HACKING Ugly

How doing things the “wrong” way sometimes turns out right

Tom Liston – tliston.phone@gmail.com
Twitter: @tliston
ELEGANCE OVERRATED

• Don’t get me wrong: elegance has it’s place
  • It’s just that it isn’t - by default – better
  • And yet, that’s our assumption…
  • So much so that oftentimes we pursue elegance for elegance sake
DIJKSTRA SPEAKS

“Elegance is not a dispensable luxury but a factor that decides between success and failure.”

-Edsger Dijkstra
HE ALSO PLAYED GORDON FREEMAN IN HALF-LIFE
LISTON SPEAKS

“Elegance is wonderful but can be kinda time-consuming. And, FYI, ‘succeeding’ is the deciding factor between success and failure.”

-Tom Liston
REMEMBER, “IT WORKS” IS ALWAYS ENOUGH

Last December, I got a phone call from some hacker “wannabe” named Ed Skoudis

• “Skodo” wanted to do some testing on an executable, but the executable wouldn’t run
• It appeared to require something “additional” before it would run (phoning home, some condition, etc…)
• Skodo wanted to know if I could “fix” the executable so… well… it would run…
GETTING TO “IT WORKS” IS – SOMETIMES – TOO EASY

• I did a little “deep inspection” of the executable.
  • Note: I never “reverse engineer” anything. That would be wrong.
• It turned out, it wasn’t phoning home
• It was checking to see if a “companion” program was running
• The check was being performed by just looking for a process with a specific name
• This creates a dilemma…
RULE #1: A GOOD MAGICIAN NEVER REVEALS HIS TRICKS

• I could just tell “Skodo” to rename calc.exe and run it at the same time as he was running his test executable
  • H@x0r 5tr33t cr3D
• Instead, I put together a do-nothing hackerish-looking program (with the correct name), and fired it off to “Skodo” with instructions to run it while doing his testing
Running my little program allowed Skodo to do his testing on the original app – which now ran like a champ…

Then Skodo asked what my executable did

I refused to tell him

“You wanted it fixed… I fixed it. If you want to know HOW I fixed it, the price goes up.”

RULE #2: IT’S ALL IN THE PRESENTATION

HACKING Ugly
RULE #3: sometimes a good magician MUST reveal his tricks

• A day later when talking with him about my “remuneration,” Skodo dropped a bomb:
  • “Since you won’t tell me, I have one of my guys reversing your executable to see what it does.”
HOW TO BECOME A HACKING GOD: LESSON 1

BEHOLD GOD
Creator of ASCII, the Batmobile, and tasty, tasty bacon
For these, we thank Him
LESSON 1: TRY DUMB STUFF

I walked up to our booth at ShmooCon to see all of the InGuardians’ staff huddled around a laptop. It had been returned by an employee and was to be reprovisioned to another. Unfortunately, we hadn’t been provided with a password for the only account on the machine.

While everyone tossed out ideas for getting in (I distinctly remember Jay Beale saying “Firewire attack” over and over…) I sat down and typed in:

password
THE PROBLEM

• Back in the day (Win2000/WinXP), while doing some testing, we could cause a Windows “system” process to write to the filesystem, but that was about it…

• While that was fun, we really wanted to turn that into something more – unfortunately our options weren’t too good
  • We could write to a startup directory and wait for a reboot
  • We could put a tasty-looking file somewhere and hope someone would run it
• We also had the ability to crash the process
• What happens when a process crashes?
  • drwatson.exe runs
• So the solution is simple: overwrite drwatson.exe and then crash our process!
DAMN YOU, WFP!

- Then...Windows File Protection reared its ugly head
- WPF “protects” important Windows files by replacing them if they’re ever altered or deleted
- In our testing, our exploit only worked about 5% of the time because when we overwrote drwatson.exe, WPF replaced it before we could crash our process and spawn it
IF I HAD A HAMMER, I’D HAMMER IN THE MORNIN’…

- Lots of elegant ideas got tossed around about ways to disable WPF
  - They were all *waaaaaaaaaaay* too much work
- I went with the “brute-force and ignorance” approach
- I Googled up a list of every file protected by WPF and added code to our attack to overwrite them all
- With WPF “otherwise occupied,” I overwrote drwatson.exe with our attack and crashed the process
- 100% effective
LESSON 2: EVEN WHEN TRYING TO BE SMART, PEOPLE STILL DO DUMB THINGS

Mike Poor and I were onsite at a client location and had just started chatting with our contact when he decided to include someone else in the meeting. Before leaving the office, he made a big show of attaching his laptop lock, “because you guys can’t be trusted…”

As he walked out the door, I quickly told Mike, “stall him.” Mike stopped him just outside the office for a second, to ask him something and I walked out a few seconds later, handed him his laptop and explained that it would probably be MUCH safer if he just took it with him.
So...

What did I see?

Note: If you’re working with someone, and when you say something like “stall him” they don’t “stall him,” – no questions asked – you need to work with a better class of people

HACKING Ugly
I love having friends and colleagues who raise the bar and make me better

802.15.4 is a wireless protocol used for point-to-point or multi-point “personal area networks”

Most notably, the Zigbee protocol uses 802.15.4 as a baseline and adds some extra routing and networking “stuff” on top

802.15.4 specifies the use of the AES encryption algorithm with a 128 bit / 16 byte key for protecting data in transit
A BEAUTIFUL MIND

• So let’s say you have some captured Zigbee wireless packets and a firmware dump from a Zigbee device
• You could attempt to disassemble the firmware and locate the key – a “high touch” / “high tech” solution…
• Or, you could just realize that there’s only SO many 16 byte chunks in that dump file (ie. FileSize – 15)… and try using ‘em ALL to see if one decodes the packets
• “zbgoodfind” – a Python script from the Beautiful Mind of Josh Wright
HOIST WITH ONE’S OWN PETARD

- Retesting a client network after two years
- WORST. PENTEST. EVER.
- Found ONE machine that somehow had missed being patched
- Found an SFTP application
- The config file contains:
  - A USERNAME!!!
  - And an encrypted password

😊
WHAT TO DO…?

• I could always reverse engineer deeply analyze the program to try to decode the password
• Debugging could work, but I didn’t have a machine set up to do .NET debugging
• I ended up using a .NET “decompiler” and a hex editor
• I swapped the pointers for the machine name with the pointer for the password
• I ran the program, and watched as it did a DNS lookup for ………
SECURITY IS ABOUT BREAKING RULES

• We were testing a Smart Grid deployment based on Data Over Powerline technology
• Data over powerline has a very limited range, so the deployment used field devices called data aggregators that had 50-100 meters feeding them
• Single board computers field mounted on poles
• I opened one up and immediately saw something interesting
RULES? WHOSE RULES?

• I pulled it out of the device, plugged it into my laptop and used “dd” to image it.
• Mounted the image, edited /etc/passwd to add a new root equivalent user
• Unmounted the image, used “dd” to copy it to another CF card
• Plugged it into an aggregator and booted it up
• Hooked up to the local ethernet port and boom… I’m root.
• From there, we were “IN” to the back-end of the utility
BOXERS OR DEBRIEFS?

• In our debrief teleconference I described exactly what I had done to a group of utility executives
• One of the executives on the phone – I don’t know who – just couldn’t hold it in
• He said: “But… that’s not fair!”
LIFE ISN’T FAIR

BUT THE CAT GETS TO POOP IN THE HOUSE
OUR JOB: STAY OUTSIDE THE BOX

- Pentesters keep their clients safe by pooping in their houses.
- The job requires us to be “not fair” because the bad guys AREN’T fair.
- The road to “not fair” is oftentimes ugly.
QUESTIONS?

HACKING Ugly