Analyzing Risks and Vulnerabilities' of Various Computer Systems and Undergoing Exploitation using Embedded Devices

Drew Branch

Kennedy Space Center

November 3, 2014

Author Note

Drew A. Branch

B.S. in Electrical and Computer Engineering, *Morgan State University*

M.P.S. Cybersecurity, *University of Maryland, Baltimore County* (In Progress)

Contact: DrewBranch@umbc.edu

Table of Contents

Abstract

Security is one of the most if not the most important areas today. After the several attacks on the United States, security everywhere has heightened from airports to communication among the military branches legionnaires. With advanced persistent threats (APT's) on the rise following Stuxnet, government branches and agencies are required, more than ever, to follow several standards, policies and procedures to reduce the likelihood of a breach. Attack vectors today are very advanced and are going to continue to get more and more advanced as security controls advance. This creates a need for networks and systems to be in an updated, patched and secured state in a launch control system environment. Attacks on critical systems are becoming more and more relevant and frequent. Nation states are hacking into critical networks that might control electrical power grids or water dams as well as carrying out APT's attacks on government entities. NASA, as an organization, must protect its self from attacks from all different types of attackers with different motives. Although the International Space Station was created, there is still competition between the different space programs. With that in mind, NASA might get attacked and breached for various reasons such as espionage or sabotage.

My project will provide a way for NASA to complete an in house penetration test which includes: asset discovery, vulnerability scans, exploit vulnerabilities and also provide forensic information to harden systems. Completing penetration testing is a part of the compliance requirements of the Federal Information Security Act (FISMA), NASA NPR 2810.1 and related NASA Handbooks. This project is to demonstrate how in house penetration testing can be conducted that will satisfy all of the compliance requirements of the National Institute of Standards and Technology (NIST), as outlined in FISMA. By the end of this project, I hope to have carried out the tasks stated above as well as gain an immense

knowledge about compliance, security tools, networks and network devices, as well as policies and procedures.

Project Description

I was given the task to conduct a scaled penetration test on a sandboxed test bed network of multiple computers with various operating systems. The goal of this test was to show proof of concept that a penetration test can be carried out by using low cost embedded devices and open source software. In the near future, a penetration test will be conducted by an outside entity and the results of both tests will be compared. The three phases of a penetration test that were focused on were: asset discovery, vulnerability discovery and vulnerability exploitation. To avoid the quality of the project being hindered, the scope of the penetration test was scaled due to time constraints.

Methodology

To perform this scaled penetration test I used a number of devices and tools. An embedded device, which is a computer system with a dedicated function, was used to run the open source penetration testing operating system. The open source penetration operating system came with a variety of penetration testing tools already installed. I conducted extensive research on various open source tools that enabled me to complete the penetration test in a guided manor. I compiled a list of these tools with a short description of their functions. These tools were then sorted and prioritized by the function of the tool and the amount of features the tool had that were useful. I installed the missing tools to make the penetration operating system installation more geared to my networked environment.

After the tools were installed, I tested the tools for full functionality. During this test, I made sure that all of the tools' dependencies were installed so that the tools could be opened successfully and operated to their full potential. After the dependencies were installed, I conducted several test runs of the programs and created a command reference guide.

Once the penetration test was started, the selected and prioritized tools were used to complete each phase of the penetration test. During each phase, documentation was thoroughly taken of the output of the tools to document the steps and for further analysis.

Results

Asset Discovery

During the asset discovery phase, I ran multiple programs and discovery scans to gain as much information as possible about the assets on the test bed network. During these scans, I found out whether an asset was up and running, the internet protocol (IP) address, which OS the system was running, which ports were open, the SSH host key and the network topology of the test bed. Figure 1 and Figure 2 displays the output of an intense scanned computer system on the network.

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-10-28 10:06 EDT
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Ping Scan at 10:06
Scanning          151 [4 ports]
Completed Ping Scan at 10:06, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:06
Completed Parallel DNS resolution of 1 host. at 10:06, 13.00s elapsed
Initiating SYN Stealth Scan at 10:06
Scanning        .151 [65535 ports]
Discovered open port 22/tcp on         .151
Discovered open port 111/tcp on         .151
Discovered open port 6000/tcp on         .151
Discovered open port 831/tcp on         .151
Completed SYN Stealth Scan at 10:06, 5.99s elapsed (65535 total ports)
Initiating Service scan at 10:06
Scanning 4 services on         .151
Completed Service scan at 10:09, 131.17s elapsed (4 services on 1 host)
Initiating OS detection (try #1) against         .151
Initiating Traceroute at 10:09
Completed Traceroute at 10:09, 0.02s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 10:09
Completed Parallel DNS resolution of 2 hosts. at 10:09, 13.00s elapsed
NSE: Script scanning         .151.
Initiating NSE at 10:09
Completed NSE at 10:11, 121.02s elapsed
Nmap scan report for         .151
```

Figure 1: Discovery Scan Part 1

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh?
| ssh-hostkey:
|   1024 23:90:48:ce:51:a4:55:c6:74:6f:64:e3:14:50:7c:ab (DSA)
|_  2048 7a:2e:e5:23:57:c6:04:83:de:9b:e9:0a:0f:70:c0:24 (RSA)
111/tcp  open  rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2           111/tcp  rpcbind
|   100000  2           111/udp  rpcbind
|   100024  1           828/udp  status
|_  100024  1           831/tcp  status
831/tcp  open  status  1 (RPC #100024)
| rpcinfo:
|   program version   port/proto  service
|   100000  2           111/tcp  rpcbind
|   100000  2           111/udp  rpcbind
|   100024  1           828/udp  status
|_  100024  1           831/tcp  status
6000/tcp open  X11     (access denied)
Device type: general purpose
Running: Linux 2.4.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.4 cpe:/o:linux:linux_kernel:3
OS details: DD-WRT v24-sp2 (Linux 2.4.37), Linux 3.2
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
```

Figure 2: Discovery Scan Part 2

After the initial discovery scans were complete, I conducted a trace route scan to determine the IP addresses of any hubs, routers, or switches that might be on the network. Knowing the IP address and/or media access control (MAC) address of a connection point

within a network would allow a non-authorized entity to conduct a man in the middle attack and monitor all network traffic. The trace route scan discovered that there was one networking device, xxx.xxx.xxx.2, on the network as depicted in Figure 3.

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-10-28 09:58 EDT
Nmap scan report for        .151
Host is up (0.00015s latency).

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.07 ms           .2
2   0.11 ms        .151

Nmap done: 1 IP address (1 host up) scanned in 26.06 seconds
```

Figure 3: Trace route

Once the IP address was known, the device was scanned using an intense scan. The device was found up and running and the MAC address was also discovered. The OS of the networking device could not be determined but suggestions were produced with the percentage of likelihood of each as depicted in Figure 4.

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-10-28 12:21 EDT
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 12:21
Scanning          .2 [1 port]
Completed ARP Ping Scan at 12:21, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:21
Completed Parallel DNS resolution of 1 host. at 12:21, 0.03s elapsed
Initiating SYN Stealth Scan at 12:21
Scanning          .2 [65535 ports]
Completed SYN Stealth Scan at 12:21, 3.67s elapsed (65535 total ports)
Initiating Service scan at 12:21
Initiating OS detection (try #1) against          .2
Retrying OS detection (try #2) against          .2
NSE: Script scanning          .2.
Initiating NSE at 12:21
Completed NSE at 12:21. 0.00s elapsed
Nmap scan report for          .2
Host is up (0.00025s latency).
All 65535 scanned ports on          .2 are closed
MAC Address:                  (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Aethra Starvoice 1042 ADSL router (91%), Bluebird SuperDOS (91%), Brother NC-130h
print server (91%), Brother HL-1870N printer (91%), Brother HL-2070N or MFC-5460CN printer (91%), Brother
HL-2070N printer (91%), Brother HL-5070N printer (91%), Brother MFC-7820N printer (91%), Brother MFC-9420CN
printer (91%), Elk ELK-M1EXP Ethernet-to-serial bridge (91%)
```

Figure 4: Discovery Scan of Networking Device

After the scans were completed on the networking device, a clear network topology was obtained as shown in Figure 5, where localhost is the embedded device.

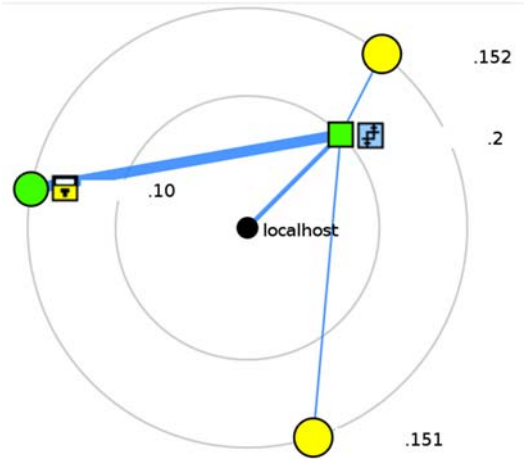

Figure 5: Test Bed Network

Vulnerability Discovery and Exploitation

The next phases of the penetration test, vulnerability discovery and exploitation, were performed in concurrence with one another. This was possible because one of the open source tools was comprised of other open source tools that had vulnerability discovery and exploitation capabilities. This program had the capability of performing discovery scans as well. After performing a discovery scan within this tool, the scanned systems are displayed in a plane with the OS icon identification as monitors as pictured in Figure 6.

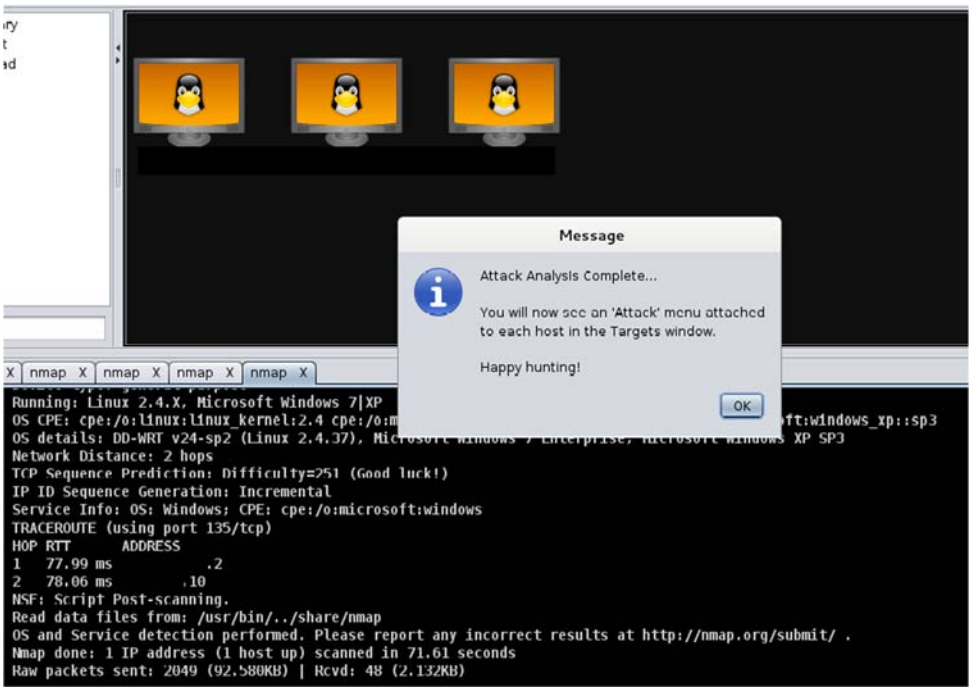

Figure 6: Discovery Scan w/ OS

After the scan was complete, I noticed that there were three open source operating system machines. This information was incorrect. I had to further investigate each host to see if I could find information that allowed me to correctly identify the right operating system. I found that the discovery scan used also scanned for services that might be running. In Figure 7, I found the computer system that was misidentified by looking at the running services.

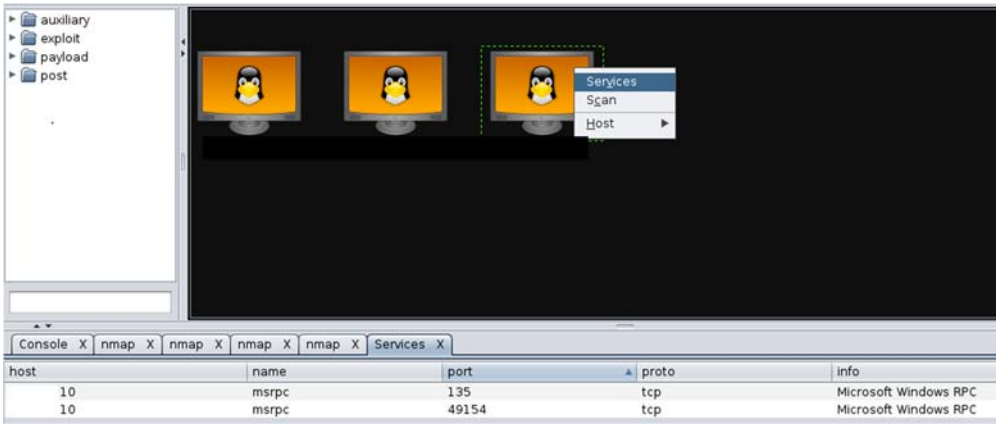

Figure 7: Running Services

Figure 8 shows how I changed the OS of a system in the plane after discovering the identification was wrong.

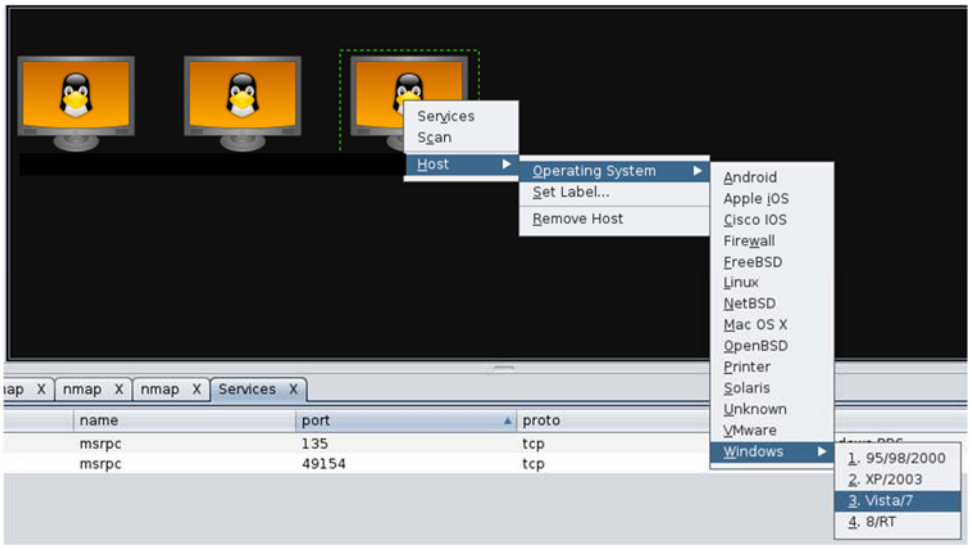

Figure 8: Changing the OS

After the system was changed, I delivered the exploitations found to compromise the systems. This tool found thirteen exploits for the machines on the network, twelve for two of the open source machines and one for the commercial machine. I was not able to compromise the systems using the exploits found. Figure 9 shows the results of exploitation attempts.

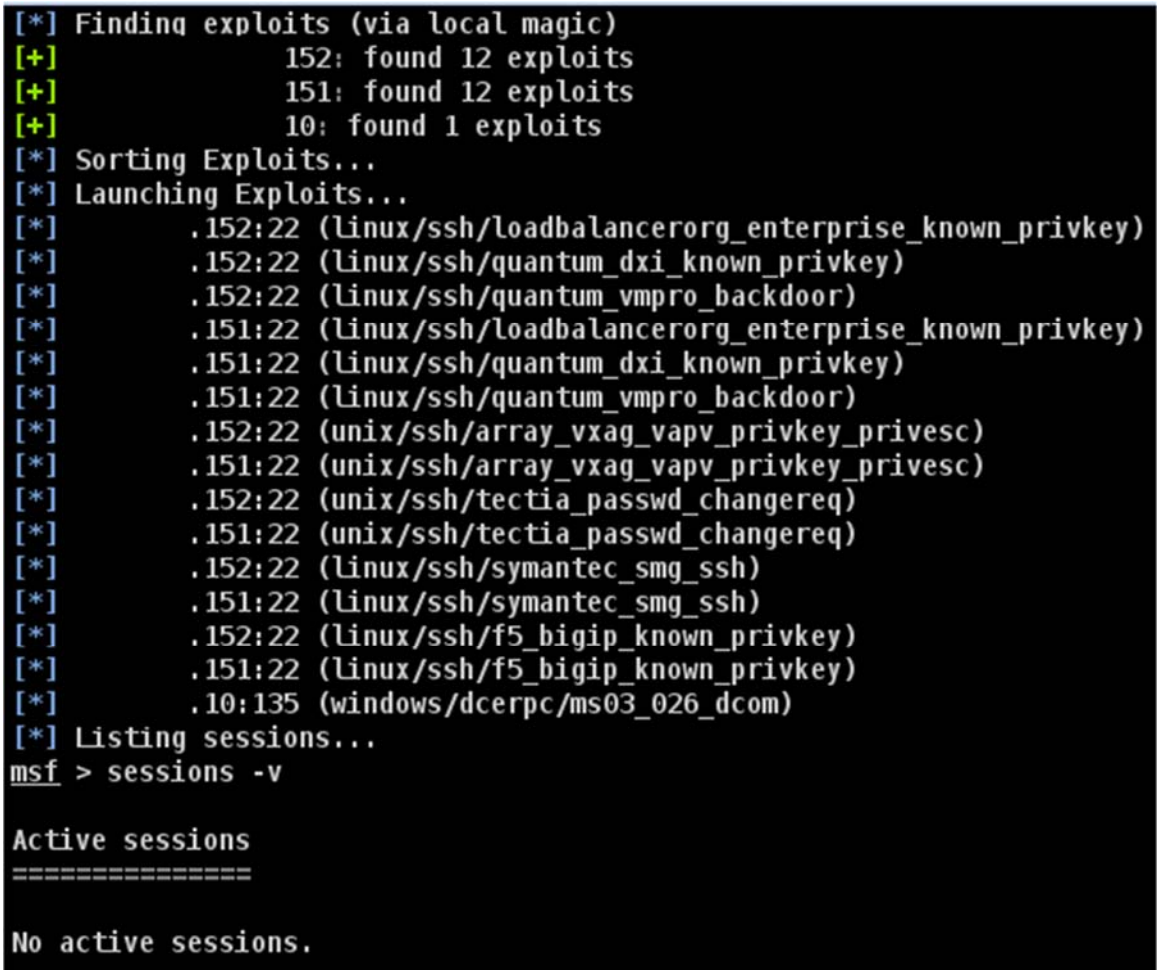

Figure 9: Exploitation Delivery

Using the exploits, I could not exploit the computer systems. The exploits above attempted to create a secure shell (SSH) session between the embedded device and the computer systems. If this would have been possible I would have had access and control of the system(s). If more ports were opened and not up to date with the latest patches, the computer systems would have be more vulnerable.

Beneficial Exposure

Currently, I am completing my master's degree in cybersecurity at UMBC. The program that I am enrolled in is geared more towards government IT security, law and policies. Fortunately for me, this internship has a direct correlation with what I learned before coming to Kennedy Space Center and what I will build on further when I leave. This experience was great and I will definitely take this experience and everything I learned while at KSC with me in my future.

Over the past semesters, this opportunity enhanced already possessed skills, exposed me to new skills and provided hands on experience with software and hardware that I will use in my career field. My communication skills, confidence in public speaking, and knowledge about numerous IT security subject matters were built by going to group meetings and actively expressing myself within them. Also by me receiving real work, I am gaining real world IT security experience. In graduate school, I learned about: mitigation, risk analysis, policy making, business continuity plans, disaster recovery plans, network devices, attack vectors, compliance laws, patch management, and various other security tools. By being here, I have gained a real world, in-depth experience on all of those topics and how they are implemented and sustained. Being involved with a project for over a year and given a vast number of projects and responsibilities really gave me the capability to see how IT solutions are researched, evaluated, purchased and then implemented.

This opportunity was a perfect opportunity for me. I am doing work that interests me, that is relevant to current security topics and I have gained experience that employers are looking for in a future employee. I am convinced that after my yearlong internship I will have a considerable advantage over the average graduates competing for the same job. This

is due to the fact that I am getting a complete experience of the IT security field and IT security insight from a government aspect.

Conclusion

To date, I have gained valuable knowledge and experience. So far, I have worked on compliance projects, a project management project carried out a secure programing service analysis and assessment, analyzed security issues using risk automation software, and completed a penetration device testing and assessment. Also, I gained valuable non-technical skills dealing with budget requirements and making decisions for products that satisfies the most security requirements. Over the four semesters I have been involved in many different facets of IT security. This experience is the highlight of my career so far. I am extremely excited to hopefully return to KSC in the future, to get new relevant projects and expand my experience and knowledge.