make sure that data is protected in case it's lost or compromised. JATF had a firewall installed with Cisco ACLs in place and functioning at a basic level. However, a crucial flaw was the failure of the backup media to provide a reasonable solution to the hack that destroyed the systems. Thus, a key component of any security assessment is to find a way that is safe and reliable to back up the data on the system that is at risk. Let's look at some data backup methods on various system types.

6.6.1 End-User Systems: End user computers typically do not have servers and services running on them, but must be backed up nevertheless to preserve user data. Currently the most reliable method for backup is the use of CD-R or CD-RW technology to “burn” copies of the system at various times and store the images in a safe location (e.g., a fire safe). It is important to make backups on a regular basis, but end users have a great deal of difficulty defining a “cycle” or other means to determine when the backups should be created. Thus, the recommendation is to back up dynamic data at least once a week on a permanent media such as CD-R. In a more advanced network setting, a scheduled network storage backup can be used to automatically save data. However, many organizations do not have this functionality in place.

6.6.2 Servers: Obviously servers should be backed up as well. The preferred method is to define the server's business cycle (such as a business day, week, month, etc.) and use that cycle as the starting point for backup management. By storing images based on the cycle on CD-R (or RW) technology, you can maintain images across a longer time horizon (a year or two). This means that a business may burn an image of dynamic data on a daily basis or whatever cycle it determines to be the key cycle for its systems. Typically in an assessment, the cycle would be determined by the length of time that could quickly be recreated using other means, such as paper invoices. Ideally, the most recent images should be stored offsite in order to protect them in the event of fire or other natural disaster befalling the location.

6.6.3 Mirror Server Backups: The creation of a true support mechanism for your server comes in the form of a mirror backup server. Hacks and other types of attacks are usually most damaging in the short run. Recovery, particularly for enterprise Web business, is critical. At JATF, the company had developed a “pull” mirror server which on a regular basis duplicated the production Web server's image onto itself. This updates the backup server with any changes that have been made. The problem is that in the compromise of the main Web server, the backup is compromised by default with no effort on the part of the hacker.

A better solution is a “push” backup. In this approach, the production Web server is placed out in the exposed position for all to see, but the main source of the server’s data is the backup server. This backup server can then be rapidly mirrored onto the production Web server as needed and the main system images stored only behind the firewall on a heavily defended machine that allows no access from the Internet. Should a hacker compromise the production Web server, the backup image can simply be written through a VPN-type connection through the firewall to the production Web server. The backup server should never accept any connection from the production server except for an authenticated backup request.

In this model, main breakpoints in the development cycle should also be burned onto permanent storage, such as optical disks. All development work should be done on the backup, behind the firewall or in a VPN scenario with authentication of individual users from outside the firewall. This authentication should be handled very carefully and logged thoroughly to prevent hackers from using this hole in the firewall to attack the backup server. The preference would be to maintain the operation behind the firewall and only allow file and system updates internally. Users would have to come into the local network to provide updates to the backup server.

7. POSSIBLE NETWORK SECURITY AND DATA PROTECTION SOLUTIONS FOR JATF'S SITUATION

Now that you know the basic security items that you need to consider when hardening a network and data from hackers, let's look at what the consultant recommended for JATF's network. As you look at the solution, take note of which concepts and techniques the solution incorporates. Also, consider how you might improve the solution. Remember, there is definitely more than one way to secure a network.

7.1 Replace FTPD

FTP is an old protocol (RFC 454, 1973). One of the most obvious yet overlooked solutions with FTP is to update the client (FTPD) to the most current version of FTP (based on RFC 959, 1985). However, even this approach does not guarantee security using FTPD as a protocol.

As was discussed earlier, one of the easiest approaches for a hacker to perform this attack would be to use Ethernet packet sniffing to procure a password to the system. As in most organizations' legacy systems, JATF failed to consider packet sniffing as an issue in its security. JATF, feeling secure in its firewalled environment, created the perfect environment for a classic "locked room" crime by not monitoring its network traffic.

The first recommendation by the consultant was to consider the need for FTP; in particular, the need for anonymous FTP as a service. While there are certainly anonymous FTP sites in existence, they need to be controlled. The consultant could see no reason why the company needed to be able to anonymously connect to the server to transfer files since only two people were doing the transfers and they both had access to the server and client doing the transfers. Thus, the first recommendation for JATF was to add user accounts for file transfers so that logins could be monitored and controlled.

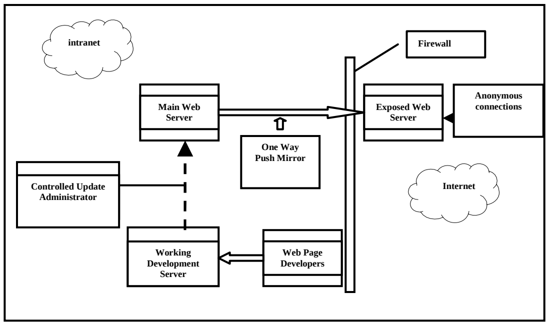
Unfortunately, this in and of itself simply opens the door for packet-sniffing-based attacks because FTPD is still a notoriously insecure protocol. Thus, the consultant recommended **migration from FTPD to SFTP** (RFC 959, 1985) as a protocol. This protocol provides an encrypted (much like SSH) environment for packet transfer between machines while maintaining the traditional FTP environment to avoid learning curves and resistance by employees. These two simple changes—**user accounts** and **SFTP**— to the procedure for file transfer at JATF might have prevented the entire attack.

7.2 Create New Backup Models to Prevent Mirror Fault

The consultant recommended a primary model for the development of a reverse mirror. Here the Main Web Server (MWS) is protected behind a firewall and the Exposed Web Server (EWS) receiving Web connections is only a mirror of the MWS as per White, et. al., 2003. This model is illustrated in Figure 2.

Essentially, the MWS is positioned behind a one-way firewall that allows only connections to the EWS. In this manner, the EWS is mirrored on a regular basis by the master copy of this disk being "pushed" through the firewall. If the connection between the two is both encrypted (VPN) and dedicated (firewall controlled), the ability of someone to corrupt data on the MWS is severely limited. In the case of JATF, the anonymous connections are from the outside network and the MWS should be housed behind a separate firewall that allows no outside contact.

Figure 2
One-way Mirror Push Backup [OMPB] (White, et. al., 2003)



If the model described by White, et. al., 2003 is followed, the Working Development Server (WDS) is the only real contact point for anyone besides the administrator who controls updates between the two servers (Controlled Updates). In this manner, the data is heavily protected from both **external hackers** and **internal malicious mischief**. Particular care should be made to limit any access to the MWS, and FTP (even SFTP) should be allowed only from a trusted IP address (the Controlled Update Administrator). All other access should be eliminated.

Furthermore, the MWS (and all other servers) should be "hardened" as a means of eliminating unneeded services and tools which are not used on that machine (White and Rea, 2003). If JATF would follow this network architecture, the likelihood of a repeated compromise is very unlikely.

Even if attacks can be prevented today, possible new attacks and exploits will be found tomorrow. To recover from attacks, system administrators must be able to conduct effective system forensics. For this, you must consider logging, mirroring, and other issues to learn from your mistakes. Hence, the consultant recommended various logging techniques.

7.3 Implement Logging Techniques to Preserve Evidence

Logging techniques vary wildly from administrator to administrator and platform to platform. One of the key problems for JATF was the loss of the logs on the server that was compromised. This is a fairly standard tactic by most hackers. Typically, a break-in will be accompanied by scripts which either "scrub" the log to remove any evidence of tampering or identifying marks from the hacker, or more commonly, the hacker simply deletes the logs from the system.

At JATF the hacker simply deleted the log files on the Windows NT server and thus, there was no record of the break-in. Likewise, JATF did not log at the router to determine internal connections to the Web server, which could have provided additional forensic information about the system from which the FTP connection originated. Granted, it is not difficult to spoof the IP on the attacking machine to either an unassigned IP or the IP belonging to someone not on the system, but regardless, the need for information to determine exactly what happened is critical if the problem is to be prevented from reoccurring.

The consultant recommended that JATF should implement one (or more) of the following logging mechanisms:

7.3.1 Entry Point Logging: This involves generating firewall logs from the point of entry to a network segment. This may not be possible if the segments are switched or hubbed. In the event the network is switched or hubbed, it is possible to log entry by the addition of firewalling tools on each segment, but this may result in great expense and maintenance. In the event the system is routed at that segment, the router firewall should be used to generate a log.

7.3.2 Remote Server Logging: This logging technique is very similar to the tactic of Entry Point Logging described above. However, now the critical issue is to ensure the Web server generates not only an internal log, but also a remote log stored on a secure machine. This machine should be firewalled from the machines or segments being logged and restricted only to log packets being sent from the servers. It is important to firewall this machine off as the hacker will likely acquire the address of the logging server after compromising the machine. Yet, if the remote logging server only accepts syslog packets, the likelihood of the hacker being able to continue the compromise on into the logging server (particularly if it is in a separate segment) is very low.

7.3.3 Intrusion Detection System (IDS) Logging: This is, by far, the most critical type of logging to generate forensic information about break-in attempts. Because IDS logging is passive, it is very difficult for the hacker to detect it or where the logs are being sent. As with the other approaches, it is preferable that the IDS send its information to a separate storage location that is protected. This will require the hacker to perform repeated break-ins across multiple firewalls and segments to compromise systems. IDS can be performed by a variety of tools on most any platform preferred by the administrator. Certainly commercial tools are widely used, but it is also common to see tools such as Snort (Snort, 2003) and Acid (Snort, 2003) being used as low-cost, powerful solutions to the problem.

8. CONCLUSION

If JATF had developed logging solutions as per the above, a great deal of forensic information about the break-in would have been available to the analyst. This would have helped determine the source of the break-in, the nature of what was done, and possibly the identity of the attacker. Logging could also be used to create a strategy that would thwart future break-ins.

JATF is not an exception to the rule. Unfortunately, many networks are vulnerable to the same type of attacks discussed in this case. Administrators and users must educate themselves on how to protect networks and data. Recovery plans must be put into place to trace activity and recover data with minimal loss. Unfortunately, network security is usually not a priority until data is lost. This mindset must be changed.

9. LEARNING OUTCOMES

In this case, you've seen how one company's network was compromised and valuable data destroyed with very little effort on the hacker's part. You've learned how the network was compromised. You've also been exposed to other methods hackers may use to get into a computer network. Most importantly, you've learned how to assess and deploy a secure network to guard system resources and protect data.

Moreover, you've also learned many rules of thumb to keep in mind as you plan new networks and audit existing architectures:

- Backups are not guaranteed if they are not properly protected.
- An exposed Web server will eventually be comprised.
    - All users should be authenticated to use system services.
    - Network security rules and audits should be conducted on a regular basis.
    - Hackers will find holes in a network if they exist.
    - Tools for protecting, as well as exposing, networks are easily acquired on the Internet.
    - Security should never be taken for granted.

10. REFERENCES

Anonymous [2003], http://www.foreverhack.net, [Only available in Chinese.]. Retrieved March 29, 2003. (This site has been removed.)

Fyoder [2003], Exploit World. Retrieved August 22, 2003, from http://www.insecure.org/sploits.html

Meinel, Carolyn P. [2003], Happy Hacker Series. Retrieved August 22, 2003, from http://www.happyhacker.org/

MIT [2003], Kerberos. Retrieved August 22, 2003, from http://web.mit.edu/kerberos/www/

Openwall [2003], John the Ripper. Retrieved August 22, 2003, from http://openwall.com/john/

Network Working Group [1973], RFC 454. Retrieved August 22, 2003, from http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc0454.html.

Network Working Group [1985], RFC 959. Retrieved August 22, 2003, from http://www.w3.org/Protocols/rfc959/Overview.html

SNORT [2003], The Open Source Network Intrusion Detection System. Retrieved August 22, 2003, from http://www.snort.org/

Song, Dug [2003], DSniff. Retrieved August 22, 2003, from http://www.monkey.org/~dugsong/dsniff/

White, Doug, Alan Rea, Lou Glorfeld Jon Anderson [2003], "A Paradigm of Network Security: A Classroom Model." Proceedings of the 24th Annual Decision Sciences Institute Meeting, November 22-25,Washington, D.C.

White, Doug and Alan Rea [2003], “Server Hardening Tactics for Increased Security.” Working Paper.

AUTHOR BIOGRAPHIES

Doug White has worked for The Federal Reserve System, Martin Marietta Energy Systems, and numerous consulting operations including those in China. Dr. White has spent 12 years teaching computer programming, security, and networking at the university level. Dr. White is currently an Associate Professor at Roger Williams University in Bristol, Rhode Island

Alan Rea is an Associate Professor of Computer Information Systems at the Haworth College of Business, Western Michigan University in Kalamazoo, MI. At WMU, Dr. Rea teaches courses in Web Design and Development, Programming, and Server Administration. His current research involves a combination of wireless computing, security, and Human Computer Interaction.

iscap

Information Systems & Computing
Academic Professionals

EDSIG

Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096