

Forum:[Hack Forums](#)

Thread: [Real Story of a Pentester #3 - Hacker's Wet Dream](#)

[BLANK] - February 12, 2018, 2:14 pm

Hacker's Wet Dream

Imagine being sent back in time, lets say to 2008, and you are equipped with a modern day computer with all the modern day exploits and methods. What havoc could you wreck? How easily would it be to pop boxes and compromise networks? Well, this happened to me. Not in the sense of actually going back in time, but I was placed on a network where I felt that I had gone back in time. Que the "What Year is It" meme, get your energy drink ready, crank that techno and lets go hack some shit!

The Beginning of the End

Early on in my penetration testing career I was put on a Best Practice test. This is the type of test where the scope is everything, where the rules are "do anything but don't break anything" and you have a set timeline to test and exploit as much as possible. Sounds amazing right, well it is, and it is what I live for at my job. Normally I am given a limited scope for some compliance test, but occasionally I get to really run wild in a network. I was given two weeks to see what was possible. My contacts were confident that I couldn't do much but were interested if I did find anything. So they put my custom device inside their network, I grabbed a dynamic IP and off I went. The only information I had to go on was a spreadsheet of where some static devices were located. This did save me time so I appreciated the help...not that I needed it.

I immediately spun up my broadcast listeners, at this time in my career I was not familiar with using Nmap for broadcast enumeration (I learned that later on, aren't you glad you read my other stories?) so I just went straight to CrackMapExec (CME) and performed a scan against the network which consisted of four separate subnets. There were so many hosts vulnerable to SMBv1 with no signing enforced it was crazy. I created my output list and spun up Responder with NTLMRelayx. Now if you are not familiar with those two tools, get familiar ASAP. Responder will poison broadcast requests to retrieve NTLMv2 credentials, these credentials are forwarded to NTLMRelayx which will attempt to authenticate to host machines over SMB (specified by single host or by target file) and if successful it will attempt to dump hashes or upload commands. The first attempt I just had it dump hash tables and I got dozens of hash tables, which meant probably 50 credential hashes. Even as a new tester I knew that I was already rocking this network.

I Like Your Cameras, Mind if I Borrow Them?

Knowing I was getting success from relaying credentials I decided to go for my first shell. One of the beautiful things you can do with NTLMRelayx is you can push entire powershell commands. So I spun up Empire, created a listener and a one line powershell stager. I pushed this one liner through NTLMRelayx and waited. It took a few minutes but eventually got an active agent on Empire. After enumerating the host I found that it was an old Windows Server 2000 and it was hosting their CCTV software. Did I really just crack their security system in under an hour? After enumerating the user hashes I was able to get a clear text password via an online hash cracker. Now normally this system did not enable RDP access...but what the hell. I enabled RDP and logged into the server. Immediately the camera system loaded and I was looking at live images of vans leaving the front gates. I could see several other screens as well. I could turn off cameras, put them in loops, delete video etc (of course I didn't). I looked at the spread sheet I was provided and saw there were two other servers hosting camera systems. And what do you know, they all accepted the same username and admin password. Awesome, complete control of the camera system. If I was Red

Teaming this would be really helpful when the physical intruders wanted to breach the building.

Patch Management, What is That?

Next I decided to see if any of the hosts were vulnerable to MS08-67 or other well known vulnerabilities. Sure enough, Windows XP boxes were found and owned as well as Windows 7 and Windows server 2008 hosts. I don't think a single host was older than Windows Server 2008 R2 on the entire network, and none of it seemed patched. While I did get shells on hosts and a couple of servers I don't like using these type of exploits if I don't have to. They can sometimes knock down a host (this may have happened during this test...don't tell anyone) so I will only use them strategically and if I can't find a better way in.

Everyone is Admin

After getting a lot of shells and credentials I decided that was good for one day. Of course I was so amped from the testing. Every tool and exploit was working. I got to try out stuff I had only read about or used in limited labs. I had my team cheering as I went from one machine to another, getting deeper and deeper. It was hard to call it a day but eventually I did.

The next day I came in early, I just wanted to get on the network and run with it. I examined my findings from the day before and noticed a similarity in the local administrator hashes, and by similarity I mean they were all exactly the same. They re-used the same local admin password for almost every single host computer...game fucking over. I spun up CME again but this time I used the Mimikatz module to dump all hashes including currently logged on. See Windows does not store your password in clear text, but the clear text password can be retrieved if the user is logged on, nice how that works right. Imagine the worst hollywood movie of data flowing down the monitor, hashes and username, the word PWND in green popping up etc, that is exactly how my screen looked because the local admin hash was getting me full access to dozens and dozens of computers.

As I scrolled through the clear text accounts I found my point of contacts, but it was only their local accounts not their Domain Administrator accounts. Still I kept them just in case. I kept scrolling and there it was, the holy grail a DA account in clear text. I immediately RDP to the Domain Controller, logged in as DA and the desktop loaded with all the beautiful Windows Server administration windows. Naturally the first thing I did was create a new domain admin for myself (this can sometimes trigger security so be careful) and started to explore the network in full. Now I could literally log into every computer.

Full Access...But Wait There is More!

Going from computer to computer, server to server is time consuming. I spent the better part of a day and a half just enumerating as much as possible. It wasn't until the fourth day that I found the gold at the end of the rainbow. I found the backup server which also hosted email and other services. All of this could be accessed via a single login, but the login was no DA because it was a program not a domain access. Remember those credentials I saved earlier from my point of contact...yep that worked. At first I was not sure what I had accessed but as I moved from window to window I found myself staring at a complete account manager. I mean every kind of account you can think of, corporate paypal, UPS, MSDN, Cisco Account, account/password to the firewalls, IPS, and other security devices. It was a ton of information, so naturally I exfiltrated the data because the manager came with a nice export feature. If I was a real attacker I could have lived off this stuff, who knows how far I could go without being caught. There was one other program I found (same login) on that server, the backup mail server. But, it was not just a backup, it was active as well. I decided to let my point of contacts know just how far I had gotten. I logged into the mail server, found one of my point of contacts accounts and emailed the other point of contact telling them how this could have been a mass phishing email.

They asked to talk to me about 20 minutes later.

The Phone Call

I got a call as I was driving home. They were wondering how I sent that email and if I could explain how I got access etc. So this was on a Thursday and I told him "I have had DA since Tuesday", he replied "Oh, that is disconcerting". Yea no shit, they had no idea that I owned the network. He said they could tell I was gaining some access because they were tracking my device IP a bit but they had no idea just how far I really was getting. The good news was they were genuinely interested to improve their security. Unfortunately they were still living in the 2008 era of network security.

And then I woke up....and realized I had to write the damn report which took nearly 10 hours. Sigh, the price to pay for awesomeness.

Mad76e - February 12, 2018, 2:59 pm

Yeah around 2-500 pages with summary detailed description of each vulnerabilities with suggestion how to patch/remove each vuln. /updates even firmware, costs, conclusion etc. Did you get back later to see if they secured the network/servers? Im familiar with the concept.. Just "that" moment had been so damn good if someone else could do. Sometimes its more writing than actually hacking

[BLANK] - February 13, 2018, 3:45 pm

I have not been asked to retest yet though I heard they may want another test done. Our reports don't typically go into cost analysis, I mainly test, find issues and write it up. Typically my part is about 30 pages depending on how many different tests there are. The CRA probably writes up more since they are performing a full risk analysis.

Nomad.Null - February 15, 2018, 9:44 am

This makes me grateful that we have tech writers that write our reports for us xD.

Well done.

[BLANK] - February 15, 2018, 2:36 pm

(Feb 15, 2018 10:44 AM)True Demon Wrote: [\[CITE\]](#) This makes me grateful that we have tech writers that write our reports for us xD.

Well done.

Yea I don't understand the reasoning of "lets take these geeky computer enthusiasts and make them write professional reports for the customers." Granted I am actually a decent report writer but it should never be an expectation. You are very lucky to have tech writers!

Mad76e - February 15, 2018, 9:05 pm

(Feb 15, 2018 10:44 AM)True Demon Wrote: [\[CITE\]](#) ...

Youre back?! Long time no see.. Nice to see you again!

Nomad.Null - February 16, 2018, 1:54 pm

@EasyGhost I'm not half bad at it myself, but yeah, tech writers who are literally word and excel wizards makes for a way better deliverable. You should pitch it to your boss. It makes a huge difference.

@Mad Never left, homie. Just been flying under the radar mostly.

Mad76e - February 16, 2018, 1:58 pm

(Feb 16, 2018 02:54 PM)True Demon Wrote: [\[CITE\]](#) @EasyGhost I'm not half bad at it myself, but

yeah, tech writers who are literally word and excel wizards makes for a way better deliverable. You should pitch it to your boss. It makes a huge difference.

@Mad Never left, homie. Just been flying under the radar mostly.

Nice.. You need anyone to talk to now and them just send me a PM, Im there for my friends :)

[BLANK] - March 2, 2018, 11:12 am

(Feb 16, 2018 02:58 PM)Mad76e Wrote: [\[CITE\]](#) Nice.. You need anyone to talk to now and them just send me a PM, Im there for my friends :)

Thanks, maybe you can get me an invite to Red Lions.

laaS - March 2, 2018, 4:45 pm

That was a fun read. One thing I am curious about is the mimikatz part, I've used mimikatz before to dump passwords but from what you've written it looks like you remotely ran mimikatz on a bunch of computers which had logged in users and it dumped their passwords, I wouldn't have thought this was possible as I'd have thought it would create a new session as the user you're running mimikatz as, is this incorrect?

One little thing to add, one of the first things I like to do when I get access to a DC is take a copy of system hive and NTDS.DIT, if you've got a decent gpu you can crack passwords way faster using hashcat than you ever could trying to dictionary attack over a network, that way if you get detected/locked out you can at least try authenticate as other cracked users :)

[BLANK] - March 2, 2018, 9:06 pm

So the CME Mimikatz module first initializes a http server on the attacker's host. Then uses the supplied credentials to log into the target machine via SMB. If access is granted and if the authenticated user has the rights to run Powershell then Mimikatz.ps1 is ran in memory and the results go to the attacker's http server.

laaS - March 3, 2018, 12:27 am

(Mar 2, 2018 10:06 PM)EasyGhost Wrote: [\[CITE\]](#) So the CME Mimikatz module first initializes a http server on the attacker's host. Then uses the supplied credentials to log into the target machine via SMB. If access is granted and if the authenticated user has the rights to run Powershell then Mimikatz.ps1 is ran in memory and the results go to the attacker's http server.

Ahh that's pretty cool, will that work under the default execution policy or does the user have to have powershell with unrestricted execution policy? I'm gonna have a play with that.