Applied Security Project
A Course on Offensive Technologies

Fabio Massacci, Luca Allodi, Stanislav Dashevsky, Daniel Ricardo Dos Santos

Offensive Technologies

- The general who wins a battle makes many calculations in his temple ere the battle is fought. The general who loses a battle makes but few calculations beforehand.
  - Sun Tzu: the Art of War (512 BC)
- Defensive warfare, therefore, does not consist of waiting idly for things to happen.
  - Von Clausewitz: Principles of War (1812 AD)
- Offense and Defense aren't peers. Defense is Offense's child.
  - John Lambert, via Twitter (2014 AD)

Course Idea

- Play with real exploits and their countermeasures (static analyzers, security monitors, intrusion detection systems) and way to defuse the latter.
- This is a project course so there are very few lectures (mostly by students themselves to present their progress)
- This is also an experiment!

Course Credits

- Various Offensive Technologies to be experimented
- Project Course → 6 credits
  - One technology with simple implementation
- Research Project → 12 credits
  - Two technologies or one and a bit of automations
- Applied Security Project → 18 credits
  - More tehcnologies or sophistricated implementations

Course Plan

- September
  - Presentation of “playgrounds” by research group
  - Warm-up exercise: “students rates vulnerabilities”
  - Students read papers, Black Hat presentations etc.
- October
  - Students present their idea, how they understood the work etc.
- November
  - Start implementing stuff
- December or later
  - Presentation of work done

Example: Red Pills, Blue Pills

- "You take the blue pill, the story ends. You wake up in your bed and believe whatever you want to believe. You take the red pill, you stay in wonderland, and I show you how deep the rabbit hole goes.”
  - Morpheus, to Neo – The Matrix
- First idea by J Rutkowska at Black Hat 2006
  - Rootkit gives Blue Pill to Antivirus so Antivirus think everything is ok.

Red Pill: a more interesting idea

- Red Pills for software
  - a software takes a red pill and discover whether it is running in a virtual machine a
- Why it’s important?
  - Normal users run real machines, security researchers, search engines etc. etc. run virtual machines
  - if you discover you are on a real machine→ unleash attack
  - If you discover you are on virtual machine → uhm, looks you are under observation → deploy innocent behavior

The Task

- Javascript Red Pill for Browsers
  - When a client contact a (malicious) server the server ships back a piece of code that tells whether the browser is running on a virtual machine or a real machine
- Develop one or more working red pills
  - Start from paper by Ho et al. WOOT’14
- Instrument a black market exploit kit with the pills
- Test it in our malware lab
- The more pills, the more automatic/realistic the process the larger the credits and the votes

NodeJS Vulnerabilities

- NodeJS → leading open source Server-side Javascript engine
- Task
  - Deploy large NodeJS application (e.g. Ghost)
  - Inject one or more vulnerabilities into it
    - E.g. replacing libraries with vulnerable ones, manually change code, find them by crawling etc. etc.
    - E.g. OWASP top 10, SANS Top 50, Nodesecurity etc.
  - Write exploit to use it
    - E.g. Simple ('alert("U'r p0wn3d!") or more complex (heap overflow)
  - Run countermeasure/detection tool
    - E.g. JS static analyzers, NodeSentry Monitor, or w3af penetration tester
  - Automate Process using TextRex/malware lab infrastructure
  - Change/refine vulns until it escape detection
- Precise set-up to be discussed

OS Vulnerabilities

- Same idea but working on OS is harder
- Task
  - Identify software with vulnerability
    - Or write your own vulnerability
  - Receive baseline exploit
    - Work under conditions (e.g. mem address or crash sw)
    - From metasploit., exploit DB etc.
  - Try detect exploit with IDS (Snort, BRO)
  - Enhance the exploit = work without restriction
    - E.g. Super-duper sigereturn-oriented exploit
  - Repeat until exploit is undetected

Example of "Course Agreement"

- PhisBuster Project
- 18-21) Base set-up od the experimental framework
  - (original web site, Phishing copy web site, clients which can contact either the clone site or the original site, alarm server, embedded "call-alarm" Js on "separate" networks). See papers.
  - One can manually set the web sites, but must be possible to launch automatically the clients with at least three browsers (IE,FF,GC) connecting to the server or the phisher and there should be a procedure to collect all data from the experiment (#number of connections etc.)
  - Manual set-up of a variety of obfuscate "alarm bells" JS into the (e.g. JS embedded as css, images etc. etc. using string concatenation etc. etc.)
- 21-24) Automatic "vacuum cleaning" of the source web site into the phished site
  - There are a number of tools on the web to do that and simple experimental analysis of the combined pipe (embedding alarm N)+(vacuum clean with tool M}+(contact client with browser K
- 25-28) automatic tool
  - It parses a JS fraction of a server and embed and randomly changes the code so that the code is equivalent plus the alarm It should use some of the obfuscation techniques described in Sandmark http://sandmark.cs.arizona.edu/
- 29-31) as one but the alarm code has to be intermixe mixed with "normal" production code
  - so it is not possible to actually use the site without calling the alarm bell
- In all cases above:
  - Brief report (2-6pages) of work done
  - Code released into the laboratory infrastructure

Next Lectures

- Presentation of topics
  - a bit more in details
- Tutorials of
  - TestREx – a platform to test exploits
  - Malware lab – a platform for experimenting with exploit kits

IMPORTANT

- You must use your own machine(s) or the lab machine(s) that we make avaliable for you in this course
- The attempt to make exploits work on machines you do not own is ILLEGAL
  - Italy ratified the Council CyberCrime Convention
  - Hacking into someone'else system is punished with a fine of 10K€ + 1-5 years of jail (3-8 if it is a State's system)