LAB5 Mitnick Shimomura

Dennis Keritsis

Abstract

This paper will outline number types of vulnerabilities from the categories of Applications, Configurations, and Active Measures. Based on this, remedies will be discussed.

Keywords: Mitnick attack

LAB5 Mitnick Shimomura

This paper will discuss the Mitnick attack from the point of view of a System Admin. System Admins must utilize multiple tools and understand, holistically, vulnerabilities of a system. Specifically, this paper will discuss vulnerabilities associated with three (3) categories, namely, Applications, Configurations, and Active Measures. Solutions will be proposed respectively.

I. Applications

I (A). Application Vulnerabilities

I (A) (1). Finger

According to the IEEE dictionary, an application is "The seventh and highest layer of the OSI model[1] providing the only interface between the user and the application program." [6]. According to the video there is an application used by Mitnick during Reconnaissance, namely, finger. Mitnick used finger to determine whether or not there was a trust relationship.[2]

I (A) (2). Rlogin

The other Application used was rlogin. The problem with rlogin was it "allow[ed] logins to their account **without any password from trusted IP address**." [3]. There are a number of vulnerabilities there. First, the trust relationship is only based off the IP address, which can be easily spoofed through simple packet forging. Second, there is no password required.

[1] We may or may not be using OSI.

[2] A trust relationship is a configuration vulnerability that will be discussed later, but if a trust relationship does exist, the attack must still do reconnaissance some way.

I (B). Application Fixes

In regards to plugging up the vulnerabilities with finger itself, the trivial case would be to close the port associated with finger, which is port 79 according to the video lecture. In regards to rlogin, it is best that the System Admin require passwords. However, given that that rlogin uses "address-based authentication," it best just to ban the application use of rlogin all together. [2]. This can be done using deep packet inspection because rlogin uses TCP and a firewall wouldn't be able to drop the TCP packet holding rlogin. With the TCP port being left open, SSH sits on top of TCP and it is a better alternative because it uses encryption and password.

II. Configurations

II (A). Configuration Vulnerabilities

II (A) (1). SYN Flooding

Mitnick needed to mute the computer that had a trust relationship with the target server. He needed to do this because he wanted to send his own ACK to the SYN/ACK, using a spoofed IP, instead of the trusted computer. In order to do this, Mitnick need to perform a DoS attack, specifically SYN flooding. RFC 4987 "TCP SYN Flooding Attacks and Common Mitigations" was issued in order to remedy this vulnerability. [8]. It defines SYN flooding as a "half-connection[] that [allows for] no resources left to establish new legitimate connections." [9].

II (A) (2). Morris Attack

As noted above, Mitnick needed to ACK to the SYN/ACK but in order to do so he needed to "guess" the right sequence number, and forge the packet accordingly. At the time, BSD 4.2 has a weakness in its initial sequence number generations. [8]. This vulnerability was not originally discovered by Mitnick himself. Instead, it was found by Robert T. Morris who authored “A Weakness in the 4.2BSD Unix TCP/IP Software.” [7]. This was called the “Morris attack.” [8]. Mitnick had to determine what was the previous sequence number and add 1 when he would send the ACK to the SYN/ACK. Although, the “initial sequence numbers are intended to be more of less random[,] RFC 793 specifies that the 32-bit counter be incremented by 1 in the low-order position about every 4 microseconds.” [2].

II (B). Configuration Fixes

II (B) (1). Common Defense to SYN Attacks

RFC 4987 outlines a plurality of “Common Defenses” to SYN attacks, for example, Filtering, Increasing Backlog, SYN Cookies, and Firewalls and Proxies. This paper will discuss “Recycling the Oldest Half-Open TCB.” According to RFC 4987, this method “works under the assumption that legitimate connections can be fully established in less time than the backlog can be filled by incoming attack SYNs.” This is quite a simple method. Therefore, when the target computer sends the SYN/ACK to the muted computer, the backlog would simply drop the oldest connection to make room for the incoming legit connection of SYN/ACK.

II (B) (2). Cryptographic Component to Sequence Numbers

Mitnick built off the work of Morris and the “Morris Attack.” The fix in this case is to prevent the guess of the sequence number based on an algorithm. RFC 1948 was soon released (possibly in view of the Mitnick attack) in order to remedy “Sequence Number Attacks” [8]. Specifically, RFC 1948 mention “Morris [not Mitnick] describes a form of attack based on guessing what sequence numbers TCP will use for new connections.” RFC 1948 goes on to propose the used of “cryptographic authentications”. It is important to note that RFC 1948 recommends MD5 as “a good choice[] since the code is widely available.” However, MD5 and even SHA1 are considered non-cryptographic hash algorithms. People are now transitioning to

SH2. Nonetheless, as long as System Admins utilize cryptographic hashes in the principles found in RFC 1948, they are able to obfuscate the sequence number which his the primary vulnerability to be exploited during TCP.

III. Active Measures

III (A). Active Measure Vulnerabilities

III (A) (1). No Aggregation

During the lecture video, Mitnick was noted to have rewritten the rhosts file instead of appending it. One possible vulnerability is that this file should have been monitored, for example, with Wiretap. Additionally, the RST bit is not logged and this would allow Mitnick to close a TCP connection without being logged. Mitnick also uses the finger command to determine if there was a trust relationship. This information may or may not have been logged.

III (A) (2). Poor Password Control

As mentioned in the section of Applications, it was noted that Sysadmins should not allow the use of rlogin and instead opt for using SSH. SSH may require a password; however, this might not be sufficient.

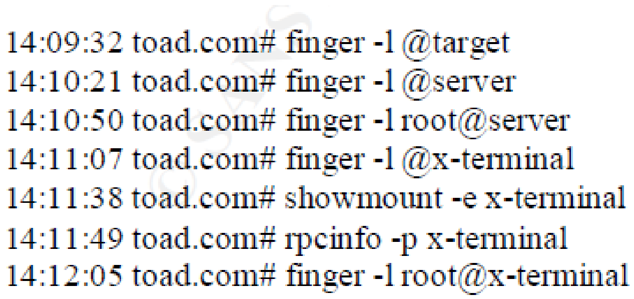
III (A) (3). Message Forging

Mitnick also utilized IP spoofing, forging the SRC, when he would send the SYN and ACK's to the target computer. While doing this, Mitnick was performing these actions off the subnet.

III (B). Active Measure Fixes

III (B) (1). Graylog Server

Log aggregation is very important for digital forensics. Having time stamps and what type of changes occurred is very important. In the example in the video, Mitnick made an error by rewriting the file instead of appending it. Nonetheless though, log aggregation would be able to detect changes to file, in addition to deletion, addition, and the like. Mitnick also used the finger to poke and prod the computer to determine if there was a trust relationship. According to SAN Institute [4], the commands would have looked something like:



```
14:09:32 toad.com# finger -l @target
14:10:21 toad.com# finger -l @server
14:10:50 toad.com# finger -l root@server
14:11:07 toad.com# finger -l @x-terminal
14:11:38 toad.com# showmount -e x-terminal
14:11:49 toad.com# rpcinfo -p x-terminal
14:12:05 toad.com# finger -l root@x-terminal
```

Using log aggregation, we would be able to use the time stamp and count() function in Graylog. We would be able to figure out how many times the finger command is being used within a given time period, and send an alert accordingly. Similarly, when Mitnick would enter the system and modify/delete/edit the rhost file, this would be logged and noted accordingly. System Admins would be able to write to the syslog file and have this information aggregated to Graylog accordingly. An alert of an important file change could be triggered too.

III (B) (2). Password Control

As establish in the first section of Applications, rlogin should not be used. Although, according to Red Hat Services, for rlogin, "[y]ou should never provide public, password-free accounts on your machine....To disable password-free login, edit the file /etc/pam.d/rlogin and /etc/pam.d/rsh and reconfigure[.]" [1]. With that noted, rlogin should be replaced with SSH. Although, SSH is cryptographically secure, the *implementation* may not be. It is important to have strong passphrase or long-passwords. Further, giving employees InfoSec training once a year and having them change their password periodically is also important.

III (B) (3). Firewall Rules

Mitnick used IP spoofing when he sent the SYN and the ACK's. Mitnick, however, was not on the same subnet. To remedy this: have a firewall block packets incoming packets that have the same IP as computers on our subnet. This solution was proposed by Mr. Morris, the author of the Morris Attack ("A workable solution might be to only trust hosts on the same physical network, and modify gateways to reject packets that claim to, but do not in fat, come from directly connected networks."). [7]. Similarly, RFC 4987 discloses that firewalls can "offload the connection establishment procedures onto a firewall that screens connection attempts until they are completed[.]" As such, a firewall could block a flurry of deleterious messages coming from outside the subnet and protect end systems accordingly.

Conclusion

Mitnick did not hack in a vacuum. Instead, he built his attack from Mr. Morris's discovered vulnerability associated with sequence number generations and incrementation. Although this paper discussed solutions to the vulnerabilities, some of them are out of date. For example, RFC 1948 discusses the use of MD5, a cryptographic component, as a remedy for sequence number predictability. It was also noted that MD5, now, is out of date and InfoSec is now transitioning to SHA2. This is the very nature of InfoSec. A game of cat and mouse. Hackers discover vulnerabilities and InfoSec responds accordingly.

References

1. (n.d.). Retrieved from https://www.distributednetworks.com/redhat-linux-system-administration/module2/insecure-remote-login.php
2. Defending Against Sequence Number Attacks. (n.d.). Retrieved from https://tools.ietf.org/html/rfc1948
3. Faking the TCP handshake: Hacker News. (n.d.). Retrieved from https://news.ycombinator.com/item?id=10654285
4. Global Information Assurance Certification Paper. (n.d.). Retrieved from http://staff.ustc.edu.cn/~guoyan/risk14/kevin-mitnick-hacking_1929_.pdf
5. Guha, B., & Mukherjee, B. (n.d.). Network security via reverse engineering of TCP code: vulnerability analysis and proposed solutions. *Proceedings of IEEE INFOCOM 96. Conference on Computer Communications*. doi: 10.1109/infcom.1996.493354
6. Institute of Electrical and Electronics Engineers. (2000). *Ieee 100: the authoritative dictionary of Ieee standards terms*. New York.
7. Morris, R. T. (n.d.). A Weakness in the 4.2BSD Unix TCP/IP Software. Retrieved from https://pdos.csail.mit.edu/~rtm/papers/117.pdf
8. TCP - Initial Sequence Number and the Mitnick Attack. (n.d.). Retrieved from https://www.youtube.com/watch?v=DR8gkI4rfMw
9. TCP SYN Flooding Attacks and Common Mitigations. (n.d.). Retrieved from https://tools.ietf.org/html/rfc4987